Back to All Articles(https://cyberscope.netally.com/blog)

Nmap Host Discovery – What is it and How do you use it?

Introduction

Nmap host discovery is a crucial, foundational capability on which to build other important activities including a network assessment or security audit. Why? Because the first step in effectively characterizing a network is to figure out what is "connected". Another way of saying this is to determine what specific IP addresses are currently in use. Once done, other more in-depth scans can be done on these addresses in use while ignoring unused IP addresses, greatly speeding execution time.

Table of Contents

What is the Definition of a Host?

According to NIST[1] a host is defined as:

"…any hardware device that has the capability of permitting access to a network via a user interface, specialized software, network address, protocol stack, or any other means. Some examples include, but are not limited to, computers, personal electronic devices, thin clients, and multi-functional devices.”

Given this expansive host definition and its access to the network resources, what makes a specific host “interesting” to a network or security team depends upon the circumstance in question. It could involve questions of services running or host performance for a network team. Alternatively, for security groups it can have significant implications from a threat perspective.

As an aside, note that a server is a special host that often provides resources to other devices—frequently clients.
Because of its open-source nature and wide usage, beyond host discovery Nmap is an excellent tool by which to perform a myriad of network and device analysis to comprehensively characterize your network, devices, endpoints, and infrastructure. In this blog, we’ll focus on introducing host discovery. *You are strongly urged to complete additional research and collaborate with all network and security staff before rolling out Nmap testing in your environment.*

What is Nmap Host Discover?

The goal of host discovery or “ping scanning” is to build a map of responsive hosts on a network. Once done, additional more targeted or in-depth scans can be done of only these specific targets. Key points to keep in mind. Host discovery…

- Identifies any device with an IP address (or are “live”) on a network
- Assists in quantifying a network before deeper scanning is performed
- Can utilize various protocols to probe hosts (ICMP, TCP, ARP, etc.).
- Streamlines unnecessary scans by excluding offline or inactive hosts or unused IP addresses

How does Nmap Perform Host Discovery?

Nmap offers various methods to discover hosts, based on the options you specify it will send different types of packets to quantify if a host is active including:

- **List Scan** – A host discovery can be initiated without doing any active scans on the target network. Rather than send traffic to any hosts, it instead attempts a reverse DNS lookup to determine if a hostname can be found for the target. It is the least intrusive host discovery method and is a great way to verify you are targeting the proper IP addresses before performing more aggressive scan methods.

  - **Here is the command: nmap -sL [targets]**
- **ICMP Echo Request (ping)** – This is the standard "ping" method. If a host replies with an ICMP Echo Reply, it's considered alive and is logged as a host. No port scanning is performed. Note this is the next level of scan intrusiveness (beyond a List Scan) providing a lighter means to scout a target network while mostly flying under the radar.
  - **Command syntax is: nmap -sn [targets]**

Expert Tip:

Since Nmap commands and scripts when executed can be disruptive using the "list scan" can be an unobtrusive means to ensure that you have proper IP addresses for your targets. If the returned hosts include domain names you do not recognize, it is wise to dig deeper before proceeding to avoid scanning an incorrect network.

- **ICMP Timestamp or Netmask Requests** – Occasionally a firewall or hosts will block Echo requests but may allow lesser-known ICMP types to pass. In this case there are three options:
  - -PE: ICMP Echo
  - -PP: ICMP Timestamp
  - -PM: ICMP Netmask

Enable Echo request via the -PE option. Timestamp and address mask queries can be sent utilizing the -PP and -PM options, respectively. The latter two options are helpful when Echo requests are blocked but other ICMP queries are still available options.

- **Command syntax is: nmap -PE -PP -PM [targets]**
- **TCP SYN Ping** – This method sends an empty TCP SYN packet to a specified port (commonly 80 or 443). If the target replies with a SYN/ACK, it's considered "live". The way this command works is interesting, a SYN flag suggests to the target an attempt to establish a connection is being made. Under typical circumstances, the destination port will be closed, and an RST (reset) packet sent back. However, should the target port be open, it will take the second step of a TCP three-way-handshake by responding with a SYN/ACK TCP packet.

- - **Syntax is: nmap -PS80 [targets]**

- **TCP ACK Ping** – Similar to the above TCP SYN Ping, but substitutes an ACK packet for the SYN. In this case, the ACK packet is mimicking an acknowledgement over an existing TCP connection. However, no connection exists so in this case the target should respond with a RST—which in the process is disclosing its existence.
  - **The command is: nmap -PA80 [target]**
- **UDP Ping** – Typically sends empty UDP packets to specific ports. If a host is alive and the port is closed, it may reply with ICMP Port Unreachable. If the port is not specified, the default one is 40125 (this port is almost never used). Reaching an unopened port is important, since if an open port is reached, many services simply ignore the empty packet and not return a response. The reason being is that when an open port is reached, many services simply ignore the empty packet and not return a response but a closed port.
  - **Syntax is: nmap -PU53 [target]**
- **ARP Ping (Only for Local Networks)** – For targets on a local Ethernet network, ARP (Address Resolution Protocol) requests are the most reliable method. Firewalls typically don't block ARP so this can be an effective way to perform host discovery.
  - **Syntax: nmap -PR [targets]**

Please note that Nmap allows these techniques to be combined or mixed to increase host discovery reliability.

