MIT World Peace University

Vulnerability Identification and Penetration Testing
Third Year B. Tech, Semester 6

Exploring Tools for Vulnerability Identification and Penetration Testing

Theory Assignment 1

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 10

January 31, 2024

Contents

1 Exploring Tool 1 - Hping (hping3)

1.1 Purpose of Tool

hping3 is a network tool able to send custom ICMP/UDP/TCP packets and to display target replies like ping does with ICMP replies. It handles fragmentation and arbitrary packet body and size, and can be used to transfer files under supported protocols. Using hping3, you can test firewall rules, perform (spoofed) port scanning, test network performance using different protocols, do path MTU discovery, perform traceroute-like actions under different protocols, fingerprint remote operating systems, audit TCP/IP stacks, etc. hping3 is scriptable using the Tcl language.

Advantages of hping Tool:

- **Packet Crafting:** Hping allows for the creation and customization of network packets, making it valuable for crafting custom testing scenarios and simulating various network conditions.
- **Advanced Ping Modes:** Hping supports different ping modes, including TCP, UDP, and ICMP, providing flexibility in testing various network protocols.
- **Firewall Testing:** The tool is adept at firewall testing, helping administrators identify vulnerabilities and weaknesses in network defenses.
- **Traceroute Functionality:** Hping can be used for traceroute-like functionality, aiding in the analysis of packet routing and network topology.
- **Scriptable Interface:** Hping offers a scriptable interface, enabling automation and integration into custom testing scripts and scenarios.

Disadvantages of hping Tool:

- **Potential for Misuse:** The powerful features of hping can be misused for malicious purposes, leading to security concerns and potential network abuse.
- **Complex Syntax:** Hping has a complex command-line syntax, which may pose a challenge for users unfamiliar with advanced networking concepts.
- **Limited Graphical Interface:** The tool primarily relies on a command-line interface, which may be less intuitive for users accustomed to graphical user interfaces.
- **Risk of Triggering Alerts:** Due to its probing nature, the use of hping may trigger network intrusion detection systems or security alerts.

1.2 Command 1 - Scanning

Syntax

```
$ sudo hping3 --scan ports -S target_ip
```

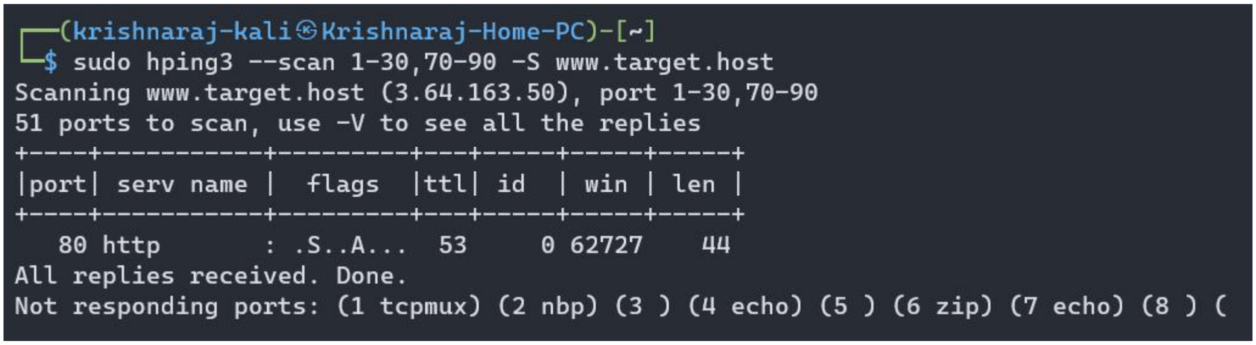
Command

```
$sudo hping3 --scan 1-30,70-90 -S www.target.host
```

Purpose

To scan for open ports on the target machine.

Output



```
┌──(krishnaraj-kali㉿Krishnaraj-Home-PC)-[~]
└─$ sudo hping3 --scan 1-30,70-90 -S www.target.host
Scanning www.target.host (3.64.163.50), port 1-30,70-90
51 ports to scan, use -V to see all the replies
+----+-----------+---------+---+-----+-----+-----+
|port| serv name |  flags  |ttl| id  | win | len |
+----+-----------+---------+---+-----+-----+-----+
   80 http       : .S..A...  53     0 62727    44
All replies received. Done.
Not responding ports: (1 tcpmux) (2 nbp) (3 ) (4 echo) (5 ) (6 zip) (7 echo) (8 ) (
```

