2.7 Nmap im autorisierten Netz – Hosts, Ports und Dienste untersuchen

Ziel dieser Seite

Nmap ist ein Werkzeug zur aktiven Netzwerkerkundung und Portanalyse. Administratoren können damit unter anderem prüfen:

- welche Systeme in einem autorisierten Netz erreichbar sind;
- welche TCP- und UDP-Ports antworten;
- welche Dienste an diesen Ports angeboten werden;
- welche Dienstversionen wahrscheinlich verwendet werden;
- ob eine Firewall Pakete verwirft oder zurückweist;
- ob ein Dienst nur lokal oder auch aus einem anderen Netzsegment erreichbar ist;
- ob sich die Erreichbarkeit nach einer Änderung verbessert oder verschlechtert hat;
- ob ein dokumentierter Sollzustand mit dem tatsächlich sichtbaren Zustand übereinstimmt.

Nmap sendet aktiv Pakete oder Verbindungsversuche an Zielsysteme. Es ist daher kein rein passives Diagnosewerkzeug.

> **Nmap darf ausschließlich gegen eigene Systeme oder mit eindeutiger Genehmigung des zuständigen Betreibers verwendet werden.** Vor einem Scan müssen Zielbereich, Zeitraum, Scanart, Quellsystem und zulässige Intensität festgelegt werden.

Kennzeichnungen

Kennzeichnung	Bedeutung
[RO]	Lokale Abfrage ohne Netzwerkprüfung
[TEST]	Aktiver Netzwerk- oder Diensttest
[PRIV]	Erhöhte Berechtigungen können erforderlich sein
[FILE]	Befehl erstellt oder verarbeitet Dateien
[SENS]	Ausgabe kann sensible Infrastrukturinformationen enthalten
[DISRUPT]	Test kann Dienste, Geräte oder Überwachungssysteme belasten
[CHANGE]	Befehl verändert eine Einstellung oder Datei

---

1. Wofür sollte Nmap in der Fehleranalyse verwendet werden?

Geeignete und ungeeignete Einsatzbereiche anzeigen

Geeignete administrative Aufgaben

Aufgabe	Beispiel
Einzelnen Dienst prüfen	Ist TCP-Port 443 vom Clientnetz erreichbar?
Erreichbare Systeme erfassen	Welche freigegebenen Testsysteme antworten im VLAN?
Firewalländerung kontrollieren	Ist der freigegebene Port nach der Regeländerung erreichbar?
Dienstinventar vergleichen	Stimmen offene Ports mit der Dokumentation überein?
Dienstidentifikation	Welcher Dienst antwortet tatsächlich auf einem Port?
TCP und UDP unterscheiden	Antwortet nur TCP oder auch UDP?
Erreichbarkeit aus verschiedenen Segmenten vergleichen	Ist der Dienst intern erreichbar, aus dem Gastnetz aber gefiltert?
Fehlerzustand dokumentieren	Welche Ergebnisse waren vor und nach einer Änderung sichtbar?

Nmap ist nicht geeignet für

- Scans fremder öffentlicher Systeme ohne Erlaubnis;
- unkontrollierte Scans kompletter Unternehmensnetze;
- das Umgehen von Firewalls oder Überwachungssystemen;
- Verschleierungs-, Spoofing- oder Tarntechniken im normalen Administrationsbetrieb;
- Brute-Force-, Exploit- oder Denial-of-Service-Tests;
- den Nachweis, dass ein Dienst fachlich korrekt funktioniert;
- die alleinige Entscheidung, ob ein System sicher ist.

Ein offener Port beweist nur, dass aus Sicht des Scanstandorts eine entsprechende Antwort empfangen wurde. Er beweist nicht, dass:

- die Anwendung vollständig funktioniert;
- die Anwendung korrekt konfiguriert ist;
- eine Benutzeranmeldung möglich ist;
- der Dienst keine Sicherheitslücken besitzt;
- alle Clients denselben Netzwerkpfad verwenden.

---

2. Wie wird Nmap auf Windows, Linux und macOS geprüft?

Installations- und Versionsprüfung anzeigen

Aufgabe	Windows	Linux	macOS
Programmpfad suchen	[RO] Get-Command nmap	[RO] command -v nmap	[RO] command -v nmap
Version anzeigen	[RO] nmap --version	[RO] nmap --version	[RO] nmap --version
Hilfe anzeigen	[RO] nmap -h	[RO] nmap -h	[RO] nmap -h
Lokale Handbuchseite	Nicht standardmäßig vorhanden	[RO] man nmap	[RO] man nmap

Windows

Die offizielle Windows-Installation von Nmap verwendet normalerweise Npcap für Raw-Packet-Funktionen. Für bestimmte Scanarten sollte PowerShell oder die Eingabeaufforderung als Administrator gestartet werden.

Linux und macOS

Scanarten, die Raw-IP-Pakete verwenden, benötigen normalerweise erhöhte Rechte:

```
[TEST][PRIV][SENS] sudo nmap -sS -p 443 192.0.2.10
```

Ein TCP-Connect-Scan kann gewöhnlich ohne erhöhte Rechte ausgeführt werden:

```
[TEST][SENS] nmap -sT -p 443 192.0.2.10
```

Die tatsächlich verfügbaren Funktionen hängen von Betriebssystem, Nmap-Version, Rechten und Paket-Capture-Treiber ab.

---

3. Welche Freigaben müssen vor einem Scan vorliegen?

Autorisierungscheckliste anzeigen

Vor einem Scan sollten mindestens folgende Punkte dokumentiert sein:

Punkt	Beispiel
Auftraggeber	Zuständige IT-Abteilung
Zielsystem	192.0.2.10

Punkt	Beispiel
Zielnetz	`192.0.2.0/28`
Erlaubte Ports	TCP 22, 80 und 443
Erlaubte Scanarten	Host Discovery, TCP Connect, Versionserkennung
Scanquelle	Administrationssystem `192.0.2.50`
Zeitraum	`14:00–14:15 Uhr`
Verantwortliche Kontaktperson	Max Mustermann
Abbruchkriterium	Dienstfehler, hohe Last oder Alarmmeldung
Protokollierung	Ausgabe als normale, XML- und grepbare Datei
Erwarteter Sollzustand	TCP 443 offen, TCP 22 gefiltert

Vor dem Start klären

- Befinden sich empfindliche Altgeräte im Zielbereich?
- Sind Drucker, Telefonanlagen, Kameras, Steuerungen oder medizinische Geräte betroffen?
- Gibt es Intrusion-Detection- oder Intrusion-Prevention-Systeme?
- Muss das Security Operations Center informiert werden?
- Gibt es eine maximale Scanrate?
- Darf eine Dienst- oder Betriebssystemerkennung durchgeführt werden?
- Dürfen NSE-Skripte verwendet werden?
- Wie lange dürfen Ergebnisse gespeichert werden?

