Nmap : Architecture et méthodologies d'un scanner réseau pour l'audit de sécurité

Stéphane FOSSE
fosse.fr

19 janvier 2026

Résumé

Nmap (Network Mapper) s'impose depuis 1997 comme l'outil de référence pour la reconnaissance réseau et l'audit de sécurité. Cette analyse technique examine l'architecture interne de Nmap, ses algorithmes de scan, les méthodes de *fingerprinting* TCP/IP et l'évolution de son moteur de scripts NSE. L'approche se concentre sur les aspects techniques permettant aux professionnels de la sécurité de maîtriser les mécanismes sous-jacents pour un usage optimal en audit d'infrastructure.

Table des matières

1 Introduction

L'audit de sécurité réseau nécessite des outils capables d'identifier avec précision les services exposés, les systèmes d'exploitation déployés et les vulnérabilités présentes sur l'infrastructure cible. Nmap répond à ces exigences par une approche multicouche combinant analyse des piles TCP/IP, détection de services et exécution de scripts spécialisés [1].

Développé initialement par Gordon Lyon (Fyodor) en 1997, Nmap exploite les ambiguïtés des RFC TCP/IP pour caractériser de manière unique les implémentations réseau [2]. Son architecture modulaire permet une extensibilité via le Nmap Scripting Engine (NSE), transformant un simple scanner de ports en plateforme d'analyse de sécurité.

2 Architecture technique et algorithmes de scan

2.1 Scan TCP SYN : fondements de la reconnaissance furtive

Le scan TCP SYN constitue la méthode par défaut de Nmap pour sa rapidité et sa discrétion relative [3]. Le principe repose sur l'envoi de paquets TCP avec le flag SYN positionné, sans compléter la *handshake* TCP complète.

L'algorithme suit cette séquence :

1. Envoi d'un paquet TCP SYN vers le port cible ;
2. Analyse de la réponse du système distant ;
3. Classification de l'état du port selon la réponse reçue.

La classification des ports s'effectue selon ces critères :

— **Ouvert** : réception d'un SYN/ACK ;

— **Fermé** : réception d'un RST (Reset the connection) ;

— **Filtré** : absence de réponse ou message ICMP d'erreur.

Cette méthode présente l'avantage de ne jamais établir de connexion complète, réduisant les traces dans les logs système tout en maintenant une vitesse d'exécution élevée.

2.2 Techniques de scan avancées

Nmap implémente plusieurs techniques spécialisées pour contourner les dispositifs de filtrage [4] :

Scan TCP FIN, NULL et Xmas : Ces méthodes exploitent le comportement spécifié dans la RFC 793 « Transmission Control Protocol » datant de septembre 1981. Un port ouvert ne doit théoriquement pas répondre à ces paquets, tandis qu'un port fermé doit envoyer un RST. L'efficacité varie selon l'implémentation de la pile TCP/IP, et donc le système qui se trouve derrière le port interrogé.

Scan TCP ACK : Utilisé principalement pour cartographier les règles de pare-feu. Ce scan permet de distinguer les ports filtrés des ports non filtrés sans révéler leur état d'ouverture.

Scan UDP : Plus complexe que TCP en raison de la nature sans connexion d'UDP. Nmap envoie des paquets UDP et analyse les réponses ICMP « Port Unreachable » pour déterminer l'état des ports.

2.3 Optimisation des performances

L'architecture de Nmap intègre plusieurs mécanismes d'optimisation [5] :

Parallélisation adaptative : Nmap ajuste dynamiquement le nombre de sondes simultanées selon les conditions réseau. Les paramètres `-min-parallelism` et `-max-parallelism` contrôlent cette parallélisation.

Templates de timing : Six profils prédéfinis (T0 à T5) équilibrent vitesse et discrétion. Le template T4 (agressif) optimise pour les réseaux fiables, tandis que T0 (paranoïaque) minimise la détection.

Gestion intelligente des retransmissions : L'algorithme adapte le nombre de nouvelles tentatives selon la fiabilité observée du réseau, évitant les délais inutiles tout en maintenant la précision.

3 Fingerprinting TCP/IP et détection d'OS

3.1 Méthodologie

La détection d'OS de Nmap repose sur l'analyse des particularités d'implémentation TCP/IP propres à chaque système [6]. Le processus utilise jusqu'à 16 sondes TCP, UDP et ICMP ciblant des ports ouverts et fermés connus.

Les tests analysent plusieurs caractéristiques :

— **Séquences ISN TCP** : Algorithmes de génération des numéros de séquence initiaux ;
— **Options TCP** : Support, ordre et valeurs des options TCP ;
— **Taille de fenêtre TCP** : Valeurs annoncées selon différents contextes ;
— **Comportement IP** : TTL initial, fragmentation, gestion du champ ID ;
— **Réponses ICMP** : Types et contenu des messages d'erreur.