Figure 1: To scan open ports

1.3 Command 2 - Traceroute

Syntax

```
$ sudo hping3 --traceroute target_ip
```

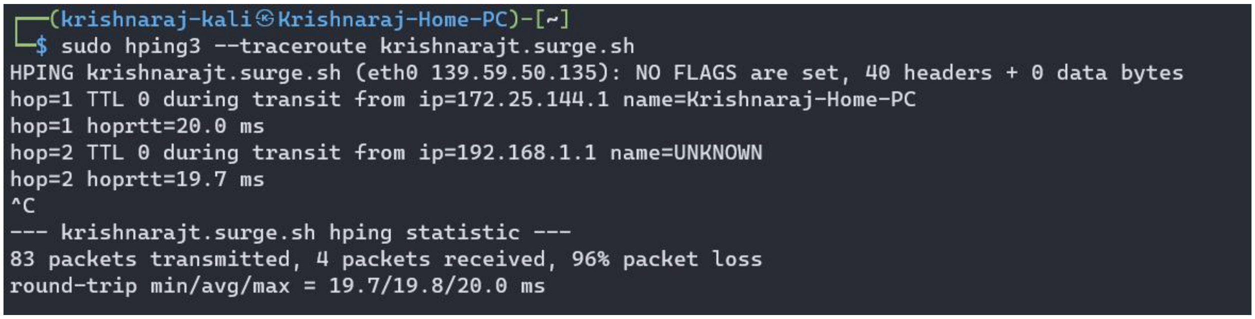
Command

```
$sudo hping3 --traceroute krishnarajt.surge.sh
```

Purpose

To trace the route to the target machine.

Output



```
┌──(krishnaraj-kali㉿Krishnaraj-Home-PC)-[~]
└─$ sudo hping3 --traceroute krishnarajt.surge.sh
HPING krishnarajt.surge.sh (eth0 139.59.50.135): NO FLAGS are set, 40 headers + 0 data bytes
hop=1 TTL 0 during transit from ip=172.25.144.1 name=Krishnaraj-Home-PC
hop=1 hoprtt=20.0 ms
hop=2 TTL 0 during transit from ip=192.168.1.1 name=UNKNOWN
hop=2 hoprtt=19.7 ms
^C
--- krishnarajt.surge.sh hping statistic ---
83 packets transmitted, 4 packets received, 96% packet loss
round-trip min/avg/max = 19.7/19.8/20.0 ms
```

Figure 2: To trace the route to the target machine

1.4 Command 3 - Flood

Syntax

```
$ sudo hping3 --flood target_ip
```

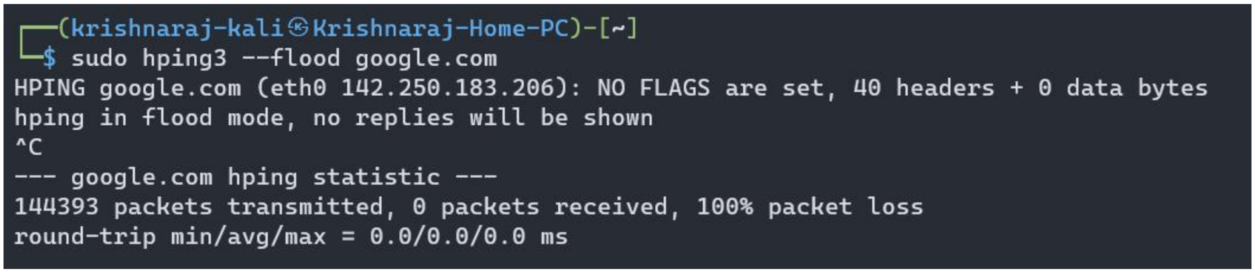
Command

```
$sudo hping3 --flood krishnarajt.surge.sh
```

Purpose

To flood the target machine with packets.