> Die Erlaubnis, einen einzelnen Server zu prüfen, ist keine Erlaubnis zum Scan des gesamten Subnetzes.

---

4. Wie werden Zielsysteme sicher angegeben?

Zielspezifikation anzeigen

Zielart	Beispiel
Einzelne IPv4-Adresse	`192.0.2.10`
Mehrere einzelne Ziele	`192.0.2.10 192.0.2.20`
DNS-Name	`server.example.com`
CIDR-Netz	`192.0.2.0/28`

Zielart	Beispiel
IPv4-Adressbereich	`192.0.2.10-20`
IPv6-Adresse	`2001:db8::10`
Ziele aus einer Datei	`-iL targets.txt`
Bestimmtes Ziel ausschließen	`--exclude 192.0.2.15`
Ausschlussliste verwenden	`--excludefile excluded.txt`

Einzelnes Ziel

```
[TEST][SENS] nmap 192.0.2.10
```

Kleines, ausdrücklich freigegebenes Netz

```
[TEST][SENS] nmap 192.0.2.0/28
```

IPv6-Ziel

```
[TEST][SENS] nmap -6 2001:db8::10
```

Ziele aus einer autorisiert erstellten Datei

```
[TEST][FILE][SENS] nmap -iL targets.txt
```

Zielbereich nur auflösen und auflisten, ohne Zielpakete zu senden

```
[RO][SENS] nmap -sL 192.0.2.0/28
```

`-sL` führt einen List Scan durch. Dabei werden die Ziele aufgelistet und standardmäßig gegebenenfalls Reverse-DNS-Abfragen durchgeführt. Es findet kein normaler Host- oder Portscan gegen die Zielsysteme statt.

Ohne DNS-Auflösung auflisten

```
[RO][SENS] nmap -sL -n 192.0.2.0/28
```

Das Auflisten des Zielbereichs vor dem aktiven Scan hilft, fehlerhafte CIDR-Angaben frühzeitig zu erkennen.

---

5. Wie wird geprüft, welche Hosts erreichbar sind?

Host Discovery anzeigen

Nur Host-Erkennung, ohne Portscan

```
[TEST][SENS] nmap -sn 192.0.2.0/28
```

`-sn` deaktiviert den anschließenden Portscan. Nmap führt aber weiterhin aktive Host-Erkennungsprüfungen durch.

Ohne Namensauflösung

```
[TEST][SENS] nmap -sn -n 192.0.2.0/28
```

Grund für die Bewertung anzeigen

```
[TEST][SENS] nmap -sn --reason 192.0.2.0/28
```

Lokales Ethernet-Netz

Im lokalen Ethernet-Netz verwendet Nmap für die Host-Erkennung normalerweise ARP beziehungsweise bei IPv6 Neighbor Discovery. Das ist häufig zuverlässiger als ein einfacher ICMP-Echo-Test.

Gezielte ICMP-Echo-Erkennung

```
[TEST][PRIV][SENS] sudo nmap -sn -PE 192.0.2.0/28
```

TCP-SYN-Erkennung auf freigegebenen Ports

```
[TEST][PRIV][SENS] sudo nmap -sn -PS22,80,443 192.0.2.0/28
```

TCP-ACK-Erkennung

```
[TEST][PRIV][SENS] sudo nmap -sn -PA80,443 192.0.2.0/28
```

UDP-Erkennung auf einem gezielt freigegebenen Port

```
[TEST][PRIV][SENS] sudo nmap -sn -PU53 192.0.2.0/28
```

Wichtige Interpretation

Ergebnis	Aussage
`Host is up`	Nmap hat eine als Erreichbarkeitsnachweis bewertete Antwort erhalten
Kein Hosteintrag	Keine ausreichende Antwort empfangen
ICMP antwortet nicht	Host kann trotzdem über TCP oder UDP erreichbar sein
ARP-Antwort im lokalen Netz	Gerät ist auf Layer 2 erreichbar
Hohe Latenz	Kann durch Netzwerk, Zielsystem oder Scanverfahren entstehen

Ein fehlender Hosttreffer beweist nicht, dass das System ausgeschaltet ist. Firewalls können die für die Erkennung verwendeten Pakete verwerfen.

---

6. Was bedeutet `-Pn`, und wann sollte es verwendet werden?

Portscan ohne vorherige Host-Erkennung anzeigen

Mit `-Pn` behandelt Nmap die angegebenen Ziele als erreichbar und überspringt die normale Host-Erkennung.

```
[TEST][SENS] nmap -Pn -p 443 192.0.2.10
```

Sinnvoller Einsatz

- der Host ist nachweislich aktiv;
- ICMP und andere Discovery-Pakete werden gefiltert;
- ein bestimmter Port soll unabhängig von der Host-Erkennung geprüft werden;
- nur ein einzelnes oder sehr kleines freigegebenes Ziel wird untersucht.

Auswirkung

Ohne erfolgreiche Host-Erkennung würde Nmap möglicherweise melden:

```
Host seems down.
```

Mit `-Pn` wird der Portscan trotzdem durchgeführt.

Risiko bei großen Zielbereichen

```
nmap -Pn 192.0.2.0/16
```

Dieser Scan würde jedes Ziel im Bereich als aktiv behandeln und entsprechend prüfen. Das kann sehr lange dauern und erhebliche Netzlast oder Sicherheitsmeldungen verursachen.

> `-Pn` sollte nicht pauschal für große Netze verwendet werden. Es eignet sich hauptsächlich für bekannte, ausdrücklich freigegebene Einzelziele oder kleine Zielgruppen.

