PARROT OS PYTHON BASE SECURITY TESTING TOOL

This document provides an overview, key features, and detailed instructions for using the interactive\_system\_tester.py tool in both its original and updated versions. The tool is designed for penetration testing, vulnerability assessment, and system auditing on Parrot OS, with enhancements to improve stealth, robustness, and comprehensiveness.

**Table of Contents**

[**1. Overview 2**](#_Toc193205921)

[**- Original Version Overview 2**](#_Toc193205922)

[**- Updated Version Overview 2**](#_Toc193205923)

[**2. Key Features 2**](#_Toc193205924)

[**Original Version Features (advanced\_system\_tester.py) 3**](#_Toc193205925)

[**Updated Version Features (interactive\_system\_tester.py) 3**](#_Toc193205926)

[**3. Summary of Platforms 4**](#_Toc193205927)

[**- Platforms for Testing (Targets): The tool can test any networked device or system, including: 4**](#_Toc193205928)

[**- Platform for Running the Tool: The tool itself runs on: 4**](#_Toc193205929)

[**4. Instructions for Use 4**](#_Toc193205930)

[**4.1. Original Version Instructions (advanced\_system\_tester.py) 5**](#_Toc193205931)

[**4.2. Updated Version Instructions (interactive\_system\_tester.py) 5**](#_Toc193205932)

[**5. Example Usage 6**](#_Toc193205933)

[**5.1. Original Version Example (advanced\_system\_tester.py) 6**](#_Toc193205934)

[**5.2. Updated Version Example (interactive\_system\_tester.py) 7**](#_Toc193205935)

[**6. Troubleshooting 8**](#_Toc193205936)

[**EXTRA: HOW TO UTILIZE PROXY CHAIN FOR TESTING 9**](#_Toc193205937)

[**1. Prerequisites for Proxy Chain Usage 9**](#_Toc193205938)

[**2. Configure proxychains 10**](#_Toc193205939)

1. Overview

* Original Version Overview

Name: advanced\_system\_tester.py

The original advanced\_system\_tester.py is a Python-based tool designed for Parrot OS to perform interactive system testing, including network scanning, vulnerability assessment, exploitation, anonymity checks, and system auditing. It leverages popular tools like nmap, Metasploit, John the Ripper, and SQLmap to identify and exploit vulnerabilities in target systems. The tool provides an interactive command-line interface with support for parallel task execution, detailed logging, and report generation in multiple formats (text, HTML, CSV).

* Updated Version Overview

Name: interactive\_system\_tester.py

The updated interactive\_system\_tester.py builds on the original by adding advanced stealth features, external CVE database integration, and improved robustness. It enhances anonymity with Tor, proxy rotation, and MAC spoofing, while integrating with the National Vulnerability Database (NVD) API to enrich vulnerability data with severity scores and descriptions. The tool now includes retry logic, comprehensive reporting, and modular testing to ensure reliability and effectiveness during penetration testing and threat detection.

1. Key Features

## Original Version Features (advanced\_system\_tester.py)

* Interactive Interface: Command-line interface with autocompletion and keybindings for user-friendly operation.
* Multiple Test Profiles: Supports profiles like network, vulnerability, exploitation, anonymity, and auditing.
* Tool Integration: Uses tools like nmap, Metasploit, John the Ripper, Nikto, SQLmap, Aircrack-ng, Lynis, and chkrootkit.
* Parallel Execution: Runs tasks in parallel with a progress bar for efficiency.
* Custom Wordlist Support: Allows users to specify custom wordlists for password cracking and exploitation.
* Vulnerability Exploitation: Detects vulnerabilities with nmap and prompts users to proceed with Metasploit exploitation.
* Reporting: Generates reports in text, HTML, and CSV formats with test results and exploitation outcomes.
* Logging: Detailed logging of all actions and errors to a dedicated log file.