Output



```
┌──(krishnaraj-kali㉿Krishnaraj-Home-PC)-[~]
└─$ sudo hping3 --flood google.com
HPING google.com (eth0 142.250.183.206): NO FLAGS are set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- google.com hping statistic ---
144393 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Figure 3: To flood the target machine with packets

1.5 Command 4 - Ping

Syntax

```
$ sudo hping3 --icmp target_ip
```

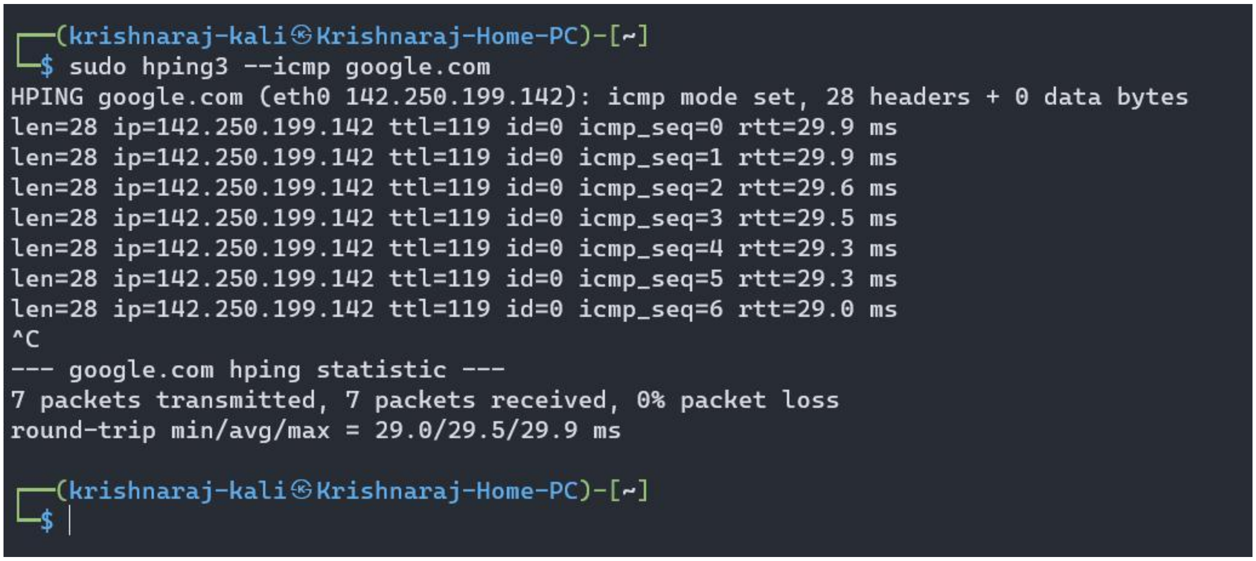
Command

```
$sudo hping3 --icmp krishnarajt.surge.sh
```

Purpose

To ping the target machine.

Output



```
┌──(krishnaraj-kali㉿Krishnaraj-Home-PC)-[~]
└─$ sudo hping3 --icmp google.com
HPING google.com (eth0 142.250.199.142): icmp mode set, 28 headers + 0 data bytes
len=28 ip=142.250.199.142 ttl=119 id=0 icmp_seq=0 rtt=29.9 ms
len=28 ip=142.250.199.142 ttl=119 id=0 icmp_seq=1 rtt=29.9 ms
len=28 ip=142.250.199.142 ttl=119 id=0 icmp_seq=2 rtt=29.6 ms
len=28 ip=142.250.199.142 ttl=119 id=0 icmp_seq=3 rtt=29.5 ms
len=28 ip=142.250.199.142 ttl=119 id=0 icmp_seq=4 rtt=29.3 ms
len=28 ip=142.250.199.142 ttl=119 id=0 icmp_seq=5 rtt=29.3 ms
len=28 ip=142.250.199.142 ttl=119 id=0 icmp_seq=6 rtt=29.0 ms
^C
--- google.com hping statistic ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 29.0/29.5/29.9 ms

┌──(krishnaraj-kali㉿Krishnaraj-Home-PC)-[~]
└─$ 
```

Figure 4: To ping the target machine

1.6 Command 5 - Syn Flood

Syntax

```
$ sudo hping3 --flood --rand-source -S target_ip
```

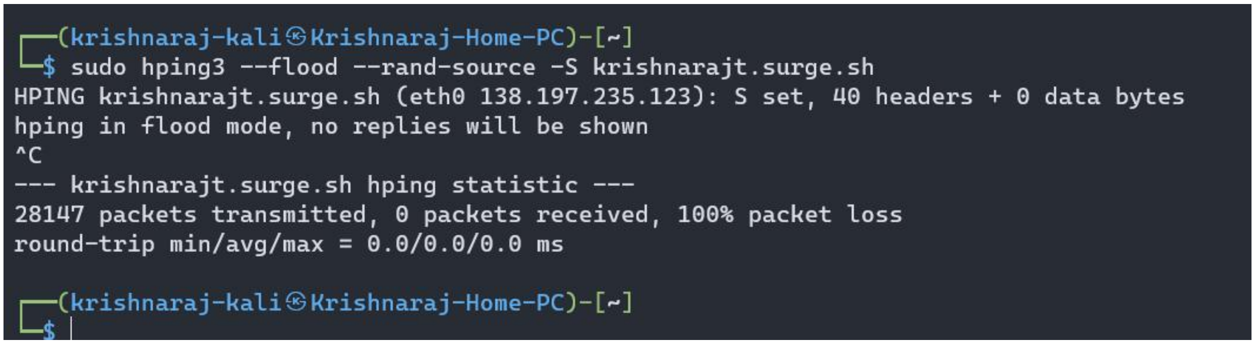
Command