---

7. Welche Portzustände zeigt Nmap an?

Portzustände und Bedeutung anzeigen

Zustand	Bedeutung
`open`	Eine Anwendung nimmt an diesem Port Verbindungen oder Pakete an
`closed`	Ziel ist erreichbar, aber an diesem Port antwortet kein Dienst
`filtered`	Nmap kann wegen eines Filters oder Netzwerkhindernisses nicht bestimmen, ob der Port offen oder geschlossen ist
`unfiltered`	Port ist erreichbar, der verwendete Scan kann aber nicht zwischen offen und geschlossen unterscheiden
`open|filtered`	Nmap kann nicht zwischen offen und gefiltert unterscheiden
`closed|filtered`	Nmap kann nicht zwischen geschlossen und gefiltert unterscheiden

Wichtig

`filtered` bedeutet nicht automatisch:

```
Eine Firewall auf dem Zielsystem blockiert den Port.
```

Mögliche Ursachen sind unter anderem:

- Host-Firewall;
- Netzwerk-Firewall;
- Access Control List;
- Routerfilter;
- Security Group;
- Paketverlust;
- asymmetrisches Routing;
- IPS oder Rate Limiting;
- falscher Rückweg;
- unvollständige Scanantwort.

Mit `--reason` zeigt Nmap an, auf welcher Antwort oder Nichtantwort die Bewertung beruht:

```
[TEST][SENS] nmap --reason -p 22,80,443 192.0.2.10
```

---

8. Welche Ports scannt Nmap standardmäßig?

Portauswahl anzeigen

Ohne ausdrückliche Portangabe untersucht Nmap standardmäßig die 1.000 häufigsten Ports des jeweiligen gescannten Protokolls. Es handelt sich nicht automatisch um alle 65.535 TCP-Ports.

Bestimmte Ports

```
[TEST][SENS] nmap -p 22,80,443 192.0.2.10
```

Portbereich

```
[TEST][SENS] nmap -p 8000-8100 192.0.2.10
```

Alle TCP-Ports

```
[TEST][SENS] nmap -p- 192.0.2.10
```

`-p-` steht für die Ports 1 bis 65.535. Ein vollständiger Portscan ist wesentlich umfangreicher als ein Standardscan und muss ausdrücklich freigegeben sein.

Die häufigsten 100 Ports

```
[TEST][SENS] nmap --top-ports 100 192.0.2.10
```

Schneller Scan der häufigsten Ports

```
[TEST][SENS] nmap -F 192.0.2.10
```

`-F` reduziert die Anzahl der normalerweise untersuchten Ports. Die genaue Portauswahl hängt von der lokalen Nmap-Dienstdatenbank ab.

Nur offene Ports anzeigen

```
[TEST][SENS] nmap --open -p 22,80,443,445,3389 192.0.2.10
```

`--open` verkürzt die Ausgabe. Für die Fehleranalyse können geschlossene und gefilterte Ports jedoch wichtige Informationen liefern. Die vollständige Ausgabe sollte deshalb häufig bevorzugt werden.

---

9. Was ist ein TCP-Connect-Scan?

TCP-Connect-Scan `-sT` anzeigen

Der TCP-Connect-Scan verwendet die normale `connect()`-Funktion des Betriebssystems. Er kann gewöhnlich ohne Raw-Packet-Rechte ausgeführt werden.

```
[TEST][SENS] nmap -sT -p 22,80,443 192.0.2.10
```

Eigenschaften

Eigenschaft	TCP Connect
Option	`-sT`

Eigenschaft	TCP Connect
Erhöhte Rechte	Normalerweise nicht erforderlich
Verbindung	Vollständiger TCP-Verbindungsaufbau zu offenen Ports
Protokollierung am Ziel	Wahrscheinlich
Eignung	Funktionale Erreichbarkeitsprüfung einzelner TCP-Ports
Belastung	Bei wenigen Ports normalerweise gering, aber aktiv

Einzelnen HTTPS-Port prüfen

```
[TEST][SENS] nmap -sT -Pn -p 443 --reason 192.0.2.10
```

Da bei offenen Ports eine vollständige Verbindung aufgebaut wird, kann der Zielserver den Versuch in seinen Logs erfassen.

Für einen gezielten administrativen Funktionstest ist das nicht grundsätzlich nachteilig: Der Scan soll nachvollziehbar und autorisiert sein.

---

10. Was ist ein TCP-SYN-Scan?

TCP-SYN-Scan `-sS` anzeigen

Beim SYN-Scan sendet Nmap ein TCP-SYN-Paket, ohne bei einem offenen Port eine vollständige Anwendungssitzung aufzubauen.

```
[TEST][PRIV][SENS] sudo nmap -sS -p 22,80,443 192.0.2.10
```

Typische Antworten

Antwort	Nmap-Bewertung
SYN/ACK	open
RST	closed
Keine Antwort nach Wiederholungen	filtered
Bestimmte ICMP-Unreachable-Antwort	filtered

Eigenschaften

Eigenschaft	TCP SYN
Option	`-sS`
Erhöhte Rechte	Auf Unix-Systemen normalerweise erforderlich
Vollständige TCP-Verbindung	Nein
Unterscheidung	`open`, `closed`, `filtered`
Eignung	Kontrollierte Portprüfung im autorisierten Netz

Der Ausdruck „halb offen“ bedeutet nicht, dass der Scan unsichtbar ist. Firewalls, IDS, IPS und Zielsysteme können SYN-Scans erkennen und protokollieren.

---

11. Wie werden UDP-Dienste geprüft?

UDP-Scan `-sU` anzeigen

UDP ist verbindungslos. Ein fehlendes Antwortpaket lässt deshalb häufig keine eindeutige Entscheidung zu.