- **Here is an example syntax: nmap -PS80,443 -PA22 -PU53 -PE [targets]**

Here are three additional practical examples:

1. **Basic Host Discovery (Ping Sweep)**
   nmap -sn [targets] – Performs a ping scan to find which hosts are up, but doesn't scan any ports
2. **Aggressive Host Discovery**
   nmap -PS22,80,443 -PA21,23,80 -PU53 -PE [targets] – Uses multiple probes to maximize the chance of identifying live hosts
3. **Host Discovery Without Ping (No Ping)**
   nmap -Pn [targets] – Treats all hosts as online and skips host discovery. Useful when ICMP is blocked but you still want to try port scanning.

What if a Host in not Online?

The above discussion pre-supposes a host is online and they will eventually respond. If you know or have reason to believe you could be missing offline hosts, try these techniques:

- **Use Multiple Probe Types**
  Some hosts block certain types of packets (especially ICMP or SYN). Try combining different discovery methods, for example:
  - **nmap -PE -PS80,443 -PA22 -PU53 [targets]**
    Any combination like this increases your chance of triggering some kind of response
- **Try an ARP Scan (If on a Local Network)**
  If you're on the same subnet, use ARP scans — they almost always work for local hosts, regardless of firewalls:
  - **nmap -PR [targets]**
    If you do see the host with -PR, it's alive but blocking other probe types.
- **Use -Pn to Skip Host Discovery**
  If you're confident the host exists but Nmap shows it as down, force Nmap to treat it as online and scan it anyway:
  - **nmap -Pn [targets]**
    This disables discovery and moves directly to port scanning.
- **Check with Other Tools**
  Use ping, traceroute, or arp to verify the host's presence:
  - **ping [targets]**
  - **arp -a | grep [targets]**
- **Ensure Nmap Isn't Being Blocked**
  Some environments may block scans, especially from antivirus, EDR, or IDS/IPS systems. Check:
  - Are you behind a firewall?
  - Is your scan interface correct (-e option)?
  - Is the host rate-limiting or filtering your packets?
- **Confirm Host is Actually Online**
  It could be powered off. Try:
  - Logging into the host physically or via remote management tools.
  - Using Wake-on-LAN if it's supposed to be online.

- **Scan the Entire Subnet**
  Sometimes IP address assumptions are wrong. The host may have a different IP. Scan the whole subnet:
  - **nmap -sn 192.168.1.0/24**
- **Look at Firewall and IDS/IPS Logs**
  If you have access, review the logs of firewalls or IDS tools to see if packets are being dropped or flagged.

Best practices for Nmap Host Discovery?

- **NEVER Perform Nmap Host Discovery without Organizational Approval & Full Scan Knowledge**
  Any Nmap command and script execution can have significant negative impact to IT resources and network performance. Therefore, it is critical that before proceeding:
  - You work with all IT stakeholders for complete buy-in and approval
  - You understand COMPLETELY how the command or script execution will potentially impact IT resources
- **Understand Your Environment**
  Know what protocols and ports are allowed or blocked in your network. Tailor your discovery method to the situation
  - For internal networks, use ARP scanning for accuracy
  - With external scans, combine TCP SYN and ICMP Echo
- **Use -sn for Lightweight Scans**
  If you're only trying to find live hosts and don't need port details, -sn is the way to go. It's faster and generates less noise.
- **Scan at Appropriate Times**
  Avoid scanning production environments during peak hours. Discovery packets, especially UDP, can cause noise or trigger alerts.
- **Use Timing Controls**
  Use -T options to control scan speed
  - -T0: Paranoid (slowest, stealthiest)
  - -T1: Sneaky
  - -T2: Polite
  - -T3: Normal
  - -T4: Aggressive
  - -T5: Insane (fastest, least stealthy)
    An example here would be: nmap -sn -T4 [targets] – This is a "light weight"

scan executed aggressively.

