actusécu

By xmco

OCTOBRE 2025

À LA UNE

L'usage offensif de l'Intelligence Artificielle par les cyberattaquants

ARTICLE

Nmap, vétéran incontournable de la cybersécurité offensive

62

Octobre 2025.

Responsable de publication : Clémence Illouz, XMCO - Direction artistique / Réalisation : Romain Mahieu, agence plusdebleu - Contributeurs : Les consultants du cabinet XMCO. Crédits photo : ©XMCO, ©AdobeStock. Contact Rédaction : actusecu@xmco.fr

L'édito

Marc BEHAR, PDG et Fondateur d'XMCO

Welcome back !
L'ActuSécu est de retour !

> Nous en sommes tous très fiers, chez XMCO, et nous espérons que vous apprécierez ce nouveau numéro qui comporte beaucoup, mais vraiment beaucoup de contenu… Bien entendu, XMCO n'a jamais cessé de travailler pour innover, développer son expertise, mais nous avons passé 2 années à nous transformer, et ce n'est pas fini.

Pour ce retour « dans les kiosques », vous trouverez dans ce numéro deux sujets aux antipodes de la modernité : l'usage offensif de l'IA par les cyber-attaquants et Nmap, un vétéran de la cybersécurité…

C'est un choix volontaire, car si Nmap constitue l'un des plus vieux outils, il est encore largement utilisé dans le cadre de la sécurité offensive.

De l'autre côté, pas une question ne nous est épargnée concernant l'IA et je dois reconnaître qu'avec l'âge, l'acceptation des progrès devient de moins en moins naturelle, fluide.

Ma génération a connu les cassettes audio qu'il fallait rembobiner avec un stylo, et l'enregistrement « live » des chansons qui passaient à la radio, en espérant que l'animateur se taise pendant l'intro…
On connaît désormais le disque SSD de plusieurs Terraoctets pour le même encombrementqu'une carte de crédit.

Notre ambition n'est pas d'effacer le passé, mais plutôt de construire un avenir qui conserve le meilleur de ce qui a déjà été accompli.

Je tiens à remercier chaleureusement mes collaborateurs pour leur passion, leur investissement personnel et leur envie de bien faire, afin qu'XMCO soit le cabinet de conseil qui combine le meilleur du passé, du présent et du futur.

Bonne lecture ! ■

Le saviez-vous ?

246

C'est le nombre d'alertes envoyées en mai 2025 aux clients Serenety sur des compromissions d'infostealer avec publication ou mise en vente d'informations de connexion volées.

+100

C'est le nombre en constante évolution des Clouds of Logs, ces canaux Telegram qui vendent et mettent à disposition les données volées par des infostealers, et dont le cout d'un abonnement coute en moyenne 200 dollars par mois.

7

C'est le nombre des principaux infostealer à l'origine des compromissions que nous observons : Lumma, Vidar, Redline, Nexus Stealer, Atomic MacOS, Stealc, Acreed.
Suite au démantèlement des infrastructures de Lumma, des infostealer montent en puissance. C'est le cas d'Acreed, apparu début 2025.

3

C'est le nombre des Marketplaces majeures qui revendent les données volées par les infostealer, depuis le démantèlement de Genesis en 2023.
Il s'agit bien sûr de RussianMarket, mais aussi de 2Easy et d'Exodus Market.

394 000

C'est le nombre de machines Windows infectées par Lumma, rien qu'entre le 16 mars et le 16 mai 2025. Cela représente des millions d'identifiants volés pour être potentiellement réutilisés par des attaquants afin de compromettre les systèmes d'information de nombreuses organisations.

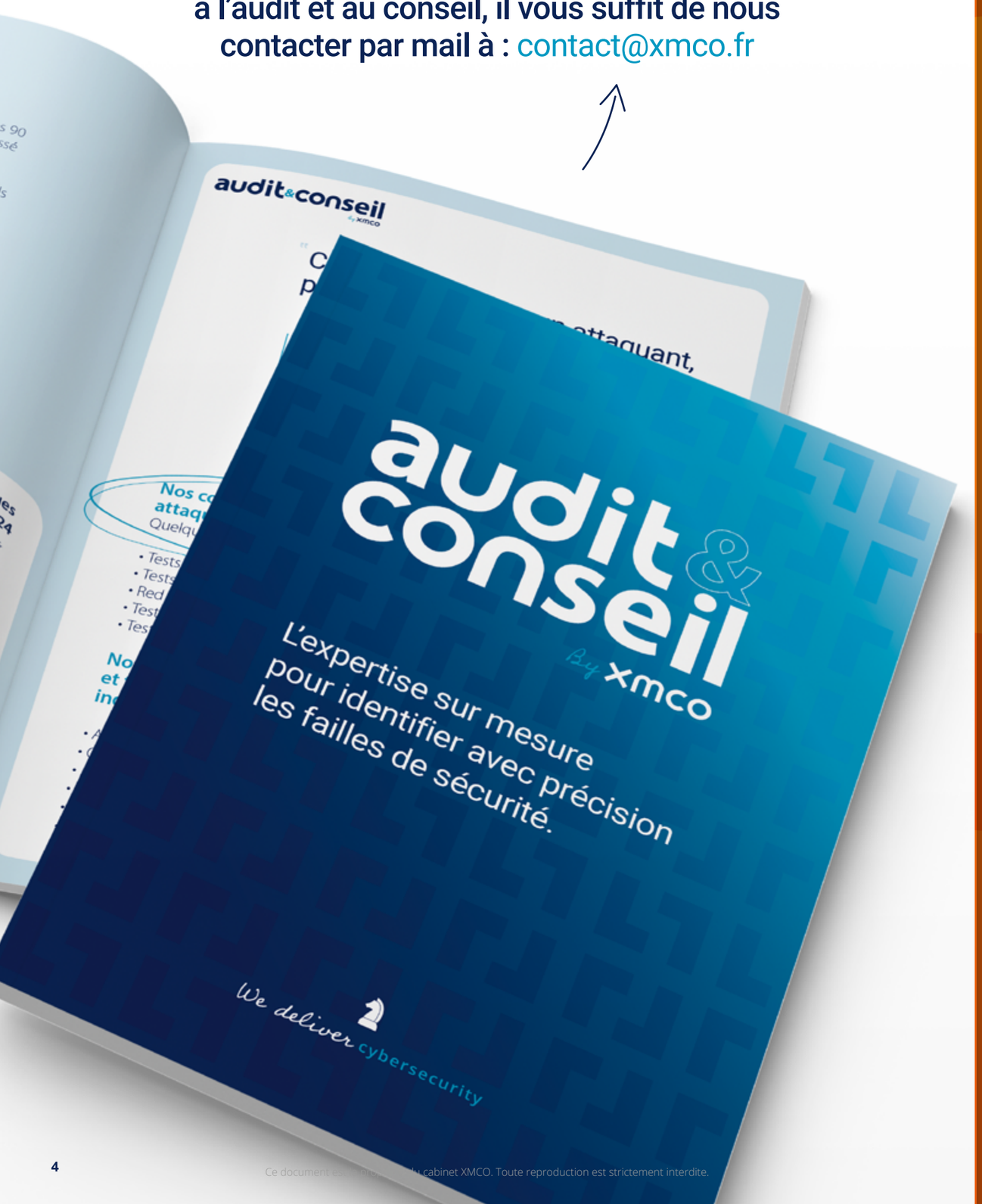
Sommaire

Numéro 62

OCTOBRE 2025

62

Pour découvrir notre nouveau livret dédié à l'audit et au conseil, il vous suffit de nous contacter par mail à : contact@xmco.fr



yuno décrypte

Que s'est-il passé dans le monde de la cyber ces dernières semaines ?

Yuno, notre service de veille cyber identifie les menaces détectées par le CERT-XMCO et revient sur l'exploitation d'**une vulnérabilité zero-day** affectant **Sitecore** et d'une seconde, **crtitique**, affectant les produits **Citrix NetScaler ADC** *et* **Gateway**.

La première, corrigée le 2 septembre 2025 et référencée **CVE-2025-53690 (CVSS : 9)**, permettait à un attaquant distant et non authentifié d'accéder à des données normalement restreintes, voire d'exécuter du code arbitraire.
La vulnérabilité provenait de la désérialisation de données non fiables dans certaines configurations exploitant un exemple de clé (machine key) exposé dans les guides de déploiement de Sitecore de 2017 ou antérieurs.
Le lendemain, **Mandiant** signalait des tentatives d'exploitation de la faille, s'appuyant sur la distribution d'une **payload ViewState** spécifiquement conçue, chargée d'exécuter en cascade le **malware WEEPSTEEL**. Cette souche de code malveillant disposerait de **fonctions similaires à celles de la backdoor GhostContainer**, utilisée pour compromettre les serveurs Exchange d'entités stratégiques en Asie en juillet dernier.
Afin d'endiguer ces risques, **Yuno a alors recommandé l'installation des correctifs de sécurité proposés par Sitecore et de suivre les IoCs fournis par Mandiant**.

La seconde, corrigée le 26 août 2025 et référencée **CVE-2025-7775 (CVSS : 9.2)**, permettait à un attaquant distant et non authentifié de provoquer un déni de service et d'exécuter du code arbitraire.
Le même jour, la **Shadowserver Fondation** a identifié plus de **28 000 instances Citrix exposées sur Internet et vulnérables à la CVE-2025-7775**.
De son côté, notre service ***Serenety*** a détecté **56 assets Citrix NetScaler ADC et Gateway** disposant d'une version vulnérable parmi ses clients. En l'état, aucune information n'est publiquement disponible sur le groupe d'attaquants à l'origine de cette campagne.

En juin dernier, **Yuno documentait déjà l'exploitation** à grande échelle des failles critiques **CVE-2025-6543** et **CVE-2025-5777** affectant les mêmes produits Citrix. Afin d'endiguer ces risques, **Yuno recommandait alors l'installation des correctifs de sécurité proposés par Citrix pour NetScaler ADC et NetScaler Gateway**.

Pour en savoir plus sur l'anticipation des menaces cyber par Yuno :
www.xmco.fr/yuno-veille-vulnerabilites-cybersecurite/

L'usage offensif de l'Intelligence Artificielle par les cyberattaquants

Introduction

En novembre 2022, l'introduction de ChatGPT par la firme OpenAI a permis une démocratisation sans précédent des technologies de Natural Language Processing (NLP) auprès du grand public et, par voie de conséquence, des cybercriminels qui les ont détournés à des fins malveillantes.
L'intégration de l'IA dans les modes opératoires des cybercriminels pourrait provoquer une inflation du volume des intrusions, facilitées par des outils d'automatisation et d'optimisation pilotés par IA.

Les organisations ciblées doivent désormais composer avec des menaces polymorphes, adaptatives, et en constante évolution, bien qu'étant encore limitées par les contraintes techniques de développement des modèles.

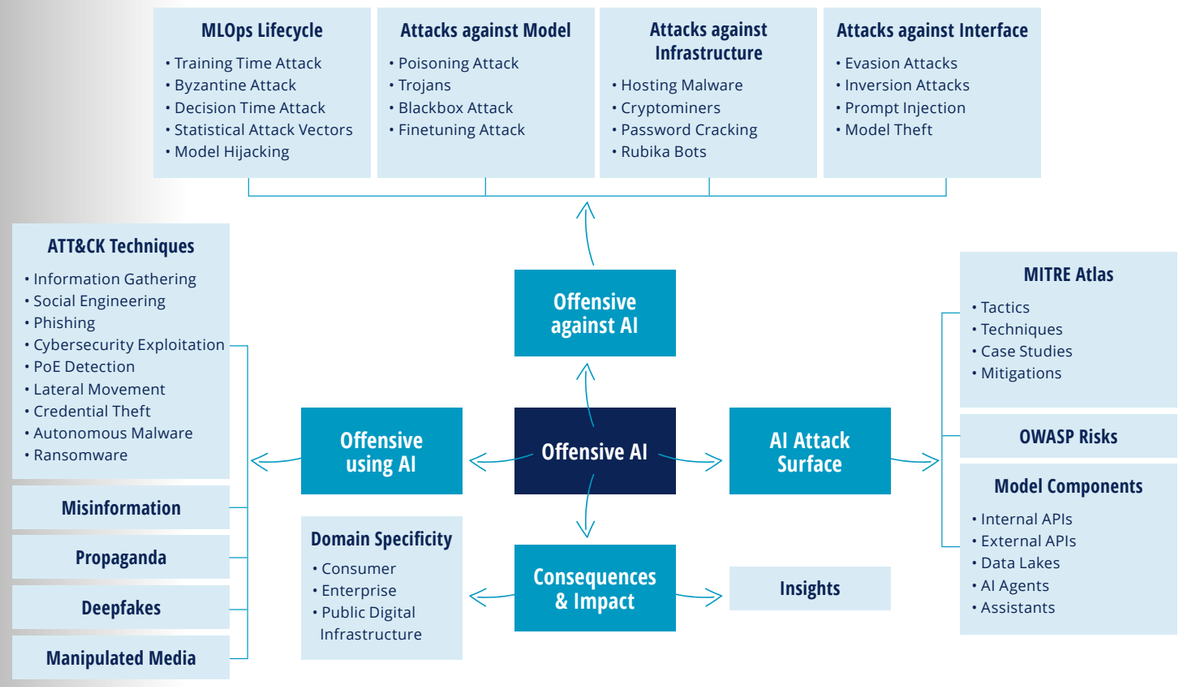
En partenariat avec OpenAI, les chercheurs de Microsoft ont annoncé le 14 février 2024 que des groupes APT liés à la Russie, à l'Iran, à la Corée du Nord et à la Chine avaient recours aux LLM (Large Language Model) dans le cadre de leurs opérations[1]. Ces groupes employaient les outils d'OpenAI pour collecter des informations en sources ouvertes, élaborer des campagnes de phishing, traduire des documents d'intérêt, détecter des erreurs de programmation et automatiser des tâches courantes.

Comprendre les leviers de cette transformation suppose d'analyser la façon dont sont construits les modèles d'IA les plus utilisés dans ce contexte, en particulier les LLM. Leur création s'articule en plusieurs étapes clefs : la collecte massive de données textuelles, leur prétraitement pour éliminer les biais et les erreurs, l'entraînement du modèle via des algorithmes d'apprentissage profond, puis le fine-tuning qui permet d'adapter le modèle à des tâches précises. La phase de déploiement s'accompagne finalement d'une supervision continue couplée à des mécanismes de retour d'expérience pensés pour améliorer la robustesse et la pertinence du modèle face à de nouveaux usages ou à des tentatives de contournement malveillantes. Malgré ces dispositifs de protection, les LLM demeurent exposés à des attaques, qu'il s'agisse d'injection de prompts, d'empoisonnement des données ou d'exploitation de failles dans leur processus de développement, ces dernières pouvant porter atteinte à leur fiabilité et à leur sécurité, limitant par conséquent leur adoption généralisée.

L'analyse développée dans cet article s'appuie sur le modèle de la kill chain cyber, un cadre méthodologique permettant de décomposer les différentes étapes d'une attaque informatique, depuis la phase de reconnaissance jusqu'à l'exfiltration des données ou la compromission prolongée du système ciblé[2].

L'usage offensif de l'intelligence artificielle bouleverse aujourd'hui chacune de ces étapes : automatisation de la phase de renseignement, production de campagnes de phishing sur mesure, développement de malware polymorphes, ou encore utilisation de deepfakes pour tromper les contrôles humains.

Cette dynamique se manifeste dès la phase de reconnaissance : traditionnellement constituée de recherches manuelles chronophages, elle connaît désormais une mutation profonde grâce à l'apport de l'IA.



Vue d'ensemble de l'IA appliquée aux méthodes offensives. Source : Arxiv[3]

Glossaire

Adversarial Attack
Technique visant à tromper un modèle d'Intelligence Artificielle (IA) ou de Machine Learning (ML) en lui fournissant des entrées spécifiquement conçues pour provoquer des erreurs de classification ou des comportements inattendus. Ces attaques exploitent les vulnérabilités des modèles, notamment des réseaux neuronaux profonds, en modifiant subtilement les données d'entrée pour induire des prédictions erronées.

Adversarial Learning
Domaine de la sécurité de l'IA qui étudie la résistance des modèles face aux attaques adversariales. Il s'agit de comprendre comment des acteurs malveillants peuvent manipuler les données d'entrée ou d'entraînement pour biaiser ou tromper un modèle, et de développer des contre-mesures pour renforcer la robustesse des systèmes d'IA.

Classification
Dans le contexte de l'intelligence artificielle (IA), la classification désigne une méthode d'apprentissage automatique consistant à attribuer une catégorie ou une classe à une donnée d'entrée selon ses caractéristiques.

Deepfake
Contenu audio, vidéo ou image généré par IA, imitant de manière réaliste une personne réelle. Les deepfakes sont utilisés pour des fraudes, des manipulations ou des attaques d'ingénierie sociale sophistiquées.

Empoisonnement de modèles *(Data Poisoning/Model Poisoning)*
Type d'attaque adversariale visant à insérer des données corrompues ou biaisées dans le jeu d'entraînement d'un modèle d'IA ou de ML. L'objectif est de manipuler le comportement du modèle, en induisant des prédictions erronées ou des failles de sécurité.

Fine-tuning
Processus d'ajustement d'un modèle d'IA pré-entraîné sur un large corpus, pour le spécialiser sur une tâche ou un domaine particulier, en utilisant un jeu de données plus restreint et spécifique.

GAN *(Generative Adversarial Network / Réseau Génératif Antagoniste)*
Architecture de deep learning composée de deux réseaux neuronaux antagonistes : un générateur, qui crée de nouvelles données et un discriminateur, qui tente de distinguer les données générées des données réelles. Les GAN sont utilisés pour produire des images, des sons ou du texte réalistes et sont à la base de nombreuses applications de deepfake.

GenAI *(Intelligence Artificielle Générative)*
Ensemble des technologies d'IA capables de générer de nouveaux contenus (texte, images, audio, vidéo) de manière autonome, à partir de modèles entraînés sur de vastes ensembles de données. GenAI inclut les LLM (Large Language Models), les GAN et d'autres architectures génératives. Il est utilisé aussi bien pour des usages légitimes que malveillants.

Groupe APT *(Advanced Persistent Threat)*
Ensemble structuré d'attaquants, souvent soutenus par des États ou des organisations criminelles, menant des opérations cyber offensives avancées, ciblées et persistantes.

Kill Chain
Modèle décrivant les étapes successives d'une cyberattaque, de la reconnaissance initiale à l'impact final. Il permet d'analyser et de structurer les différentes phases d'une intrusion pour mieux comprendre le mode opératoire des attaquants.

Large Language Model *(LLM)*
Modèle d'IA de grande taille, entraîné sur d'immenses volumes de textes, capable de comprendre, générer et manipuler du langage naturel. Les LLM sont utilisés pour la génération de texte, l'analyse sémantique, la traduction, etc.

Malware polymorphe
Logiciel malveillant qui modifie son code ou sa structure à chaque infection, afin d'échapper aux systèmes de détection basés sur des signatures statiques.

MLOps *(Machine Learning Operations)*
Ensemble des pratiques visant à industrialiser, déployer, superviser et maintenir les modèles de machine learning en production, tout en assurant leur sécurité et leur robustesse.

NLP *(Natural Language Processing / Traitement Automatisé du Langage Naturel)*
Branche de l'IA qui permet aux ordinateurs d'analyser, de comprendre et de générer le langage humain. Le NLP est à la base des assistants vocaux, des traducteurs automatiques et des LLM.

Prompt
Instruction ou question formulée par un utilisateur pour guider une intelligence artificielle dans la génération d'une réponse ou d'un contenu spécifique.

Une phase de reconnaissance automatisée

La collecte d'informations représente une phase préliminaire au cours de laquelle les attaquants rassemblent des renseignements sur leurs cibles pour com-prendre l'environnement, identifier les vulnérabilités potentielles et adapter leurs stratégies d'attaque.

Cette phase comprend deux approches principales : la collecte active, impliquant une interaction directe avec les systèmes cibles via des outils comme Nmap pour les scans de réseau, et la collecte passive, plus dis-crète, reposant sur l'analyse de données publiquement accessibles (OSINT) comme les bases WHOIS, les moteurs de recherche et les médias sociaux.

Collecte de données et cartographie des données

L'IA a introduit de nouvelles approches dans la phase de reconnaissance et de collecte de données, en facilitant l'automatisation et l'analyse des informations. Jusqu'à l'intégration de systèmes intelligents au sein du processus, cette étape reposait principalement sur des méthodes manuelles ou semi-automatisées[4]. L'IA offre désormais des capacités avancées de traitement des données collectées, permettant une cartographie précise et une visualisation globale de l'entreprise ciblée, fournissant ainsi aux auditeurs de sécurité ainsi qu'aux potentiels attaquants une vue d'ensemble structurée de leur cible.

Les outils légitimes, enrichis par l'apport de modèles d'IA, offrent aujourd'hui des fonction-nalités avancées susceptibles d'être exploitées aussi bien par des professionnels de la cybersécurité que dans le cadre d'un usage détourné, par des acteurs de la menace. C'est le cas de Nmap, initialement conçu comme un scanner réseau pour administrateurs système, permettant de découvrir les hôtes et services sur un réseau informatique[5]. Le projet nmap.ai améliore les fonctionnalités classiques de Nmap en y intégrant une couche d'IA analysant automatiquement les résultats techniques du scan[6]. Grâce à l'utilisation d'un modèle GPT-3.5, nmap.ai transforme les sorties brutes de Nmap en points clefs simplifiés et fournit des recommandations de sécurité qui, détournées de leur usage initial, peuvent être exploitées par les attaquants.

L'émergence de bots IA dédiés au scraping de plateformes comme LinkedIn offre de nouvelles perspectives dont certains groupes cybercriminels pourraient bénéficier. Une étude réalisée par l'Université de Stanford en 2024 démontre que l'utilisation de scrapers ayant recours à l'IA permet aujourd'hui d'extraire et de structurer les informations de plus de 100 000 profils LinkedIn en moins de 24 heures, là où une opération manuelle équivalente aurait nécessité un travail humain autrement plus conséquent[7].

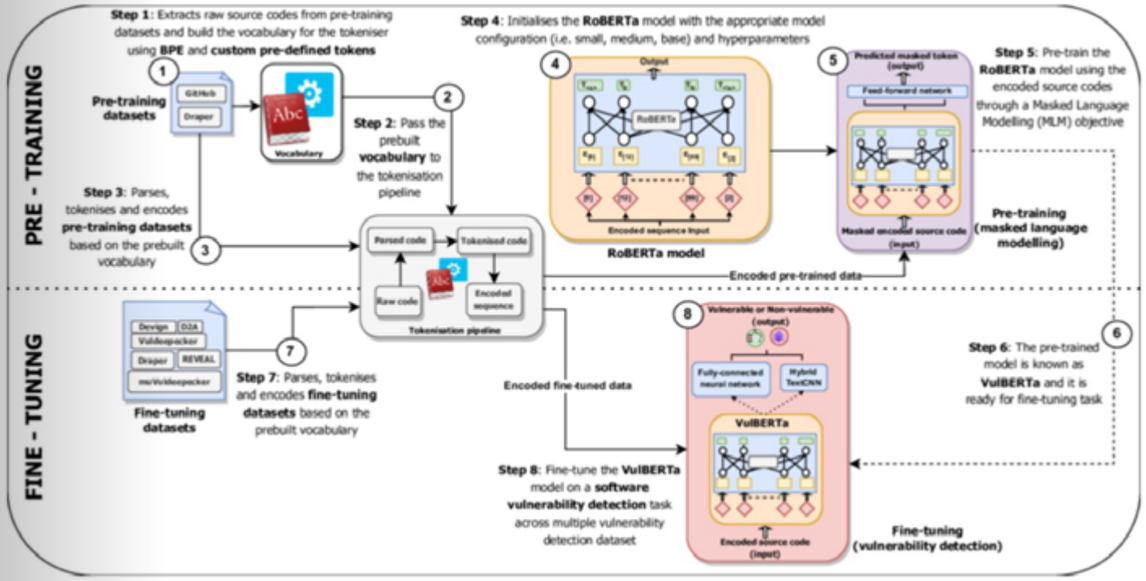
Les données collectées permettent ensuite aux attaquants de cartographier les stru-ctures d'entreprises ciblées et d'identifier des points d'entrée probants, notamment via le croisement de ces informations avec des bases de données issues de fuites ou des moteurs de recherche spécialisés dans l'indexation de ressources exposées sur internet (Shodan, Censys). Cette approche permet aux attaquants de dresser un portrait détaillé de leurs cibles, facilitant la préparation d'attaques sur mesure et la sélection des vecteurs d'intrusion les plus pertinents.

