



Network Enumeration Tool for Host Exploration and Recon

Network Enumeration Tool for Host Exploration and Recon

Background and Purpose

N-ETHER (Network Enumeration Tool for Host Exploration and Reconnaissance) is a robust and highly automated **Bash script** designed to streamline the critical initial phases of network security auditing and penetration testing.

It's core purpose is to perform **fast, comprehensive, and consistent** host and port discovery across single targets or large lists of IP addresses. N-ETHER functions as an **intelligent, opinionated wrapper** around the powerful `nmap` tool, bundling best-practice scanning techniques—including service versioning, NSE scripting, and intelligent port selection—into a single, easy-to-use utility. This significantly reduces auditor workload and potential for human error compared to manually executing multiple complex `nmap` commands.

Key Features

N-ETHER provides the following functionalities, configurable via command-line arguments:

Feature	Description	Argument
Concurrent List Scanning	Safely scans multiple targets simultaneously (default 5 parallel jobs) using a named pipe (<code>mkfifo</code>), drastically reducing total scan time without corrupting log output.	<code>-L <target_file></code>
Comprehensive TCP Scan	Performs two-stage TCP enumeration: initial port discovery followed by a detailed service and script scan on <i>only</i> the discovered open ports.	<code>-f</code> (Full Scan) or <code>-q</code> (Quick Scan)
Conditional UDP Scan	Automatically includes a UDP scan of the top 200 ports unless the quick mode (<code>-q</code>) is selected or a SOCKS proxy is specified (due to UDP unreliability over SOCKS).	Default (<code>-f</code>)
Advanced Auditing	Enables specific Nmap Scripting Engine (NSE) scripts known for valuable information gathering (e.g., <code>ftp-anon</code> , <code>smb-enum-shares</code> , <code>ssl-enum-ciphers</code> , <code>dns-brute</code>).	<code>-a</code>
Stealthy Discovery	Defaults to stealthy operation (<code>-Pn</code> / No Ping) to bypass potential firewalls/security devices, with an option to revert to standard ping/ARP host discovery.	Default: <code>-Pn</code> / Override: <code>-d</code>
Operational Flexibility	Supports IPv6 scanning and traffic routing through a SOCKS5 proxy. Outputs clean, columnar Summary Reports alongside the raw Nmap log files.	<code>-6</code> , <code>-x <proxy></code>

Value for OpSec and Differentiation

Aspect	Nmap (Base Tool)	N-ETHER (Wrapper Script)
Operational Security (OpSec)	Default settings often use aggressive host discovery that is easily blocked.	Default settings prioritize stealth (-Pn) and include proxy support (-x) for anonymity.
Concurrency & Logs	Requires external tooling (like GNU Parallel) and results in interleaved output.	Built-in, robust, pipe-based concurrency with automatic log segregation into per-target files.
Workflow	Requires multiple, manually chained commands (discovery, then scan, then script).	Single command automates a two-stage TCP scan, conditional UDP scan, and optional advanced scripts.
Reporting	Outputs raw XML or text that requires separate parsing.	Generates a concise, columnar, human-readable Summary Report by parsing the raw Nmap output.

Installation and Usage

1. Prerequisites

The only dependency is **Nmap**.

- **Install Nmap** (Example for Debian/Ubuntu):

```
Bash
sudo apt update
sudo apt install nmap
```

2. Setup

1. **Save the script:** Save the code to a file named `n-ether.sh`.
2. **Add execute permission:**

```
Bash
chmod +x n-ether.sh
```

3. Command Examples

Scenario	Goal	Command Example
Single Target (Full Scan)	Full TCP (all ports) and UDP (top 200) scan on one IP.	<code>./n-ether.sh -t 192.168.1.100 -f</code>
Quick Scan	Scan only the top 100 TCP ports and skip UDP.	<code>./n-ether.sh -t example.com -q</code>
Concurrent List Scan	Scan targets listed in <code>targets.txt</code> concurrently.	<code>./n-ether.sh -L targets.txt</code>
Advanced Audit	Full scan with specific NSE vulnerability/discovery scripts.	<code>./n-ether.sh -t 192.168.1.10 -a -f</code>

4. Running via Proxy (Tor Example)

To route scan traffic through a SOCKS proxy (e.g., Tor, which typically runs on `127.0.0.1:9050`), use the `-x` flag.

1. **Ensure Tor is running** and its SOCKS proxy is active.
2. **Execute N-ETHER:**

```
Bash
./n-ether.sh -t target-site.org -f -x socks5://127.0.0.1:9050
or for a list:
```

```
Bash
./n-ether.sh -L targets.txt -q -x socks5://127.0.0.1:9050
```

Note: When the `-x` proxy flag is used, N-ETHER will **automatically skip the UDP scan** as Nmap cannot reliably perform UDP scanning through SOCKS proxies.

Integration with OWASP ZAP

N-ETHER is a network enumerator (Layer 3/4), while **OWASP ZAP** is a web application scanner (Layer 7). The best way to integrate N-ETHER's results into ZAP is through **ZAP's Scripting Functions** or the **Automation Framework (AF)**.

The most effective method is using a **Python Active Scan Script** within ZAP to:

1. **Execute N-ETHER** via the `subprocess` module against the host IP.
2. **Read and Parse** N-ETHER's generated summary files (`summary_*.txt`).
3. **Generate ZAP Alerts or Update Scope** based on the network findings (e.g., if N-ETHER finds an open, anonymous FTP service, the ZAP script raises a high-priority alert).

This approach allows N-ETHER to serve as a powerful **pre-scan reconnaissance engine**, feeding critical network context directly into ZAP's web application security workflow.