- **Scan Specific Hosts or Ranges**
  Avoid scanning the entire internet unless you have a specific reason (and permission). Target IPs or ranges intelligently. An added benefit here is that should you inadvertently execute an incorrect Nmap command, you can limit the potential "damage" to hosts (e.g., making them crash) or the greater network (e.g., seriously degrading performance)
- **Respect Legal and Ethical Boundaries**
  Only scan networks you own or are authorized to audit. Unauthorized scanning is illegal and unethical—you can get in REAL trouble with law enforcement, or your current employer, should you perform unauthorized scans.
- **Save and Parse Results**
  Use output options to save your results for analysis. Sample syntax:
  - **nmap -sn -oN scan_results.txt [targets]**
- **Other options:**
  - o -oX: XML
  - o -oG: Grepable
  - oA: All formats

Common Nmap Host Discovery Mistakes to Avoid

- **Assuming All Hosts Respond to ICMP**
  Many modern systems block ICMP pings. Relying solely on -PE (ICMP Echo) will miss hosts. Use multiple techniques (-PS, -PA, -PU) to address this.
- **Not Using ARP for Local Networks**
  Skipping -PR on a LAN will miss many hosts. ARP is nearly always reliable within a subnet.
- **Overloading the Network**
  Aggressive scans (-T5, large port ranges, or multiple concurrent scans) can cause performance issues or alert intrusion detection systems. Use -T4 or T5 timing controls sparingly!
- **Forgetting to Adjust Firewall Rules**
  If you're scanning a host behind a firewall, ensure your scanning machine is allowed through.
- **Not Checking for Rate Limiting**
  Some networks throttle repeated probes. Watch for this and adjust your timing or scan in batches.

In Summary

Congratulations, you have completed the short introduction or primer for Nmap Host Discovery! As you probably already know, Nmap commands and scripting are powerful open-source tools to enumerate your network and better understand hosts and their various attributes.

In this blog, we’ve shown how Nmap host discovery can be an excellent way to identify live hosts on a network. Whether you’re performing network inventory, preparing for a security assessment, or just managing your own infrastructure, host discovery is often the first and most important step.

Congratulations you’ve just mastered Nmap Discovery like a true hero!

By combining multiple probe techniques, tailoring your scans to the network environment, and following best practices, you can achieve reliable and accurate host discovery with minimal disruption. Just as importantly, avoid common pitfalls like relying solely on ICMP, skipping ARP on LANs, or scanning unauthorized networks.

It’s also important you duly take note of the best practices and common Nmap host discovery mistakes, as several of them—specifically #1 “NEVER Perform Nmap Host Discovery Without Organizational Approval & Full Scan Knowledge” and #3 “Overloading the Network”—could make you very unpopular within your business should you not follow these recommendations!

That all being said, mastering host discovery not only makes your scanning more efficient – it sets the stage for everything else Nmap can do and opens you up to leverage the enormous open-source community available to help you.

[1]host – Glossary | CSRC (https://csrc.nist.gov/glossary/term/host)

Author Bio – Brad Reinboldt (https://cyberscope.netally.com/blog/contributor/brad-reinboldt)

Product Manager – CyberScope®

As a Product Manager at NetAlly, Brad Reinboldt is responsible for wired and cybersecurity solutions. He has more than 30 years’ of experience in the computing, networking, and storage sectors in various development and technical management roles. He holds a master’s degree in electrical engineering as well as an MBA in management.

netAlly(https://cyberscope.netally.com)

(/products/cyberscop

CyberScope®

Edge Network Vulnerability Scanner

CyberScope empowers you to quickly discover, identify, and test edge infrastructure and IoT, OT, and ICS devices, wired (Ethernet/Fiber) and WiFi networks, then assess cybersecurity posture against policies, generate reports and perform ongoing monitoring—all without deploying agents.

Learn More

(/products/cyberscope-air)

CyberScope® Air

WiFi Vulnerability Scanner & Tester

CyberScope Air enables SecOps or NetOps teams to discover, validate, and scan edge infrastructure and IoT, OT, and ICS devices whether WiFi or Bluetooth/BLE. Assessing cybersecurity posture of WLANs against policies, generating reports, and performing ongoing monitoring for changes has never been easier.

Learn More

NetAlly(https://cyberscope.netally.com/blog/author/na-kacy)

April 22, 2025(https://cyberscope.netally.com/blog/2025/04/22)

More Posts

(https://cyberscope.netally.com)

Network Security Solutions 101: Simple Measures to Take (https://cyberscope.netally.com/blog/network-security-solutions-101-simple-measures-to-take)

So optimieren Sie die Netzwerkleistung und sichern gleichzeitig den Netzwerkrand – Folge 1 (https://cyberscope.netally.com/blog/so-optimieren-sie-die-netzwerkleistung-und-sichern-gleichzeitig-den-netzwerkrand-folge-1)

CONTACT

☎ 1-844-878-2559(tel:1-844-878-2559) ☎ 1-719-755-0770 (International)(tel:+1-719-755-0770)

(https://www.facebook.com/netally19) (https://bsky.app/profile/netally.com)
(https://www.linkedin.com/company/netally/)
(https://www.youtube.com/channel/UCR1A6GilqQEVhO9yFq0ZorA/featured)
(https://www.instagram.com/netally_official/)

Website Terms of Use Data Privacy Center EULA Subscription Preferences

Do Not Sell My Info