Gezielter UDP-Scan

```
[TEST][PRIV][SENS] sudo nmap -sU -p 53,123,161 192.0.2.10
```

Mit Begründung und Versionserkennung

```
[TEST][PRIV][SENS] sudo nmap -sU -sV --reason -p 53,123,161 192.0.2.10
```

Typische Ergebnisse

Antwort	Mögliche Bewertung
Gültige UDP-Antwort	`open`
ICMP Port Unreachable	`closed`
Bestimmte andere ICMP-Fehler	`filtered`
Keine Antwort	`open|filtered`

Warum UDP-Scans länger dauern

- viele offene UDP-Dienste antworten nur auf gültige Protokollanfragen;
- gefilterte Pakete erzeugen häufig keine Antwort;

- ICMP-Fehlermeldungen können begrenzt werden;
- Nmap muss Zeitüberschreitungen und Wiederholungen abwarten.

Nur die tatsächlich benötigten Ports prüfen

```
[TEST][PRIV][SENS] sudo nmap -sU -p 53 192.0.2.10
```

Ein ungeplanter Scan aller 65.535 UDP-Ports kann sehr lange dauern und unnötige Last verursachen.

---

12. Wie werden TCP und UDP gemeinsam untersucht?

Kombinierten Scan anzeigen

TCP-SYN- und UDP-Scan können kombiniert werden:

```
[TEST][PRIV][SENS] sudo nmap -sS -sU -p T:53,80,443,U:53,123 192.0.2.10
```

Bedeutung

Ausdruck	Geprüfte Ports
`T:53,80,443`	TCP 53, 80 und 443
`U:53,123`	UDP 53 und 123

Ohne die Präfixe `T:` und `U:` kann die Portzuordnung bei einem kombinierten Scan missverständlich werden.

DNS über TCP und UDP prüfen

```
[TEST][PRIV][SENS] sudo nmap -sS -sU -p T:53,U:53 192.0.2.10
```

Ein offener TCP- und UDP-Port bestätigt noch nicht, dass DNS-Anfragen korrekt beantwortet werden. Danach sollte ein protokollspezifischer Test folgen, beispielsweise mit `dig`, `Resolve-DnsName` oder `nslookup`.

---

13. Wie werden Dienste und Versionen erkannt?

Service- und Versionserkennung `-sV` anzeigen

Die Versionserkennung sendet zusätzliche, teilweise protokollspezifische Anfragen an offene oder wahrscheinlich offene Ports.

```
[TEST][SENS] nmap -sV -p 22,80,443 192.0.2.10
```

Leichtere Versionserkennung

```
[TEST][SENS] nmap -sV --version-light -p 22,80,443 192.0.2.10
```

Intensität festlegen

```
[TEST][SENS] nmap -sV --version-intensity 2 -p 22,80,443 192.0.2.10
```

Der Wertebereich für `--version-intensity` reicht von `0` bis `9`:

Intensität	Verhalten
`0`	Nur wenige, besonders wahrscheinliche Prüfungen
`2`	Reduzierte, häufig ausreichende Prüfung
`7`	Standardintensität
`9`	Alle verfügbaren Versionserkennungs-Probes

Interpretation

```
443/tcp open  https  nginx 1.24.0
```

Dabei sind zu unterscheiden:

Feld	Aussage
`443/tcp`	Untersuchte Port-/Protokollkombination
`open`	Nmap erhielt eine Antwort für einen offenen Port
`https`	Erkannter oder anhand der Portzuordnung vermuteter Dienst
`nginx 1.24.0`	Von Nmap ermittelte oder geschätzte Produktversion

Versionsergebnisse können unvollständig oder irreführend sein, beispielsweise durch:

- Reverse Proxies;
- Load Balancer;
- TLS-Termination;
- absichtlich veränderte Banner;
- Backports von Sicherheitskorrekturen;
- herstellerspezifische Builds;
- Protokollweiterleitungen;
- IDS- oder IPS-Antworten.

> Ein Nmap-Versionshinweis sollte mit der lokalen Paketverwaltung, der Anwendungskonfiguration oder der Herstelleroberfläche bestätigt werden.

---

14. Wie wird eine Betriebssystemerkennung durchgeführt?

OS-Erkennung `-O` anzeigen

Nmap versucht anhand verschiedener Netzwerkmerkmale, das Betriebssystem des Zielsystems zu bestimmen.

```
[TEST][PRIV][SENS] sudo nmap -O 192.0.2.10
```

Nur gezielte Ports und OS-Erkennung

```
[TEST][PRIV][SENS] sudo nmap -O -p 22,80,443 192.0.2.10
```

Erkennungsversuche begrenzen

```
[TEST][PRIV][SENS] sudo nmap -O --osscan-limit 192.0.2.0/28
```

`--osscan-limit` beschränkt die OS-Erkennung auf Ziele, bei denen Nmap dafür geeignete Bedingungen erkennt.

Voraussetzungen für gute Ergebnisse

Nmap erzielt bessere Resultate, wenn mindestens:

- ein offener TCP-Port;

- ein geschlossener TCP-Port;
- ausreichende Antworten des Zielsystems

vorhanden sind.

Mögliche Ungenauigkeiten

- Firewall verändert Antworten;
- NAT oder Proxy liegt zwischen Quelle und Ziel;
- mehrere Systeme teilen eine IP-Adresse;
- Netzwerkgerät beantwortet Pakete stellvertretend;
- zu wenige geeignete Ports antworten;
- Betriebssystem-Fingerprint ist nicht eindeutig.

Die Ausgabe ist eine Netzwerkerkennung beziehungsweise Schätzung und kein sicherer Beweis für das installierte Betriebssystem.

---

15. Welche einfachen NSE-Skripte können für die Diagnose verwendet werden?

Nmap Scripting Engine sicher einordnen

Die Nmap Scripting Engine, kurz NSE, erweitert Nmap um Skripte. Diese können von einfachen Informationsabfragen bis zu Sicherheits- und Belastungstests reichen.

Standard-Skripte

```
[TEST][DISRUPT][SENS] nmap -sC -p 22,80,443 192.0.2.10
```

`-sC` entspricht:

```
--script=default
```

Auch Standardskripte senden zusätzliche Anfragen an Dienste und sollten nur nach Freigabe verwendet werden.

Bestimmtes, zuvor geprüftes Skript

```
[TEST][SENS] nmap --script http-title -p 80,443 192.0.2.10
```

Skriptbeschreibung anzeigen

```
[RO] nmap --script-help http-title
```

Mehrere freigegebene Informationsskripte

```
[TEST][SENS] nmap --script http-title,http-headers -p 80,443 192.0.2.10
```

Vor einem NSE-Einsatz müssen geprüft werden:

- Skriptname;
- Skriptbeschreibung;
- Kategorie;
- benötigte Argumente;
- mögliche Auswirkungen;
- Zielport;
- Umfang der Autorisierung.

