accenture

NIKTO & HPING3

Name
Fatemah Kamil Alshammasi
Fatimah Ahmed Bin Dous
Alanoud Alghannam

accenture

Table of Contents

Introduction

In this rapidly evolving landscape of cybersecurity, the tools used to safeguard systems and networks are critical to mitigate modern cybersecurity threats. This report focuses on two significant tools, Nikto and HPING3, which are instrumental in identifying and mitigating vulnerabilities within computer systems. These tools not only help in security assessments but also simulate various attack scenarios to prepare and harden systems against potential malicious activities.

Nikto is a specialized web server scanner that is designed to find outdated software, misconfigurations, and dangerous server vulnerabilities, which could be exploited through attacks. Examples of such attacks include, but are not limited to, cross-site scripting (XSS) and SQL injections. Nikto serves as a preventive measure by scanning for vulnerabilities that could lead to critical security breaches.

On the other hand, HPING3 is a command-line-oriented TCP/IP packet assembler/analyzer. This tool is used for network testing, firewall testing, manual packet crafting, and network security audits. HPING3 can perform a variety of attacks, including, but not limited to, port scanning, network testing under unusual conditions, and denial of service (DoS) attacks. By simulating these attacks, HPING3 enables security professionals to analyze the response of networks to aggressive traffic conditions and unexpected packet structures.

Both tools are crucial in cybersecurity, providing the capabilities necessary to perform comprehensive security assessments and implement robust defenses. This report examines the detailed functionalities, usage procedures, and preventive actions that can be taken based on the results derived from these tools. Throughout this analysis, we aim to show some of the practical applications and effectiveness of Nikto and HPING3 in maintaining cybersecurity standards.

Nikto

Nikto is a popular tool for scanning web servers and web applications to detect possible security weaknesses. It conducts thorough assessments on web servers to identify outdated software, misconfigurations, and common security vulnerabilities, enabling various attacks such as detecting outdated software through Nikto scans. Nikto scans web servers for possibly sensitive files and directories, including configuration files, backup files, and hidden directories. Nikto conducts vulnerability assessments by scanning for typical web server vulnerabilities like cross-site scripting (XSS), SQL injection, insecure configurations, and others.

Tool Presentation

Comprehensive Scanning

- Vulnerability Detection: Nikto helps to investigate and detect vulnerabilities of web servers and associated software
- Server Misconfigurations: Nikto helps to detect the servers' misconfigurations, like incorrect directory permissions, server type, and more.
- Outdated Software: Nikto helps to detect outdated versions of server software that could lead to security risks.

Open-source and Extensible

- Open Source: Nikto is free and open source, allowing users to inspect and modify its code
- Plugins: Users can create custom plugins to extend their functionality and accommodate specialized needs.

Ease of Use

- Command-Line Interface: Nikto operates from a command-line interface, making it suitable for scripting and automation.
- Configuration: Nikto allows timeout values, port numbers, and more to be set.

Extensive Report Generation

- Output Formats: Generated reports can be setup using various formats like simple text, HTML, XML, and CSV, facilitating the meshing with other tools and systems.
- Detailed Reports: Nikto offers the generation of thorough and detailed reports that lists the vulnerabilities and problems encountered throughout the scanning

SSL Support

- HTTPS Scanning: The tool examines the security of both HTTP and HTTPS websites, offering users the ability to assess said websites.
- Certificate Information: Nikto assists in pinpointing problems that are related to certificate configuration as it can elicit and report SSL certificate details.

Customizable Scanning

- Custom Scan Options: Users can choose the type of tests they want to run, ignore tests, or emphasize certain types of vulnerability.
- Rate Limiting: Nikto enables the control of the request rate to lighten the possible overloading of the target server.

Integrated Security Testing

- Cross-Site Scripting (XSS): Nikto examines XSS vulnerability, helping to solve a frequently faced web security issue.
- SQL Injection: It also looks for SQL injection vulnerabilities that can endanger databases.

Integration with Other Tools

- Metasploit Integration: Nikto offers the direct outputting of results into Metasploit for extra exploitation.
- Nmap Integration: It offers the choice of using Nmap output to select target hosts, aiding the testing of combined network and application layer.

Nikto Commands Introduction

It is also great to mention the command “nikto -h,” which lists the basic commands to the user to help them learn and remember those commands.