Reconnaissance en vulnérabilité

L'intégration de l'IA démultiplie également l'efficacité des outils utilisés dans la détection des vulnérabilités affectant les systèmes ciblés. De manière similaire, cette étape reposait traditionnellement sur des analyses manuelles ou semi-automatisées via des scripts.
Désormais, des systèmes spécifiquement entraînés à l'analyse des descriptions de vul-nérabilités informatiques tels que VulBERTA sont capables de traiter des milliers de documentations techniques et d'identifier sans intervention humaine les brèches critiques propres à chaque organisation, tout en recommandant des stratégies d'exploitation adaptées à l'infra-structure technologique de la victime[8]. Utilisé en parallèle d'outils réalisant des scans massifs, VulBERTA permet de croiser les données collectées avec une analyse sémantique approfondie du code source ciblé. Selon l'étude publiée par l'ETH Zurich à l'origine du logiciel, VulBERTA a permis d'augmenter de 35 % la rapidité de détection des vulnérabilités exploitables par rapport aux méthodes traditionnelles[9].

Au-delà de la simple identification, l'IA corrèle les vulnérabilités repérées avec les renseigne-ments collectés durant la phase de recon-naissance afin de cibler en priorité les failles réellement présentes dans l'environnement technique de la victime, puis interroge auto-matiquement les bases de données de vulnérabilités pour identifier les failles non corrigées correspondantes.

Système d'entraînement de VulBERTA
Source : ResearchGate[10]

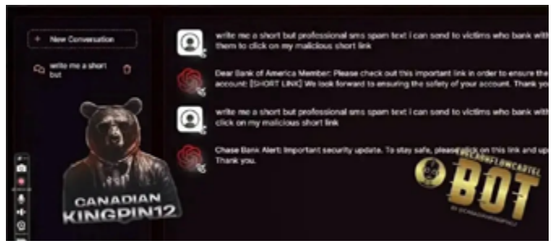


Accès initial et social engineering

Deux tendances majeures se dégagent de l'implémentation de l'IA au sein des techniques d'ingénierie sociale : le phishing hyper person-nalisé et l'utilisation de deepfakes audio/vidéo désormais accessibles à moindre coût. ■

Phishing

Les modèles de traitement du langage naturel tels que ChatGPT, ou l'un de ses dérivés malveillants tels que FraudGPT, sont désormais employés pour générer des messages personnalisés, adaptés au contexte et au profil psychologique des cibles[12]. Les indices traditionnels utilisés pour détecter les emails de phishing tels que les erreurs grammaticales et les tons inadéquats employés par les attaquants sont aujourd'hui des écueils largement atténués par l'intégration de l'IA dans le processus de rédaction. Le fait que ces emails se déclinent en plusieurs variantes permet également de mettre à mal les règles de détection anti-phishing n'incluant pas de mesures anti-IA[13]. Une étude menée par l'Université de San Antonio révèle que les taux d'efficacité des emails générés par IA sont très proches, voire supérieurs à ceux des emails écrits par des humains et reproductibles à volonté[14]. Une seconde étude démontre qu'au premier trimestre 2025, 82,6% des emails de phishing analysés utilisaient l'IA, avec une augmentation de 17,3% du volume global par rapport au semestre précédent[15].



Génération de SMS de phishing à l'aide de FraudGPT
Source : Netenrich[12]

Les campagnes de phishing s'appuient fréquemment sur des événements d'actualité dans le but de renforcer l'illusion de légitimité. Dans le cadre des Jeux Olympiques s'étant tenus à Paris au cours de l'été 2024, les chercheurs de Trend Micro ont analysé une campagne incitant les potentielles victimes à investir dans des cryptomonnaies frauduleuses[16].

Les attaquants ont bâti cette campagne autour de sites web frauduleux, agrémentés d'images produites par intelligence artificielle pour accroître leur crédibilité. Ce recours à l'ingénierie sociale et à la contextualisation des attaques se retrouve également dans d'autres formes de fraudes.

Le compte X (Twitter) vérifié d'Olympics_Solana diffusant son site frauduleux. Source : Trend Micro[16]

Les attaques de type *Business Email Compromise* (BEC) illustrent parfaitement cette évolution. Dans ce type de fraude, les cybercriminels usurpent l'identité d'un dirigeant ou d'un partenaire commercial pour manipuler un employé en créant un sentiment d'urgence ou de confidentialité pour le pousser à procéder à des virements frauduleux ou à l'envoi de données sensibles vers des comptes contrôlés par l'attaquant. Ces mails, générés automatiquement grâce à des modèles d'IA générative en fonction des informations collectées lors de la phase de reconnaissance, constituent aujourd'hui l'un des scénarios de compromission présentant le plus fort taux de réussite[14]. Selon les données du FBI, cette menace a causé des pertes financières colossales atteignant 55,5 milliards de dollars entre octobre 2013 et décembre 2023, avec plus de 305 033 cas signalés[17]. L'essor de l'IA dans la création de campagnes de phishing a contribué à une augmentation rapide du nombre d'attaques, celles-ci devenant de plus en plus crédibles et sophistiquées[18].

L'intelligence artificielle facilite désormais l'orchestration d'attaques multicanales sophistiquées combinant phishing, vishing (voice phishing) et deepfake. Cette méthode d'attaque particulièrement efficace associe l'envoi initial d'un email frauduleux suivi par un contact téléphonique lors duquel les attaquants emploient des voix générées artificiellement pour manipuler leurs victimes. Les chercheurs de Sophos ont observé cette méthode lors de 15 incidents de sécurité entre novembre 2024 et la mi-janvier 2025. Les opérateurs du ransomware 3AM ont d'abord diffusé massivement des emails sur une courte période, puis contacté les utilisateurs ciblés par téléphone pour les informer d'un incident de sécurité et les accompagner dans la remédiation, facilitant ainsi la prise de contrôle des systèmes visés[19].

Cette convergence technologique permet aux attaquants de renforcer la crédibilité de leurs tentatives d'ingénierie sociale en créant une cohérence entre les différents vecteurs d'attaque. L'IA coordonne ces différentes phases, adaptant le discours téléphonique en fonction des réactions au message initial, créant ainsi une expérience d'ingénierie sociale fluide et convaincante.

Deepfakes

Ces technologies permettent de créer des imitations quasi parfaites de voix et de visages, rendant les attaques d'ingénierie sociale considérablement plus convaincantes[20].

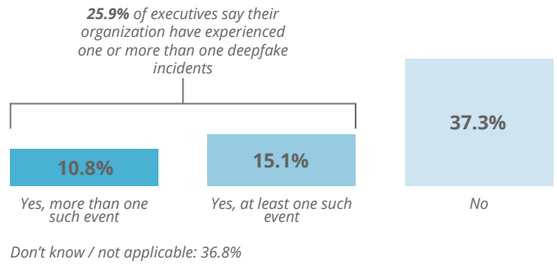
En février 2024, l'entreprise d'ingénierie britannique Arup a perdu plus de 25 millions de dollars suite à une fraude sophistiquée impliquant l'usage d'un deepfake au cours d'une vidéoconférence au sein de laquelle le directeur financier et d'autres collègues étaient générés par IA[21]. Initialement méfiant face à un message demandant une transaction secrète, l'employé a finalement été convaincu par le réalisme des participants virtuels à l'appel vidéo. Loin d'être isolé, cet incident s'inscrit dans un contexte global marqué par la multiplication des attaques ayant recours à l'IA pour contourner les dispositifs de sécurité, notamment à travers la compromission de données biométriques destinées à alimenter des deepfakes.

Le 15 février 2024, Group-IB a par exemple mis au jour un nouveau Trojan bancaire baptisé GoldPickaxe.iOS, ciblant les utilisateurs iOS et étant capable de collecter les données de reconnaissance faciale de ses victimes[22]. Les acteurs de la menace exploitent ensuite ces données biométriques pour générer des deepfakes via des outils de face-swapping leur permettant de remplacer leur visage par celui des victimes et accéder à des informations sensibles ou des comptes bancaires[23].

Cette capacité à générer des deepfakes à partir de données biométriques volées ouvre désormais la voie à des attaques encore plus sophistiquées, notamment grâce à l'émergence de technologies de deepfake en temps réel.

En l'état, la réalisation d'un deepfake live est encore réservée aux acteurs sophistiqués disposant d'une puissance de calcul conséquente. Des outils tels que *"Deep-Live-Cam"* permettent aujourd'hui d'en réaliser mais sont encore limités par leurs performances. Les caractéristiques physiques poussées, à l'image des lunettes ou de la pilosité compliquent significativement la réalisation[24]. À ce jour, l'attaque la plus accessible impliquant un deepfake demeure la réalisation d'une vidéo préalable ou d'un message audio pouvant être envoyé à la victime. Il est néanmoins plausible que les technologies de deepfake live se développement et se démocratisent dans les prochaines années, dans le sillon des progrès réalisés par l'IA.

La menace demeure significative : d'après un sondage réalisé par Deloitte, 25,9 % des dirigeants interrogés ont révélé que leur organisation avait connu un ou plusieurs incidents impliquant des deepfake ciblant des données financières et comptables au cours des 12 mois précédents[25].



En réponse à la question : « During the past 12 months, did your organization experience any deepfake incidents targeting financial andaccounting data? ». Source : Deloitte[25]

Exploitation, déploiement, obfuscation

Génération de code malveillant par IA

L'intelligence artificielle générative (GenAI) permet de générer des solutions ou des contenus inédits, pour autant l'accessibilité des modèles génératifs n'a pas produit d'explosion du nombre de nouveaux malware dans la nature[26]. Les systèmes actuels de GenAI ne disposent pas des capacités spécifiques pour créer de manière indépendante des malware opérationnels, et requièrent de fait une intervention humaine afin de corriger et diriger le processus de création[27]. L'efficacité du modèle dépendant de ses données d'entraînement, la qualité de la génération en pâtit car les exemples de malware sophistiqués sont rarement accessibles publiquement.
En outre, pour qu'un système d'intelligence artificielle générative puisse concevoir un malware sophistiqué fonctionnel, il doit non seulement être apte à produire du code robuste mais également disposer d'informations relatives à des failles exploitables concernant les systèmes ciblés. Les exigences poussées liées à la génération de malware limitent donc son utilisation à un cercle restreint d'acteurs possédant ces capacités et informations[28]. Les cybercriminels utilisent néanmoins la GenAI pour créer des malware ou des scripts simples, améliorer les compétences des malware existants, ou en créer des variantes.

```
$ie = New-Object -ComObject "InternetExplorer.Application"
$ie.visible = $false
$ie.navigate("https://www.example.com/login")
while ($ie.Busy -eq $true) { Start-Sleep -Milliseconds 100 }
$usernameField = $ie.Document.getElementByID("username")
$usernameField.value = "username"
$passwordField = $ie.Document.getElementByID("password")
$passwordField.value = "password"
$submitButton = $ie.Document.getElementByID("submit")
$submitButton.click()
Start-Sleep -Seconds 5
$cookie = $ie.Document.cookie
$cookie | Out-File -FilePath "C:\Path\To\WebSessionCookie.txt"
```

Prompt demandant à ChatGPT de créer du code à partir d'une technique MITRE ATT&CK. Source : Trend Micro[26]

Des exemples de génération de malware dans la nature ont déjà été observés par des chercheurs. HP Wolf Security a mis au jour une campagne ciblant des utilisateurs francophones au cours de laquelle des JavaScript et VBScript malveillant ont été rédigés à l'aide d'une IA générative[29]. La structure, la présence de commentaires détaillant chaque ligne de code, ainsi que le choix de noms de fonctions a permis aux chercheurs de déterminer que de la GenAI avait été utilisée par les cybercriminels afin de développer ces scripts. Cette campagne visait à infecter les victimes avec AsyncRAT, un infostealer capable d'enregistrer les frappes du clavier et l'écran de l'utilisateur. Si ce cas met en lumière la manière dont l'IA générative abaisse les barrières techniques lors de la conception d'attaques, il convient de souligner que cette technologie demeure à un stade précoce de développement, et que ses applications futures pourraient démultiplier l'ampleur et la complexité des menaces.
OpenAI a annoncé en mai 2025 mettre à disposition du public son agent d'ingénierie logicielle Codex, capable d'écrire, comprendre et corriger du code informatique. Entraîné via un apprentissage par renforcement sur des tâches de programmation réelles afin de produire du code conforme aux standards humains, cet outil pourrait être détourné de ses usages légaux afin de servir les intérêts de cybercriminels. Comme observé avec les versions jailbreakées de ChatGPT, les restrictions de sécurité implémentées par OpenAI pourraient très probablement être contournées et les capacités de Codex seraient alors utilisées à des fins malveillantes[30].
Le Center for Emerging Technology and Security a publié en juillet 2024 une analyse des potentiels usages de l'IA générative dans le cadre de la génération de malware [28]. Selon les chercheurs, l'IA générative pourrait permettre à des agents malveillants de modifier leur code à l'exécution, voire de se réécrire entièrement afin d'échapper à la détection. Ces agents seraient également capables de rédiger eux-mêmes des payloads ou de créer de nouveaux outils pour surmonter des obstacles inédits, adaptant ainsi leurs tactiques en temps réel et opérant de manière autonome, avec un besoin réduit de supervision humaine. La coopération entre plusieurs agents offrirait une persistance renforcée, chaque entité apprenant et s'adaptant continuellement à son environnement.
Enfin, leur capacité à raisonner sur leur contexte et à ajuster leurs communications pour se fondre dans le trafic légitime conférerait à ces malware une furtivité et une persistance sans précédent.

Techniques d'obfuscation, malware polymorphiques

L'IA offre la possibilité de multiplier les variantes de malware et de scripts, compliquant ainsi la tâche des systèmes de détection fondés sur les règles YARA. Recorded Future l'ont démontré en adaptant le code de l'infostealer STEELHOOK, utilisé par le groupe d'attaquants APT28 attribué à la Russie, pour contourner les règles YARA en étudiant précisément leurs modes de détection[31].
Une équipe de chercheurs de l'Indian Institute of Technology Madras a conduit une analyse approfondie de l'usage de l'IA dans le cadre d'actions cyber offensives[32]. Le papier analyse comment cette nouvelle technologie pourrait obliger la cybersécurité à repenser son arsenal de détection, en distinguant trois évolutions majeures rattachées à des phases distinctes de la killchain :

- **L'accès initial** *(Point of Entry)* : d'un point de vue défensif, cette phase initiale est essentielle pour identifier et bloquer les menaces dès leur apparition. L'article présente Deeplocker, développé en tant que proof-of-concept par IBM Research, illustrant l'utilisation de l'IA pour créer des malware hautement furtifs[33].
  L'innovation clef réside dans l'intégration d'un réseau de neurones profond (DNN) pour dissimuler une charge malveillante au sein d'une application légitime. La payload est dissimulée grâce à l'IA, son déchiffrement dépendant d'une clef générée dynamiquement par le DNN. Aucune clef n'étant stockée dans le code, la rétro-ingénierie devient inefficace. La payload demeure inactive jusqu'à ce que le malware détecte une combinaison précise d'attributs (biométriques, de configuration). DeepLocker matérialise le risque des attaques IA embarquées, où l'IA sert à la fois de verrou (dissimulation) et de clef (activation ciblée se basant sur de multiples variables), nécessitant une évolution des paradigmes de détection.

- **L'évasion :** cette phase se concentre sur les techniques utilisées par les cyberattaquants pour échapper à la détection au sein d'un réseau après l'avoir pénétré. L'IA, et notamment l'aspect d'apprentissage par renforcement reposant sur le principe d'essai et erreur, permet au malware de modifier sa structure interne afin de contourner la détection statique lors de phases de test contre des antivirus. Des agents IA peuvent interagir dynamiquement avec des échantillons de malware pour appliquer des transformations binaires qui préservent la fonctionnalité du code tout en contournant les systèmes de détection statique. Ces manipulations incluent par exemple l'ajout de caractères ou d'octets aléatoires, ou encore la suppression de signatures, permettant au malware de rester indétecté.

- **Prise de décision lors du déploiement de la charge :** le malware autonome se caractérise par son utilisation de l'IA afin de prendre des décisions éclairées et instantanées, conduisant à des cyberattaques autonomes[34]. Il est capable d'évaluer l'environnement et prendre des décisions avisées grâce à des instructions dynamiques et d'adapter ses décisions en temps réel en fonction des besoins.

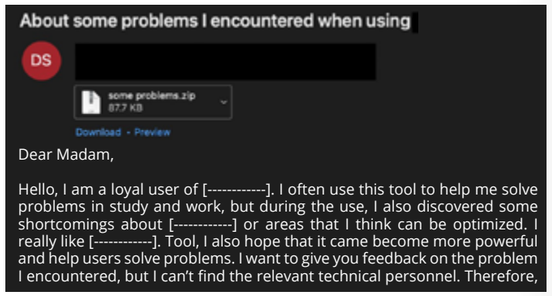
Cette nouvelle technologie de génération permet aux malware d'évoluer vers des formes polymorphes, générées dynamiquement pour échapper aux signatures des antivirus et s'adapter en temps réel à son environnement.
Chaque nouvelle itération d'un ransomware peut par exemple présenter des variations dans son code source, ses chaînes de caractères ou ses méthodes de chiffrement, ce qui rend inefficaces les bases de signatures classiques. Cette automatisation permet aux attaquants de lancer des campagnes massives, tout en réduisant le risque d'être détectés par les outils de sécurité statiques.

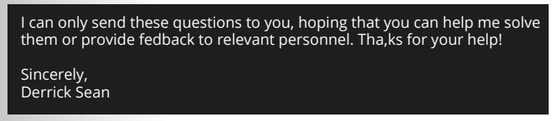
De la sécurité des modèles d'IA

Le MLOps, ou Machine Learning Operations, est une fonction clef de l'ingénierie du machine learning englobant tous les processus nécessaires pour mettre les modèles d'IA en production et les maintenir. Le cycle de vie MLOps comporte plusieurs phases essentielles : le développement, la phase de test, le déploiement, la supervision et enfin les retours d'information. Toutes ces phases sont susceptibles d'être victimes d'attaques de différentes natures[35].

Les outils basés sur l'IA exploités comme vecteur d'accès initial

À mesure que l'intelligence artificielle s'implante dans les environnements professionnels, les outils qui en découlent font désormais l'objet d'attaques visant à les compromettre afin de s'introduire dans les systèmes d'information liés. Ces offensives ont pour principal objectif de dérober des informations sensibles, d'autant plus que les systèmes d'IA eux-mêmes présentent désormais des vulnérabilités documentées et activement exploitées, telles que la vulnérabilité référencée CVE-2025-31693 au sein de Drupal AI, permettant de compromettre l'intégrité ou la confidentialité des données manipulées par ces systèmes[36].
Plus encore, le 24 avril 2024, des chercheurs de Cornell Tech, de l'Israel Institute of Technology et d'Intuit ont présenté Morris II, un ver informatique inédit exploitant les services d'intelligence artificielle générative pour se propager et mener des actions malveillantes[39]. Construit autour d'un prompt autoréplicateur, Morris II exploite la capacité des LLM à générer et propager automatiquement des instructions malveillantes, à la manière d'une injection SQL. Ce ver est capable de s'auto-répliquer et de se diffuser dans des écosystèmes d'agents GenAI interconnectés, notamment via des assistants de messagerie dotés d'IA, sans nécessiter d'interaction de la part de l'utilisateur (« zero-click »). Les chercheurs ont démontré que Morris II peut exfiltrer des données sensibles telles que noms, numéros de téléphone, informations bancaires ou numéros de sécurité sociale, et lancer des campagnes de spam, en exploitant les failles des assistants de messagerie alimentés par des modèles comme ChatGPT, Gemini ou LLaVA.
Des vulnérabilités référencées ciblant des outils intervenant dans l'étape de développement de modèles d'IA ont également commencé à faire leur apparition. C'est le cas de la vulnérabilité critique CVE-2025-3248 affectant Langflow, un outil open source populaire pour la création de workflows d'agents IA, permettant un attaquant distant non authentifié d'exécuter du code arbitraire sur le serveur cible[40][41].
Une exploitation réussie pourrait mettre en péril la confidentialité des modèles, des jeux de données, des flux d'agents IA en développement et de toute la chaîne d'approvisionnement logicielle. L'analyse de cette vulnérabilité souligne la nécessité de restreindre l'exposition des outils IA récents et de privilégier leur déploiement en environnement isolé, afin d'éviter tout risque d'empoisonnement de modèles.
Du fait des opportunités présentées par la compromission de la chaîne d'approvisionnement logicielle associée à l'IA, le secteur devient une cible privilégiée pour des cyberattaques cherchant à perturber les processus scientifiques[37]. Le 16 mai 2024, Proofpoint a publié un rapport détaillant la distribution du malware SugarGh0st RAT par des acteurs malveillants visant spécifiquement des organisations américaines engagées dans la recherche et le développement en intelligence artificielle, dont des universités, des entreprises privées et des agences gouvernementales[38].





Mail frauduleux distribuant SugarGh0st RAT
Source : Proofpoint[38]

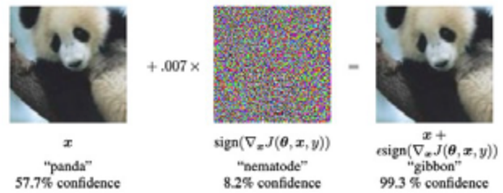
Cette campagne illustre une tendance croissante : les entités impliquées dans la recherche sur l'IA représentent désormais un enjeu majeur pour des opérations de cyberespionnage, les attaquants cherchant à obtenir des renseignements sensibles et non publics sur les avancées technologiques en intelligence artificielle. Elles pourraient en outre permettre à des attaquants d'empoisonner le code et / ou les données sur lesquelles sont entraînés les modèles, afin de compromettre en cascade l'ensemble de ses utilisateurs.

Des modèles vulnérables à l'adversarial learning

L'apprentissage adversarial *(adversarial machine learning)* est un domaine à l'intersection de la sécurité informatique et de l'intelligence artificielle. Il étudie les vulnérabilités des modèles d'apprentissage automatique face à des attaques malveillantes. Dans les faits, un acteur de la menace peut manipuler des données d'entrée afin de contourner la classification ou de mettre en évidence les limites de décision du modèle LLM attaqué[42].
Cette pratique soulève des préoccupations de sécurité, ces techniques pouvant être utilisées pour attaquer des systèmes d'apprentissage automatiques, et ce même si l'acteur de la menace n'a pas accès au code source.
Le 27 mai 2025, l'entreprise américaine Meta possédant notamment Facebook et Instagram a décidé que toutes les publications, photos, commentaires et interactions publiques des utilisateurs adultes de Facebook et Instagram en Europe seraient collectées pour entraîner ses modèles d'intelligence artificielle[43]. Ce choix expose l'entreprise à un risque accru de compromission de ses modèles par des attaques d'empoisonnement collaboratif, où des acteurs malveillants pourraient volontairement injecter, à grande échelle, des contenus biaisés, trompeurs ou malveillants dans les données publiques collectées.

