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 communautaire 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 considèrent 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, comment 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 :

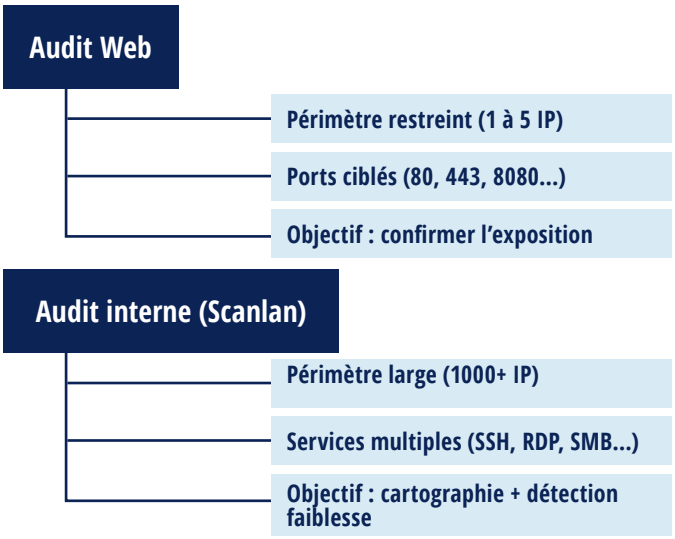
- cibler des plages IP précises plutôt que tout le réseau,
- prioriser les ports courants (22, 80, 443, 445, 3389),
- ajuster le parallélisme et les délais de retransmission.

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.

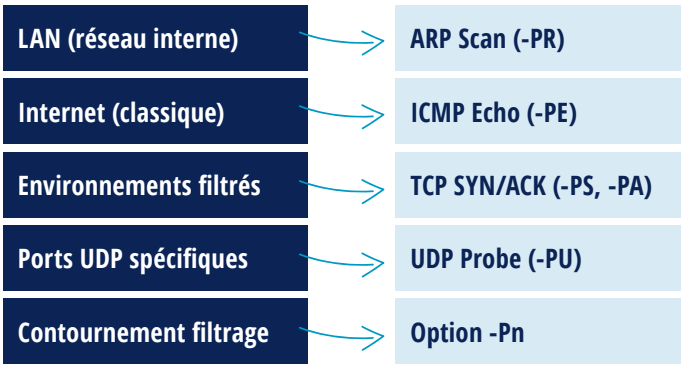
À l'inverse, certains dispositifs de sécurité peuvent générer des **faux positifs** en renvoyant systématiquement des réponses, même pour des adresses inexistantes. Cela complique l'interprétation et oblige à recouper les résultats avec d'autres sources (logs réseau, outils complémentaires).

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



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.