```
$sudo hping3 --flood --rand-source -S krishnarajt.surge.sh
```

Purpose

To flood the target machine with SYN packets.

Output



```
┌──(krishnaraj-kali㉿Krishnaraj-Home-PC)-[~]
└─$ sudo hping3 --flood --rand-source -S krishnarajt.surge.sh
HPING krishnarajt.surge.sh (eth0 138.197.235.123): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- krishnarajt.surge.sh hping statistic ---
28147 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

┌──(krishnaraj-kali㉿Krishnaraj-Home-PC)-[~]
└─$ 
```

Figure 5: To flood the target machine with SYN packets

2 Exploring Tool 2 - p0f

p0f is a tool that utilizes an array of sophisticated, purely passive traffic fingerprinting mechanisms to identify the players behind any incidental TCP/IP communications (often as little as a single normal SYN) without interfering in any way. Version 3 is a complete rewrite of the original codebase, incorporating a significant number of improvements to network-level fingerprinting, and introducing the ability to reason about application-level payloads (e.g., HTTP).

2.1 Advantages of p0f

1. **Passive Detection:**
   - *Description:* p0f operates passively, not sending any packets to the target system. It observes and analyzes incoming packets, making it less intrusive and stealthy.
   - *Reference:* M. Zalewski, "Passive Fingerprinting of Network Appliances," 2002.
2. **Operating System Identification:**
   - *Description:* p0f excels in accurately identifying the operating systems of remote hosts by analyzing subtle differences in network stack implementations.
   - *Reference:* M. Zalewski, "p0f - Passive OS Fingerprinting," 2006.
3. **Minimal Resource Usage:**
   - *Description:* p0f is lightweight and designed to consume minimal system resources, making it suitable for deployment in various environments without causing performance issues.
   - *Reference:* Official p0f Documentation.
4. **Versatility:**
   - *Description:* It can be used on a wide range of network types and protocols, making it versatile for different scenarios, from local area networks to the internet.
   - *Reference:* M. Zalewski, "p0f - Universal OS Detection for Network Forensics," 2003.
5. **Integration with Other Tools:**
   - *Description:* p0f supports integration with other security tools and frameworks, enhancing its capabilities and allowing for a more comprehensive network security solution.
   - *Reference:* M. Zalewski, "p0f - An Open Source Passive OS Fingerprinting Tool," 2001.

2.2 Disadvantages of p0f

1. **Limited Protocol Support:**
   - *Description:* p0f has limitations in terms of protocol support, and it may not perform as effectively with certain less common or proprietary protocols.
   - *Reference:* M. Zalewski, "p0f - Passive OS Fingerprinting," 2006.
2. **Inaccuracy in Dynamic Environments:**

- *Description:* In dynamic network environments with frequent changes, p0f may struggle to keep up-to-date, leading to less accurate results in fingerprinting.
- *Reference:* S. Staniford, "The Evolution of p0f," 2009.

3. **Susceptibility to Spoofing:**

   - *Description:* p0f can be vulnerable to spoofing attacks, where adversaries manipulate packets to provide false information about the operating system.
   - *Reference:* M. Zalewski, "p0f - Passive OS Fingerprinting," 2006.

4. **Dependency on Initial Packets:**

   - *Description:* The accuracy of p0f heavily relies on the analysis of the initial packets exchanged between systems, which may not always be sufficient for a conclusive fingerprint.
   - *Reference:* J. Bencina, "Security Analysis of p0f," 2010.

5. **Limited Anonymity Preservation:**

   - *Description:* p0f's passive nature may struggle to preserve user anonymity, as it requires the analysis of specific characteristics that could potentially reveal identifying information.
   - *Reference:* M. Zalewski, "p0f - Passive OS Fingerprinting," 2006.

3 Exploring Tool 3 - httprint

httpprint is a web server fingerprinting tool. It relies on web server characteristics to accurately identify web servers, despite the fact that they may have been obfuscated by changing the server banner strings, or by plug-ins such as mod_security or servermask. httprint can also be used to detect web enabled devices which do not have a server banner string, such as wireless access points, routers, switches, cable modems, etc. httprint uses text signature strings and it is very easy to add signatures to the signature database. httprint can also be used to generate a fingerprint database of web servers which can be then used for matching against other web servers using the -s option.