L'empoisonnement de modèles collaboratifs consiste à manipuler les données d'entraînement de LLMs afin d'induire des prédictions biaisées ou inexactes au modèle. L'adversaire cherche à orienter sélectivement la donnée de sortie du modèle, l'objectif étant d'obtenir des prédictions erronées pour certaines entrées tout en maintenant la précision pour d'autres[44]. Plusieurs schémas d'attaques sont envisagés, dont voici deux exemples :

- **Label Contamination Attacks :** une étiquette (label) d'entraînement désigne l'annotation ou la valeur cible associée à chaque exemple lors de l'apprentissage supervisé.[45] Concernant des tâches de classification, l'étiquette correspond à la catégorie attribuée à un texte, par exemple positif, neutre ou négatif. L'enjeu sera de manipuler ces étiquettes, par exemple en associant une étiquette « négatif » à un avis manifestement positif, afin que le modèle apprenne de mauvaises associations entre les textes et leurs catégories.
- **Decision Time Attack :** cette attaque consiste à modifier de façon ciblée les caractéristiques des données d'entrée pour tromper un modèle d'IA lors de la prédiction du résultat[46]. Les effets concrets de ces attaques sont très variés : elles peuvent provoquer des erreurs de reconnaissance simples, par exemple lorsqu'un modèle confond un panda avec un gibbon, mais aussi permettre des actions plus dangereuses, comme manipuler le comportement de voitures autonomes ou échapper à des systèmes de sécurité basés sur la vidéo, l'audio ou les empreintes digitales[47]. À titre d'exemple, des chercheurs ont piégé l'Autopilot d'une Tesla Model X en projetant brièvement un panneau stop sur un écran publicitaire, provoquant l'arrêt du véhicule[48].



Une image de panda à laquelle on ajoute un bruit imperceptible, faisant croire au modèle qu'il s'agit d'un gibbon. Source : Cybernews[47]

Les méthodes d'attaques adversariales sont applicables aux LLMs collaboratifs comme ChatGPT, notamment lors des phases de fine-tuning ou via des mécanismes de retours d'expérience de la part des utilisateurs. Toutefois, l'efficacité et la faisabilité de ces attaques dépendent fortement des politiques de gestion des données, des contrôles de sécurité, et de la vigilance des opérateurs de ces plateformes. Les attaques adversariales sont théoriquement plus efficaces contre des développements de modèles open-source, la structure du modèle et l'accès aux données d'entraînement étant nécessaires pour parvenir à créer de réels impacts. Si la phase de fine-tuning a pour objectif de rendre les modèles open-source moins exposés aux attaques, les systèmes développés dans un environnement cloisonné restent moins susceptibles d'être compromis.

Conclusion

L'intégration de l'intelligence artificielle dans les modes opératoires malveillants est devenue un élément central d'une part toujours croissante des stratégies d'attaque actuellement observées. Pour autant, leur potentiel demeure restreint par des barrières techniques, bien que celles-ci reculent avec l'amélioration continue des modèles.

L'IA permet néanmoins d'industrialiser des tâches qui nécessitaient une expertise humaine. Les phases de reconnaissance, de collecte d'informations et de cartographie des cibles sont désormais accélérées par des outils bénéficiant des apports de l'IA, capables de traiter et de structurer des volumes massifs de données, offrant ainsi aux acteurs malveillants une représentation globale des vecteurs de compromission exploitables de leurs cibles. Les techniques d'ingénierie sociale ont également bénéficié des progrès rapides de l'IA. Les campagnes de phishing ciblées et automatisées sont désormais modulables à volonté, adaptant leurs discours en fonction des données collectées lors de la phase de reconnaissance. Cette sophistication s'étend aux attaques multicanales, où phishing, vishing et deepfakes vocaux sont orchestrés de manière cohérente, renforçant la crédibilité des scénarios d'ingénierie sociale.

La conception de logiciels malveillants sophistiqués impose de concilier en permanence des exigences de furtivité, de persistance et performance. Or, ces ajustements stratégiques exigent une forme de raisonnement tactique et une flexibilité que les LLM ne possèdent pas à ce jour. Il en va de même pour la capacité de la GenAI à détecter et exploiter des failles de manière autonome[49]. Parallèlement, la multiplication des vulnérabilités ciblant les modèles d'intelligence artificielle et leurs outils de développement expose la recherche et les organisations à de nouveaux risques d'attaques sophistiquées, incluant le vol de données sensibles et la compromission de chaînes d'approvisionnement logicielles.

Les capacités actuelles de l'intelligence artificielle ne suffisent pas à instaurer un écosystème cybercriminel pleinement automatisé, capable de mener sans supervision chacune des phases de la kill chain. Si la tendance se dirige vers une plus grande automatisation, l'intervention humaine demeure requise à la fois pour entraîner, purifier et guider les modèles, mais aussi pour orienter l'IA à chaque étape de la kill chain, depuis la reconnaissance initiale jusqu'au déploiement et à l'adaptation des attaques en fonction des réactions de la cible.

À l'avenir, l'évolution des techniques d'Intelligence Artificielle pourrait toutefois bouleverser ce panorama.

Les progrès en matière d'auto-apprentissage, de génération automatique de code malveillant et d'adaptation dynamique aux contre-mesures laissent entrevoir la possibilité d'attaques plus autonomes, capables de s'auto-adapter sans intervention humaine directe. L'IA pourrait ainsi prendre en charge des décisions tactiques, optimiser les vecteurs d'attaque, voire gérer la résilience des opérations face à la détection ou à la neutralisation. Cette automatisation progressive accroîtrait la rapidité, la discrétion et la capacité d'innovation des attaquants, rendant les menaces plus difficiles à anticiper et à contrer.

À l'horizon 2025 et au-delà, l'intelligence artificielle devrait donc profondément transformer le paysage de la défense en cybersécurité, en réponse à la sophistication croissante des attaques automatisées. Les dispositifs d'analyse prédictive, fondés sur l'exploitation de vastes corpus de données historiques et le traitement en temps réel des flux d'information, pourraient permettre d'identifier précocement les vulnérabilités et d'anticiper l'émergence de menaces[50]. L'IA permettrait alors d'automatiser non seulement la détection des incidents, mais également les réponses aux attaques, en neutralisant rapidement les tentatives de compromission sans intervention humaine directe. L'une des options potentiellement envisagées par les organisations pourrait être d'intégrer massivement des architectures de type Zero Trust, où chaque accès serait validé en continu[51]. Les outils d'analyse comportementale, alimentés par l'IA, favoriseraient une surveillance proactive des endpoints et des flux de données, tout en isolant efficacement les menaces dès leur apparition.

Toutefois, comme ce papier a pu le démontrer, la généralisation de ces technologies ne s'effectuerait pas sans risques : l'IA pourrait elle-même devenir une cible privilégiée des attaquants, qui tenteraient de manipuler les modèles ou de contaminer les données d'entraînement pour contourner les défenses.
À l'avenir, la capacité des organisations à anticiper et à se prémunir contre les attaques adversariales s'imposera comme une priorité absolue, la robustesse des modèles d'intelligence artificielle constituant dès lors un enjeu stratégique central pour la cybersécurité de demain. ■

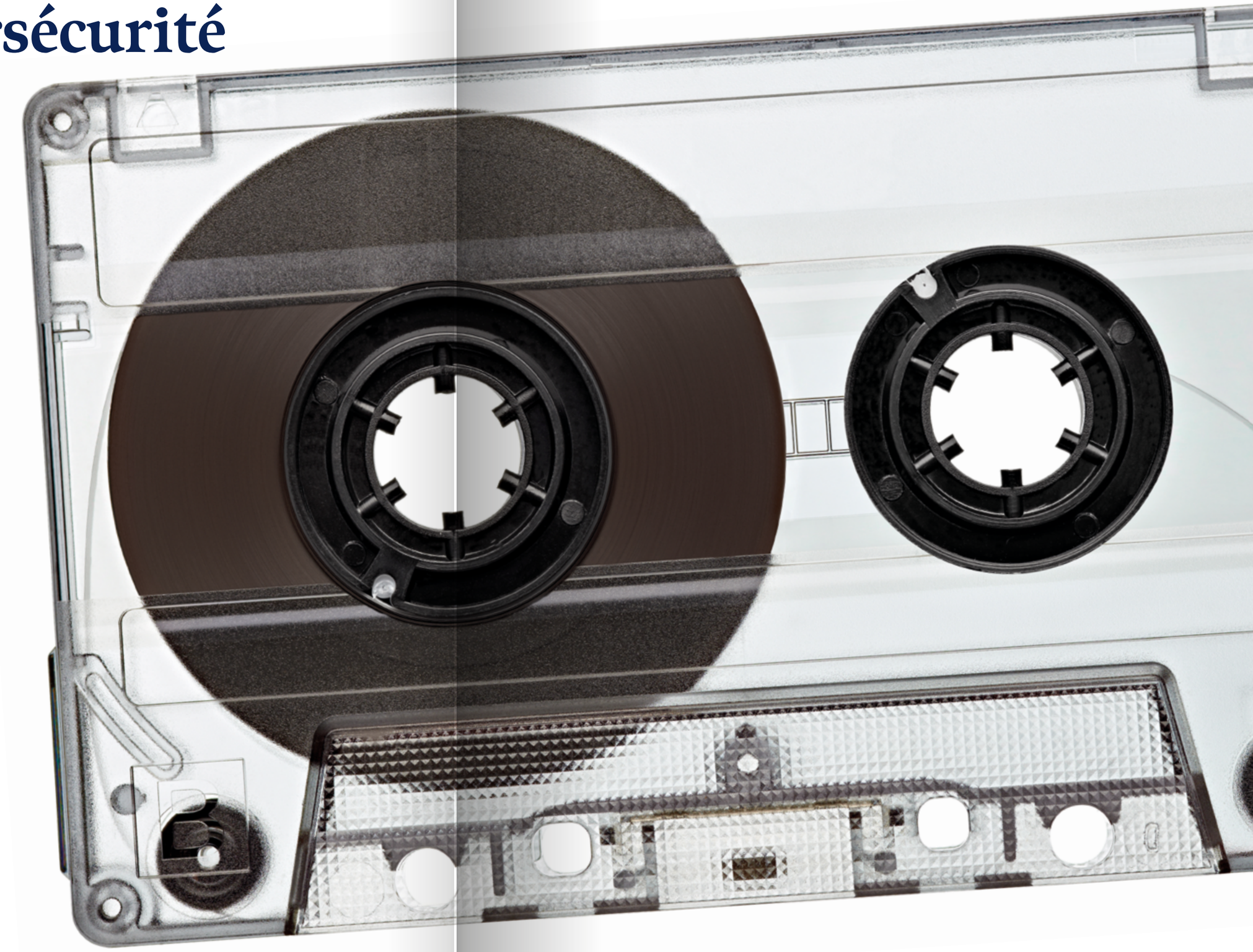
Bibliographie

[1] M. T. Intelligence, «Staying ahead of threat actors in the age of AI,» 2024. [En ligne] **https://www.microsoft.com/en-us/security/blog/2024/02/14/staying-ahead-of-threat-actors-in-the-age-of-ai/**
[2] «Qu'est-ce que la cyber-kill chain ?,» [En ligne] **https://www.microsoft.com/fr-fr/security/business/security-101/what-is-cyber-kill-chain**
[3] Arxiv, «A Survey on Offensive AI Within Cybersecurity,» 7 Octobre 2024. [En ligne] **https://arxiv.org/pdf/2410.03566**
[4] CMSWIRE, «LinkedIn Scraped by Bad Bots in Massive Scale Attack,» 2016. [En ligne] **https://www.cmswire.com/information-management/linkedin-scraped-by-bad-bots-in-massive-scale-attack/**
[5] WebAsha, «How Hackers Use AI for Reconnaissance : The Role of Artificial Intelligence in Cybersecurity Threats and Data Gathering,» 2025. [En ligne] **https://www.webasha.com/blog/how-hackers-use-ai-for-reconnaissance-the-role-of-artificial-intelligence-in-cybersecurity-threats-and-data-gathering**
[6] Ronantakizawa, «Github project, nmap.ai,» [En ligne] **https://github.com/ronantakizawa/nmap.ai**
[7] S. University, «Automated Social Media Reconnaissance in Modern Cyber Threats.,» 2024.
[8] S. M. Hazim Hanif, «VulBERTa:Simplified Source Code Pre-Training for Vulnerability Detection,» 2022. [En ligne] **https://arxiv.org/abs/2205.12424**
[9] E. Zurich, «Large Language Model for Vulnerability Detection: Emerging Results and Future Directions.,» 2024. [En ligne] **https://arxiv.org/abs/2401.15468**
[10] ResearchGate, «VulBERTa training pipeline. Steps are taken in order from 1 to 8.,» [En ligne] **https://www.researchgate.net/figure/ulBERTa-training-pipeline-Steps-are-taken-in-order-from-1-to-8_fig1_364069820**
[11] PortSwigger, «Burp AI,» [En ligne] **https://portswigger.net/burp/documentation/desktop/burp-ai**
[12] Netenrich, «FraudGPT: The Villain Avatar of ChatGPT,» 2023. [En ligne] **https://netenrich.com/blog/fraudgpt-the-villain-avatar-of-chatgpt**
[13] Darktrace, «Business Email Compromise (BEC) in the Age of AI,» 2024. [En ligne] **https://www.darktrace.com/blog/business-email-compromise-bec-in-the-age-of-ai**
[14] S. A. University, «Lateral Phishing With Large Language Models: A Large Organization Comparative Study,» 2025. [En ligne] **https://www.researchgate.net/publication/390288076_Lateral_Phishing_with_Large_Language_Models_A_Large_Organization_Comparative_Study**
[15] KnownBe4, «Phishing Threat Trends Report,» 2025. [En ligne] **https://www.knowbe4.com/hubfs/Phishing-Threat-Trends-2025_Report.pdf**
[16] T. Micro, «ICO Scams Leverage 2024 Olympics to Lure Victims, Use AI for Fake Sites,» 2024. [En ligne] **https://www.trendmicro.com/en_us/research/24/f/ico-scams-leverage-2024-olympics-to-lure-victims-use-ai-for-fake.html**
[17] FBI, «Business Email Compromise: The $55 Billion Scam,» 2024. Federal Bureau of Investigation, 2024, Business Email Compromise: The $55 Billion Scam (I-091124-PSA).
[18] SlashNext, « The State of Phishing 2024,» 2025. [En ligne] **https://slashnext.com/the-state-of-phishing-2024/**
[19] Sophos, «A familiar playbook with a twist: 3AM ransomware actors dropped virtual machine with vishing and Quick Assist,» 2025. [En ligne] **https://news.sophos.com/en-us/2025/05/20/a-familiar-playbook-with-a-twist-3am-ransomware-actors-dropped-virtual-machine-with-vishing-and-quick-assist/**
[20] Trustpair, « Deepfake : porte ouverte à la fraude en entreprise ?,» 2024. [En ligne] **https://trustpair.com/fr/blog/deepfake-porte-ouverte-a-la-fraude-en-entreprise/**
[21] C. World, « Finance worker pays out $25 million after video call with deepfake 'chief financial officer',» 2024. [En ligne] **https://edition.cnn.com/2024/02/04/asia/deepfake-cfo-scam-hong-kong-intl-hnk**
[22] Group-IB, «Face Off: Group-IB identifies first iOS trojan stealing facial recognition data,» 2024. [En ligne] **https://www.group-ib.com/blog/goldfactory-ios-trojan/**
[23] Toolify.ai, « Create Realistic Face Swaps with SimSwap,» 2024. [En ligne] **https://www.toolify.ai/ai-news/create-realistic-face-swaps-with-simswap-2749440**
[24] Hacksider, «Deep-Live-Cam,» 2023. [En ligne] **https://github.com/hacksider/Deep-Live-Cam**
[25] Deloitte, «Generative AI and the fight for trust,» 2024. [En ligne] **https://www2.deloitte.com/content/dam/Deloitte/us/Documents/Advisory/us-generative-ai-and-the-fight-for-trust.pdf**
[26] T. Micro, «A Closer Look at ChatGPT's Role in Automated Malware Creation,» 2023. [En ligne] **https://www.trendmicro.com/en_us/research/23/k/a-closer-look-at-chatgpt-s-role-in-automated-malware-creation.html**
[27] D. Harry, «LLM-enabled Developer Experience (as of April 2024),» 2024. [En ligne] **https://www.linkedin.com/pulse/llm-enabled-developer-experience-april-2024-drew-harry-h0k4c**
[28] C. f. E. T. a. Security, « Evaluating Malicious Generative AI Capabilities,» 2024. [En ligne] **https://cetas.turing.ac.uk/publications/evaluating-malicious-generative-ai-capabilities**
[29] HP, « HP Wolf Security Uncovers Evidence of Attackers Using AI to Generate Malware,» 2024. [En ligne] **https://www.hp.com/us-en/newsroom/press-releases/2024/ai-generate-malware.html**
[30] Coolaj86, «Chat GPT «DAN» (and other «Jailbreaks»),» 2023. [En ligne] **https://gist.github.com/**

Vétéran incontournable de la cybersécurité offensive.

Par Gauthier PETITJEAN expert XMCO

Dans l'univers de la cybersécurité offensive, certains outils sont devenus des classiques indétrônables. **Nmap (Network Mapper)** en est sans doute l'exemple le plus emblématique. **Créé en 1997 par Gordon Lyon**, alias *Fyodor*, il a été conçu pour répondre à un besoin simple mais universel : découvrir les machines présentes sur un réseau et déterminer les services accessibles à distance. Vingt-cinq ans plus tard, malgré l'arrivée de solutions plus spécialisées et plus « modernes », Nmap reste un passage obligé dans la trousse à outils de tout pentester ou administrateur système. ■



Un outil historique et communautaire

Dès sa sortie, Nmap s'est distingué par sa simplicité d'utilisation et son efficacité. La possibilité d'identifier rapidement les hôtes actifs, de lister les ports ouverts et d'associer ces résultats à des services applicatifs a fait de lui un outil révolutionnaire à la fin des années 1990. À une époque où l'offre en matière de scanners réseau était limitée, Nmap a ouvert la voie à des pratiques de reconnaissance structurées.

Très vite, une **communauté internationale** s'est formée autour du projet. Hébergé sur GitHub, il totalise aujourd'hui plus de **13 000 commits** et bénéficie de contributions régulières, qu'il s'agisse de correctifs, de nouvelles signatures pour la détection de systèmes ou d'extensions de son moteur de scripts. L'ouvrage de référence, *Nmap Network Scanning*, écrit par Fyodor lui-même, est devenu une sorte de bible pour les étudiants en cybersécurité et les pentesters débutants.

La notoriété de l'outil a dépassé le cercle strictement technique. On retrouve Nmap dans des films cultes comme **Matrix Reloaded**, dans des romans de science-fiction et dans de nombreux cours universitaires. Pour beaucoup, « *apprendre Nmap* » est synonyme d'entrer dans l'univers du hacking éthique.

Un modèle de licence original

Sur le plan juridique, Nmap est distribué sous une **double licence**. La version com-munautaire repose sur la GNU GPLv2, ce qui garantit son usage gratuit pour tous. Mais le créateur a également mis en place la **Nmap Public Source Licence (NPSL)** : celle-ci impose, en cas d'intégration dans un produit commercial, d'acquérir **une licence OEM**. Ce modèle hybride, peu courant dans l'open source, a permis de **pérenniser financièrement** le projet sans limiter son adoption massive dans le monde académique et professionnel. Concrètement, un administrateur système, un chercheur ou un auditeur peuvent utiliser Nmap librement, mais une société éditrice qui souhaiterait l'intégrer dans une suite logicielle de cybersécurité devra s'acquitter de droits spécifiques.

Un écosystème riche

Au fil du temps, un écosystème entier s'est construit autour de Nmap.
Le site **Insecure.org** reste la vitrine historique du projet, tandis que **SecTools.org** propose un classement communautaire des meilleurs outils de sécurité et que **SecLists.org** fournit des listes de dictionnaires et de payloads indispensables aux pentesters. Ces ressources font de Nmap bien plus qu'un simple utilitaire : il est le cœur d'un environnement pédagogique et pratique que beaucoup d'auditeurs con-sidèrent comme un standard.

La documentation officielle, traduite en plus de 15 langues, est l'une des plus fournies du domaine. Elle comprend des manuels détaillés, des guides pratiques, et même un livre complet de plus de 500 pages. Certes, certaines sections sont datées et mentionnent des outils disparus, mais l'ensemble reste **une source de savoir incontournable**.

Un outil ancien mais vivant

À première vue, Nmap peut sembler daté. Son site web au design figé dans les années 2000, ses exemples parfois dépassés et certaines recommandations obsolètes donnent l'impression d'un outil en fin de vie. Mais cette apparence est trompeuse. Le cœur de Nmap, écrit en **C/C++**, est robuste, performant et continuellement mis à jour. Son moteur de scripts (**NSE**, basé sur Lua) permet d'ajouter facilement de nouvelles fonctionnalités. Plus de **600 scripts** sont aujourd'hui disponibles, couvrant des usages allant de la découverte réseau à la détection de vulnérabilités célèbres comme EternalBlue.

Cette modularité explique sa longévité : Nmap n'est pas figé, il évolue avec les besoins et les contributions de sa communauté. Il ne rivalise pas toujours avec des scanners commerciaux en termes d'ergonomie ou de rapidité, mais il conserve un avantage majeur : **la confiance**. Les auditeurs savent ce que fait Nmap, com-ment il le fait, et peuvent vérifier chaque étape.

L'écosystème Nmap

Astuce

Malgré son apparence rétro, Nmap reste le point de départ incontournable de tout audit réseau. Sa force réside moins dans les résultats bruts que dans la capacité à les interpréter. C'est pourquoi il est utilisé aussi bien dans les grandes entreprises que dans les formations universitaires : il apprend à raisonner comme un attaquant, en partant d'une cartographie précise avant d'exploiter une vulnérabilité.

Cadre d'utilisation : du scan web aux audits internes

Si Nmap a acquis une telle longévité, c'est avant tout grâce à sa **polyvalence**. L'outil s'adapte à des contextes très différents, qu'il s'agisse d'un test applicatif ciblé ou d'un audit interne à grande échelle. Comprendre ces différences est essentiel pour bien l'utiliser et éviter des interprétations trompeuses.

Audit applicatif web : un périmètre restreint

Dans le cadre d'un audit web, l'auditeur connaît généralement les adresses IP ou les noms de domaine à analyser. Les objectifs sont précis : identifier les services exposés sur les ports standards (HTTP/80, HTTPS/443, parfois 8080 ou 8443), vérifier les redirections, déceler la présence d'un proxy ou d'un service inattendu. Nmap sert alors d'outil de vérification initiale, avant d'entrer dans une analyse plus approfondie avec un scanner applicatif ou des tests manuels.

La contrainte principale est ici **le temps** : il s'agit de confirmer rapidement l'exposition de services. Un scan trop large serait inutile. Nmap, grâce à sa rapidité et à ses options de filtrage (scan de ports ciblés, détection de service), s'intègre parfaitement dans cette logique.

Audit interne : la cartographie à grande échelle

À l'opposé, un audit interne, souvent appelé *scanlan*, place l'auditeur dans un environnement inconnu. Il reçoit un accès réseau (parfois sans information préalable) et doit identifier l'ensemble des systèmes actifs. Le périmètre n'est plus un site web unique mais **des milliers d'adresses IP à explorer**.

La première étape est **la découverte d'hôtes** : quels équipements répondent sur le réseau ? Vient ensuite **la détection des services** : ports ouverts, protocoles actifs (SSH, SMB, RDP, LDAP, DNS, etc.). L'enjeu est double : établir **une cartographie exhaustive** et détecter des services mal sécurisés ou inattendus. Ici, Nmap devient central, car il est capable de balayer rapidement de larges plages IP et d'orienter la suite de l'audit.

Contraintes et choix méthodologiques

La principale difficulté réside dans **l'équilibre entre exhaustivité et performance**. Un scan trop large peut durer des heures, saturer le réseau, voire déclencher des alertes de

sécurité. À l'inverse, un scan trop restrictif risque de laisser passer des systèmes critiques.
L'auditeur doit donc ajuster ses paramètres :

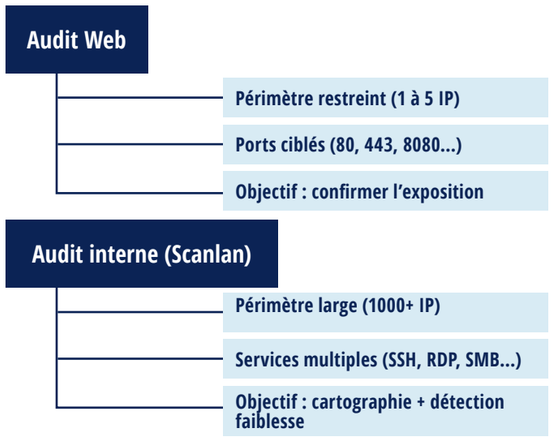
- cibler des plages IP précises plutôt que tout le réseau,
- prioriser les ports courants (22, 80, 443, 445, 3389),
- ajuster le parallélisme et les délais de retransmission.

