**Penetration Testing: A Comprehensive Report on Exploits and Mitigations**

Student’s name

Institution

Course

Professor’s name

Due Date

**Table of Content**

1. Introduction/Overview
2. Test 1: Obtaining a Meterpreter Shell Using MSF
3. Test 2: Java Exploit
4. Test 3: Reverse TCP Shell Exploit
5. Test 4: PDF Exploit
6. Test 5: Aurora Exploit
7. Conclusion
8. References

**Introduction/Overview**

Pentesting is an important activity that involves the use of professional cyber attackers to attempt to breach an organization’s systems with a view of establishing their weak points. It involves anticipating threats which prevent nasty players from capitalizing on cracks that already exist. The emphasis of this work is made on a series of pentests by applying various exploits in the context of one case and testing the corresponding countermeasures. The testing environment consists of two primary machines: a prototype and a Kali Linux system with Metasploit Framework to launch the exploitation, and a target Windows XP virtual machine with intentionally installed outdated programs. Here are the tests carried out in this project: getting a meterpreter shell with MSF, exploiting Java weakness, executing a reverse TCP shell, PDF prober and the Aurora one. Every test shows various attack techniques employed by hackers to infiltrate networks, extort information, or persist. Thus, the presented project tries to embrace and offer the useful knowledge about the emerging of the opportunities for proper control over the weak spots and making security more robust.

**Test 1: Obtaining a Meterpreter Shell Using MSF**

**Vulnerability Being Tested**

The first test of this report aims at taking advantage of a weakness in the windows XP operating system with help of the Metasploit Framework (MSF). More specifically, the attacker was able to take advantage of the MS08-067 capability found in the SMB service to have unauthorized access to of the target machine (Microsoft. 2008). This weakness results in an attacker being able to run code of their choice on a compromised system, which results to RCE. The MS08-067 has it that this is one of the most documented bugs that was very popular in penetration testing since the hole was very easy to exploit and always yielded positive results. The problem is lay in a buffer overflow within the Microsoft Server Message Block(SMB) service that used for sharing files and printers over network. The specific situation is that attackers can transmit the specific and specially constructed packet to the SMB service of the target machine. After the vulnerability is activated, an attacker is capable of installing a payload, for instance, the meterpreter shell to achieve control over the target.

**Configuration of Devices**

The testing setup involved two machines: a Kali Linux machine with Metasploit Framwork and a Windows XP target with installed MS08-067 exploit. On the attacker side, they use Kali Linux machine and therefore Metasploit as a framework in which the exploit is launched and the payload is transmitted (Microsoft. 2008). This Windows XP source was using a deliberately old operating system version with no security fixes at all, meaning it could be spammed with default known vulnerabilities such as MS08-067.

**Exploit Process**

The process was started with the use of Nmap to determine the open ports and the services running on the windows XP machine. It was known that the SMB service (port 445) was taken and the target machine was running Windows XP which was out dated and an exploit was found for it ms08-067. Exploiting the SMB protocol, the particular exploit module in Metasploit was chosen as windows/smb/ms08\_067\_netapi. After the exploit was launched an attacker machine used specially crafted packets to the target machines SMB service that caused the buffer overflow. In the case of the exploit concerning Supeofs, after the exploit is run and completes it successfully, Metasploit introduced a meterpreter payload onto the target’s system which allowed remote access. The attacker was also able to communicate with the meterpreter shell, enter command, and access the system file.

**Test Results**

The exploitation was successful, and the meterpreter shell has been acquired. In that specific session meterpreter, the attacker had complete control over the victim’s computer operating system besides running commands, increasing the privileges and infiltration of data. This vulnerability is well exploited by attackers and proves how these attackers can easily take control of outdated systems provided care isn’t taken to organize defenses and updates.

**Mitigations**

To reduce such vulnerability, it is recommended that organization should update their systems with the most current version of patches. In the case of MS08-067 Microsoft issued an update for what was found to be unsafe in the SMB systems and it is strongly advised that the patch be installed on all XP and lower operating system based computers (Metasploit Project. (n.d.). Third, firewall settings need to be set to reject any inbound connections to unnecessary services on a LAN such as SMB from other networks. Introduction of IDS to the system can allow the detection of network traffic that may be attributed to attempts at exploitation. These should also be implemented on a regular basis as part of vulnerability assessment and Penetration testing in order to optimise security (Microsoft. 2008). Last but not the least, organizations which are still in the use of operating systems should upgrade to the current ones since the new operating systems boast of lesser exposed vulnerabilities.