Im normalen Troubleshooting nicht pauschal verwenden

```
--script vuln
--script exploit
--script brute
--script dos
--script intrusive
```

Diese Kategorien können Schwachstellenprüfungen, Anmeldeversuche oder belastende Tests enthalten. Sie gehören nicht in eine gewöhnliche Netzwerkdiagnose ohne gesonderten Auftrag.

> Auch die Kategorie `safe` ist keine absolute Garantie für Wirkungslosigkeit. Das konkrete Skript muss vor dem Einsatz geprüft werden.

---

16. Warum sollte `-A` nicht unüberlegt verwendet werden?

Aggressive Erkennung erklären

Die Option `-A` aktiviert mehrere Funktionen gleichzeitig:

- Betriebssystemerkennung;

- Versionserkennung;
- Standardskript-Scan;
- Traceroute.

```
[TEST][PRIV][DISRUPT][SENS] sudo nmap -A 192.0.2.10
```

Für eine gezielte Fehleranalyse ist dieser Befehl häufig zu breit. Er sendet mehr unterschiedliche Prüfungen als ein einfacher Portscan und erzeugt umfangreichere Logs und mögliche Sicherheitsmeldungen.

Besser ist eine schrittweise Auswahl:

```
[TEST][SENS] nmap -sT -p 443 192.0.2.10
```

Danach bei Bedarf:

```
[TEST][SENS] nmap -sV --version-light -p 443 192.0.2.10
```

Und nur bei begründetem Bedarf:

```
[TEST][PRIV][SENS] sudo nmap -O -p 22,80,443 192.0.2.10
```

Dadurch bleibt erkennbar, welche Scanoption welche Reaktion verursacht hat.

---

17. Wie wird die Namensauflösung kontrolliert?

DNS-Optionen anzeigen

Keine DNS-Auflösung durchführen

```
[TEST][SENS] nmap -n -p 22,80,443 192.0.2.10
```

DNS-Auflösung für alle Ziele erzwingen

```
[TEST][SENS] nmap -R -p 22,80,443 192.0.2.10
```

Bedeutung

Option	Wirkung
`-n`	Keine Reverse-DNS-Auflösung
`-R`	Reverse-DNS-Auflösung für alle Ziele
Keine Option	Nmap entscheidet abhängig vom Scanergebnis

Für reproduzierbare Portprüfungen ist `-n` häufig sinnvoll:

- Scan wird nicht durch DNS verzögert;
- keine zusätzlichen DNS-Abfragen;
- Ergebnisse zeigen die tatsächlichen IP-Adressen;
- DNS- und Portprobleme werden getrennt untersucht.

Wenn die DNS-Zuordnung selbst Teil der Diagnose ist, sollte sie bewusst separat geprüft werden.

---

18. Wie wird die Scanrate kontrolliert?

Timing und Belastungsbegrenzung anzeigen

Nmap bietet Timing-Vorlagen:

Vorlage	Name	Einordnung
`-T0`	Paranoid	Extrem langsam
`-T1`	Sneaky	Sehr langsam
`-T2`	Polite	Reduzierte Geschwindigkeit
`-T3`	Normal	Standard
`-T4`	Aggressive	Schneller, nur in stabilen und freigegebenen Netzen
`-T5`	Insane	Sehr aggressiv; für normales Troubleshooting ungeeignet

Normale Geschwindigkeit

```
[TEST][SENS] nmap -T3 -p 22,80,443 192.0.2.10
```

Rücksichtsvollere Prüfung

```
[TEST][SENS] nmap -T2 -p 22,80,443 192.0.2.10
```

Maximale Paketrate begrenzen

```
[TEST][SENS] nmap --max-rate 50 -p 1-1000 192.0.2.10
```

`--max-rate 50` fordert Nmap auf, im Durchschnitt nicht schneller als 50 Pakete pro Sekunde zu senden. Abhängig von Scanart und Betriebssystem können trotzdem kurzfristige Abweichungen auftreten.

Host-Zeitüberschreitung

```
[TEST][SENS] nmap --host-timeout 2m -p 22,80,443 192.0.2.10
```

Wenn ein Ziel nach zwei Minuten nicht vollständig geprüft wurde, beendet Nmap die weitere Bearbeitung dieses Hosts.

Für empfindliche Systeme

- nur einzelne Ziele;
- nur benötigte Ports;
- `-T2` oder `-T3`;
- niedrige `--max-rate`;
- keine umfassende Versionserkennung;
- keine pauschalen NSE-Skripte;
- System- und Dienstüberwachung parallel beobachten.

---

19. Wie werden Ergebnisse gespeichert?

Ausgabeformate anzeigen

Option	Format	Verwendung
`-oN DATEI`	Normale Textausgabe	Menschliche Auswertung
`-oX DATEI`	XML	Automatisierte Verarbeitung
`-oG DATEI`	Grepbares Format	Älteres zeilenorientiertes Format
`-oA BASISNAME`	Alle drei Formate	Vollständige Dokumentation

Normale Ausgabe speichern

```
[TEST][FILE][SENS] nmap -sT -p 22,80,443 -oN server-scan.txt 192.0.2.10
```

XML-Ausgabe

```
[TEST][FILE][SENS] nmap -sT -p 22,80,443 -oX server-scan.xml 192.0.2.10
```

Alle Hauptformate

```
[TEST][FILE][SENS] nmap -sT -p 22,80,443 -oA server-scan 192.0.2.10
```

Dabei entstehen typischerweise:

```
server-scan.nmap
server-scan.xml
server-scan.gnmap
```

Zusätzlich Terminalausgabe mit `tee` speichern

Linux und macOS:

```
[TEST][FILE][SENS] nmap -sT -p 22,80,443 192.0.2.10 | tee server-scan.txt
```

PowerShell:

```
[TEST][FILE][SENS] nmap -sT -p 22,80,443 192.0.2.10 | Tee-Object -FilePath server-scan.txt
```

`tee` beziehungsweise `Tee-Object` speichert nur die sichtbare Textausgabe. Für strukturierte Verarbeitung sollte `-oX` oder `-oA` verwendet werden.