Le contexte légal et contractuel est également à prendre en compte. Un audit externe doit respecter les plages autorisées par le client, sous peine d'être assimilé à une attaque. En interne, les tests doivent être coordonnés avec l'équipe IT pour limiter les risques de perturbation (ralentissements, déni de service involontaire).

Nmap comme point de départ

Dans tous les cas, Nmap ne doit pas être considéré comme une fin en soi. Ses résultats bruts, liste d'hôtes et états de ports, sont avant tout **des indicateurs**. Leur valeur dépend de l'interprétation qui en sera faite et des corrélations établies avec d'autres outils (Wireshark, Metasploit, scanners de vulnérabilités).
Ainsi, dans un audit web, un port 443 ouvert pourra mener à l'analyse d'un certificat TLS ou d'un reverse proxy. Dans un audit interne, un port 445 exposé sur plusieurs hôtes pourra indiquer une mauvaise segmentation réseau et ouvrir la voie à des attaques SMB.

Deux scénarios d'audit avec Nmap



Astuce

Toujours adapter le paramétrage de Nmap au contexte réel de l'audit. Un scan massif peut être pertinent en interne, mais catastrophique sur un site en production exposé sur Internet. Mieux vaut un scan progressif et raisonné qu'une cartographie rapide mais incomplète ou risquée.

NOTE : Veuillez vous référer à l'article complet pour une information plus détaillée.

Détection des hôtes : identifier ce qui est « up »

Avant de scanner des ports ou d'identifier des services, l'auditeur doit d'abord répondre à une question simple : **quelles machines sont actives sur le réseau ?** Cette étape, appelée *host discovery*, est essentielle car un hôte non détecté sera totalement absent de l'analyse. Nmap propose plusieurs techniques pour répondre à ce besoin, chacune ayant ses avantages et ses limites.

Principe de base

La logique est simple : envoyer une sonde, attendre une réponse. Si l'hôte répond d'une manière ou d'une autre, il est considéré comme actif *(up)*. À défaut de réponse, il est marqué comme inactif *(down)*. Ce mécanisme peut paraître trivial, mais dans la réalité il se heurte à de nombreux obstacles : pare-feu, règles de filtrage, IDS/IPS, comportements spécifiques des systèmes.

Méthodes disponibles

Nmap permet de tester la disponibilité d'un hôte avec différentes approches, adaptées à chaque contexte :

- **ARP Scan (-PR) :** utilisé en réseau local (LAN). Très fiable car un hôte doit répondre aux requêtes ARP pour être accessible. Incontournable en audit interne.
- **ICMP Echo (-PE) :** équivalent d'un *ping* classique. Simple mais souvent bloqué par les pare-feu ou désactivé par les administrateurs.
- **ICMP Timestamp / Address Mask (-PP, -PM) :** variantes plus rares, parfois utiles pour contourner des restrictions.
- **TCP SYN/ACK Probe (-PS, -PA) :** envoi d'un paquet TCP sur un port donné. Si le système répond par un SYN/ACK ou un RST, on sait qu'il est présent. Méthode efficace sur des ports standards (22, 80, 443, 445).
- **UDP Probe (-PU) :** moins courante mais intéressante pour tester des services UDP (DNS, SNMP). Les résultats sont plus difficiles à interpréter car de nombreux systèmes ignorent les paquets non sollicités.

Option -Pn : contourner la découverte

Nmap intègre l'option **-Pn**, qui permet de **désactiver la phase de découverte d'hôtes**. Tous les systèmes de la plage spécifiée sont alors considérés comme « up » et directement scannés. Cette approche peut être utile lorsque l'on sait que les pare-feu bloquent les pings ICMP ou les probes TCP, mais elle est risquée : l'outil perd du temps à analyser des adresses inactives, ce qui ralentit considérablement les scans sur de larges plages IP.

Pièges et faux négatifs

Un problème fréquent est la génération de **faux négatifs** : un hôte actif n'est pas détecté car il ne répond pas aux sondes envoyées. Ce cas se produit notamment sur Internet, où les administrateurs bloquent les réponses ICMP et filtrent les paquets suspects.
L'auditeur doit en être conscient et multiplier les méthodes de détection pour maximiser ses chances de succès.

À l'inverse, certains dispositifs de sécurité peuvent générer des **faux positifs** en renvoyant systématiquement des réponses, même pour des adresses inexistantes. Cela complique l'interprétation et oblige à recouper les résultats avec d'autres sources (logs réseau, outils complémentaires).

Meilleures pratiques

En pratique, un auditeur expérimenté combine plusieurs approches :

- Sur un LAN : privilégier l'ARP Scan (-PR), rapide et fiable.
- Sur Internet : mixer ICMP (-PE) et TCP (-PS, -PA) sur des ports courants.
- En environnement filtré : tenter l'option -Pn, mais uniquement sur des plages limitées pour éviter les pertes de temps.

Il est également conseillé d'analyser les résultats avec un regard critique. Un hôte considéré comme *down* par Nmap n'est pas nécessairement hors ligne : il peut être simplement protégé par un pare-feu restrictif.

Méthodes de découverte d'hôtes

LAN (réseau interne)	ARP Scan (-PR)
Internet (classique)	ICMP Echo (-PE)
Environnements filtrés	TCP SYN/ACK (-PS, -PA)
Ports UDP spécifiques	UDP Probe (-PU)
Contournement filtrage	Option -Pn

Astuce

Ne jamais se contenter d'une seule méthode de découverte. Multiplier les sondes augmente la fiabilité de la cartographie initiale. En cas de doute, mieux vaut considérer un hôte comme potentiellement actif plutôt que de l'écarter trop tôt de l'audit : certaines failles critiques se trouvent souvent sur des systèmes que l'on pensait invisibles.

Analyse des ports : comprendre leurs états et leurs implications

Une fois les hôtes identifiés comme actifs, l'étape suivante consiste à déterminer **quels services sont accessibles**. Chaque service écoute sur un ou plusieurs ports réseau, et l'analyse de leur état permet d'établir la surface d'exposition d'un système.

Nmap excelle dans ce rôle en classant les ports scannés selon plusieurs statuts, mais leur interprétation exige une véritable expertise.

Les six états définis par Nmap

Contrairement à l'idée reçue, un port n'est pas simplement « ouvert » ou « fermé ». Nmap distingue six états principaux :

- **Open (ouvert) :** un service répond activement. C'est l'indicateur clé, car il représente une porte d'entrée exploitable pour un attaquant.
- **Closed (fermé) :** aucun service n'écoute, mais le port est accessible et renvoie un paquet RST. Ce statut prouve que l'hôte est bien joignable.
- **Filtered (filtré) :** Nmap ne reçoit aucune réponse. Le trafic est probablement bloqué par un pare-feu ou un filtre réseau.
- **Open|Filtered :** impossible de trancher. Typique des scans UDP, où l'absence de réponse peut indiquer à la fois un service actif ou un filtrage.
- **Unfiltered (non filtré) :** le port est accessible, mais Nmap n'a pas pu déterminer son état précis.
- **Closed|Filtered :** état indéfini, généralement rencontré avec des scans spécialisés (comme l'Idle Scan).

Cette granularité est l'un des points forts de Nmap : elle incite l'auditeur à réfléchir à la nature de chaque réponse plutôt que de se contenter d'un binaire ouvert/fermé.

Influence de la méthode de scan

Le statut d'un port dépend de la technique employée :

- Un **TCP SYN Scan (-sS)** enverra un paquet SYN et interprétera la réponse (SYN/ACK = ouvert, RST = fermé, pas de réponse = filtré).
- Un **TCP Connect (-sT)** réalisera une connexion complète *(three-way handshake)*, ce qui augmente la fiabilité mais ralentit le processus.
- Un **UDP Scan (-sU)** se heurtera souvent à des silences, donnant beaucoup d'« open|filtered ».

Ainsi, le même port peut apparaître sous différents statuts selon la méthode utilisée. L'auditeur doit en tenir compte et, au besoin, croiser les résultats.

Limites et ambiguïtés

Les infrastructures modernes compliquent l'interprétation. De nombreux pare-feu bloquent silencieusement les connexions non autorisées, générant un grand nombre de ports « filtrés ». De plus, certains équipements (load balancers, reverse proxies, WAF) peuvent répondre à la place du serveur réel, brouillant les pistes.
Il est fréquent, par exemple, qu'un port 80 soit détecté comme « ouvert » alors qu'il renvoie systématiquement une redirection vers HTTPS.

De même, un port 22 marqué « fermé » peut cacher un service SSH accessible uniquement depuis des plages IP autorisées.

Exploiter les résultats

La valeur d'un audit ne se mesure pas au nombre de ports identifiés, mais à la manière dont ils sont exploités pour construire un raisonnement.

Quelques exemples :

- Un port **445 (SMB)** exposé en interne peut révéler des partages non sécurisés ou permettre une attaque de type EternalBlue.
- Un port **3389 (RDP)** détecté sur un serveur externe est un signal d'alerte majeur, car il expose directement un service critique sur Internet.
- Un port **53 (DNS)** ouvert en UDP peut indiquer un résolveur mal configuré et exposé aux attaques par amplification.

Ces constats n'apparaissent pas dans la sortie brute de Nmap : ils nécessitent une interprétation contextualisée.

États possibles d'un port selon Nmap

Open	**Service actif, exploitable**
Closed	**Pas de service, mais hôte joignable**
Filtered	**Paquets bloqués, état incertain**
Open I Filtered	**Silence ambigu (ex. UDP)**
Unfiltered	**Port accessible, état inconnu**
Unfiltered	**État indéfini (scans avancés)**

Astuce

Ne jamais se fier aveuglément à l'étiquette « open ». Derrière un port marqué ouvert peut se cacher un service protégé ou restreint. À l'inverse, un port filtré peut s'ouvrir dans certaines conditions. L'analyse critique et croisée est la clé : il faut toujours replacer les résultats de Nmap dans le contexte du réseau audité et des scénarios d'attaque plausibles.

Méthodes de scan principales : avantages et limites

L'efficacité de Nmap repose en grande partie sur la variété de ses méthodes de scan. Chaque technique s'appuie sur une logique réseau différente et produit des résultats distincts. L'auditeur doit donc choisir la méthode adaptée à son contexte, car ce choix impacte à la fois **la précision** des résultats, **la rapidité** du scan et **le niveau de discrétion** vis-à-vis des systèmes de défense.

TCP Connect Scan (-sT)

Le **TCP Connect** est la méthode la plus simple : Nmap établit une connexion complète en effectuant le *three-way handshake* (SYN → SYN/ACK → ACK). Si la connexion aboutit, le port est considéré comme ouvert. S'il reçoit un RST, il est fermé.

Avantages :

- Ne nécessite pas de privilèges root/administrateur, fonctionne dans tous les environnements.
- Résultats fiables, car la connexion est réellement établie.
- Peu de risques d'erreur d'interprétation.

Limites :

- Plus lent, car chaque tentative implique une connexion complète.
- Détectable facilement par les journaux système (le service voit la tentative).

En pratique, le TCP Connect est recommandé pour la majorité des audits professionnels : il est sûr, exhaustif et reproductible.

TCP SYN Scan (-sS)

Aussi appelé *half-open scan* ou *stealth scan*, le **TCP SYN Scan** envoie un paquet SYN et interprète la réponse :

- SYN/ACK → port ouvert
- RST → port fermé
- Pas de réponse → filtré

Nmap n'envoie pas l'ACK final, ce qui évite de compléter la connexion.

Avantages :

- Plus rapide que le TCP Connect.
- Historiquement perçu comme plus « discret » car la connexion n'est pas finalisée.

Limites :

- Nécessite des privilèges root/administrateur pour forger les paquets.
- Détectable par la plupart des IDS/IPS modernes, qui repèrent les tentatives incomplètes.
- Peut causer des perturbations : un scan massif SYN peut saturer un pare-feu ou générer un déni de service (SYN flood).

Aujourd'hui, le SYN Scan n'est plus véritablement furtif. Dans un contexte professionnel, il doit être utilisé avec précaution, voire évité si le périmètre audité est sensible.

UDP Scan (-sU)

Contrairement au TCP, l'UDP est sans état *(stateless)*. Le UDP Scan envoie un paquet UDP vide (ou minimal) vers un port. Trois cas principaux se présentent :

- Réponse ICMP « port unreachable » → port fermé.
- Réponse spécifique du service (DNS, SNMP) → port ouvert.
- Pas de réponse → port marqué « open|filtered ».

Avantages :

- Indispensable pour détecter certains services critiques (DNS/53, SNMP/161, DHCP/67).
- Permet d'identifier des surfaces d'attaque souvent négligées.

Limites :

- Très lent, car les systèmes ignorent souvent les paquets UDP non valides.
- Résultats ambigus, beaucoup de « open|filtered ».
- Peut générer un trafic lourd si le nombre de ports est important.

En pratique, l'UDP Scan doit être ciblé : inutile de balayer 65 535 ports UDP, mieux vaut viser quelques services connus.

Autres méthodes (plus marginales)

Nmap propose également des techniques spécialisées comme l'**ACK Scan (-sA)** pour tester les règles de filtrage, le **FIN/Xmas/Null Scan (-sF, -sX, -sN)** pour contourner certains pare-feu, ou encore l'**Idle Scan (-sI)** permettant d'effectuer un scan en usurpant l'identité d'un hôte tiers. Ces méthodes sont aujourd'hui surtout utilisées à des fins pédagogiques ou dans des contextes très spécifiques.

Choisir la bonne méthode

Le choix ne doit jamais être laissé au hasard. Sur un périmètre large et sensible, privilégier le **TCP Connect** pour sa robustesse. Pour compléter, exécuter un **UDP ciblé** sur les services critiques. Le **SYN Scan**, bien que populaire dans les guides en ligne, est rarement adapté dans un audit professionnel en raison des risques de perturbation et de sa détectabilité accrue.

Résumé des principales méthodes de scan

TCP Connect (-sT)	Fiable, lent, visible dans les logs
TCP SYN (-sS)	Rapide, nécessite root, détectable
UDP Scan (-sU)	Utile, lent, résultats ambigus
Autres (ACK, FIN, Idle...)	Cas spécifiques

Astuce

> **Ne jamais céder à la tentation du « tout-ouvrir » avec Nmap. Le choix d'une méthode doit toujours tenir compte du contexte opérationnel : périmètre restreint ou large, tolérance aux perturbations, dispositifs de défense en place. Un bon auditeur sait adapter sa stratégie de scan plutôt que d'appliquer aveuglément des recettes trouvées en ligne.**

Identification, scripts et bonnes pratiques : la valeur ajoutée de Nmap

La détection d'hôtes et l'analyse des ports constituent la base du travail avec Nmap. Mais pour un audit efficace, il faut aller plus loin : **identifier les services qui s'exécutent derrière ces ports**, reconnaître les systèmes d'exploitation et, si nécessaire, exploiter les capacités avancées offertes par **le Nmap Scripting Engine (NSE)**.

Identifier les services applicatifs (-sV)

Un port ouvert n'indique pas seulement la présence d'un service : il est crucial de déterminer **quel service** fonctionne et, si possible, dans **quelle version**. L'option **-sV** envoie des sondes spécifiques et compare les réponses à une base de signatures. Nmap utilise ainsi plus de 180 « probes » capables de différencier un serveur Apache d'un Nginx, ou un simple proxy d'une application métier.

L'identification peut se baser sur deux approches :

- **Méthode « table » :** correspondance statique entre ports/protocoles et services connus (fichier *nmap-services*). Rapide mais peu fiable, car basée uniquement sur les usages par défaut.

- **Méthode « probed » :** envoi de sondes spécifiques (paquets de test) et analyse des réponses avec des expressions régulières (fichier *nmap-service-probes*). Plus lente mais beaucoup plus précise, car elle agit au niveau applicatif.

Ces données sont essentielles : savoir qu'un service SSH écoute en version 7.2p2 peut orienter un auditeur vers une recherche de vulnérabilités spécifiques.

Fonctionnement

- 187 sondes sont disponibles (103 TCP, 84 UDP).
- L'identification se fait en trois étapes :
  - Essai d'une sonde « NULL » (sans données), pour capter une bannière éventuelle.
  - Envoi des sondes associées au port testé.
  - Envoi des sondes dont la rareté est ≤ à un seuil paramétré *(--version-intensity)*.
- Les résultats peuvent être :
  - **match** (identification précise),
  - **softmatch** (probabilité, résultat approximatif),
  - ou **fallback** (méthodes plus basiques).

Fiabilité et réglages

- Nmap attribue un **niveau de confiance (1–10)** selon la qualité de l'identification.
- Options utiles :
  - **--version-all** pour tester toutes les sondes,
  - **--version-light** pour accélérer en limitant les sondes,
  - **-oX** pour générer une sortie XML et exploiter le champ de confiance.

Cas particuliers

- **Service Fingerprint (SF) :** le service répond mais ne correspond à aucun modèle connu → Nmap propose de soumettre l'empreinte à sa base.
  NOTE : Veuillez-vous référer à l'article complet pour savoir comment l'exploité.
- **Tcpwrapped :** connexion acceptée puis immédiatement fermée (TCP RST), typiquement lié à des restrictions d'accès ou à un IDS/IPS.
- **« http? » :** service détecté via la méthode « table » mais non confirmé par la méthode « probed ».

L'option **-sV** permet d'aller au-delà de la simple détection de ports ouverts, en identifiant plus ou moins précisément le service exposé et parfois sa version.

Toutefois, la fiabilité varie selon la méthode utilisée, la configuration réseau, et les éventuels équipements intermédiaires (WAF, reverse-proxy, etc.).

Détection des systèmes d'exploitation (-O)

Nmap peut tenter d'identifier l'OS d'une machine grâce au fingerprinting. Il analyse des caractéristiques comme la taille de la fenêtre TCP, la gestion des flags ou la valeur du TTL. Le résultat est ensuite comparé à une base d'empreintes.

Cette fonctionnalité est ingénieuse, mais elle atteint aujourd'hui ses limites. Les systèmes modernes standardisent leurs comportements, rendant la détection moins fiable. De plus, les équipements intermédiaires (pare-feu, load balancers) biaisent souvent les résultats. L'OS detection doit donc être utilisée comme **un indice complémentaire**, jamais comme une preuve absolue.

Nmap Scripting Engine (NSE)

Le NSE constitue l'une des évolutions majeures de Nmap. Basé sur le langage **Lua**, il permet d'automatiser des analyses poussées via plus de **600 scripts** classés en catégories :

- **Discovery :** découverte réseau avancée.
- **Auth :** tests d'authentification (brute force, comptes par défaut).
- **Vuln :** détection de vulnérabilités connues (ex. SMBv1/EternalBlue).
- **Exploit :** scripts intrusifs exploitant des failles spécifiques.

L'avantage est évident : en un scan, l'auditeur peut tester la présence de vulnérabilités critiques. Mais les risques sont réels : certains scripts sont intrusifs, obsolètes ou dangereux. Ils peuvent bloquer des comptes (tests d'authentification répétés) ou provoquer des dénis de service involontaires. Dans un cadre professionnel, l'usage du NSE doit donc rester **ciblé et prudent**.

Optimisation et bonnes pratiques
Nmap offre de nombreux paramètres pour ajuster ses scans : vitesse, parallélisme, délais de retransmission. Accélérer un scan peut être tentant, mais au prix d'une fiabilité moindre. L'auditeur doit trouver **le juste équilibre entre rapidité et exhaustivité**.

Il est également essentiel d'exploiter les options de débogage comme **--packet-trace**, qui permet de visualiser les paquets envoyés et reçus. Ces fonctionnalités aident à comprendre les comportements inattendus, qu'ils soient dus au réseau ou aux systèmes audités.

La place de Nmap aujourd'hui

Nmap n'est pas un scanner de vulnérabilités tout-en-un. C'est **un outil de reconnaissance et de cartographie** qui, bien utilisé, reste incontournable. Ses forces résident dans sa robustesse, sa transparence et sa communauté active.

Dans un audit, **il constitue la première brique** : découvrir les hôtes, identifier les services, préparer le terrain pour des analyses plus poussées avec d'autres outils.

Son apparente vétusté ne doit pas tromper : c'est un vétéran encore redoutablement efficace.

Trois usages avancés de Nmap

Identification (-sV)	Services et versions précises
Fingerprinting (-O)	Indices sur l'OS
Scripting (NSE)	Automatisation et détection ciblée

Astuce

Utiliser Nmap non pas comme une fin en soi, mais comme le point de départ d'une réflexion stratégique.
Les résultats bruts ne sont que des indices : c'est l'interprétation humaine, croisée avec d'autres outils et méthodes, qui transforme un scan en véritable analyse de sécurité.

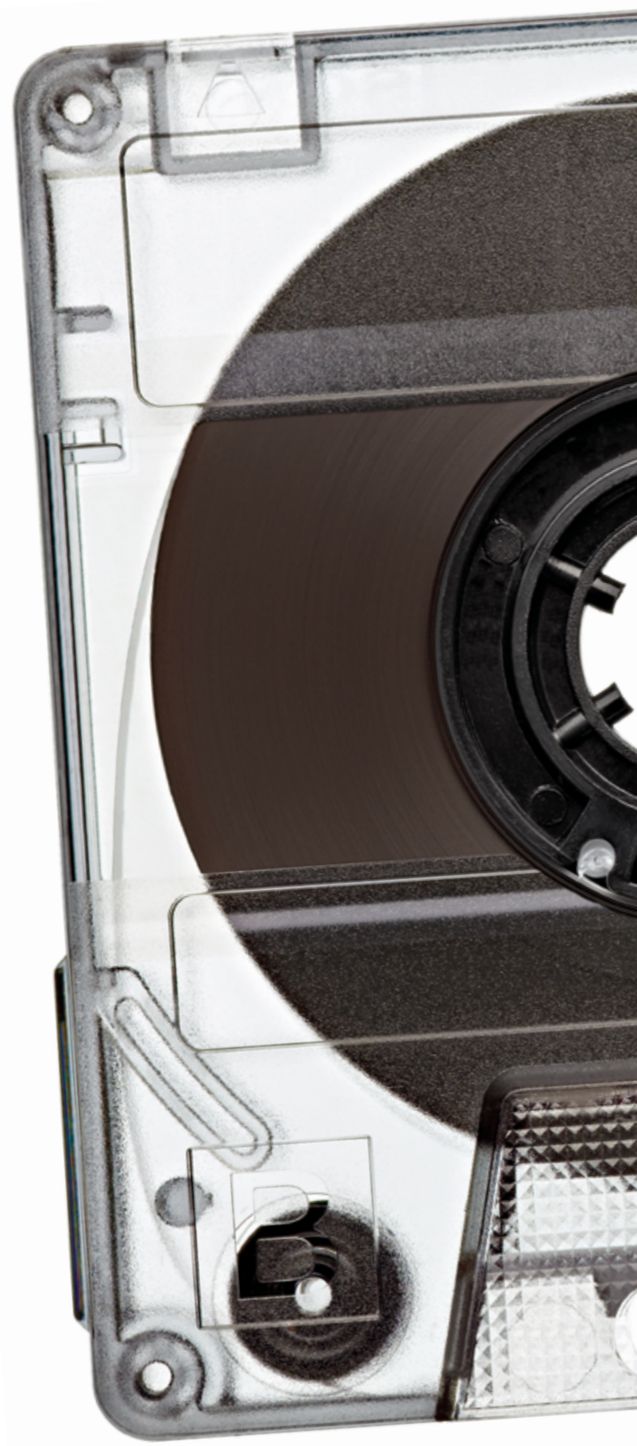
Conclusion

Nmap s'impose ainsi comme un outil de référence dans le domaine de la cartographie réseau, largement utilisé lors de tests d'intrusion. Reconnu comme un incontournable durant l'étape de reconnaissance, Nmap est un outil ayant traversé les décennies et gagné en popularité au fil du temps. Les raisons de son adoption dans le milieu sont nombreuses ; comme vu au sein de cet article, Nmap tire sa force de sa grande capacité de personnalisation, sa polyvalence et la fiabilité des résultats qu'il permet d'obtenir.