## Updated Version Features (interactive\_system\_tester.py)

* Advanced Stealth Features:
  + Fragmented Packets: Uses nmap’s --fragment and --data-length to split packets, evading IDS/IPS.
  + Proxy Rotation: Rotates Tor exit nodes or external proxies via proxychains to mask the source IP.
  + Randomized User-Agent: Applies random User-Agent strings to HTTP-based scans (e.g., Nikto, SQLmap).
  + Idle Scan: Implements nmap idle scans using a zombie host to obscure the origin.
  + Timing Randomization: Adds random delays and timing variations across all tools to avoid detection patterns.
* External CVE Database Integration:
  + Integrates with the NVD API to fetch detailed CVE information, including CVSS scores and descriptions.
  + Enriches vulnerability data from nmap and Metasploit with NVD details.
* Enhanced Robustness:
  + Implements retry logic with up to 3 attempts and random delays for failed commands.
  + Modular test execution ensures failures in one test do not halt others.
* Comprehensive Threat Assessment:
  + Assigns severity scores to vulnerabilities based on NVD CVSS data.
  + Displays detailed CVE descriptions in reports for better decision-making.
* Improved Reporting:
  + Includes CVE details, severity scores, and descriptions in HTML reports with severity highlighting.
  + Supports structured CSV output for vulnerability data.
* Stealth Metrics: Logs stealth-related actions (e.g., MAC spoofing, proxy usage) for auditing.

1. Summary of Platforms

* Platforms for Testing (Targets): The tool can test any networked device or system, including:
  + Computers (Linux, Windows, macOS)
  + Smartphones (Android, iOS) with exposed network services
  + IoT devices (e.g., cameras, routers)
  + Network infrastructure (e.g., routers, switches)
  + Cloud servers and virtual machines
  + Web applications and APIs on any platform
* Platform for Running the Tool: The tool itself runs on:
  + Primary: Parrot OS (recommended for minimal setup)
  + Secondary: Other Linux distributions (e.g., Kali Linux, Ubuntu) with manual dependency installation

1. Instructions for Use

* Prerequisites: Both versions of the tool require the following setup:
* System Requirements:
  + Operating System: Parrot OS (or a Linux distribution with similar tools).
  + Privileges: Must be run with sudo (root privileges required).
  + Python Version: Python 3.6 or higher.
* Install Dependencies:
  + Python Packages:

*pip3 install prompt\_toolkit tqdm jinja2 requests*

* + Parrot OS Tools:

*sudo apt update*

*sudo apt install nmap masscan nikto sqlmap john hashcat aircrack-ng metasploit-framework lynis chkrootkit rkhunter macchanger tor torsocks anonsurf proxychains*

* + Configure proxychains (interactive\_system\_tester.py version Only):

Edit /etc/proxychains.conf to enable Tor:

*socks5 127.0.0.1 9050*

* + File Setup, Make it executable:

*chmod +x interactive\_system\_tester.py*

* 1. Original Version Instructions (advanced\_system\_tester.py)
* Run the Script:

*sudo ./interactive\_system\_tester.py*

* Main Menu Options:
  + configure: Set up test settings (port range, profile, verbosity).
  + targets: Select target IPs (single or from a file).
  + run: Execute tests based on the selected profile.
  + export: Generate and save reports.
  + exit: Exit the tool.
* Configure Settings:
  + Choose a port range (e.g., 1-1024).
  + Select a test profile (e.g., exploitation).
  + Enable verbose mode for detailed output.
* Select Targets:
  + Enter a single IP (e.g., 192.168.1.101) or a file with a list of IPs.
* Run Tests:
  + Execute the run command to start testing.
  + For exploitation profiles, you’ll be prompted to proceed with exploitation and provide credentials/wordlists.
* Export Reports:
  + Choose formats (text, html, csv) to save the results.
  1. Updated Version Instructions (interactive\_system\_tester.py)
* Run the Script:

*sudo ./interactive\_system\_tester.py*

* Main Menu Options: Same as the original, with additional configuration options for stealth and anonymity.
* Configure Settings:
  + Set the port range and test profile as in the original.
  + Enable verbose mode.
  + Enable Stealth Mode: Choose yes to activate stealth features (MAC spoofing, fragmented packets, etc.).
  + Use Tor for Anonymity: Choose yes to route traffic through Tor or proxies.
  + If stealth mode is enabled, provide a network interface (e.g., eth0) for MAC spoofing.
* Select Targets: Same as the original: enter a single IP or a file with IPs.
* Run Tests:
  + Execute the run command.
  + The tool will use stealth features (e.g., proxy rotation, random User-Agents) if enabled.
  + Vulnerabilities will be enriched with NVD data, including CVSS scores and descriptions.
  + For exploitation profiles, you’ll be prompted to proceed with exploitation.
* Export Reports:
  + Choose formats (text, html, csv).
  + HTML reports include detailed CVE information with severity scores.