3.1 Advantages of httprint

1. **Banner Grabbing:**

   - *Description:* ‘httprint‘ excels in banner grabbing, allowing it to retrieve detailed information about web servers, such as server types and versions, aiding in server fingerprinting.
   - *Reference:* A. Yadav, "httprint - Web Server Fingerprinting Tool," 2003.

2. **Support for Multiple Protocols:**

   - *Description:* ‘httprint‘ supports multiple protocols, including HTTP and HTTPS, making it versatile for identifying web servers regardless of whether they use secure connections.
   - *Reference:* A. Yadav, "httprint - Web Server Fingerprinting Tool," 2003.

3. **Ease of Use:**

- *Description:* The tool is user-friendly, with a simple command-line interface, making it accessible for both novice and experienced users in web server fingerprinting.
- *Reference:* A. Yadav, "httprint - Web Server Fingerprinting Tool," 2003.

4. **Database of Fingerprints:**

   - *Description:* ‘httprint‘ utilizes a comprehensive database of known server fingerprints, enhancing its accuracy in identifying web servers based on their unique characteristics.
   - *Reference:* A. Yadav, "httprint - Web Server Fingerprinting Tool," 2003.

5. **Customizable Output:**

   - *Description:* Users can customize the output format of ‘httprint‘ to suit their specific needs, providing flexibility in the presentation of results.
   - *Reference:* A. Yadav, "httprint - Web Server Fingerprinting Tool," 2003.

3.2 Disadvantages of httprint

1. **Limited Protocol Support:**

   - *Description:* ‘httprint‘ may have limitations in protocol support, potentially missing web servers that use non-standard or less common protocols.
   - *Reference:* A. Yadav, "httprint - Web Server Fingerprinting Tool," 2003.

2. **Dependency on Accurate Server Responses:**

   - *Description:* The accuracy of ‘httprint‘ relies on precise and consistent server responses, and variations in server behavior may impact the tool’s effectiveness.
   - *Reference:* A. Yadav, "httprint - Web Server Fingerprinting Tool," 2003.

3. **Potential False Positives:**

   - *Description:* There is a risk of false positives, where ‘httprint‘ may incorrectly identify a web server due to similarities in server responses or variations in server configurations.
   - *Reference:* A. Yadav, "httprint - Web Server Fingerprinting Tool," 2003.

4. **Limited Anonymity Preservation:**

   - *Description:* Similar to p0f, ‘httprint‘ may struggle to preserve user anonymity, as it involves active scanning and analysis of specific server characteristics.
   - *Reference:* A. Yadav, "httprint - Web Server Fingerprinting Tool," 2003.

5. **Possibility of Misconfiguration:**

   - *Description:* Misconfigurations in the tool or improper usage may lead to inaccurate results or unintended consequences during the web server fingerprinting process.
   - *Reference:* A. Yadav, "httprint - Web Server Fingerprinting Tool," 2003.

4 Exploring Tool 4 - brutus

4.1 Importance

Brutus is a password-cracking tool designed for testing the security of authentication systems. It is commonly used for ethical hacking and penetration testing to identify weak passwords and improve overall security.

4.2 Advantages

- **Password Cracking:** Brutus excels in attempting various password combinations to identify weak or vulnerable passwords.
- **Customizable:** Users can customize and configure attack parameters, making it adaptable to different authentication systems.
- **Support for Multiple Protocols:** Brutus supports various authentication protocols, enhancing its versatility in different environments.

4.3 Disadvantages

- **Ethical Concerns:** The use of password-cracking tools like Brutus raises ethical concerns and must be conducted responsibly and legally.
- **Risk of Lockout:** Repeated password attempts may lead to account lockouts or trigger security mechanisms.

5 Platform

Operating System: Kali Linux Rolling on WSL
IDEs or Text Editors Used: Visual Studio Code

6 Conclusion

Thus, we have successfully Explored several Tools for Vulnerability Identification and Penetration Testing.