Mais si Nmap reste toujours autant utilisé, c'est surtout parce qu'il reste parfaitement adapté depuis plus de deux décennies à un besoin indispensable et qui n'a pas changé depuis le jour de sa sortie, le 1er septembre 1997 : connaître ce qui est exposé sur un réseau. À l'image d'une boussole, Nmap aide à naviguer au sein de réseaux inconnus et sert de référence pour savoir quelle direction emprunter.

L'outil s'est toutefois enrichi avec le temps et a tenté d'offrir de nouvelles fonctionnalités comme la détection de systèmes d'exploitation ou l'exécution de scripts en tout genre. Ces deux mécanismes tirent leur force de la collaboration des développeurs et illustrent la volonté du fondateur Gordon Lyon de créer un outil open source, communautaire et collaboratif. ■

Retrouvez notre dossier complet sur ce sujet en page n°34 de ce numéro.



We are hiring!

Manager Audit

4 ans d'expérience minimum - CDI - Paris & Nantes
Ce poste allie encadrement d'équipe et expertise technique, avec une forte implication dans la réalisation d'audits. Vous contribuerez activement au développement de notre offre de tests d'intrusion (Pentest).

Consultant Senior/Confirmé Audit Technique

5 ans d'expérience minimum - CDI - Paris
Vous réaliserez des audits de sécurité et des tests d'intrusion sur des environnements variés et complexes. Vous participerez également à la montée en compétences des nouveaux collaborateurs.

Consultant Senior/Confirmé Red Team

5 ans d'expérience minimum - CDI - Paris
Intégré à notre équipe Red Team, vous conduirez des campagnes offensives de bout en bout et interviendrez ponctuellement sur d'autres types d'audits (web, poste de travail, tests internes...).

Consultant Senior - GRC

5 ans d'expérience minimum - CDI - Paris
Vous réaliserez des audits de conformité (notamment PCI DSS), de sécurité organisationnelle, physique et d'architecture. Vous rédigerez des rapports et proposerez des plans de remédiation adaptés.

Manager - Services Managés

6 ans d'expérience minimum - CDI - Paris
Vous encadrerez l'équipe Services Managés au quotidien et contribuerez à renforcer son autonomie. Vous veillerez à l'alignement des objectifs opérationnels avec la vision stratégique de XMCO.

Analyste Cybercriminalité - Darkweb

1 à 2 ans d'expérience minimum - CDI - Paris
Vous détecterez et analyserez des incidents de sécurité (exposition de données, vulnérabilités, etc.) pour nos clients, dans le cadre de notre offre de cybersurveillance Serenety.

Chef de Projet - Cyber Threat Intelligence (CTI)

4 ans d'expérience minimum - CDI - Paris
Vous contribuerez à la définition de la stratégie de notre activité CTI, en identifiant les cas d'usage les plus pertinents pour nos équipes internes et nos clients.

Responsable Support IT

4 ans d'expérience minimum - CDI - Paris
Vous piloterez la structuration du support interne et assurerez la disponibilité des outils du quotidien (postes, téléphonie, bureautique), tout en améliorant l'expérience utilisateur des collaborateurs.

Responsable de la Sécurité des Systèmes d'information (RSSI)

5 ans d'expérience minimum - CDI - Paris
Vous définirez la stratégie de sécurité interne et piloterez les dispositifs de gouvernance, conformité et gestion des risques. Vous accompagnerez également les équipes commerciales sur les sujets sécurité.

08-11 OCTOBRE 2025

LES ASSISES

MONACO

Nous serons présents :

Stand J32

Ce document est la propriété du cabinet XMCO. Toute reproduction est strictement interdite.

Nmap - Dossier complet

Nmap, un vétéran incontournable de la cybersécurité offensive.

1. Introduction

1.1 Propos introductifs

Cet article traite d'un des outils les plus vieux et pourtant toujours largement utilisé dans le milieu de la sécurité offensive : **Nmap** (*«Network mapper»*).

Nmap, en quelques détails historiques et légaux, c'est un projet :

- Créé en 1997 par Gordon Lyon (pseudonyme «Fyodor»). Désormais maintenu par la communauté (+13k commits sur GitHub, 10.8k stars), mais Fyodor est toujours actif notamment sur des sujets de modération.
- Open source, basé sous la licence GNU GPLv2, mais contient toutefois certaines disparités. La licence officielle est la NPSL («Nmap Public Source licence»).
- Gratuit et libre d'utilisation pour les utilisateurs finaux.
- Payant, en revanche, si besoin d'être embarqué dans des produits commercialisés. Pour cela, l'édition «Nmap OEM Edition» (licence «OEM») doit être pourvue.
- Communautaire regroupant des passionnés, développeurs, chercheurs de vulnérabilités, blogueurs, etc., autour d'autres sites comme SecTools.org, SecLists.org et Insecure.org.
- Qui a largement marqué son temps et le milieu au point de faire l'objet de centaines d'articles dans des magazines, son apparition dans plusieurs films (dont Matrix Reloaded), de dizaines de livres et d'un nombre incalculable de « talks », ainsi que d'une série de bandes dessinées.

De par cet historique, Nmap c'est aussi :

- Un site Web à l'ancienne, avec une UI « old school » et une UX assez douteuse (testez, vous verrez…) et utilisant une version d'Apache datant de 2013 impacté (soi-disant) par 66 CVE.
- Une vieille fondation recommandant des outils rétro qu'il devient de plus en plus rare d'utiliser (ex. Cain and Abel).
- Un manuel avec des interprétations et conclusions qu'il faut remettre dans le contexte en raison du côté caduc de certaines.

D'un point de vue technique, Nmap reste le taulier et demeure un outil incontournable pour apprendre et travailler:

- Permettant de conduire des scans de services, de la découverte d'hôte, de l'identification et exploitation de vulnérabilités, du fingerprinting et bien plus encore.
- Codé en C/C++.
- Modulaire, les développeurs le souhaitant peuvent ajouter des scripts via le NSE « Nmap Scripting Engine ». Ces scripts sont développés en Lua, un langage léger conçu pour être extensible. Ils visent à automatiser des tâches de reconnaissance, de détection et d'exploitation de vulnérabilités.
- Une documentation très complète de dizaines de milliers de lignes présentant l'outil dans ses moindres détails et traduite en plus de 15 langues, facilitant également le développement et l'amélioration de l'outil à travers le monde. Le livre officiel d'Nmap dispose à ce titre de plus de 500 pages d'explications et d'interprétations (dont certaines sont par ailleurs aujourd'hui obsolètes).

Vous l'aurez compris, Nmap s'est ainsi hissé parmi les outils les plus utilisés depuis sa création en 1997. Son utilisation demande toutefois de bien connaitre l'outil pour y adopter une interprétation adéquate.

1.2 L'utilisation d'Nmap dépend du besoin, de l'objectif et des contraintes de l'audit

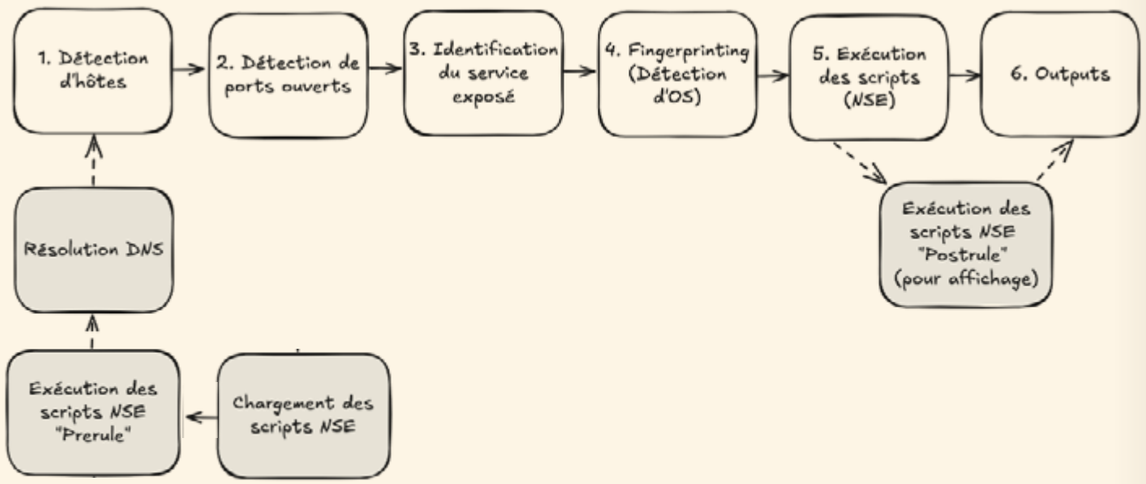
En fonction de l'objectif, des contraintes et des besoins d'un audit, la nature et les méthodes de scan à effectuer diffèrent. Par exemple dans le cas d'un audit Web, l'objectif est de tester une application Web généralement exposée sur les ports 80/TCP (HTTP) et 443/TCP (HTTPS) et un nombre limité de serveurs sera dans le périmètre. Leur adresse est fournie en amont et les serveurs sont déjà considérés comme en lignes *(up)* lors du démarrage de la mission.
Les tests applicatifs peuvent ainsi démarrer en parallèle du scan de ports.

À l'inverse pour un audit interne (scanlan), des centaines voire des milliers de systèmes sont à scanner, sans savoir au préalable lesquels sont en lignes (des plages IP étant généralement adressées). Les tests se portent sur les services exposés en interne (SSH, FTP, SMB, RDP, LDAP, HTTP, etc) : ces derniers ne peuvent ainsi pas démarrer sans les premiers résultats du scan de ports.

Par conséquent, en fonction de la nature même de l'audit et des contraintes et besoins qui en découlent, la phase de détection de serveurs, de ports et d'identification de services diffère. L'approche à adopter peut également varier en fonction du nombre de systèmes à auditer et du temps mis à disposition pour les tests.

Enfin, Nmap peut être paramétré avec certaines options pour désactiver la découverte d'hôte et de ports et réaliser d'autres opérations, comme des tentatives d'énumération du système ciblé et l'exécution de scripts divers et variés.

Voici une chaine complète décrivant le déroulé des actions effectuées par Nmap. Les actions grisées sont transparentes et non contrôlables par l'utilisateur final :



Cet article traitera principalement les 5 premières étapes de cette chaine d'actions, dans l'ordre. Il ne vise pas à être un « n-ième » papier décrivant les flags et les méthodes proposées par l'outil, mais dégrossit le fonctionnement des opérations à connaître et fournit un retour d'expérience suite à leur utilisation dans des cas réels d'audit de sécurité. La dernière partie de cet article s'attellera à fournir une liste de conseils d'utilisation et autres « tricks » souvent méconnus en vue d'assurer la fiabilité des résultats ou d'accélérer la phase de scan.

2. Partie 1 : Hôtes et Services

2.1 Détection d'hôtes

La première question à se poser lorsque nous souhaitons détecter un serveur au sein d'une plage IP est : sur quelle condition doit-on se baser pour déterminer si l'hôte est up ? De manière générale, Nmap détermine qu'un hôte est up si une réponse est obtenue aux paquets de détection, qu'importe la nature de la réponse *(TCP RST, TCP SYN/ACK, ICMP type 3 - Destination unreachable)*. Dans le seul cas où aucune réponse n'est obtenue, Nmap interrompt la chaine d'exécution et considère que l'hôte est *down*. L'objectif de cette première réside ainsi dans la réception d'une réponse de la part de l'hôte ciblé, afin d'effectuer la phase de scan ultérieure.

Astuce d'auditeur
Pour forcer Nmap à considérer l'hôte comme étant up, l'option « -Pn » peut être indiquée.

Dans ce cas, l'outil initiera directement l'étape suivante, à savoir le scan de port.

Cette première section concerne ainsi uniquement les hôtes dont l'auditeur ne peut certifier le statut (ex. fourniture d'une range d'IP seulement, comme **172.16.0.0/16**).

Plusieurs méthodes sont alors nativement proposées par Nmap pour envoyer des sondes et tenter d'obtenir une telle réponse. Celles-ci permettent d'opérer à différentes couches du modèle OSI. En voici quelques exemples :

Couche modèle OSI	Paramètre Nmap	Requête envoyée	Réponse associée
ARP (2)	**PR**	*Who has [IP]*	*[IP] is at [MAC]*
IP (3)	**PO[n]**	*Paquet IP* avec **[n]** placé au sein de *l'en-tête IP « Protocol »*	Dépend du protocole spécifié, *généralement réponse ICMP*
ICMP (3)	**PE** **PP** **PM**	*ICMP Type 8* (echo request) *ICMP Type 13* (timestamp request) *ICMP Type 17* (address mask request)	*ICMP Type 0* (echo reply) *ICMP Type 13* (timestamp reply) *ICMP Type 18* (address mask reply)
TCP (4)	**PS[port]** **PA[port]**	*TCP SYN* sur le **[port]** *TCP ACK* sur le **[port]**	*TCP ACK / TCP RST /* *ICMP type 3* (Destination Unreachable) *ICMP Type 3* (Destination Unreachable)
UDP (4)	**PU[port]**	*Datagramme UDP* sur le **[port]**	*ICMP Type 3* (Port Unreachable)

Toutes ces options offrent aux utilisateurs une forte capacité de personnalisation dans la méthode à utiliser, et peuvent être combinées ensemble. Il est cependant plutôt conseillé de n'utiliser qu'une seule méthode par commande afin de contrôler avec précision la nature des opérations réalisées. Toutes les méthodes proposées ne se valent pas. Un paramétrage intéressant pourrait être, dans l'ordre :

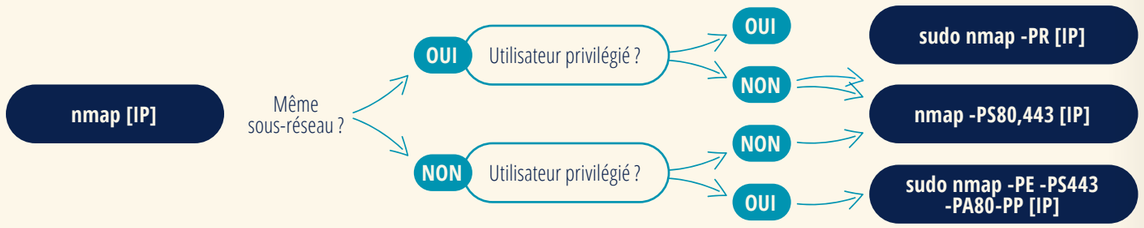
Paramètre Nmap	But visé
nmap -PS21,22,53,80,139, 3389,443,445,8080-8084,8443 [IP-RANGE]	Recherche opérant couche TCP, à la recherche de système en tout genre.
nmap -PR [IP-RANGE]	Recherche des hôtes situés dans le même sous réseau. Cette méthode a l'avantage d'être très rapide mais ne concerne qu'une faible partie d'hôtes potentiels.
nmap -PE [IP-RANGE]	Recherche des hôtes répondant aux ping (ICMP). Cette méthode permet dans certains cas d'avoir une réponse immédiate, même si sa fiabilité est limitée, le protocole ICMP étant souvent filtré.

Astuce d'auditeur
L'avantage de l'option « -PS » provient surtout du fait que les ports par défauts sont très généralement adoptés au sein des réseaux internes (leur personnalisation demanderait aux administrateurs des entreprises un suivi et un maintien fastidieux)

Par conséquent, la commande précédente permet de cibler la plupart du temps :
- des imprimantes (ports 21, 80/443) ;
- des serveurs Linux (port 22) ;
- des serveurs Windows (port 139, 445, 3389) ;
- des postes de travail Windows (port 139, 445) ;
- des serveurs de production ou tout autre équipement exposant un service Web (port 80,443, 8080, 8081, 8443, etc.).

Finalement, il devient primordial de comprendre les options proposées par l'outil sous peine de passer à côté de certains systèmes qui pourraient exposer des vulnérabilités. Bien que le but d'un audit ne soit pas d'assurer l'exhaustivité, il est toujours plus appréciable d'identifier un maximum de système et de couvrir le plus de chemins de compromission.

D'autant plus quand on sait que le comportement par défaut de Nmap est très limité, et dépend en plus de plusieurs facteurs. En effet, en fonction des privilèges d'exécution accordés à l'outil et au fait que la cible soit dans le même LAN ou non, exécuter la commande « Nmap [IP] » revient à positionner les paramètres ci-dessous :



Spécificité Nmap
Le comportement de découverte d'hôte diffère selon les privilèges accordés lors de l'exécution de Nmap, et peut même ne pas prendre en compte les paramètres spécifiés sans en avertir l'utilisateur final.
Par exemple, même si l'option « -PE » est indiquée et qu'Nmap est lancé en tant qu'utilisateur privilégié, seules des requêtes ARP (équivalent à « -PR ») seront émises pour des systèmes présents dans le même réseau local. Pour forcer le comportement spécifié (ex. émission de requêtes ICMP avec « -PE »), il faut utiliser l'option « --send-ip ».

Encore, lancer Nmap avec l'option « -PS21 » en sudo équivaut à réaliser un TCP SYN sur ce port, alors qu'un TCP Connect sera réalisé si l'outil n'est pas lancé avec ces privilèges.
Pour s'assurer du comportement adopté par l'outil, l'option « *--packet-trace* » détaillée dans la SECTION 2 de cet article s'avèrera forte utile.

2.2 État des ports

Avant d'étudier l'étape de détection de ports, il faut comprendre les états attribués par Nmap aux ports « potentiellement » ouverts sur un système.

Lorsqu'une application est exécutée et configurée pour exposer un service sur le réseau, elle écoute sur un port permettant à un système (client) d'établir une connexion (TCP/UDP). Par exemple sur Linux, le démarrage du processus *sshd* fait appel à l'ouverture d'une socket réseau sur le port 22/TCP par défaut. Dès lors, n'importe quel utilisateur situé sur le même réseau (ou ayant un accès à ce réseau) peut communiquer avec ce serveur sur ce port, sous réserve que les configurations déployées par les équipements intermédiaires comme les routeurs, switch, firewall ou reverse-proxy (WAF, serveur de cache, CDN, « load-balancer », etc.) permettent de laisser les flux transiter via les configurations adéquates (règles firewall, routage, NAT, etc.).

Ainsi, en plus de devoir être **ouverts**, les ports doivent être **accessibles**.

L'objectif de cette étape étant de déterminer si un port expose un service, l'outil définit les 6 états suivants pour un port donné. Ils sont classés du plus au moins fréquent :

État d'un port	Description
Ouvert	Un service est exposé et sa réponse est parvenue à Nmap (ex. *TCP ACK*). Le port est donc ouvert et les flux ne sont pas bloqués.
Fermé	Une réponse est parvenue à Nmap indiquant qu'aucun service n'est en écoute sur le port (ex. *TCP RST*).
Filtré	Nmap ne peut pas déterminer si le port est ouvert ou fermé, car aucune réponse du serveur n'a été obtenue (l'erreur ICMP « *destination unreachable* » n'est pas valable ici, contrairement à la phase de détection d'hôte). Les paquets de réponses sont bloqués par un équipement intermédiaire comme un pare-feu.
Ouvert / Filtré	Comme pour l'état Filtré, aucune réponse n'a été obtenue par Nmap. Toutefois, il est possible qu'il s'agisse du comportement attendu, car le paquet envoyé (ex. TCP FIN) ou le protocole utilisé par le service (UDP) ne soit pas configuré pour répondre aux paquets envoyés par Nmap (cf. section suivante).
Non-filtré	Le port est accessible, car une réponse est obtenue, mais Nmap ne peut pas déterminer si ce dernier est ouvert ou fermé.
Fermé / Filtré	Le port est soit fermé, soit filtré. Seuls les scans « UDP », « IP », « FIN », « NULL » et « Xmas » classent un port dans cet état.

Nmap se base donc sur la réponse qu'il obtient des paquets qu'il envoie **pour déterminer si un port est ouvert**.
Le résultat renvoyé par l'outil ne dépend donc pas seulement de l'état même du port et de son accès réseau, mais aussi de la méthode de scan utilisée.

Astuce d'auditeur
Il est primordial de comprendre que la méthode de scan utilisée conditionne les résultats obtenus. Afin de ne pas commettre d'erreurs d'interprétation, il est nécessaire d'étudier et de comprendre en détail le fonctionnement des méthodes de scan proposées par Nmap et ainsi savoir adapter l'outil à son besoin.

Spécificité Nmap
La sémantique de l'état du port obtenu n'est pas à prendre au pied de la lettre et il convient de garder un esprit critique lors de leur analyse.
Par exemple, ce n'est pas parce qu'un port est marqué « ouvert|filtré » qu'il faut le considérer soit « ouvert », soit « filtré ».

Ce résultat « ouvert|filtré » indique seulement qu'au vu du scan utilisé, le comportement obtenu ne permet pas de conclure avec certitude sur l'état du port.

Celui-ci pourrait tout à fait être :
- « ouvert », mais le paquet envoyé n'a pas provoqué de réponse de la part du service ciblé (ex. fonctionnement d'UDP, cf. section suivante) ;
- « filtré », expliquant pourquoi aucune réponse n'ait été reçue (comportement standard d'un port « filtré »);

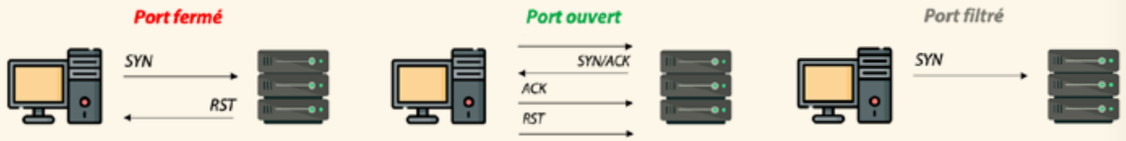
Ce port pourrait même être en réalité « fermé » sur l'hôte ciblé, la réponse (ex. *TCP RST*) ayant pu être bloquée par un hôte intermédiaire.

Ainsi, bien qu'il ne soit pas toujours possible d'attester avec certitude de l'état d'un port, certaines méthodes proposées par Nmap sont plus fiables que d'autres et à ce titre doivent être privilégiées.

2.3 Détection des ports ouverts

Nous ne rentrerons pas dans les détails de chacune des méthodes proposées au sein de cet article, car toutes n'ont pas la même efficacité. Certaines d'entre elles n'ont par ailleurs pas vocation à identifier un port ouvert.
Ainsi, seules les 3 principales méthodes visant à statuer sur l'accès à un port seront détaillées. Le paramètre « -sn » vise à passer cette étape et à désactiver le scan de ports.

2.3.1 Scan TCP Connect
Le scan le plus important à connaître est le scan TCP Connect (-sT). Cette méthode crée une connexion TCP complète (« three-way handshake ») sur chaque port spécifié via l'appel au système d'exploitation. Plus précisément, la fonction en C « connect » est utilisée par Nmap (bibliothèque winsock2.h pour Windows, sys/socket.h pour Linux).
Il s'agit du seul type de scan TCP proposé par Nmap qu'il est recommandé d'utiliser.
Ce dernier classe de la façon suivante l'état du port testé :



Aussi, il s'agit du seul type de scan ne nécessitant pas de permissions élevées sur le système (administrateur local/root) pour être effectué. Toutes les autres méthodes proposées par Nmap envoient des paquets forgés sur mesure directement par l'outil, ce qui nécessite de plus hauts privilèges sur le système.