---

20. Wie werden Ergebnisse vor und nach einer Änderung verglichen?

Vergleichsdiagnose anzeigen

Vor der Änderung

```
[TEST][FILE][SENS] nmap -n -sT -p 22,80,443 --reason -oA server-before 192.0.2.10
```

Nach der Änderung

```
[TEST][FILE][SENS] nmap -n -sT -p 22,80,443 --reason -oA server-after 192.0.2.10
```

Linux und macOS:

```
[RO][FILE][SENS] diff -u server-before.nmap server-after.nmap
```

PowerShell:

```
[RO][FILE][SENS] Compare-Object (Get-Content .\server-before.nmap) (Get-Content .\server-after.nmap)
```

Für einen aussagekräftigen Vergleich müssen identisch sein:

- Scanquelle;
- Zieladresse;
- Scanart;
- Portliste;
- Nmap-Version;
- Berechtigungsstufe;
- DNS-Einstellung;
- Timing-Einstellung;
- Netzwerkpfad.

Andernfalls können Unterschiede durch die Testmethode statt durch die eigentliche Änderung entstehen.

---

21. Wie wird ein bestimmter Dienst systematisch geprüft?

Diagnosefolge für einen einzelnen Dienst anzeigen

Beispiel: HTTPS auf TCP-Port 443.

1. Namensauflösung getrennt prüfen

Windows:

```
[RO] Resolve-DnsName server.example.com
```

Linux oder macOS:

```
[RO] dig server.example.com
```

2. Host Discovery durchführen

```
[TEST][SENS] nmap -sn --reason 192.0.2.10
```

3. Port unabhängig von Host Discovery prüfen

```
[TEST][SENS] nmap -Pn -n -sT -p 443 --reason 192.0.2.10
```

4. Dienst vorsichtig identifizieren

```
[TEST][SENS] nmap -Pn -n -sV --version-light -p 443 192.0.2.10
```

5. Anwendungsebene prüfen

```
[TEST][SENS] curl -vkI https://192.0.2.10/
```

Bei namensabhängigen virtuellen Hosts sollte der korrekte DNS-Name verwendet werden:

```
[TEST][SENS] curl -vI https://server.example.com/
```

6. TLS gesondert prüfen

```
[TEST][SENS] openssl s_client -connect 192.0.2.10:443 -servername server.example.com
```

Interpretation

Ergebnis	Nächster Schritt
Port `closed`	Dienststatus und lokale Listener prüfen
Port `filtered`	Firewall, ACL, Routing und Paketmitschnitt prüfen

Ergebnis	Nächster Schritt
Port `open`, curl fehlschlägt	HTTP-, TLS-, Proxy- oder Anwendungsproblem untersuchen
Port `open`, TLS falsch	Zertifikat, SNI, TLS-Version und Zeit prüfen
Nmap erkennt falschen Dienst	Direkten Protokolltest durchführen
Unterschiedliche Ergebnisse aus zwei Netzen	Firewall- und Routingpfad vergleichen

Nmap beantwortet hauptsächlich die Frage:

```
Wie reagiert dieser Port aus Sicht dieses Scanstandorts?
```

Für die vollständige Fehlerursache sind weitere Werkzeuge erforderlich.

---

22. Wie wird ein Firewallproblem eingegrenzt?

Vergleich zwischen Netzsegmenten anzeigen

Ein Scan sollte von genau den Standorten erfolgen, deren Erreichbarkeit untersucht wird.

```
Clientnetz A → Firewall → Servernetz
Clientnetz B → Firewall → Servernetz
```

Scan aus Netz A

```
[TEST][FILE][SENS] nmap -Pn -n -sT -p 443 --reason -oN netz-a.txt 192.0.2.10
```

Scan aus Netz B

```
[TEST][FILE][SENS] nmap -Pn -n -sT -p 443 --reason -oN netz-b.txt 192.0.2.10
```

Mögliche Beobachtung

Netz A	Netz B	Mögliche Schlussfolgerung
`open`	`filtered`	Unterschiedliche Firewall-, ACL- oder Routingbehandlung

Netz A	Netz B	Mögliche Schlussfolgerung
`open`	`closed`	Möglicherweise anderes Ziel, NAT oder unterschiedliche Antwortquelle
`filtered`	`filtered`	Gemeinsamer Filter, fehlender Rückweg oder Ziel nicht erreichbar
`open`	`open`	Grundlegende Portfreigabe aus beiden Netzen vorhanden

Zusätzlich sollten geprüft werden:

- Quell-IP des Scanhosts;
- Ziel-IP nach NAT;
- Firewall-Logs;
- Routingtabellen;
- Security Groups;
- lokale Host-Firewall;
- Paketmitschnitt auf Client, Firewall oder Server;
- Rückweg zum jeweiligen Clientnetz.

Ein Nmap-Ergebnis allein zeigt nicht, an welchem Gerät ein Paket verworfen wurde.

---

23. Welche Scanarten sollten im normalen Troubleshooting vermieden werden?

Nicht empfohlene Optionen anzeigen

Folgende Funktionen gehören normalerweise nicht zur alltäglichen Fehleranalyse:

Scanart oder Option	Grund
Idle Scan `-sI`	Verwendet ein fremdes System als sogenanntes Zombie-System
FTP Bounce `-b`	Missbraucht einen FTP-Server als Relay
Eigene TCP-Flags `--scanflags`	Kann Filter- und IDS-Umgehung dienen
Decoys `-D`	Verschleiert die tatsächliche Scanquelle
Quelladress-Spoofing `-S`	Erschwert Rückverfolgung und kann Netzstörungen verursachen
Fragmentierung `-f`	Wird häufig zur Filterumgehung eingesetzt
`--badsum`	Erzeugt absichtlich ungültige Prüfsummen
NSE-Kategorie `brute`	Führt Anmeldeversuche durch

Scanart oder Option	Grund
NSE-Kategorie `exploit`	Versucht Sicherheitslücken auszunutzen
NSE-Kategorie `dos`	Kann Dienste beeinträchtigen
NSE-Kategorie `intrusive`	Kann Systeme oder Anwendungen belasten
Ungeprüftes `-A` gegen viele Systeme	Kombiniert mehrere aktive Erkennungsverfahren
`-T5`	Sehr aggressive Zeitsteuerung

Solche Funktionen werden in diesem Troubleshooting-Buch nicht als Standardbefehle empfohlen. Für einen gesonderten Sicherheitstest sind ein eigener Auftrag, eine genaue Risikoanalyse und abgestimmte Abbruchkriterien erforderlich.