3.2 Analyse des algorithmes de séquençage

L'analyse des numéros de séquence TCP révèle des patterns caractéristiques [7]. Nmap évalue :

— **Plus Grand Diviseur Commun** : Calcul du PGDC entre séquences successives pour identifier les algorithmes basés sur des compteurs ;
— **Indice de prédictibilité (SP)** : Mesure de la difficulté à prédire les séquences futures, crucial pour évaluer la vulnérabilité aux attaques de spoofing ;
— **Taux de compteur (ISR)** : Vitesse d'incrémentation du générateur de séquences.

3.3 Classification et précision

La base de données d'empreintes de Nmap contient plus de 6 000 signatures couvrant systèmes d'exploitation, équipements réseau et dispositifs embarqués [8]. L'algorithme de classification compare l'empreinte collectée aux signatures référence selon un système de pondération.

La précision dépend de plusieurs facteurs :

— Disponibilité de ports ouverts et fermés ;
— Interférence des pare-feux intermédiaires ;
— Mise à jour de la base de signatures.

4 Nmap Scripting Engine : extensibilité et automatisation

4.1 Architecture du moteur de scripts

Le NSE transforme Nmap en plateforme extensible pour l'audit de sécurité [9]. Basé sur Lua 5.4, il exécute des scripts en parallèle selon quatre phases :

1. **Prerule** : Exécution avant tout scan ;
2. **Hostrule** : Exécution par hôte ;
3. **Portrule** : Exécution par port ouvert ;
4. **Postrule** : Exécution après scan complet.

4.2 Catégorisation et sélection des scripts

Les scripts NSE se répartissent en catégories fonctionnelles [10] :

— **Safe** : Scripts sans risque d'interruption de service
— **Intrusive** : Scripts pouvant affecter le système cible
— **Vuln** : Détection de vulnérabilités spécifiques
— **Auth** : Tests d'authentification et de force brute
— **Discovery** : Énumération de services et ressources

La sélection s'effectue via expressions booléennes permettant des combinaisons complexes :

```
nmap --script "vuln and not intrusive" target
nmap --script "http-* and safe" target
```

4.3 Développement et contribution

L'écosystème NSE compte plus de 600 scripts couvrant protocoles modernes, vulnérabilités récentes et systèmes industriels [11]. Le processus de développement suit des standards stricts :

— Documentation complète des fonctionnalités ;
— Tests de non-régression automatisés ;
— Révision par la communauté pour les scripts critiques.

5 Applications en audit de sécurité

5.1 Reconnaissance d'infrastructure

L'audit débutent typiquement par une reconnaissance d'infrastructure utilisant les capacités de découverte de Nmap [12] :

```
# Decouverte d'hotes actifs
nmap -sn 192.168.1.0/24

# Scan complet avec detection d'OS et services
nmap -sS -sV -O -A target_range

# Enumeration de services web
nmap --script http-enum,http-headers target
```

5.2 Détection de vulnérabilités

Le NSE automatise la détection de vulnérabilités connues [13] :

```
# Scan de vulnerabilites SSL/TLS
nmap --script ssl-cert,ssl-enum-ciphers target

# Detection de vulnerabilites SMB
nmap --script smb-vuln-* target

# Audit de configurations faibles
nmap --script auth target
```

5.3 Intégration dans les processus DevSecOps

Nmap s'intègre dans les pipelines CI/CD pour la validation continue de sécurité [14] :

— Vérification automatique des configurations ;
— Détection de services non autorisés ;
— Validation des règles de pare-feu.

6 Contournement et évasion

6.1 Techniques d'évasion des IDS/IPS

Nmap propose plusieurs méthodes pour contourner les systèmes de détection [15] :
Fragmentation IP : Division des paquets en fragments plus petits :

```
nmap -f target  # Fragmentation 8 octets
nmap --mtu 16 target  # MTU personnalise
```

Utilisation de leurres : Masquage de l'origine réelle du scan :

```
nmap -D decoy1,decoy2,ME,decoy3 target
```

Modification du port source : Exploitation des règles de pare-feu permissives :

```
nmap --source-port 53 target  # Port DNS
nmap -g 80 target  # Port HTTP
```

6.2 Scan zombie (Idle scan)

La technique de scan zombie représente la méthode la plus furtive, utilisant un hôte intermédiaire pour masquer l'origine du scan [16]. Cette technique exploite la prévisibilité des identifiants IP pour déduire l'état des ports sans émettre de paquets directement vers la cible.