Spécificité Nmap
Tout programme logiciel est légitime d'ouvrir une connexion afin d'envoyer et recevoir des données sur un réseau (ex. navigateur web, jeux vidéo, etc.). Aucun privilège n'est requis pour se faire, car la gestion des données envoyées et reçues est gérée par le système d'exploitation via la mise à disposition d'API système (ex. fonction en C « connect »). En effet, le programme ne gère pas lui-même les paquets envoyés et leur contenu : il ne fait qu'appeler ces fonctions. La manipulation des paquets / datagrammes réseau et des bits constituant leurs en-têtes est ensuite réalisée par le système d'exploitation, pour des raisons de facilité de développement (abstraction), et de sécurité.

Or, certaines méthodes proposées par Nmap ont justement besoin de manipuler certaines valeurs des en-têtes et bloc de données couche TCP, UDP ou IP et d'avoir le contrôle total sur la socket réseau courante, afin d'envoyer des paquets de tests sur mesure. La méthode « socket » avec la structure « SOCK_RAW » du code source est utilisée à cet effet.

Par conséquent, la nécessité d'être lancé en tant qu'utilisateur privilégié est synonyme de comportement spécifique, non ordinaire et potentiellement à risque.

2.3.2 Scan UDP
Un scan UDP (*-sU*) peut également être pertinent à réaliser, à la recherche de service comme *SNMP, DHCP* ou *DNS*.

Avant toute chose, une particularité inhérente au protocole UDP est primordiale à comprendre : dans la majorité des cas où un service UDP est potentiellement en écoute, il y a de grandes chances que le port soit marqué « *ouvert|filtré* » au lieu d'être simplement marqué « *ouvert* ».

D'une part pour des protocoles assez conventionnels comme les 3 mentionnés précédemment *(SNMP, DHCP, DNS)*, **Nmap envoie un datagramme UDP spécifique au protocole**.
Par exemple :
- **Get-Request** demande au serveur *SNMP* la valeur d'un objet spécifique ;
- **DHCPINFORM** permet d'obtenir la configuration auprès d'un serveur *DHCP* sans demander à renouveler ou demander une nouvelle adresse IP ;
- **Query version.bind** demande au serveur DNS la version du service utilisé.

Ces datagrammes sont normalement compréhensibles par les services associés, provoquant la plupart du temps une réponse de leur part. Dans ce cas, le port sera marqué « *ouvert* ».

À l'inverse pour d'autres protocoles moins conventionnels pour lesquels il n'existe pas de datagrammes préconçus (ex. protocole propriétaire, application sur mesure…), **Nmap envoie un datagramme UDP vide**. En cas de non-réponse, le port sera marqué « *ouvert|filtré* ».

L'enregistrement réseau suivant illustre cette différence de comportement entre 3 ports UDP « conventionnels » et autre pour lequel Nmap n'a pas prévu de datagramme préconçu.

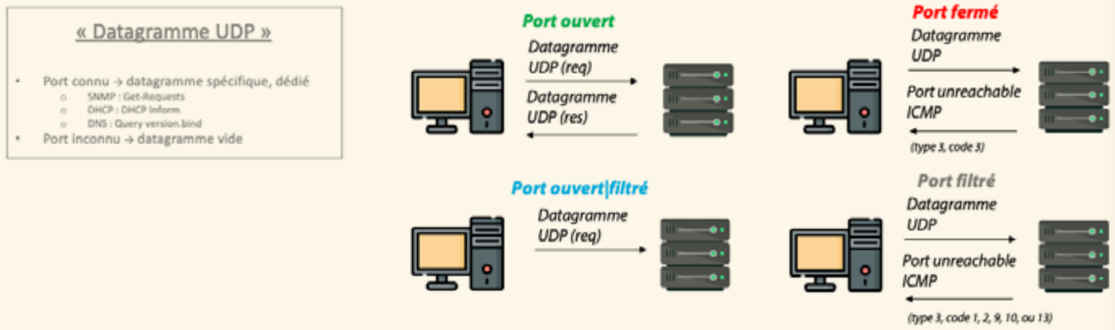
Source	Destination	TCP Port	Info
192.168.1.38	45.33.32.156		161 get-request 1.3.6.1.2.1.1.5.0
192.168.1.38	45.33.32.156		161 get-request
192.168.1.38	45.33.32.156	(1)	67 DHCP Inform - Transaction ID 0x1234567
192.168.1.38	45.33.32.156		67 DHCP Inform - Transaction ID 0x1234567
192.168.1.38	45.33.32.156		53 Standard query 0x0006 TXT version.bind
192.168.1.38	45.33.32.156		53 Server status request 0x0000
192.168.1.38	45.33.32.156		53 Standard query 0x0000 PTR _services._dns-sd._udp.local
192.168.1.38	45.33.32.156	(2)	1337 35807 → 1337 Len=0
192.168.1.38	45.33.32.156		13337 43955 → 13337 Len=0

Si un paquet est spécifiquement conçu pour le protocole utilisé, il sera envoyé comme paquet de détection (1).
Dans le cas contraire, un datagramme UDP vide sera envoyé (2).

Lorsqu'un paquet *vide, malformé* ou *incompréhensible* est envoyé à un service en UDP, **il est très rare qu'un message d'erreur soit renvoyé**. La pile réseau cible transmet simplement le datagramme vide à l'application en écoute, qui le rejette immédiatement sans renvoyer de message d'erreur.

Par conséquent lorsqu'aucune réponse n'est obtenue par Nmap, l'outil marque le port comme « *ouvert|filtré* », car il n'est pas capable de déterminer si cette absence de réponse provient d'un problème lié au datagramme lui-même (le port pourrait alors être considéré *ouvert*), ou d'un équipement intermédiaire filtrant la réponse renvoyée comme pour tout service fonctionnant sur TCP (le port serait alors considéré *filtré*).

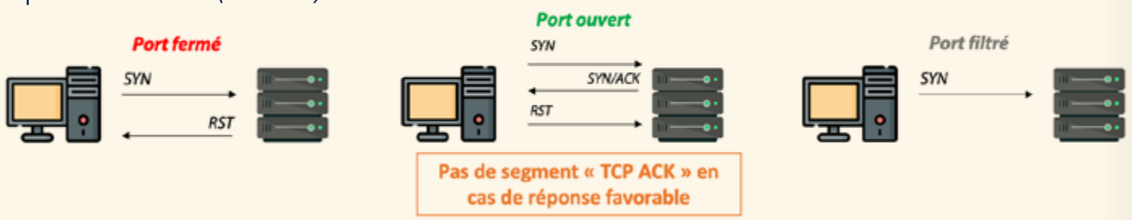
Le scan UDP classe ainsi de la façon suivante l'état du port testé :



Astuce d'auditeur
Malgré cette difficulté de classification inhérente au protocole UDP, ce type scan reste pertinent s'il est utilisé de façon plus ciblée et contrôlée à la recherche de services dits « conventionnels » (DHCP, DNS, SNMP, NTP, mDNS, etc.).
À ce sujet, le paramètre « --open » permet d'afficher les ports marqués « ouvert » seulement.

2.3.3 TCP Stealth

Enfin, le scan *TCP SYN (-sS)* initie un « *three-way handshake* », mais ne le termine pas en cas de réponse favorable *(TCP ACK)* du serveur :



Cette méthode est recommandée par Nmap, car effectivement plus rapide, et apparemment plus discrète que son homologue *TCP Connect*. Cette assertion était peut-être vraie il y a 25 ans, mais ne l'est plus aujourd'hui. Les équipements de détection (IPS, IDS, NDR, XDR, etc.) ont largement évolué et cette métrique ne permet plus de ne pas être détecté. Bien au contraire, le paquet envoyé est immuable et similaire pour tous les utilisateurs de Nmap, qu'importe l'OS utilisé. Ce dernier est donc facilement reconnaissable via un mécanisme de signature intégré par un équipement de détection.

Astuce d'auditeur
Malgré son avantage en termes de rapidité, cette méthode est à proscrire dans le cadre d'un audit interne. L'utilisation du paramètre « -sS » est fortement déconseillée. Dans le cadre de l'utilisation d'un VPN de niveau 3 par exemple (dits « TUN », la plupart le sont, comme Wireguard et OpenVPN), cette méthode échouera.

Les paquets forgés directement par un logiciel tiers comme Nmap puis envoyés sur le réseau sans passer par la pile TCP/IP du système d'exploitation peuvent ne pas être pris en charge par les clients VPN.

Enfin et surtout, cette méthode de scan peut conduire au déni de service d'un pare-feu intermédiaire, pouvant bloquer l'accès à de nombreuses ressources internes.

En effet, les pare-feux aujourd'hui sont des pare-feux « à état » (ou « stateful ») : ils gardent en mémoire l'état des connexions réseau établies qui le traversent. Ce mécanisme assure le suivi de l'état des connexions réseau à des fins de facilité d'administration et de sécurité. L'état de ces connexions est stocké au sein d'une table de session, pouvant être surchargé lors de l'utilisation d'un scan ***TCP Stealth*** en fonction de la configuration adoptée par le pare-feu.

Dans le cas où un nombre important de connexions étaient autorisées sur un tel pare-feu, et que l'établissement de ces dernières venait à être initié (*TCP SYN* : état du pare-feu « **SYN_RECEIVED** » ou « **SYN_SENT** ») sans être proprement terminé (3-way handshake complété : état du pare-feu « **ESTABLISHED** »), un incident pourrait survenir.

En général, pour ne pas surcharger la table de sessions, les connexions « **ESTABLISHED** » sont automatiquement supprimées lors d'une durée d'inactivité trop importante (ex. 1h). Il s'agit d'un comportement de nettoyage par défaut.

Cependant, ce n'est pas aussi simple pour les connexions dont le handshake TCP n'est pas encore terminé, c'est-à-dire toujours dans l'état « **SYN_RECEIVED** » ou **SYN_SENT** ». En général, il est peu commun qu'un handshake TCP ne termine pas. Si le segment final *(TCP ACK)* n'est pas reçu, c'est qu'un problème réseau survient et le pare-feu n'a qu'à attendre que celui-ci soit renvoyé ultérieurement par le client (le protocole TCP étant fiable) pour clôturer le handshake TCP. Pendant ce court laps de temps, ces connexions en mémoire ne peuvent pas être nettoyées : le pare-feu doit les garder en mémoire (comportement par défaut), et doit attendre que le segment final soit renvoyé.

Si ce comportement apparaît pour un seul port, cela ne posera aucun souci. En revanche, la conclusion n'est pas la même si des milliers, dizaines voire centaines de milliers de connexions sont dans l'attente d'une clôture du handshake TCP…

Dans ce cas, cette anomalie peut potentiellement mener à une surcharge de la table de session puis au déni de service du pare-feu via une consommation excessive des ressources allouées ou à une incapacité à traiter d'autres flux. D'ailleurs, ce problème porte un nom : l'attaque SYN Flood. Pour éviter de mener ce type d'attaque au sein du réseau d'entreprises, il est préférable d'utiliser un scan *TCP Connect* dont les connexions proprement terminées peuvent légitimement être nettoyées au sein des tables de sessions.

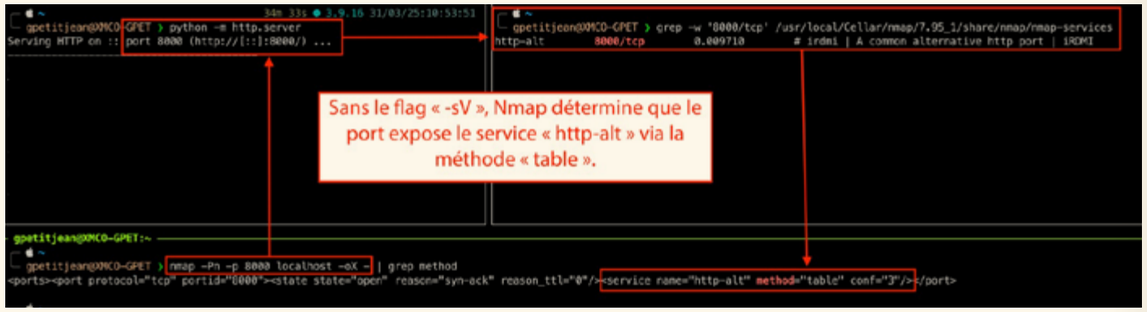
2.4 Identification du service exposé

Une fois un port **ouvert** (ou « **ouvert|filtré** ») détecté suite à l'utilisation du scan *TCP Connect* ou UDP, on peut utiliser Nmap pour tenter d'identifier dynamiquement le service qui s'y cache si le paramètre **–« sV »** est spécifiée. Des sondes de tests spécifiques aux protocoles les plus conventionnels seront alors envoyées afin d'identifier le service exposé.

2.4.1 Cas généraux

Pour identifier un service, Nmap dispose de deux méthodes principales : la méthode « **table** » et la méthode « **probed** ».

Pour la méthode **table**, Nmap dispose d'une liste de paires « **protocole / port** » utilisées par défaut (fichier « **share/nmap/nmap-services** »). Par exemple, il est courant de retrouver des applications **http** sur le port **80/tcp**, ou des services **ssh** sur le port **22/tcp**. Si le port est considéré **ouvert** et que Nmap dispose d'une correspondance au sein de ce fichier, l'outil considérera le service associé comme exposé.



Cette méthode n'est évidemment pas fiable, car elle repose seulement sur les comportements par défaut (et défini par Nmap). Agissant à la couche TCP seulement, elle ne permet pas non plus d'identifier un service fonctionnant sur UDP ou de récupérer de potentielles informations renvoyées par le service, comme sa bannière.

La deuxième méthode, appelée « **probed** », consiste alors à envoyer des paquets (des **sondes** de détection, ou « **probe** ») pour identifier dynamiquement le service TCP ou UDP exposé, à l'aide d'expressions régulières basées sur la réponse reçue par ces sondes. Bien que plus longue, cette méthode est considérée plus fiable et plus précise, agissant à la couche applicative (ex. une requête *HTTP GET*).

Les 187 sondes de détection (103 pour TCP, 84 pour UDP) sont renseignées dans le fichier « **share/nmap/nmap-service-probes** ». Elles suivent la syntaxe suivante, illustrée via un exemple pour un joindre un port web (ex. 80/tcp) :

```
Probe TCP GetRequest q|GET / HTTP/1.0\r\n\r\n|
Totalwaitms 1000
Rarity 1
Ports 1,70,[...],80,[...]
Match http m|^HTTP/1\.0 302 FOUND\r\nContent-Type: text/html; charset=utf-8\r\nLocation: http://([\w._-]+):\d+/
login\?next=%2F\r\n(?:[^\r\n]+\r\n)*?Server: Werkzeug/([\w._-]+) Python/([\w._-]+)\r\n|s p/Werkzeug httpd/ v/$2/ i/Flask web
framework; Python $3/ h/$1/ cpe:/a:python:python:$3/
[...]
Softmatch http m|^HTTP/1\.[01] (?!400)\d\d\d.*\r\nDate: .*\r\nServer: Apache ([^\r\n]+)\r\n| p/Apache httpd/ i/$1/ cpe:/
a:apache:http_server/
Fallback OptionRequest
```


D'une part, en ce qui concerne la sonde :

- **Probe** détails sur la requête envoyée ;
  - fonctionnant sur TCP (pourrait être UDP);
  - nommée « *GetRequest* » ;
  - la donnée envoyée en ASCII est « *GET / HTTP/1.0\r\n\r\n* » ;
- **Totalwaitms** durée pendant laquelle Nmap patiente de recevoir une réponse
- **Rarity** indicateur de rareté entre 1 (très courant) et 9 (très rare), utile pour le paramétrage de l'envoi des sondes (détaillé plus tard) ;
- **Port** liste de ports « communs » sur lesquels on pourrait s'attendre à recevoir une réponse. Si le port couramment scanné est spécifié via cette liste, le paquet sera envoyé.

D'autre part, concernant la réponse à cette sonde :

- **Match** expression régulière visant à déterminer **exactement** (au caractère près) le service exposé;
- **Softmatch** expression régulière visant à déterminer **approximativement** le service exposé;
- **Fallback :** si aucune sonde ne donne un match concluant, Nmap peut faire appel à une sonde générique ou revenir à une méthode plus basique pour tenter d'identifier le service.

Un point technique concernant les regex des directives **match** et **softmatch** :

- L'expression spécifiée après le protocole (ici, http) via « m|regex|s » correspond à un serveur web Python Werkzeug avec une version arbitraire **Werkzeug/([\w._-]+) Python/([\w._-]+)**;
- L'expression spécifiée ensuite correspond au résultat qui sera retourné dans la colonne « service ». Les développeurs ici ont choisi d'y inclure les informations suivantes, le service Python Werkzeug renvoyant par défaut ces données :
  - La nature du service **(p/vendorproductname/) : Werkzeug httpd**
  - Son numéro de version **(v/version/) : ([\w._-]+)**
  - Des données complémentaires **(i/info/) : Flask web framework ; Python ([\w._-]+)**
  - Le nom d'hôte **(h/hostname/) : ([\w._-]+)**
  - Le CPE associé **(cpe:/cpename/) :python:python:([\w._-]+)**

Astuce d'auditeur
À l'aide de ces directives, la méthode « probed » vise à obtenir une réponse de la part du service cible. Une telle réponse peut contenir des informations pertinentes, comme le type d'application, un numéro de version ou la nature du système d'exploitation de l'hôte sous-jacent.

Une telle description est ainsi définie pour les 187 sondes définis par Nmap. Pour déterminer quelle sonde de détection est à envoyer, Nmap se comporte comme suit. Les étapes se succèdent tant qu'aucun résultat (**match/softmatch**) n'est obtenu.

1. La sonde **NULL** est toujours essayée en première. Cette sonde n'envoie en fait aucune donnée, et attend que le service cible renvoie sa bannière (ce qui est le cas d'OpenSSH ou MS-SQL par exemple, ce qui explique pourquoi on obtient généralement leur version) ;
2. Toutes les sondes dont le port analysé est listé comme port « ***probable*** » (i.e. sont listées par la directive « **Port** ») sont essayées dans l'ordre où elles apparaissent dans le fichier « **nmap-service-probes** » ;
3. Toutes les autres sondes dont la valeur de rareté est **inférieure ou égale** à la valeur paramétrée pour l'analyse courante sont essayées, également dans l'ordre où elles

apparaissent dans le fichier « **nmap-service-probes** ». La valeur par défaut est positionnée à **7** et peut être modifiée via le paramètre « *--version-intensity [n]* ».

Astuce d'auditeur
Si aucune contrainte de temps n'est à considérer, l'option « --version-all » permet d'abaisser l'indice de rareté à 1, et d'envoyer toutes les sondes définies par Nmap. À l'inverse, l'option « --version-light » l'augmente à 9 pour n'envoyer que les sondes les plus basiques, comme « \r\n\r\n » ou « GET / HTTP/1.0\r\n\r\n » pour TCP.

Si les contraintes de temps le permettent, il est ainsi recommandé d'utiliser la méthode « probed » via « -sV » et d'augmenter l'indice de rareté le plus possible via le flag « --version-intensity [n]».

Prenons donc le cas d'un service peu commun exposé sur le port 80/tcp: celui-ci ne renvoie pas sa bannière **(1)**, Nmap envoie donc dans l'ordre toutes les sondes spécifiant le port 80 au sein de sa liste **(2)**. Si aucun **match** n'est obtenu, Nmap va alors envoyer tous les autres paquets dont l'indice de rareté est inférieur ou égal à l'indice spécifié **(3)**.

Si la valeur obtenue dans le procédé décrit ci-dessus correspond à une expression régulière **match**, la valeur est renvoyée selon les règles d'analyse syntaxique associées.

Dans le cas où un **softmatch** est d'abord identifié, Nmap va continuer ses recherches pour espérer affiner son résultat, permettant ainsi à un **match** d'être trouvé par la suite. Ce comportement permet d'une part d'envoyer le résultat le plus précis obtenu, d'optimiser la recherche par la suite, et de quand même envoyer un résultat .

En effet, si une regex **softmatch http** est valide, Nmap sait alors que le service est probablement http. Dans ce cas, il n'essaiera plus les autres protocoles (ex. **match/softmatch ssh**), ni les autres **softmatch http** pour gagner du temps. Si un **match** est ensuite identifié, ce résultat est utilisé. Sinon, les informations identifiées sont renvoyées comme décrit par la directive **softmatch**.

Spécificité Nmap
En fonction de la confiance de Nmap sur la nature du service qu'il identifie, l'outil attribue une valeur de 1 (peu confiant) à 10 (très confiant). Globalement :

- Une confiance à 3 correspond généralement à la méthode « table » ;
- Une confiance à 7 ou supérieure peut correspondre à la méthode « probed » via un softmatch ;
- Une confiance à 10 peut correspondre à la méthode « probed » via un match.

Cet attribut se retrouve dans le fichier de sortie XML activé via le paramètre « -oX » ou « -oA » (attribut « conf »).

Astuce d'auditeur
Il arrive que certains équipements intermédiaires comme des WAF ou des reverse-proxy soient paramétrés pour compléter le 3-way handshake TCP d'un serveur qu'il protège, qu'importe le port spécifié. L'ensemble des ports scannés apparaissent alors « ouvert » si seule la méthode « table » est utilisée.

Dans ce cas, un moyen plus fiable pour déterminer si un port expose réellement un service est d'utiliser le paramètre « -sV » avec l'option de sauvegarde XML à la recherche de ports dont l'attribut de confiance est supérieur à 7, c'est-à-dire ceux qui ont réellement répondu à une sonde de test sur la couche applicative.

Les détails résumés ci-dessus permettent de cette manière à Nmap de déterminer, avec plus ou moins de fiabilité et sur des couches réseau différentes si un port ouvert expose un service. Dans la plupart des cas, l'utilisation de Nmap par un auditeur s'arrête dès lors qu'il a déterminé quels systèmes sont joignables, quels ports sont ouverts et dans le meilleur des cas, quels services sont exposés. Cependant, davantage d'informations peuvent être récupérées via l'activation d'autres fonctionnalités, dont les détails seront abordés et nuancés dans les sections suivantes.

2.4.2 Cas particuliers
Certains cas particuliers peuvent subvenir lors d'un scan de port avec « *-sV* ». Les plus courants sont cités ci-dessous.

2.4.2.1 Service Fingerprint (SF)
Dans le cas où un service répond à une sonde envoyée sans pour autant que Nmap ne puisse déterminer sa nature, donc que la réponse ne valide aucun **match/softmatch**, un résultat comme le suivant sera renvoyé :

```
SF-Port21-TCP:V=3.40PVT16%D=9/6%Time=3F5A961C%r(NULL,3F,»220\x20stage\x20F
SF:TP\x20server\x20\(Version\x202\.1WU\(1\)\+SCO-2\.6\.1\+-sec\)\x20ready\
SF:.\r\n»)%r(GenericLines,81,»220\x20stage\x20FTP\x20server\x20\(Version\x
SF:202\.1WU\(1\)\+SCO-2\.6\.1\+-sec\)\x20ready\.\r\n500\x20”:\x20command\
SF:x20not\x20understood\.\r\n500\x20”:\x20command\x20not\x20understood\.\
SF:r\n»);
```

Ce résultat correspond à un « *Service Fingerprint* », comportement obtenu lorsque la réponse obtenue au paquet de test ne permet pas à Nmap d'identifier le service. Nmap propose alors à la communauté de soumettre les informations connues (nature du service, version, etc.) via une URL renvoyée sur la sortie standard (https://nmap.org/cgi-bin/submit.cgi?new-service). Même si peu lisible, son contenu peut facilement être parsé pour analyser le résultat.

2.4.2.2 TCPWrapped
« *Tcpwrapped* » fait référence à « *tcpwrapper* », un programme de contrôle d'accès au réseau basé sur l'hôte sous Unix et Linux.

Un port peut être marqué tcpwrapped lorsque le **3-way handshake TCP** est complété, mais que l'hôte a ensuite directement fermé la connexion (**TCP RST**). Ce comportement peut typiquement arriver lorsque le service distant rejette la connexion, car le paquet envoyé par Nmap n'est pas du format attendu par l'application, que la machine exécutant Nmap n'est pas autorisée à joindre le service scanné (ex. whitelist d'adresse IP) ou qu'un IDS/IPS bloque la connexion.