---

24. Welche typischen Fehlinterpretationen gibt es?

Fehlertabelle anzeigen

Fehlinterpretation	Korrekte Einordnung
`Host seems down` bedeutet ausgeschaltet	Discovery-Pakete können gefiltert worden sein
`open` bedeutet Anwendung funktioniert	Nur der Port reagiert; Anwendungstest folgt
`closed` bedeutet Firewall blockiert	Das Ziel antwortet normalerweise aktiv mit „kein Listener“
`filtered` bedeutet lokale Host-Firewall	Filter kann an jeder Stelle des Pfads liegen
`open|filtered` bedeutet offen	Nmap kann offen und gefiltert nicht unterscheiden
Dienstname in der Porttabelle ist sicher erkannt	Ohne `-sV` kann er nur aus der Portzuordnung stammen
Angezeigte Version ist garantiert installiert	Banner, Proxy oder Backport können abweichen
OS-Erkennung ist ein sicherer Beweis	Es handelt sich um eine Fingerprint-Schätzung
Kein Ergebnis bei UDP bedeutet geschlossen	Häufig lautet der Zustand `open|filtered`
`-Pn` macht den Scan passiv	Portscan bleibt aktiv; nur Host Discovery wird übersprungen
SYN-Scan ist unsichtbar	IDS, IPS, Firewalls und Systeme können ihn erkennen
Ein Scan aus dem Servernetz gilt auch für das Clientnetz	Unterschiedliche Netzwerkpfade können andere Ergebnisse liefern
Nmap ist ein vollständiger Schwachstellenscanner	Nmap ist primär ein Netzwerk- und Portscanner
Viele Optionen liefern automatisch bessere Ergebnisse	Mehr Prüfungen erhöhen Last, Logs und Fehlinterpretationsrisiko

---

25. Wie werden Nmap-Ergebnisse verifiziert?

Gegenprüfungen anzeigen

Nmap-Beobachtung	Geeignete Gegenprüfung
TCP-Port offen	`Test-NetConnection`, `nc`, `curl` oder anwendungsspezifischer Client
TCP-Port geschlossen	Lokale Listener mit `Get-NetTCPConnection`, `ss` oder `lsof` prüfen
Port gefiltert	Firewall-Logs, Routing und Paketmitschnitt prüfen
DNS-Dienst offen	`Resolve-DnsName`, `dig` oder `nslookup`
HTTP/HTTPS offen	`curl`
TLS erkannt	`openssl s_client`
SSH offen	Autorisierten SSH-Verbindungsaufbau durchführen
SMB offen	SMB-Client und Serverlogs prüfen
Dienstversion erkannt	Lokale Paket- oder Anwendungsversion prüfen
Betriebssystem erkannt	Lokale Systeminformationen oder Inventarsystem prüfen
Host nicht erkannt	ARP/Neighbor-Tabelle, Ping, TCP-Test und Switchinformationen prüfen

Lokale Listener prüfen

Betriebssystem	Befehl
Windows	`[RO] Get-NetTCPConnection -State Listen`
Linux	`[RO] ss -lntup`
macOS	`[RO] sudo lsof -nP -iTCP -sTCP:LISTEN`

Ein Port kann lokal lauschen und trotzdem aus dem untersuchten Netzsegment nicht erreichbar sein. Umgekehrt kann NAT oder Portweiterleitung einen extern erreichbaren Port zeigen, obwohl auf der Zieladresse selbst kein entsprechender lokaler Listener sichtbar ist.

---

26. Wie sieht ein sicherer Nmap-Diagnoseablauf aus?

Empfohlene Schrittfolge anzeigen

Vorbereitung

1. Auftrag und Ziel der Prüfung dokumentieren.
2. Zulässige Zieladressen und Ports festlegen.
3. Scanquelle und Netzwerksegment dokumentieren.
4. Empfindliche Systeme identifizieren.
5. Erlaubte Scanarten und maximale Rate festlegen.
6. Ansprechpartner und Abbruchkriterium festlegen.
7. Systemzeit und Speicherort der Ergebnisse prüfen.

Zielkontrolle

8. Zielbereich zuerst mit `-sL -n` auflisten.
9. CIDR-Präfix und Ausschlüsse kontrollieren.
10. Mit einem einzelnen Testsystem beginnen.

Schrittweise Prüfung

11. Host Discovery mit `-sn` durchführen.
12. Falls erforderlich einen bekannten Port mit `-Pn` prüfen.
13. Nur benötigte Ports mit `-p` angeben.
14. Zunächst TCP Connect oder SYN Scan verwenden.
15. UDP nur für konkret benötigte Ports prüfen.
16. `--reason` zur Interpretation hinzufügen.
17. Versionserkennung nur bei Bedarf verwenden.
18. NSE-Skripte einzeln prüfen und freigeben.
19. Scanrate und Zielsystemüberwachung beobachten.

Dokumentation

20. Ausgabe mit `-oA` speichern.
21. Nmap-Version und verwendeten Befehl dokumentieren.
22. Beginn, Ende und Scanquelle festhalten.
23. Ergebnisse als Beobachtung und nicht sofort als Ursache dokumentieren.

Validierung

24. Offene Ports mit einem Protokollwerkzeug prüfen.
25. Gefilterte Ports mit Firewall-Logs und Paketmitschnitten untersuchen.
26. Lokale Listener auf dem Zielsystem prüfen.
27. Nach einer Änderung denselben Scan erneut ausführen.
28. Ergebnisse vergleichen.
29. Zuständige Personen über Abschluss oder Auffälligkeiten informieren.
30. Scanresultate gemäß Schutz- und Löschrichtlinie behandeln.