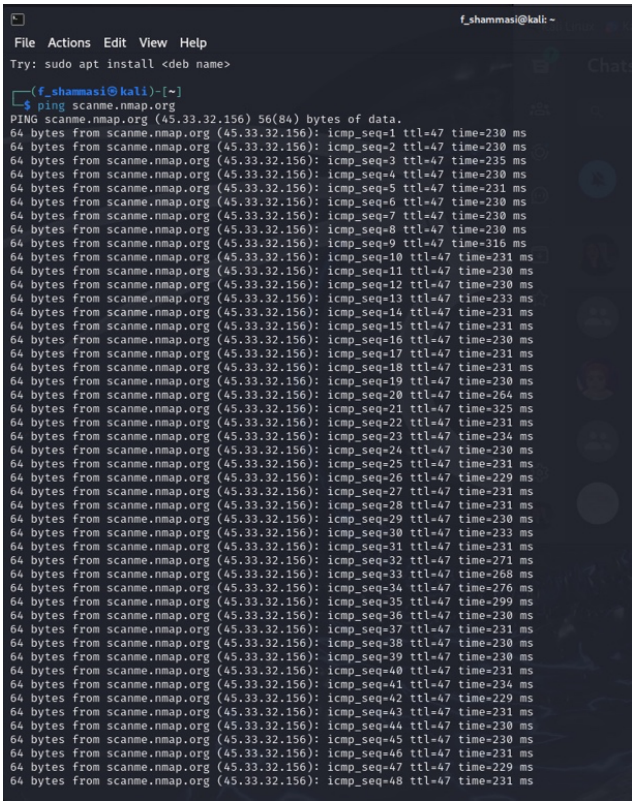
Used Procedure

Suppose a security professional is required by their company to check for any vulnerabilities and misconfigurations in their website and suppose that the company's website is "scanme.nmap.org". Here we will discuss the steps the employee followed to check their website.

Scenario Setup:
Objective: Identify security vulnerabilities in the company's website.
Target: scanme.nmap.org.
Tools: Nikto scanner on Kali Linux.

Step 1: Preparation
Installing Nikto on a Kali Linux system.
By using the following command on the terminal: sudo apt install nikto

Step 2: Reconnaissance
By using the ping command on terminal, the professional can check the connectivity with the website and getting useful information about the network structure of the target.
Using the command: ping scanme.nmap.org