2.4.2.3 « http? »
Enfin, si un résultat est trouvé via la méthode « **table** » mais qu'il n'a pas pu être enrichi via la méthode « **probed** », un point d'interrogation est rajouté au résultat initial.

2.5 Identification du système d'exploitation

L'approche à adopter lors de tests d'intrusion dépend fortement de la nature du ou des systèmes d'exploitation identifiés. Nmap intègre une fonctionnalité visant justement à identifier le système d'exploitation du serveur ciblé. Ce mécanisme peut être activé en utilisant le flag « **-O** ». Il est également embarqué, entre autres, via l'activation du paramètre « **-A** ».

2.5.1 Approche adoptée par Nmap

L'identification de système d'exploitation repose en partie sur de **subtiles ambiguïtés dans l'implémentation des protocoles *TCP, UDP* et *ICMP* entre les différents systèmes d'exploitation**, notamment au niveau des **en-têtes** des réponses reçues. La principale raison pour laquelle les systèmes d'exploitation ont des valeurs différentes est que **les RFC associées ne stipulent pas de valeurs par défaut**.

Par exemple pour TCP, l'en-tête de réponse *TCP Window Size pour Windows XP* était historiquement de ***65535***, et celle de *Windows 7* était de *8192*. Encore, couche IP, la valeur *TTL* (Time To Live) est historiquement paramétrée par défaut à *128* sur *Windows*, et à *64* pour les distributions *Linux*.

De fait, il devient possible d'exploiter ce type de subtilités pour tenter de reconnaître l'OS utilisé. Les choix des développeurs ayant programmé les systèmes d'exploitation dans l'attribution de ces valeurs se révèlent ainsi être d'excellents indices !

Compte tenu du nombre de systèmes d'exploitation disponibles, s'appuyer uniquement sur cette méthode s'avèrerait néanmoins insuffisant. Que se passerait-il si deux systèmes d'exploitation venaient à attribuer les mêmes valeurs par défaut ? Comment différencier deux systèmes d'exploitation similaires, comme deux distributions Linux ?

En plus de tirer profit des différences dans l'implémentation des RFC, l'identification d'OS par **Nmap repose surtout l'envoi de sondes spécifiquement conçues** et la **génération d'une empreinte** se basant sur les réponses obtenues. Cette empreinte se veut être vue comme la signature d'un seul système d'exploitation. Une fois obtenue, elle est ensuite comparée avec celles préalablement enregistrées par la communauté de développeurs d'Nmap au sein du fichier servant de base de données : « **share/nmap/nmap-os-db** ».

Spécificité Nmap
Étant open source, l'adoption de cette démarche démontre ici aussi que Nmap se base sur la communauté pour améliorer la précision de ses résultats.

L'objectif affiché avec ces sondes devient ainsi d'exploiter l'ensemble des valeurs non attribuées par défaut par les RFC afin de créer en sortie les résultats les plus hétérogènes possibles. En augmentant cette hétérogénéité, Nmap améliore l'unicité de chaque signature, facilitant ainsi la reconnaissance de chaque système d'exploitation testé.

2.5.2 Zoom technique : génération d'une empreinte

L'indication du paramètre « **-O** » active ainsi **l'envoi de 16 nouvelles sondes** : *13 segments TCP, 1 datagramme UDP* et *2 paquets ICMP*. Pour chacune de ces sondes, les champs des en-têtes sont minutieusement paramétrés. Nmap joue surtout avec les en-têtes *TCP Window Size*, les drapeaux (*flags*, comme *SYN, URG, PSH*) et l'indicateur IP de fragmentation (*IP DF*).
La génération d'une empreinte ne fait appel à aucun calcul cryptographique, comme nous aurions pu penser, à l'aide de fonction de hachage par exemple. En effet pour chaque sonde envoyée, Nmap stocke plutôt le résultat sous forme de couple « clé/valeur » bien définies, délimitée par le séparateur « **%** ».

- Les clés correspondent aux en-têtes manipulées par Nmap (ex. « **T** » pour le *TTL IP*);
- Les valeurs sont soit des caractères définissant un comportement (par exemple « **Y** » pour « les bits DF sont définis »), soit la valeur même du champ de l'en-tête, en hexadécimal (par exemple, « **W1=FFFF** » pour un *TCP Window Size à 65535*).

Concaténées bout à bout, 13 chaines de caractères sont générées et représentent la signature finale de l'OS.
Le bloc suivant représente ainsi la signature du système d'exploitation « *Microsoft Windows 10 1709 - 21H2* » :

```
Fingerprint Microsoft Windows 10 1709 - 21H2
Class Microsoft | Windows | 10 | general purpose
CPE cpe:/o:microsoft:windows_10 auto
SEQ(SP=FD-10C%GCD=1-6%ISR=103-111%TI=I%CI=I%II=I%SS=S%TS=U)
OPS(O1=M[54D-5BC]NW8NNS%O2=M[54D-5BC]NW8NNS%O3=M[54D-5BC]NW8%O4=M[54D-5BC]NW8NNS%O5=M[54D-5BC]NW8NNS%O6=M[54D-5BC]NNS)
WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FF70)
ECN(R=Y%DF=Y%T=7B-85%TG=80%W=FFFF%O=M[54D-5BC]NW8NNS%CC=N%Q=)
T1(R=Y%DF=Y%T=7B-85%TG=80%S=O%A=S+%F=AS%RD=0%Q=)
T2(R=Y%DF=Y%T=7B-85%TG=80%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)
T3(R=Y%DF=Y%T=7B-85%TG=80%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)
T4(R=Y%DF=Y%T=7B-85%TG=80%W=0%S=A%A=O%F=R%O=%RD=0%Q=)
T5(R=Y%DF=Y%T=7B-85%TG=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
T6(R=Y%DF=Y%T=7B-85%TG=80%W=0%S=A%A=O%F=R%O=%RD=0%Q=)
T7(R=Y%DF=Y%T=7B-85%TG=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
U1(DF=N%T=7B-85%TG=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
IE(DFI=N%T=7B-85%TG=80%CD=Z)
```

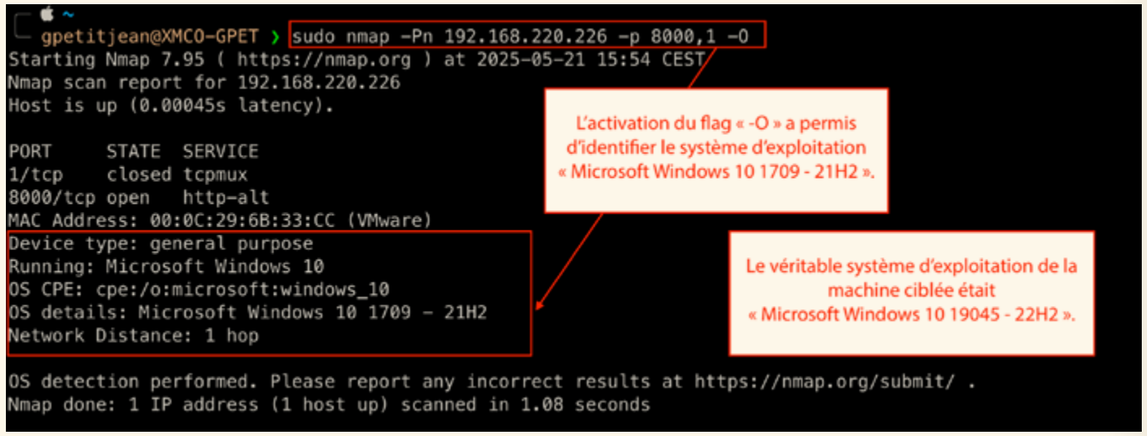
Les lignes **SEQ, OPS, WIN, ECN, T1, T2, T3, T4, T5, T6, T7, U1, IE** correspondent à l'analyse stockée sous format couple « clé/valeur ».

Nous ne rentrerons pas dans les détails de chacune de ces 13 analyses, mais en voici l'explication à des fins d'illustration pour l'une d'entre elles (**T1**, ligne 9) :

Clé/Valeur	Détail
R=Y	L'hôte a répondu (Y = Yes)
DF=Y	Le bit *Don't Fragment* est activé dans l'en-tête IP
T=7B-85	Plage des valeurs *TTL* calculées dans la réponse de plusieurs sondes (intervalle entre 0x7B et 0x85 = 123–133, confirmant la probabilité d'être un Windows)
TG=80	Valeur estimée de la granularité *TTL* (en général, 128 ou 64 ou 255) en fonction de la plage T obtenue
S=0	La valeur du numéro de séquence du segment *TCP SYN* de l'hôte est une valeur arbitraire
A=S+	La valeur du numéro de séquence du segment *TCP ACK* de l'hôte correspond au segment SYN de la sonde envoyée, augmentée de 1 (comportement attendu)
F=AS	Les flags *TCP* obtenus dans la réponse à la sonde sont S (*SYN*) et A (*ACK*), typiques d'un SYN-ACK
RD=0	Le calcul du CRC32 (algorithme de somme de contrôle) du message d'erreur d'un éventuel segment RST est nul. Le résultat est cohérent au vu de la valeur *F* précédente (*SYN-ACK*).
Q=	Champ vide, utilisé parfois pour des options spéciales ou remarques

Une fois la signature générée pour l'hôte scanné, Nmap compare ses valeurs avec celles de chaque système d'exploitation parmi les 6000 et quelques enregistrés dans sa base de données. À l'aide

d'un système de score, il enregistre les résultats les plus probables. Ce score permet également de pondérer certaines comparaisons, le TTL constituant par exemple un meilleur indice que le *TCP Window Size*.



Par ailleurs, deux prérequis doivent être réunis pour utiliser l'option « **-O** » :

- Les paquets étant forgés, Nmap doit être utilisé en tant qu'**utilisateur privilégié** (administrateur local/root) ;
- L'analyse se basant notamment sur la différence entre un port ouvert et un port fermé, au moins **un port doit être ouvert**. L'identification d'un autre port, **fermé**, augmentera grandement la qualité de l'analyse.

Enfin, l'utilisation de ce mécanisme augmente la durée du scan (sondes supplémentaires envoyées et traitement / parsing associé) et n'est donc pas à utiliser par défaut si des contraintes de temps fortes sont à considérer. De la même manière, pour **les infrastructures sensibles** comme les systèmes industriels ou certains environnements internes, il est très fortement recommandé de **ne pas utiliser ce type de scan** qui se repose sur l'envoi de paquets volontairement malformés.

2.5.3 Approche à favoriser pour des tests d'intrusion

Bien qu'ingénieuse et plutôt élégante, cette approche présente certaines limites et **il convient de nuancer la fiabilité des résultats**, à contrario de ce qui est écrit dans le manuel. Malgré la présence réelle d'ambiguïtés présentées précédemment entre les différents OS, la méthode proposée par Nmap ne peut donner qu'une indication approximative se reposant sur des données collaboratives (comme prouvée sur la capture d'écran précédente).

De plus, les OS modernes adoptent de plus en plus des comportements réseau similaires par défaut, rendant leurs empreintes plus difficiles à distinguer. Par exemple, les *TCP Window Size* sont désormais dynamiques, et non codés en dur dans la pile TCP de l'OS comme par le passé. D'autre part, les réseaux se sont grandement complexifiés et de plus en plus de composants intermédiaires perturbent aujourd'hui les données des en-têtes. Certaines machines virtuelles ou environnements cloud appliquent également des politiques réseau qui masquent ou uniformisent les réponses TCP/IP, qu'importe l'OS de l'hôte en question. Enfin, cette approche repose sur une base de données limitée (même si plutôt complète avec ses quelques 6000 enregistrements).

Voici d'autres indices pouvant être utilisés et croisés pour identifier l'OS utilisé, du plus au moins fiable :

- La **bannière** de certains services **divulgue** par défaut le système d'exploitation pour lequel ils sont compilés. Le service *DNS « BIND »* renvoie par exemple sa version et celle du système d'exploitation : **9.11.4-P2 (RedHat Enterprise Linux 7)**.
- Certains **services** ou **applications** sont typiques de systèmes d'exploitation (cf. *section 1.2.1, Détection d'hôtes*), par exemple :
  - **Linux :** SSH (22/TCP), RPC (111/TCP), NFS (2049/TCP) ; Apache (80/TCP), Oracle (1521/TCP) et Tomcat (8080/TCP) ;
  - **Windows :** services IIS et HTTPAPI (80/TCP), RDP (3389/TCP), RPC (135/TCP) et SMB (139 et 445/TCP) ;
  - **MacOS / iOS :** AirTunes rtspd (5000 et 7000/TCP)
- Les **infrastructures réseau** sont souvent créées de façon cohérente au sein d'un réseau interne. Par exemple, les postes de travail Windows des collaborateurs sont souvent dans un ou plusieurs VLAN dédiés.
- Les **préfixes d'adresse MAC**, ou « *Organizationally Unique Identifier* » (« OUI ») peuvent également révéler la nature de l'OS (exemple : préfixe « 70-F8-AE »)
- La **nomenclature des noms DNS** peut parfois révéler la nature de l'OS selon le type de serveur (surtout vrai pour les contrôleurs de domaines, par exemple *SRVDC01*).

Astuce d'auditeur
La méthode adoptée par Nmap pour déterminer le système d'exploitation de sa cible ne doit ainsi pas être considérée à elle seule comme un moyen fiable ou déterministe d'identifier un système d'exploitation. Couplée à d'autres indicateurs, elle demeure néanmoins forte utile et pertinente.

2.6 Nmap Script Engine (NSE)

Une fois un service déterminé sur un hôte donné, la phase de recherche et d'exploitation de vulnérabilités peut démarrer. Nmap met à disposition une fonctionnalité dédiée à cette étape : le « **NSE** », ou *Nmap Scripting Engin*e, permettant d'exécuter des scripts *Lua*. De la découverte réseau à l'identification de service, en passant par de la détection et l'exploitation de vulnérabilités, ces scripts visent à automatiser une partie ou l'entièreté de ces processus à l'aide d'un langage flexible et léger, maintenus par la communauté.

Lua est un langage léger, rapide, portable, distribué sous la licence libre MIT, et qui possède des coroutines pour une exécution parallèle efficace de scripts. Il s'agit d'un langage ayant été conçu pour être intégré, comme c'est le cas par Nmap qui embarque un interpréteur *Lua*.

Ce langage profite également d'une documentation fournie, bien qu'il soit considéré de niche, facilitant ainsi la tâche aux développeurs souhaitant s'atteler au développement de scripts pour NSE.

Au total, plus de 600 scripts ont été rédigés par la communauté et peuvent être retrouvés au sein du dossier « *share/nmap/scripts* ». Pour être publié dans la base de données existante, un script doit être tagué par une ou plusieurs des 15 catégories définies par Nmap, telles que :

- **Broadcast :** scripts incluant des opérations de broadcast réseau (ex. *llmnr-resolve.nse* tente de résoudre le nom d'hôte via le protocole LLMNR);
- **Auth :** scripts ciblant les mécanismes d'authentification du service concerné (ex. *ms-sql-empty-password.nse* tente de s'authentifier avec l'utilisateur « sa » et un mot de passe vide sur service de base de données MS-SQL) ;
- **Intrusive :** scripts ne pouvant pas être catégorisés comme « **safe** » (ex. *http-enum.nse* tente d'énumérer les répertoires utilisés par les applications et serveurs web les plus courants) ;
- **Vuln :** scripts vérifiant l'existence de vulnérabilités (ex. *smb-vuln-ms17-010.nse* tente de détecter si vulnérable à EternalBlue) ;
- **Exploit :** scripts visant à exploiter des vulnérabilités (ex. *http-majordomo2-dir-traversal.nse* tente d'exploiter la vulnérabilité CVE-2011-0049).

```
/usr/local/Cellar/nmap/7.95_1/share/nmap/scripts
gpetitjean@XMCO-GPET ❯ grep -H "categories =" *.nse
acarsd-info.nse:categories = {"safe","discovery"}
address-info.nse:categories = {"default", "safe"}
afp-brute.nse:categories = {"intrusive", "brute"}
afp-ls.nse:categories = {"discovery", "safe"}
afp-path-vuln.nse:categories = {"exploit", "intrusive", "vuln"}
afp-serverinfo.nse:categories = {"default", "discovery", "safe"}
afp-showmount.nse:categories = {"discovery", "safe"}
ajp-auth.nse:categories = {"default", "auth", "safe"}
ajp-brute.nse:categories = {"intrusive", "brute"}
ajp-headers.nse:categories = {"discovery", "safe"}
ajp-methods.nse:categories = {"default", "safe"}
ajp-request.nse:categories = {"discovery", "safe"}
```

Classement des scripts via les 15 catégories existantes.

L'utilisation d'un script *Lua* par le moteur NSE est assez simple. Un script peut être chargé de plusieurs façons :

```
# Scripts par défaut
nmap -sC [IP]

# Scripts spécifiques
nmap ---script [script-name] [IP]

# Par préfixe de nom (inclura tous les scripts « smb-*.nse »
nmap –-script 'smb-*' [IP]

# Par catégorie
nmap –-script 'vuln and not auth' [IP]
```

L'ensemble des scripts *Lua* proposés par Nmap sont disponibles directement depuis la documentation NSE sur le site officiel. Pour chacun d'entre eux est proposé un *bref sommaire*, les *potentiels arguments* requis ainsi que les *conditions d'exécution* nécessaires à son déclenchement, notamment **hostrule** ou **portrule**. En résumé :

- Une règle **portule** définit une condition ciblant un ou plusieurs ports. Grâce à cette règle, les scripts « *http-** » ne s'exécuteront que sur les ports définis comme *http* par Nmap, comme *80/tcp, 443/tcp* ou *8080/tcp*. Il s'agit des scripts qui peuvent donc s'exécuter plusieurs fois sur un même hôte.

- Une règle **hostrule** définit une condition sur un hôte. Grâce à cette règle, certains scripts ne s'exécuteront qu'une seule fois par hôte. Il s'agit notamment des scripts qui ne dépendent pas de ports spécifiques (ex. *whois-ip.nse*), ou dont on ne s'attend à les exécuter une seule fois (ex. scanner de vulnérabilité Eternal Blue sur SMB, *smb-vuln-ms17-010*).

Les règles **prerule** et **postrule** existent également, et s'exécutent respectivement avant et après la phase de scan. Elles concernent par exemple les scripts de broadcast (**prerule**) ou d'affichage des résultats (**postrule**).

Bien que NSE soit un module pouvant s'avérer utile, il reste un outil automatique réalisant des opérations potentiellement non contrôlées. Dans le cadre d'un audit professionnel et notamment pour des environnements sensibles, l'utilisation d'outil de ce type est à bannir dû au manque certain de maitrise associé et au risque qui en découle.

Certains scripts *Lua* peuvent à ce titre :
- Bloquer des comptes applicatifs (ex. *ms-sql-brute*) ;
- Saturer un service web via du fuzzing http (ex. *http-enum*) ;

- Contacter des services tiers (ex. *whois-ip*) ;
- Exploiter des vulnérabilités mémoire dangereuses, telles que du dépassement de tas (ou « *heap overflow* », ex. *smtp-vuln-cve2010-4344*) ;
- Porter volontairement atteinte à la disponibilité de l'hôte (ex. tous les scripts tagués « *dos* »).

Astuce d'auditeur
Nmap n'échappe pas à une règle capitale à laquelle se soumettre lors d'audits professionnels : il est essentiel de comprendre en détail le comportement adopté par les outils utilisés, et de pouvoir justifier l'intégralité des actions menées et leur innocuité sur un système d'information client.

L'utilisation du NSE peut toujours être considéré dans la mesure où son recours reste contrôlé. Il est donc préférable de sélectionner manuellement les scripts nécessaires, après lecture complète de leur contenu et validation de leur innocuité.

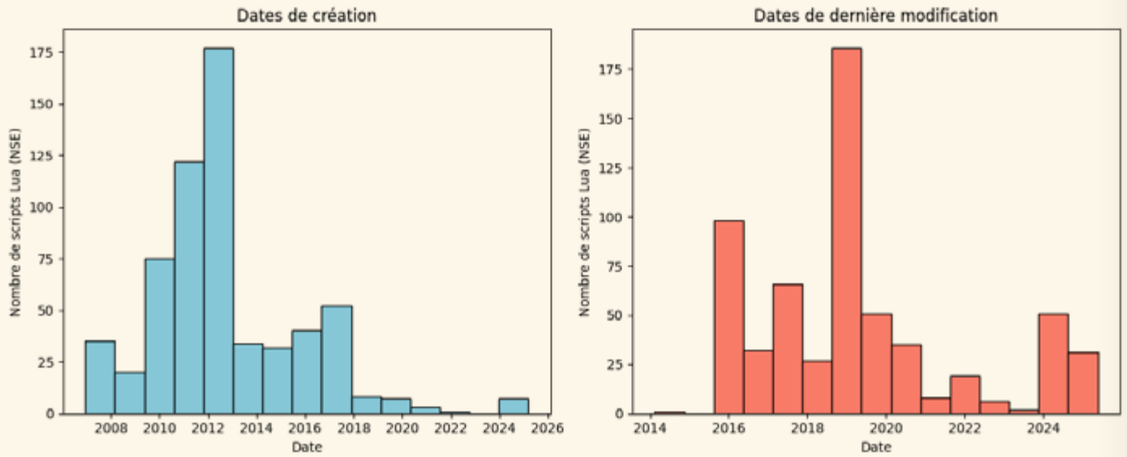
```
# Exemple à bannir

## Scripts par préfixe
nmap --script 'http-*' [IP]

## Scripts de catégorie à risque
nmap --script 'intrusive' [IP]

# Exemple à privilégier

## Script spécifique
nmap --script 'http-cookie-flags [IP]
```

En plus des risques liés à son utilisation non contrôlée, NSE demeure relativement peu utilisé en tant que scanner de vulnérabilité en tant que tel. De nombreuses raisons peuvent expliquer ce constat.

Parmi elles, nous pouvons noter qu'il ne s'agit que d'un module additionnel proposé par Nmap. NSE ne doit pas être considéré comme un scanner de vulnérabilité à part entière visant à identifier voire exploiter des failles de sécurité, contrairement à d'autres outils dédiés. Nmap reste avant tout un scanner de port, largement utilisé afin de déterminer la nature et tout au mieux la version d'un service exposé par un hôte. Nmap est en effet plutôt utilisé à des fins découverte voire de débogage réseau plutôt qu'à des fins d'exploitation logicielle.

En outre, le moteur NSE tente de centraliser des opérations de nature différente (fuzzing, bruteforce, fingerprinting, exploitation, etc.) et l'aspect couteau suisse d'un seul outil n'est généralement pas une approche privilégiée, car il est impossible de centraliser toutes les opérations d'une chaine d'exploitation. On préférera alors plutôt réaliser chaque tâche avec un outil dédié ou un script sur mesure. La syntaxe hétérogène entre les différents paramètres des scripts *Lua* complexifie également l'utilisation de ces derniers, freinant largement son utilisation.

Une autre explication réside dans le caractère désuet de certains scripts. La plupart des codes d'exploitation (catégorie « *exploit* ») concernent des CVE datant des années 2010, le plus récent étant un script pour une CVE datant de 2017. Le moteur de scripts peut toujours s'avérer ponctuellement utile pour de la collecte d'information, mais l'outil n'est plus tout à fait adapté à la recherche et à l'exploitation de vulnérabilité de nos jours. À cet égard, maintenir à jour des « *wordlists* » et scripts d'exploitation de CVE au fil du temps est une tâche exigeante et difficile qui semble avoir été abandonnée par Nmap.
La grande majorité des scripts *Lua* ont en effet été publiés dans les années 2010, et un nombre important d'entre eux n'ont pas reçu de mise à jour depuis, comme l'illustrent les graphiques suivants (statistiques basées sur le repository officiel de Nmap) :



Pour finir, un autre élément contribue à justifier le recours limité à NSE dans un cadre d'audit de sécurité. Comme tout autre scanner de vulnérabilité, NSE reste avant tout un outil automatique. La réalisation de tests d'intrusion est un exercice se démarquant justement de ce genre d'outil par de nombreux facteurs (compréhension du contexte métier, des contraintes liées à l'environnement, des besoins d'un client et des objectifs des tests), rendant l'utilisation de scanner de vulnérabilités peu populaire. Un pentest manuel offre une profondeur d'analyse, une capacité d'adaptation et de corrélation de failles qu'un outil automatique ne saurait égaler.