---

27. Kurzreferenz – sichere Basisbefehle

Nmap-Befehlstabelle anzeigen

Aufgabe	Befehl
Version anzeigen	`[RO] nmap --version`
Hilfe anzeigen	`[RO] nmap -h`
Zielbereich nur auflisten	`[RO][SENS] nmap -sL -n 192.0.2.0/28`
Hosts erkennen	`[TEST][SENS] nmap -sn -n 192.0.2.0/28`
Grund anzeigen	`[TEST][SENS] nmap -sn -n --reason 192.0.2.0/28`
Einzelnen TCP-Port prüfen	`[TEST][SENS] nmap -Pn -n -sT -p 443 --reason 192.0.2.10`
Mehrere TCP-Ports	`[TEST][SENS] nmap -sT -p 22,80,443 192.0.2.10`
SYN-Scan	`[TEST][PRIV][SENS] sudo nmap -sS -p 22,80,443 192.0.2.10`
UDP-Ports	`[TEST][PRIV][SENS] sudo nmap -sU -p 53,123,161 192.0.2.10`
TCP und UDP	`[TEST][PRIV][SENS] sudo nmap -sS -sU -p T:53,80,443,U:53,123 192.0.2.10`
Häufigste 100 Ports	`[TEST][SENS] nmap --top-ports 100 192.0.2.10`
Alle TCP-Ports	`[TEST][SENS] nmap -p- 192.0.2.10`
Nur offene Ports ausgeben	`[TEST][SENS] nmap --open -p 22,80,443 192.0.2.10`
Dienstversionen vorsichtig prüfen	`[TEST][SENS] nmap -sV --version-light -p 22,80,443 192.0.2.10`
Betriebssystem schätzen	`[TEST][PRIV][SENS] sudo nmap -O 192.0.2.10`
IPv6-Ziel	`[TEST][SENS] nmap -6 -p 22,80,443 2001:db8::10`
Scanrate begrenzen	`[TEST][SENS] nmap --max-rate 50 -p 1-1000 192.0.2.10`
Ergebnisse vollständig speichern	`[TEST][FILE][SENS] nmap -sT -p 22,80,443 -oA server-scan 192.0.2.10`
Skriptbeschreibung anzeigen	`[RO] nmap --script-help http-title`
HTTP-Titel prüfen	`[TEST][SENS] nmap --script http-title -p 80,443 192.0.2.10`

---

28. Kurzreferenz – wichtigste Optionen

Optionstabelle anzeigen

Option	Bedeutung
`-sL`	Ziele nur auflisten
`-sn`	Nur Host Discovery, kein Portscan

Option	Bedeutung
`-Pn`	Host Discovery überspringen
`-n`	Keine DNS-Auflösung
`-R`	Reverse-DNS-Auflösung erzwingen
`-sT`	TCP Connect Scan
`-sS`	TCP SYN Scan
`-sU`	UDP Scan
`-p`	Ports angeben
`-p-`	Alle Ports von 1 bis 65.535
`-F`	Schneller Scan mit reduzierter Portauswahl
`--top-ports N`	N häufigste Ports prüfen
`--open`	Nur offene oder möglicherweise offene Ports anzeigen
`--reason`	Grund für Host- und Portbewertung anzeigen
`-sV`	Dienst- und Versionserkennung
`--version-light`	Reduzierte Versionserkennung
`-O`	Betriebssystemerkennung
`-6`	IPv6 verwenden
`-iL`	Ziele aus Datei lesen
`--exclude`	Ziel ausschließen
`--excludefile`	Ausschlüsse aus Datei lesen
`-T2`	Rücksichtsvolle Timing-Vorlage
`-T3`	Normale Timing-Vorlage
`--max-rate`	Maximale durchschnittliche Paketrate begrenzen
`--host-timeout`	Zeitlimit pro Host
`-oN`	Normale Ausgabe speichern
`-oX`	XML-Ausgabe speichern
`-oG`	Grepbare Ausgabe speichern
`-oA`	Alle Hauptformate speichern
`-sC`	Standard-NSE-Skripte ausführen
`--script`	Bestimmte NSE-Skripte auswählen
`--script-help`	Beschreibung eines Skripts anzeigen

Merksätze

- Nmap ist ein aktives Diagnosewerkzeug und benötigt eine eindeutige Autorisierung.
- Zuerst den Zielbereich prüfen, dann mit einem einzelnen Ziel und wenigen Ports beginnen.
- `-sn` führt Host Discovery ohne anschließenden Portscan durch.
- `-Pn` macht einen Scan nicht passiv, sondern überspringt nur die Host-Erkennung.
- `-sT` verwendet vollständige TCP-Verbindungen und benötigt normalerweise keine Raw-Packet-Rechte.
- `-sS` verwendet Raw-Pakete und benötigt auf Unix-Systemen normalerweise erhöhte Rechte.
- `filtered` bedeutet, dass Nmap den Portzustand wegen fehlender oder filternder Antworten nicht bestimmen kann.
- UDP-Ergebnisse sind häufig weniger eindeutig als TCP-Ergebnisse.
- Ein Dienstname ohne Versionserkennung kann lediglich aus der bekannten Portzuordnung stammen.
- Versions- und Betriebssystemergebnisse sind Hinweise und müssen lokal bestätigt werden.
- `-A`, NSE-Skripte und aggressive Timing-Optionen dürfen nicht unüberlegt eingesetzt werden.
- Ein offener Port bestätigt noch keine funktionierende Anwendung.
- Ergebnisse müssen aus dem tatsächlich betroffenen Netzwerksegment erhoben werden.
- Nmap-Ergebnisse sollten mit Protokolltests, lokalen Listenern, Logs und Paketmitschnitten verifiziert werden.

Quellen

- Offizielle Nmap-Referenz
- Nmap – Target Specification
- Nmap – Host Discovery
- Nmap – Port Scanning Basics
- Nmap – Port Scanning Techniques
- Nmap – Port Specification and Scan Order
- Nmap – Service and Version Detection
- Nmap – OS Detection
- Nmap – Nmap Scripting Engine
- Nmap – Timing and Performance
- Nmap – Output
- Nmap – Legal Notices and Inappropriate Usage
- Offizielle Npcap-Dokumentation