7 Performance et optimisation

7.1 Stratégies d'optimisation réseau

L'optimisation des scans Nmap nécessite l'adaptation aux contraintes réseau [17] :

— **Scan par lots** : Division des cibles en groupes pour parallélisation efficace ;
— **Limitation de portée** : Restriction aux ports critiques via `-top-ports` ;
— **Exclusion conditionnelle** : Utilisation de `-host-timeout` pour éviter les hôtes lents.

7.2 Métriques et monitoring

Le suivi des performances s'effectue via plusieurs indicateurs :

— Débit de paquets par seconde (contrôlé par `-min-rate`/`-max-rate`) ;
— Temps de réponse moyen (RTT) ;
— Taux de perte de paquets.

8 Évolutions et perspectives

8.1 Adaptation aux environnements cloud

Les versions récentes de Nmap intègrent des capacités spécifiques aux infrastructures cloud [18] :

— Détection des services AWS, Azure, GCP ;
— Support des protocoles cloud-native (HTTP/2, gRPC) ;
— Intégration avec les API de métadonnées cloud.

8.2 Support des protocoles émergents

L'évolution continue de Nmap couvre les protocoles modernes :

— HTTP/3 et QUIC ;
— Protocoles IoT et industriels ;
— Technologies de conteneurisation.

9 Conclusion

Nmap demeure l'outil de référence pour l'audit de sécurité réseau grâce à son architecture modulaire, ses algorithmes sophistiqués et son écosystème de scripts extensible. Sa capacité d'adaptation aux évolutions technologiques, combinée à une précision de *fingerprinting* inégalée, en fait un composant indispensable des méthodologies d'audit modernes.

Comme tout logiciel de sécurité, il peut être utilisé dans le cadre d'attaques informatiques. Ce sont les fameuses technologies à double usage. Connaître son fonctionnement peut donc vous aider à mieux vous protéger aussi, en sachant comment un attaquant s'y prendra pour venir sniffer à vos ports...

Références

[1] G. Lyon, Nmap: the Network Mapper - Free Security Scanner, Site officiel du projet Nmap, 2025.

[2] G. Lyon, The History and Future of Nmap, Documentation officielle Nmap, 2025.

[3] G. Lyon, TCP SYN (Stealth) Scan (-sS), Documentation technique Nmap Network Scanning, 2025.

[4] G. Lyon, Port Scanning Techniques, Guide des techniques de scan Nmap, 2025.

[5] G. Lyon, Timing and Performance, Documentation optimisation Nmap, 2025.

[6] G. Lyon, Remote OS detection via TCP/IP Stack FingerPrinting, Article technique fondateur sur l'empreintage TCP/IP, 1998.

[7] G. Lyon, TCP/IP Fingerprinting Methods Supported by Nmap, Méthodes d'empreintage TCP/IP, 2025.

[8] G. Lyon, OS Detection, Documentation détection d'OS Nmap, 2025.

[9] G. Lyon, Chapter 9. Nmap Scripting Engine, Documentation complète du moteur de scripts NSE, 2025.

[10] G. Lyon, Nmap Scripting Engine (NSE), Guide d'utilisation NSE, 2025.

[11] N. Project, NSEDoc Reference Portal: NSE Scripts, Portail de documentation des scripts NSE, 2025.

[12] BitSight, Nmap in Cybersecurity, Guide usage Nmap en cybersécurité, 2025.

[13] N. Team, How to Detect CVEs Using Nmap Vulnerability Scan Scripts, Guide détection de vulnérabilités avec NSE, 2025.

[14] Encora, Enable Security into CI/CD pipeline with DevSecOps, Intégration Nmap dans DevSecOps, 2020.

[15] G. Lyon, Firewall/IDS Evasion and Spoofing, Techniques d'évasion Nmap, 2025.

[16] Cyber.H0und, The Art of Stealth Scanning: Master these 3 Nmap Techniques, Techniques de scan furtif, 2024.

[17] G. Lyon, Timing Templates (-T), Templates de timing pour optimisation, 2025.

[18] G. Lyon, Nmap: the Network Mapper - Free Security Scanner, Fonctionnalités cloud de Nmap 7.97, 2025.

[19] G. Lyon, Nmap Network Scanning : The Official Nmap Project Guide to Network Discovery and Security Scanning. Insecure.com, 2009, Guide officiel complet du projet Nmap, isbn : 978-0979958717.

[20] P. Kacherginsky, Writing Nmap NSE scripts for vulnerability scanning, Développement de scripts NSE avancés, 2018.

[21] G. Lyon, Nmap Introduction - Phrack 51, Article 11, Publication originale de Nmap dans Phrack Magazine, 1997.