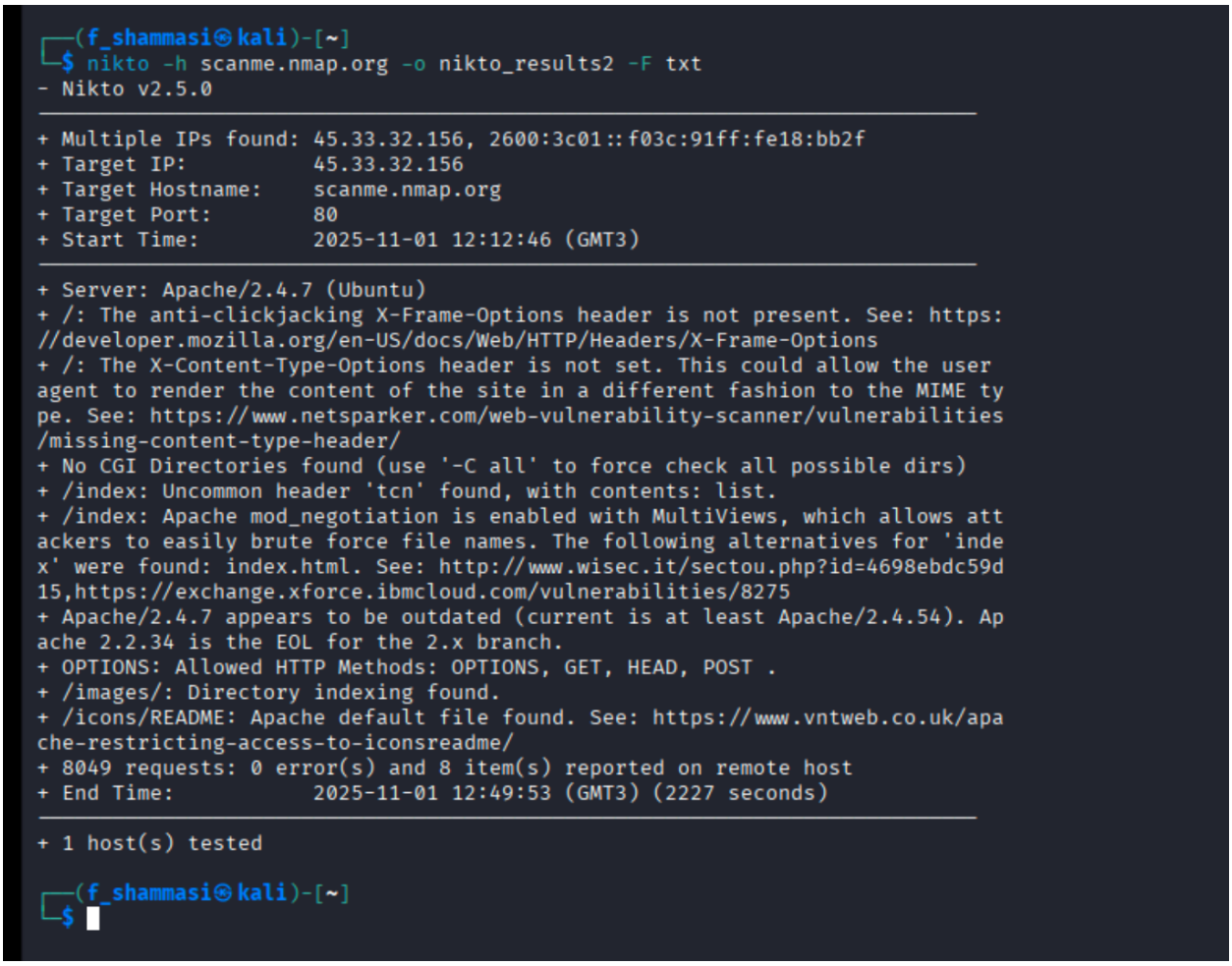


Step 3: Starting the attack

To gather information about the website, the security professional uses Nikto to scan for vulnerabilities.

The following image shows the results of the potential issues in the system.

By using the command: nikto -h scanme.nmap.org



Result and Analysis

Based on the results below, many vulnerabilities were detected, including issues related to server configurations, outdated software, and directory indexing.

```
┌──(f_shammasi㉿kali)-[~]
└─$ nikto -h scanme.nmap.org -o nikto_results2 -F txt
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Multiple IPs found: 45.33.32.156, 2600:3c01::f03c:91ff:fe18:bb2f
+ Target IP:          45.33.32.156
+ Target Hostname:    scanme.nmap.org
+ Target Port:        80
+ Start Time:         2025-11-01 12:12:46 (GMT3)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https:
//developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user
agent to render the content of the site in a different fashion to the MIME ty
pe. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities
/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows att
ackers to easily brute force file names. The following alternatives for 'inde
x' were found: index.html. See: http://www.wisec.it/sectou.php?id=4698ebdc59d
15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.54). Ap
ache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: OPTIONS, GET, HEAD, POST .
+ /images/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apa
che-restricting-access-to-iconsreadme/
+ 8049 requests: 0 error(s) and 8 item(s) reported on remote host
+ End Time:           2025-11-01 12:49:53 (GMT3) (2227 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested

┌──(f_shammasi㉿kali)-[~]
└─$
```

Here's a breakdown of the analysis:

Basic Information

- Both IPv4 and IPv6 addresses were found, indicating the server is dual-stacked.
- The server is running on port 80 with Apache version 2.4.7 on Ubuntu.

Header Vulnerabilities

- Missing X-Frame-Options: This header prevents clickjacking (clickjacking is to trick the user the click on something else to unintentionally e.g. download a malware) attacks but was not found, making the site potentially vulnerable.
- Missing X-Content-Type-Options: Without this header, the browser might execute style or script content, which might lead to MIME-type confusion attacks.

Server Outdated Vulnerability

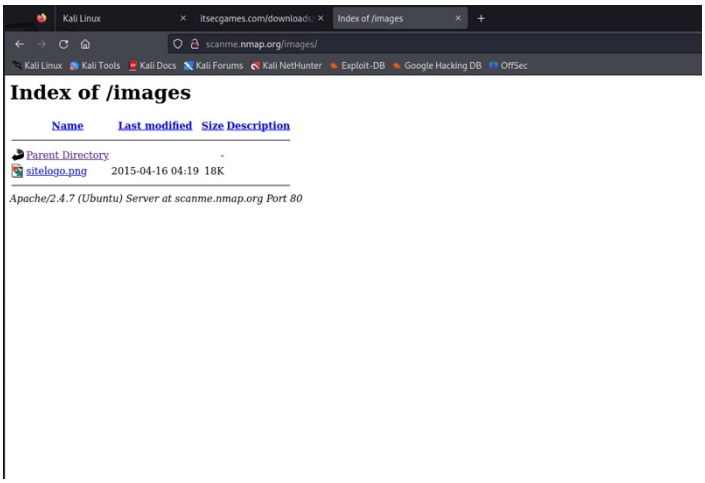
- The Apache version 2.4.7 is outdated (The latest version is 2.4.54).