Astuce d'auditeur
L'utilisation du moteur NSE s'avère ainsi peu pertinente. Certains scripts « quick-win » peuvent toutefois s'avérer intéressants, notamment :

- « smb-security-mode » permet de déterminer si la signature SMB est requise ou activée. L'absence de signature permet notamment le relai NTLM vers ce protocole.
- « smb-vuln-ms17-010 » permet de déterminer si l'hôte est vulnérable à la vulnérabilité RCE black box Eternal Blue.
- « snmp-brute » peut s'avérer utile pour mener des authentifications par force brute sur SNMP.
- « ftp-anon » peut être intéressant pour tester l'authentification anonyme FTP sur un nombre important de serveurs en une seule fois.

3. Partie 2 : Pour aller plus loin

Nmap propose des options de paramétrage permettant aux utilisateurs d'adapter les scans à leurs besoins, qu'il s'agisse de rapidité ou de précision accrue. Les deux sections suivantes traiteront des options visant ainsi à optimiser l'un de ces deux facteurs.

Astuce d'auditeur
Il est important de noter que la rapidité d'exécution est étroitement liée à la précision des résultats. Améliorer l'un revient généralement à dégrader l'autre.

Enfin, les différentes fonctionnalités de débogage pouvant faciliter l'analyse de comportements inattendus de Nmap ou de problèmes réseau feront l'objet de la dernière section de cet article.

3.1 Accélérer un scan

Nmap met à disposition **17 options** permettant aux utilisateurs un paramétrage granulaire de son temps d'exécution. De nombreuses techniques permettent ainsi d'optimiser les performances et

d'améliorer la rapidité d'exécution des scans, ce qui peut s'avérer utile lorsque plusieurs dizaines de milliers de serveurs sont à scanner. Les valeurs par défauts assignées par Nmap peuvent être modifiées à l'aide d'un seul paramètre via l'utilisation de **templates**.

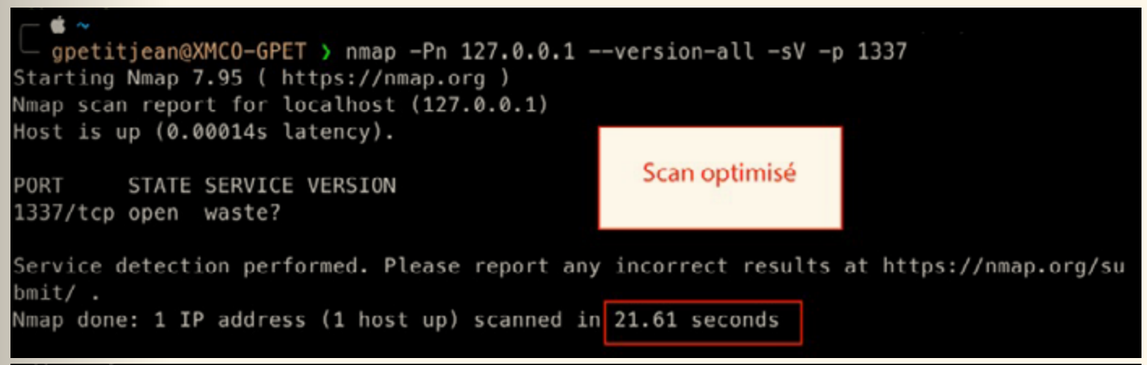
Cette partie traite des options recommandées afin d'améliorer les performances de scan d'un ou plusieurs sous-réseaux. Leur utilisation n'est à considérer qu'en cas d'excellentes performances réseau.

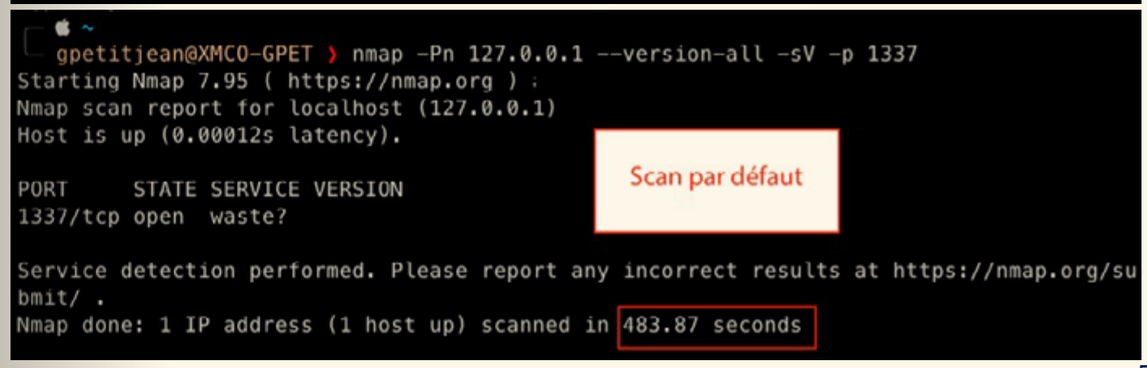
3.1.1 Options de performance sur les services

La première technique consiste à tirer profit d'une bonne réactivité du réseau en modifiant les directives *totalwaitms* et *tcpwrappedms* des fichiers de configuration. Ces deux mots clés définissent respectivement le temps d'attente de réponse des sondes envoyées via l'option « *-sV* » et pour les ports *tcpwrapped*. Initialement configurées sur des durées de **5 secondes ou plus par sonde**, elles forcent Nmap à attendre volontairement en cas de non-réponse aux requêtes envoyées (ce qui arrive la plupart du temps). Les sondes étant envoyées de façon consécutive sur un même hôte, l'absence de réponse rallonge considérablement la durée totale du scan.
Généralement, il n'est pas nécessaire d'attendre aussi longtemps pour obtenir une réponse d'un service applicatif. En diminuant ces deux directives, on considère que si un service ne répond pas rapidement, il ne répondra tout alors pas du tout.

```
# Supprimer les totalwaitms
sed -E -i '' '/^totalwaitms [0-9]+$/d' nmap-service-probes

# Ajouter pour chaque probe une durée spécifique (200)
sed -i '' '/Probe/ a\
totalwaitms 200
' nmap-service-probes

# Modifier la durée tcpwrappedms
sed -iE 's/^tcpwrappedms [0-9]+$/tcpwrappedms 100/' nmap-service-probes
```

En modifiant ces durées à *200* et *100ms*, un scan de version complet dure désormais **21 secondes**, contre **483 secondes** initialement.





3.1.2 Paramétrage un groupe d'hôtes

Les paramètres présentés ci-dessous traitent du parallélisme et du timeout apportés par Nmap sur les hôtes scannés.

Paramètre	Détail	Vleur par défaut	Valeur optimisée
--min-hostgroup	Pour le san de port et de service, Nmap rassemble les hôtes scannés par groupe. Chaque groupe est scanné itérativement, mais tous les hôtes sont scannés parallèlement au sein de ces groupes. Ce paramètre définit le nombre minimum d'hôtes insérés dans chaque groupe.	**1** puis **augmente dynamiquement** en fonction du réseau	**64, 128** voire **256**
--host-timeout	Définit la durée maximale tolérée par Nmap pour réaliser l'ensemble des opérations de scan sur un hôte (découverte d'hôte, détection de port, scans de service, etc.)	**Aucune** infini	**10 à 30m**
--script-timeout	Définit la durée maximale tolérée par Nmap pour exécuter un script *Lua* (NSE)	**Aucune** infini	**10 à 30m**

3.1.3 Paramétrage sur un hôte : couche applicative

Les paramètres présentés ci-dessous visent à limiter les opérations effectuées sur la couche applicative des hôtes scannés.

Paramètre	Détail	Vleur par défaut	Valeur optimisée
version-intensity	Réduit le nombre de paquets de tests envoyé lors de l'identification de service sur un port ouvert.	**7**	**3**
-n	Désactive la résolution inverse des hôtes considérés up. Bien qu'utile, s'affranchir de cette information aide également à accélérer un scan Nmap.	**Activée**	**Désactivée**

3.1.4 Paramétrage sur un hôte : couche TCP

Les paramètres présentés ci-dessous visent à optimiser les flux TCP échangés entre Nmap et les hôtes scannés.

Paramètre	Détail	Vleur par défaut	Valeur optimisée
--max-retries	Définit le nombre de sondes (ex. *SYN*) renvoyée par Nmap durant la phase de découverte d'hôte (« *Ping scan* ») en cas de non-réponse (ex. *SYN/ACK*) sur un port donné. Sur un réseau performant, il est rare qu'une non-réponse soit synonyme de perte d'un segment TCP.	**10**	**0** voire **1**
-sS	Active le « **Stealth scan** ». **Note : ce type de scan est fortement déconseillé au sein d'environnement sensible ou professionnel.**	**Activé** si *sudo*	**Activé**
--min-parallelism	Définit le nombre de sockets réseau minimum que Nmap va essayer de maintenir par hôte. **Note : Ce paramètre dépend fortement du réseau.**	**Dynamique**	**20** à **50**
--min-rate	Définit le nombre de sockets réseau minimum que Nmap va essayer de maintenir au global. **Note : Ce paramètre dépend fortement du réseau.**	**Dynamique**	**20** à **100**

3.2 Assurer une meilleure couverture de scan

Des astuces visant à collecter un maximum d'informations ont été proposées tout au long de l'article. Cette section adresse quelques combines supplémentaires permettant d'augmenter ses chances d'identifier un port ou un service.

Paramètre Nmap	Détail	Valeur par défaut
--version-all	Envoie toutes les sondes de détection du moteur d'identification de service «(*-sV*) proposées par Nmap.	**version-intensity à 7**
-sU	Active la reconnaissance de service UDP.	**Ports TCP seulement**
--script=+[script].nse	Rajouter un « **+** » permet de forcer l'utilisation d'un script NSE qu'importe le port spécifié	**Script non exécuté si le port n'est pas compris dans la liste de ports définis (règles *postrule* et *hostrule*).**
--all-ports	Active le moteur d'identification de service (*-sV*) sur les ports 9100 à 9107.	**Ports 9100 à 9107 non scannés**
-p$(cat modern_ports.txt)	Utiliser une liste de ports moderne et personnelle plutôt que d'utiliser « *--top-ports* ».	**Liste de port définie par Nmap pouvant être améliorée**

3.3 Déboguer un scan

3.3.1 Options de verbosité

En fonction notamment de la performance réseau et du volume d'hôte à considérer, il peut arriver qu'un scan Nmap ralentisse ou bloque, et ne termine pas. Un débogage rapide et clair aide alors à identifier la cause du problème.

Comme sur de nombreux autres outils en ligne de commande, Nmap dispose de plusieurs niveaux de verbosité : 3 exactement, pouvant être activés via le paramètre « **-vvv** ». La verbosité permet d'afficher des informations supplémentaires, mais ces dernières ne relèvent pas du débogage du scan. Son activation alerte par exemple l'utilisateur des étapes en cours (résolution DNS, scripts NSE, Ping scan, Stealth scan, etc.).

Pour autant, l'activation de la verbosité n'en reste pas moins capitale. Elle offre en effet aux utilisateurs les plus vigilants accès à des informations pouvant être perdues au fil du scan.

Spécificité Nmap

Malgré le faible avantage de cette option pour le débogage, cette dernière s'avère indispensable dans un cas bien précis, mais qu'il n'est pas rare de rencontrer lors de tests internes.

Comme indiqué précédemment, l'option « --host-timeout » défini la durée maximale tolérée par Nmap pour réaliser l'ensemble des opérations de scan sur un hôte.

Si cette valeur est dépassée, Nmap arrête le scan et abandonne cet hôte. L'ensemble des résultats associés sont alors complètement perdus : ils ne sont ni enregistrés dans les fichiers de sortie, ni dans l'affichage final dès le scan terminé.

Astuce d'auditeur

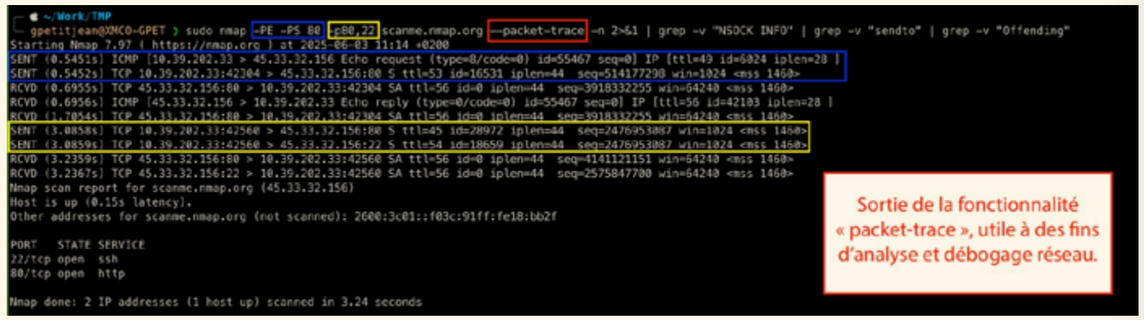
Pour cette raison, il est conseillé de toujours activer au moins le premier niveau de verbosité, et d'enregistrer les résultats au fur et à mesure du scan dans un fichier distinct. Un script de parsing dédié pourrait être utilisé pour assurer la fiabilité des résultats enregistrés.

3.3.2 Autres options d'affichage

En plus de la verbosité à 3 niveaux, Nmap dispose de cinq autres options de débogage. Comme la verbosité, ces paramètres sont désactivés par défaut.

Paramètre Nmap	Détail
-d	Fournit des informations de débogage (à 9 niveaux maximum « **-d9** »), notamment la valeur attribuée pour les paramètres d'optimisation, comme **min-hostgroup** ou **host-timeout**. Le premier niveau « **-d** » active également la verbosité standard (« **-v** »).
--packet-trace	Indique quels paquets sont envoyés sur quels hôtes/ports. Cette option permet notamment de s'affranchir de Wireshark pour un débogage rapide.
--version-trace	Indique quels paquets de détection sont activés via « *-sV* ».
--script-trace	Indique quels paquets sont activés via *NSE*.
--traceroute	Indique le chemin qu'ont pris les paquets envoyés par Nmap (« *hops* »), utile au sein d'un réseau complexe pour analyser le routeur de sortie des flux réseau.

Parmi toutes ces options, packet-trace est particulièrement efficace pour les débogages réseau. Elle permet notamment de confirmer que Nmap envoie bien le type de sonde désiré, d'identifier de potentiels blocages réseau, ou de comprendre le comportement adopté par l'outil lors de tests.



Astuce d'auditeur
Les paramètres de verbosité, de débogage et de suivi des paquets peuvent être augmentés dynamiquement au cours du scan via les touches « v », « d » puis « p ».
La combinaison avec la touche « ⇧ Shift » permet de les diminuer.

3.3.3 Interruption d'un scan

Un scan sur un nombre de ports ou d'hôtes important exige une durée d'exécution considérable, sans interruption. Au moins deux astuces combinant plusieurs paramètres peuvent être adoptées pour interrompre un scan et le reprendre par la suite.

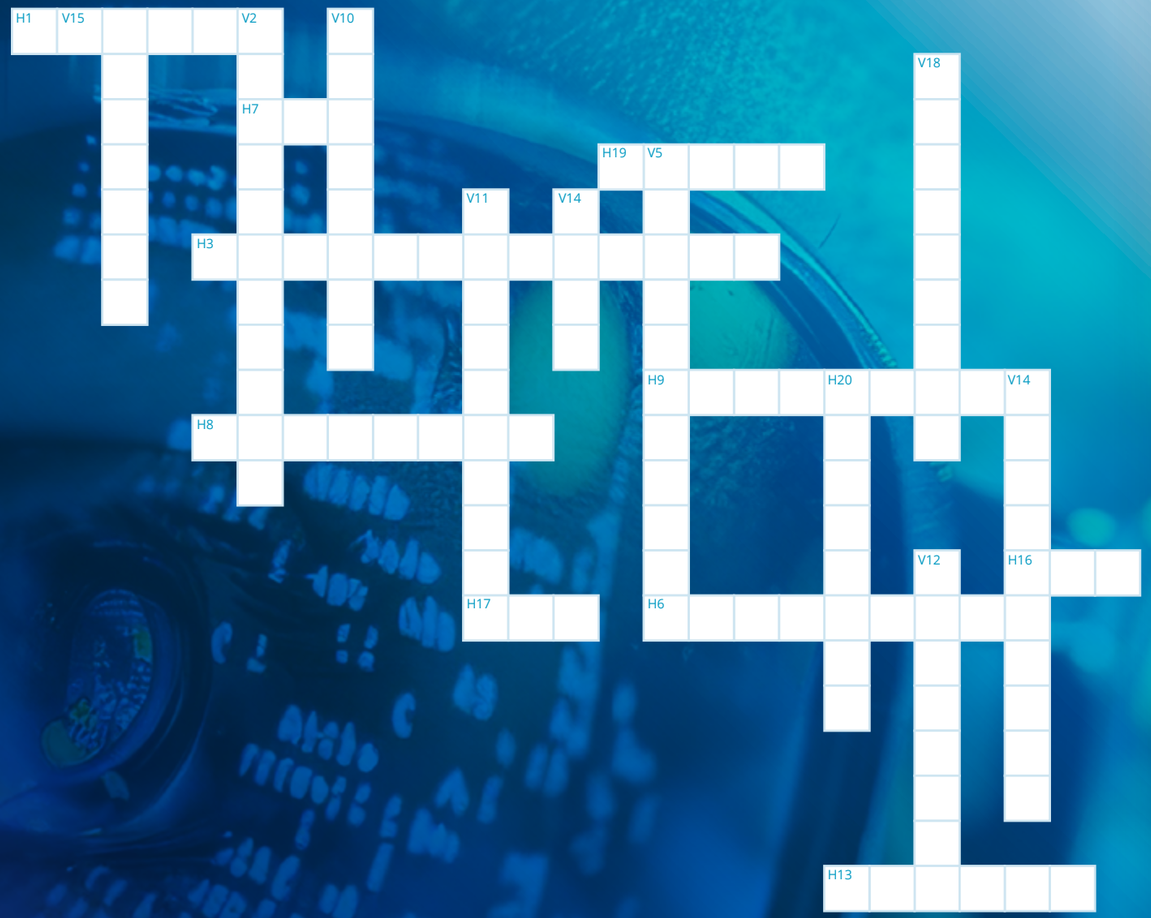
Paramètre Nmap	Détail	Astuce
-max-hostgroup	Définit le nombre maximum d'hôtes insérés dans chaque groupe.	En spécifiant un **max-hostgroup** faible (ex. 4, 8 ou 16), les résultats seront inscrits régulièrement dans le fichier de sortie. Le scan peut ainsi être interrompu puis redémarré en temps voulu, en repartant du dernier groupe d'hôte scanné.
-oN,- oX, -oG	Écrit les résultats de chaque groupe scanné au fur et à mesure dans le fichier spécifié.	
--resume	Reprend un scan à partir d'un fichier de sortie Nmap passé en argument, sans scanner à nouveau les hôtes pour lesquels des résultats existent déjà.	
-p- -r	Réalise un scan port complet de façon itérative au lieu de se baser sur la probabilité d'apparition (comportement par défaut).	En réalisant un scan de port itératif sur un hôte avec le suivi des paquets activés, il devient possible d'interrompre le scan en cours puis de le redémarrer en temps voulu, en repartant du dernier port scanné.
--packet-trace	Indique quels paquets sont envoyés sur quels hôtes/ports.	
-p[port]-	Réalise un scan redémarrant au **port** spécifié.	

4 Conclusion

Nmap s'impose ainsi comme un outil de référence dans le domaine de la cartographie réseau, largement utilisé lors de tests d'intrusion. Reconnu comme un incontournable durant l'étape de reconnaissance, Nmap est un outil ayant traversé les décennies et gagné en popularité au fil du temps. Les raisons de son adoption dans le milieu sont nombreuses ; comme vu au sein de cet article, Nmap tire sa force de sa grande capacité de personnalisation, sa polyvalence et la fiabilité des résultats qu'il permet d'obtenir. Mais si Nmap reste toujours autant utilisé, c'est surtout parce qu'il reste parfaitement adapté depuis plus de deux décennies à un besoin indispensable et qui n'a pas changé depuis le jour de sa sortie, le 1er septembre 1997 : connaître ce qui est exposé sur un réseau. À l'image d'une boussole, Nmap aide à naviguer au sein de réseaux inconnus et sert de référence pour savoir quelle direction emprunter.

L'outil s'est toutefois enrichi avec le temps et a tenté d'offrir de nouvelles fonctionnalités comme la détection de systèmes d'exploitation ou l'exécution de scripts en tout genre. Ces deux mécanismes tirent leur force de la collaboration des développeurs et illustrent la volonté du fondateur Gordon Lyon de créer un outil open source, communautaire et collaboratif.
Comme énoncé en préambule, cet article n'avait pas vocation à étudier en profondeur le fonctionnement de Nmap et de rédiger un panorama des méthodes et paramètres proposés. Les fonctionnalités de contournement d'équipements de blocage et de détection n'ont par exemple pas été abordées, n'étant aujourd'hui plus adaptées aux besoins modernes de furtivité. Comme tout sujet passionnant, ces techniques mériteraient toutefois d'être étudiées et pourraient bel et bien faire l'objet d'un deuxième article…

Mots croisés



VERTICAL

- V2 : Vecteur d'attaque exploitant des dépendances logicielles ou fournisseurs tiers.
- V4 : Interface matérielle utilisée pour déboguer ou reprogrammer des puces.
- V5 : On dit crypter c'est ça ?
- V10 : Accès complètement oublié par inadvertance permettant de contourner l'authentification normale.
- V11 : Orchestrateur de conteneurs utilisé dans les environnements cloud.
- V12 : Leurre conçu pour attirer, détecter et étudier les attaquants.
- V14 : Tu veux revoir tes fichiers ? Donne les bitcoins.
- V15 : Endroit où il est possible d'acheter une pizza 4 fromages et un lance-roquette en supplément.
- V18 : Outil permettant de se rappeler de tout ce que l'on saisit sur un clavier même si ça n'est pas nous qui tapons.
- V20 : Si Gandalf était un outil de cybersécurité «You shall not pass !».

HORIZONTAL

- H1 : Protocole de communication utilisé dans les réseaux industriels (et svp on évite de l'exposer sur Internet).
- H3 : Faute de frappe opportuniste.
- H6 : Outil d'infrastructure as code pour gérer le cloud.
- H7 : Contrôleur logique programmable utilisé dans les systèmes indutriels.
- H8 : Chaîne d'automatisation dans les environnements CI/CD.
- H9 : Type d'outil utilisé par l'ANSSI pour capturer les trames réseau.
- H13 : Plateforme DevOps opensource intégrant CI/CD et gestion de code source.
- H16 : Dont l'indisponibilité pourrait fortement impacter l'économie, la sécurité ou la population.
- H17 : Protocole série utilisé pour communiquer avec des périphériques embarqués.
- H19 : Système de contrôle industriel supervisant des processus physiques.

Notes

xmco

We deliver cybersecurity expertise

Nos consultants pensent comme les attaquants pour mieux les contrer, puis vérifient manuellement chaque vulnérabilité potentielle afin de livrer une vision claire et exploitable des risques Cyber. Audits, pentests, réponse à incident, conformité PCI DSS, veille CERT et CTI : nous couvrons tout le cycle de vie de la cybersécurité. **Cette exigence transforme la sécurité en levier de performance mesurable.** Certifiés PASSI et PCI QSA, nous demeurons indépendants et engagés pour la réussite numérique de nos clients.

Date de création :	2002
Effectif salariés :	plus de 100
Qualifications :	PASSI, QSA et CERT officiel
Clients actifs :	plus de 450
dont clients CERT :	plus de 100
Secteurs :	Banque, Assurance, Industrie, Institutions, Transports, Médias, Luxe, etc.

Renseignement :

info@xmco.fr

01 79 35 29 30

www.xmco.fr