1. Example Usage
   1. Original Version Example (advanced\_system\_tester.py)

Scenario: Test a target for vulnerabilities and attempt exploitation.

* Start the Tool: sudo ./advanced\_system\_tester.py
* Configure Settings:

A computer screen with white text

AI-generated content may be incorrect.

* Select Targets:

A computer screen with white text

AI-generated content may be incorrect.

* Run Tests:

A screenshot of a computer

AI-generated content may be incorrect.

View Report: Open interactive\_test\_report\_20250318\_001456.html in a browser to see the results.

* 1. Updated Version Example (interactive\_system\_tester.py)

Scenario: Perform a stealthy vulnerability scan with NVD integration.

* Start the Tool



* Configure Settings:

A computer screen with white text

AI-generated content may be incorrect.

* Select Targets:

A computer screen with white text

AI-generated content may be incorrect.

* Run Tests:

A screenshot of a computer screen

AI-generated content may be incorrect.

View Report: Open interactive\_test\_report\_20250318\_001456.html to see detailed CVE information and severity scores.

1. Troubleshooting

Common Issues

* Tool Not Found:
  + Error: [!] Error: <tool> is not installed.
  + Solution: Install the missing tool using sudo apt install <tool> (e.g., sudo apt install nmap).
* Permission Denied:
  + Error: [!] This script requires root privileges.
  + Solution: Run the script with sudo.
* NVD API Failure (Updated Version):
  + Error: Failed to fetch CVE details: ...
  + Solution: Check internet connectivity or NVD API status. The tool will continue with basic vulnerability data if the API fails.
* Tor/Proxy Issues (Updated Version):
  + Error: torsocks or proxychains fails to connect.
  + Solution: Ensure Tor is running (sudo systemctl start tor) and proxychains.conf is correctly configured.

# EXTRA: HOW TO UTILIZE PROXY CHAIN FOR TESTING

The updated version of interactive\_system\_tester.py includes proxy chain support via proxychains to enhance anonymity by routing traffic through a chain of proxies, such as Tor exit nodes. Below is a detailed guide on how to correctly set up and utilize a proxy chain with the tool.

1. Prerequisites for Proxy Chain Usage

Before using proxy chains, ensure the following requirements are met:

* Install proxychains: On Parrot OS or a similar Linux distribution, install proxychains:

|  |
| --- |
| sudo apt update  sudo apt install proxychains |

|  |
| --- |
| sudo apt install tor  sudo systemctl start tor  sudo systemctl enable tor |

* Install Tor (Optional but Recommended): Tor is commonly used as the proxy in the chain. Install and start the Tor service:
* Verify Tor is Running: Check the Tor service status:

|  |
| --- |
| sudo systemctl status tor |

1. Configure proxychains

The proxychains configuration file (/etc/proxychains.conf) determines how traffic is routed. Configure it to use Tor or other proxies:

* Edit the Configuration File: Open the configuration file in a text editor:

|  |
| --- |
| sudo nano /etc/proxychains.conf |

* Enable Proxy Settings:

Comment out the default SOCKS4 proxy by adding a #:

|  |
| --- |
| # socks4 127.0.0.1 9050 |

Uncomment or add the SOCKS5 proxy for Tor (default port 9050)

|  |
| --- |
| socks5 127.0.0.1 9050 |

Optionally, enable dynamic\_chain for flexible proxy routing:

|  |
| --- |
| dynamic\_chain |

Disable strict\_chain (if enabled) to allow fallback if a proxy fails:

|  |
| --- |
| # strict\_chain |

Ensure proxy\_dns is enabled to route DNS queries through the proxy:

|  |
| --- |
| proxy\_dns |

* Save and Exit: Save the file (Ctrl+O, then Ctrl+X in nano).
* Best Practices for Using Proxy Chains

|  |
| --- |
| proxychains curl https://check.torproject.org/api/ip |