Apache Configuration Vulnerabilities

- Apache mod negotiation with Multiview: This module was found enabled, which could allow an attacker to brute-force filenames of files.
- Uncommon 'tcn' Header: Found on the /index page, which could potentially be exploited.

Exposed Directories Vulnerabilities

- Directory Indexing: Enabled on /images/, which could allow an attacker to list files in directories not typically visible. In the example scanme.nmap.org, when the URL scanme.nmap.org/images/, it showed the files and directories in that directory as shown in the picture below.



- Apache Default File: An Apache server default file was found in /icons/README, which should be restricted as it can provide attackers with unnecessary information.

Why This Attack Was Successful?

The nikto tool has successfully shown vulnerabilities on the website “scanme.nmap.org” and the outdated software. Also, the tool has shown an automated insight into the links that need to be fixed and a deeper understanding of each issue.

How Was Nikto Helpful?

- Quick Identification: Nikto provided an overview of potential security issues, which is crucial for the initial phases of a security assessment.
- Detailed Reporting: Each finding was detailed, enabling investigation and mitigation.
- Adaptability: The tool's ability to handle different types of web servers and configurations demonstrates flexibility and effectiveness.

Countermeasures: Preventing Attacks Identified by Nikto

It might be impossible to eliminate all the vulnerabilities in the system, but it is possible to eliminate the vulnerabilities identified by Nikto and enhance the security of the web server. The following countermeasures can be implemented.

Update and Patch Server Software

- Upgrade the Apache server to the latest stable version to ensure all known vulnerabilities are patched.

Configure Security Headers

- X-Frame-Options: Implement the X-Frame-Options header to prevent clickjacking attacks.
- X-Content-Type-Options: Set this header to "nosniff" to prevent MIME type sniffing.

Restrict Directory Access and Indexing

- Disable Directory Listing: Configure the server to disable directory listing unless it's necessary.
- Secure Sensitive Directories: Ensure that directories containing sensitive information are protected with authentication.

Secure Apache Configuration

- Disable Unnecessary Modules: Modules like mod_negotiation should be disabled if not required to reduce the server's attack surface.
- Limit HTTP Methods: Restrict the HTTP methods that the server will accept.

Regular Security Audits and Monitoring

- Conduct Regular Security Audits: Regularly review and audit the server configuration and hosted applications for security weaknesses.

HPING3

Hping3 is a tool used via the command line for crafting packets, enabling users to alter TCP/IP packets for network testing, firewall avoidance, and reconnaissance. Advanced features are available for creating personalized packets and conducting different network scans. Different attacks can be executed, such as Firewall Evasion, where hping3 can generate packets with unique TCP flags and options to bypass firewalls or intrusion detection systems (IDS). Hping3 can conduct network scanning and port scanning by sending packets to be designated hosts and evaluating the feedback they receive. DoS Attacks: hping3 can create a large number of TCP, UDP, or ICMP packets in order to flood target systems and interfere with their services.

Tool Presentation

Versatile Packet Crafting

- Costume packet creating: Hping3 offers users customizable packets like TCP, ICMP, and UDP to send to help them with testing and fixing.
- Protocol exploration: It aids in the searching of various protocols which help users in having a better comprehension and analysis of the network behaviors
- Flexible packet parameter: Precise adjustment of packet parameters, including packet size, time to live, and source and destination addresses, is allowed to customize testing scenarios.

Comprehensive Network Testing

- Trace-route functionality: Close to the traceroute command which can trace the route packets and allow having a closer look into the network topology and performance.
- Port scanning capabilities: Hping3 allows a strong method of scanning, including TCP, ICMP, and UDP scans which facilitates for user the identification of vulnerable ports and services on targeted systems.
- Operating system detection: It offers a detection feature of the operating system of remote hosts based on the response they have for specific packets, helping in network scouting and security evaluation.

Security Assessment and Attack Simulation

- Firewall Testing: It allows for sending multiple types of packets and inspecting responses to evaluate the efficiency of firewalls and intrusion detecting and prevention systems.
- Denial of Service Attacks (DoS): Hping3 have the ability to imitate DoS attacks by continuously sending high volume packets to a targeted system, which helps in testing their resilience to similar attacks.
- OS Fingerprinting: The tool helps in verifying targeted system’s OS by analyzing the pattern in which they response, which betters the network’s assessment.

Advanced Operation Options

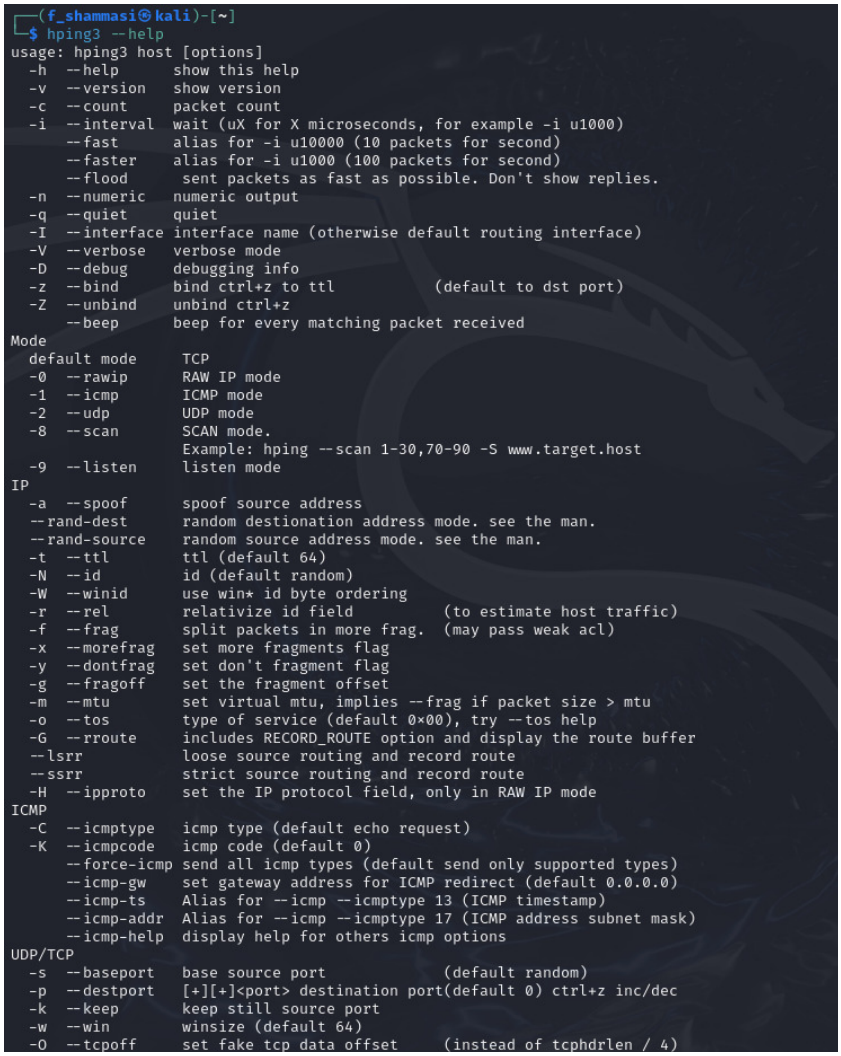
- Packet timing control: It allows users to control the timing of the packets to have accurate packet testing and transmission
- Source addresses spoofing: Hping3 can fake source addresses, allowing anonymity and attacking network defense.

Command Line Interface

- Efficient operation: As the tool works from the command line, it is suitable for scripting, automation, and blending with different systems.
- Easy configuration: It has an easy mechanism to configure options, enabling users to set parameters.

HPING3 Commands Introduction

It is also great to mention the command “hping3 --help” which lists the basic commands to the user to help them learn and remember those commands.

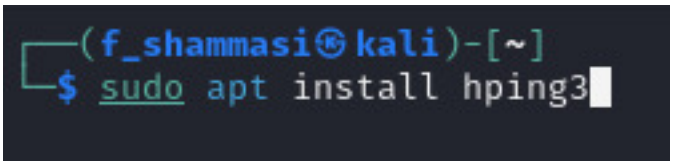


Used Procedure

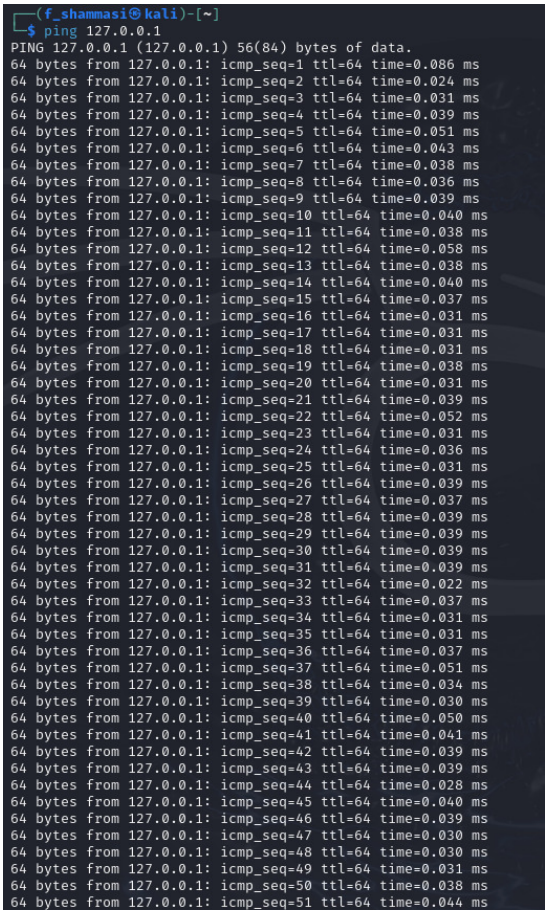
A security professional at XYZ company has been hired conduct a penetration test on their network to identify potential vulnerabilities. The primary objective is to detect open ports on the web server without alerting the intrusion detection systems (IDS) that monitor the network for malicious activities. The security professional decides to use hping3 to perform this test. Here we will discuss the steps the employee followed.

Scenario Setup:
Objective: Detect open ports on the web server
Target: 127.0.0.1
Tools: Hping3 on Kali Linux.

Step 1: Preparation
Installing Hping3 on a Kali Linux system.
By using the following command on the terminal: sudo apt install hping3



Step 2: Reconnaissance
By using the ping command on terminal, the professional can check the connectivity with the server and getting useful information about the network structure of the target.
Using the command: ping 127.0.0.1



Step 3: Starting the attack

To check the network security posture, the security professional uses Hping3 to scan for vulnerabilities.

The following image shows the results of the potential issues in the system.

By using multiple commands.

```
┌──(f_shammasi㉿kali)-[~]
└─$ sudo hping3 127.0.0.1 -c 7 -F
HPING 127.0.0.1 (lo 127.0.0.1): F set, 40 headers + 0 data bytes
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=8.1 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=3.9 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=2.4 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=3 win=0 rtt=1.3 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=4 win=0 rtt=4.6 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=5 win=0 rtt=2.3 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=6 win=0 rtt=8.8 ms

── 127.0.0.1 hping statistic ──
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 1.3/4.5/8.8 ms
```

```
┌──(f_shammasi㉿kali)-[~]
└─$ sudo hping3 127.0.0.1 -c 7 -S
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=10.1 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=1.6 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=2 win=0 rtt=4.9 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=3 win=0 rtt=4.6 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=4 win=0 rtt=3.1 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=5 win=0 rtt=3.7 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=6 win=0 rtt=3.0 ms

── 127.0.0.1 hping statistic ──
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 1.6/4.4/10.1 ms
```

```
┌──(f_shammasi㉿kali)-[~]
└─$ sudo hping3 127.0.0.1 -c 7 -S -p 1-65535
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=1 flags=RA seq=0 win=0 rtt=2.7 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=1 flags=RA seq=1 win=0 rtt=2.6 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=1 flags=RA seq=2 win=0 rtt=1.9 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=1 flags=RA seq=3 win=0 rtt=0.6 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=1 flags=RA seq=4 win=0 rtt=11.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=1 flags=RA seq=5 win=0 rtt=7.9 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=1 flags=RA seq=6 win=0 rtt=1.7 ms

── 127.0.0.1 hping statistic ──
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 0.6/4.1/11.0 ms
```

```
┌──(f_shammasi㉿kali)-[~]
└─$ sudo hping3 127.0.0.1 -c 1 -S -f -d 155
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 155 data bytes
[]tcp]
[]tcp]
[]tcp]
[]tcp]
[]tcp]
[]tcp]
[]tcp]
[]tcp]
[]tcp]
[]tcp]
[]tcp]
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=7.1 ms

── 127.0.0.1 hping statistic ──
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 7.1/7.1/7.1 ms
```

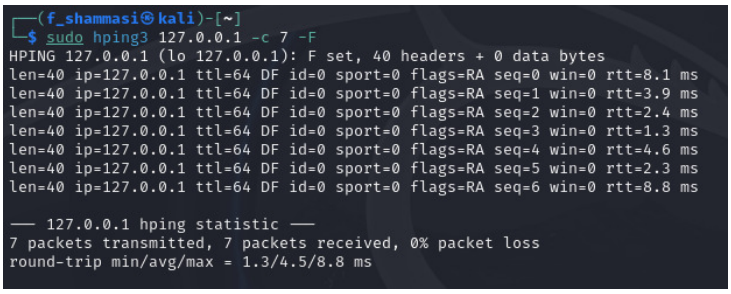
```
┌──(f_shammasi㉿kali)-[~]
└─$ sudo hping3 127.0.0.1 -c 1 -S -d 1000 -k --icmp-ts
HPING 127.0.0.1 (lo 127.0.0.1): icmp mode set, 28 headers + 1000 data bytes
len=40 ip=127.0.0.1 ttl=64 id=10687 icmp_seq=0 rtt=8.6 ms
ICMP timestamp: Originate=48422670 Receive=48422670 Transmit=48422670
ICMP timestamp RTT tsrtt=8


── 127.0.0.1 hping statistic ──
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 8.6/8.6/8.6 ms
```

```
┌──(f_shammasi㉿kali)-[~]
└─$ sudo hping3 -S -p 80 -c 1 127.0.0.1
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=3.6 ms

── 127.0.0.1 hping statistic ──
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 3.6/3.6/3.6 ms
```

```
┌──(f_shammasi㉿kali)-[~]
└─$ sudo hping3 -F -p 80 -c 1 127.0.0.1
HPING 127.0.0.1 (lo 127.0.0.1): F set, 40 headers + 0 data bytes
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=6.3 ms

── 127.0.0.1 hping statistic ──
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 6.3/6.3/6.3 ms
```

Result and Analysis

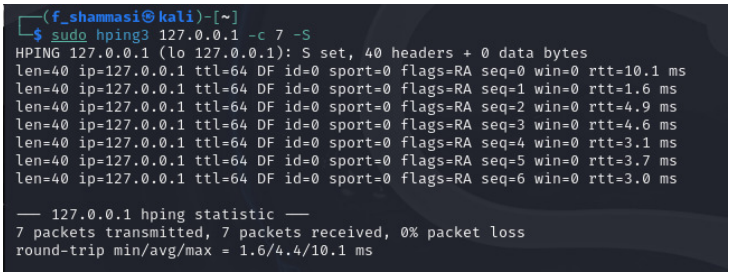
Based on the results below, some information related to TCP/IP packet manipulations and network test targeting was captured. Here's a breakdown of the analysis:



Command 1: sudo hping3 127.0.0.1 -c 7 -F

Purpose: This command is typically used to check how a system reacts to TCP FIN packets, which are used to close a TCP connection.

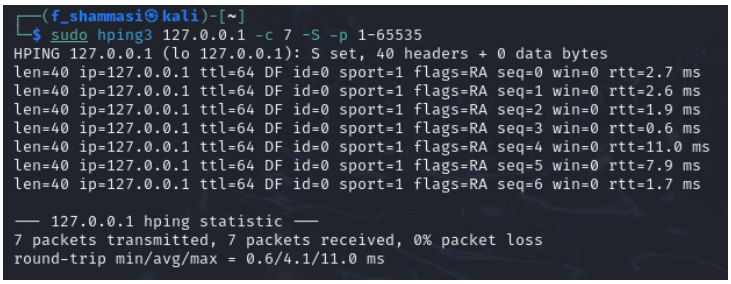
Result Analysis: The responses indicate that the host is actively rejecting the attempt to establish a connection using a FIN packet, which is expected behavior for closed ports.



Command 2: sudo hping3 127.0.0.1 -c 7 -S

Purpose: This command is used to perform a SYN scan, which is useful for identifying open ports on the target system.

Result Analysis: The responses indicate that there is no active listener on these ports, leading to the rest of the attempted connections.



all ports closed

Command 3: sudo hping3 127.0.0.1 -c 7 -S -p 1-65535

Purpose: This command attempts to SYN scan every single TCP port on the host.

Result Analysis: Similar to the previous SYN scan, the RA (Reset and Acknowledge) flags indicate no listeners on these ports except for specifying port range.

```
┌──(f_shammasi㉿kali)-[~]
└─$ sudo hping3 127.0.0.1 -c 1 -S -f -d 155
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 155 data bytes
[tcp]
[tcp]
[tcp]
[tcp]
[tcp]
[tcp]
[tcp]
[tcp]
[tcp]
[tcp]
[tcp]
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=7.1 ms

── 127.0.0.1 hping statistic ──
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 7.1/7.1/7.1 ms
```

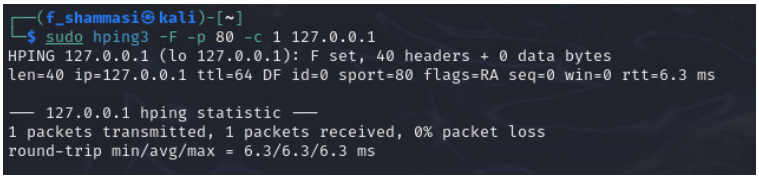
Command 4: sudo hping3 127.0.0.1 -c 1 -S -f -d 155
Purpose: This command is to check how the system handles fragmented packets that attempt to establish a TCP connection.
Result Analysis: The RA response indicates that the packet was received and reset, suggesting that the target can handle fragmented SYN packets.

```
┌──(f_shammasi㉿kali)-[~]
└─$ sudo hping3 127.0.0.1 -c 1 -S -d 1000 -k --icmp-ts
HPING 127.0.0.1 (lo 127.0.0.1): icmp mode set, 28 headers + 1000 data bytes
len=40 ip=127.0.0.1 ttl=64 id=10687 icmp_seq=0 rtt=8.6 ms
ICMP timestamp: Originate=48422670 Receive=48422670 Transmit=48422670
ICMP timestamp RTT tsrtt=8


── 127.0.0.1 hping statistic ──
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 8.6/8.6/8.6 ms
```

Command 5: sudo hping3 127.0.0.1 -c 1 -S -d 1000 -k --icmp-ts
Purpose: This command sends a SYN packet with a large payload and an ICMP timestamp, which can be used to gain information about the host's clock.
Result Analysis: The ICMP timestamp response provides the system's timestamp values.

```
┌──(f_shammasi㉿kali)-[~]
└─$ sudo hping3 -S -p 80 -c 1 127.0.0.1
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=3.6 ms

── 127.0.0.1 hping statistic ──
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 3.6/3.6/3.6 ms
```

Command 6: sudo hping3 -S -p 80 -c 1 127.0.0.1
Purpose: This tests the reaction to a SYN packet on HTTP port 80.
Result Analysis: The RA flag indicates that no service is listening on port 80; the connection attempt is reset.



Command 7: sudo hping3 -F -p 80 -c 1 127.0.0.1
Purpose: Like the first command but specifically targeting port 80 to check for its response to a FIN packet.
Result Analysis: The RA flag response confirms that port 80 is closed as it resets the connection attempt initiated by the FIN flag.

Why This Attack Was Successful?

- The "attack" can be considered successful in terms of it being a valuable test or audit rather than compromising the system. It successfully demonstrated the system's preparedness and response to various common and uncommon network events.
- All packets (TCP and ICMP) sent received responses, indicating that the target is actively responding to and processing the packets.
- No Disruption: There was no evidence of system disruption or degradation, indicating robust security and network handling configurations.

How Was HPING3 Helpful?

- Hping3's ability to manipulate packet flags, types, and sizes is instrumental in testing network behavior under diverse conditions.
- Immediate statistics and feedback on packet loss, round-trip time, and behavior under flooding conditions provide invaluable data for real-time security assessment.

Countermeasures: Preventing Attacks Identified by HPING3

It might be impossible to eliminate all the vulnerabilities in the system, but it is possible to eliminate the vulnerabilities identified by this test and enhance the security of the related TCP/IP packet manipulations and network tests targeting, the following countermeasures can be implemented.

Firewall Configuration

- Configure firewalls to allow only necessary ports and protocols based on the specific needs of your network. This reduces the attack surface that can be exploited.

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS)

- These systems can detect and prevent suspicious activities by analyzing traffic patterns and identifying anomalies that may indicate an attack, such as unexpected packet flags or flood attempts.
- Keep the IDS/IPS rules updated to minimize false positives while effectively catching malicious activities.

Anti-DDoS Solutions

- Utilize specialized DDoS protection hardware and software solutions that can absorb and mitigate large-scale DDoS attacks.

Conclusion

The report reviewed the functionality and effectiveness of two cybersecurity tools: Nikto and HPING3. We identified key vulnerabilities within the systems tested. Our analysis highlighted the vulnerabilities and outlined practical countermeasures to prevent these security risks.

Nikto demonstrated effectiveness in scanning web servers and identifying various security weaknesses such as outdated software and misconfigurations etc. HPING3 was used in packet crafting and network testing, showing potential network defenses' against various attack vectors. Overall, both tools provided insight into system vulnerabilities and countermeasure effectiveness.

Experimenting with these tools helped us understand more about the cybersecurity field. The challenges encountered during the testing phase deepened our understanding of network security architecture and its vulnerabilities.

In conclusion, this exercise showed the importance of regular security assessments and continuously updating the security measures to guard against cyber threats. It has been a valuable educational experiment to understand and utilize cybersecurity tools in real-world settings.