**Test 2: Java Exploit**

**Vulnerability Being Tested**

In the second test, the possibility of using Java vulnerability, Java Runtime Environment version that is outdated to perform a RCE attack is examined. Java vulnerabilities have remained popular within the attackers’ circles because Java has become a popular platform for both corporate and Internet applications. Of the file one is CVE-2013-2465 where attackers can gain the ability to execute code on the victim’s computer by loading Java runtime with an applet containing a dangerous code. Java applets are actual applications coded in Java that are embedded into web browsers and other Java compatibility platforms. When end-users access a page with a contaminated applet, further subversion in the JRE is done to run code on the user’s computer. This can lead to full system compromise or theft of the data or even more exploitation of the network.

**Configuration of Devices**

The test setup consisted of one machine using Kali Linux and another that had Windows 7 operating system with an insecure Java version. Psexec was employed through the Metasploit Framework on the Kali machine to launch the Java exploit that sought to attack the Java Runtime Environment on the Windows computer. Furthermore, to make the exploitation work, the target system was installed with an unpatched vulnerable version of Java to be exploited.

**Exploit Process**

To exploit this Jason chose to scan for a channel, which is done with the help of ids with Metasploit auxiliary modules to identify the version of Java that the target computer uses. When the vulnerable version had been established, the Metasploit module multi/browser/java\_rce was used. This module was intended to take advantage of a weakness in Java applets as the means of installing a payload. The exploit was initiated by making the target user access a given site with the Java applet instrumental in the exploitation. On revisiting the site, the Java applet was run and given the loophole, the attacker was in a position to launch a code at the site’s Transient Code Cache on the target machine. The payload in this case delivered was a reverse TCP shell, which allowed the attacker to continuously connect to the client system.

**Test Results**

Java incident was accomplished: got a reverse TCP shell on the target machine. The attacker was able to execute commands on the target system, read on sensitive files and gain further privileges. This goes to show the dangers of running end of support Java environments along with unpatched environments where attackers can readily compromise systems.

**Mitigations**

In order not to be a victim of Java related vulnerabilities, the user needs to make certain the Java Runtime Environment that they are using has the latest security patches. Java users should only run Java in web browsers when necessary, because current browsers have incorporated measures of preventing dangerous applets. Further, an organization should consider using of Java security sandboxes to confine the operation of untrusted code. Web browsers should also be set so that Java applets are prevented from running automatically and JavaScript should also be utilized sparsely in web applications (Mandiant. 2010). An organization can for instance use web filters or IPS that can easily identify and prevent Java applets from running on a system.

**Test 3: Reverse TCP Shell Exploit**

**Vulnerability Being Tested**

The third test is based on the type of the Reverse TCP Shell, an important method used in the penetration testing to create a channel of communication from the vulnerable computer back to the penetration tester system (Mandiant. 2010). The vulnerability being tested involves such services as payable services or firewall misconfigurations creating a window that enables a connection from an attacker machine to that of the victim’s machine. In this exploit, it becomes easy for the attacker to convince the target machine to connect to the attacker’s machine, which can overcome firewall barriers that limit the allowance of connection from the attacker. This is especially helpful where the targeted system is behind a reserved firewall or NAT (Network Address Translation).

**Configuration of Devices**

The Kali Linux machine was configured as the attacker and the Windows XP virtual machine was to be targeted. The target system had an insecure service being run with the potential of allowing the attacker to use the Reverse TCP Shell method. On the attacker’s system Metasploit was used to generate a reverse shell payload which would connect back to the Kali Linux system.

**Exploit Process**

The attacker first of all got an identification of an open port that he would use to connect to the target machine. To test the second method, the Metasploit module exploit/windows/misc/hta\_reverse\_tcp was used that aims at using an HTTP attack in order to establish the reverse shell. After the payload has been created, the attacker unleashed the exploit and by doing so the victim’s machine dial back to the attacker’s machine to the identified port. On achieving a connection the attacker was provided with a shell on the target machine where he could issue commands.

**Test Results**

The exploit was performed by running the reverse TCP shell payload into the target system through the FTP service. After the target machine executed the payload, it sent back troj connection to the Kali Linux system giving the attacker a remote shell. The attacker got all the privileges of the target, connect the command line, list the files of the targeted machine, and even switch the control of that machine to the privileged mode. The reverse shell create a continuous connection between the two systems a feature that would allow the attacker control the target even in the case of rebooting.

**Mitigations**

Restrict Outbound Connections: Redirecting firewall restrictions to the number of connections initiated from the target system is also effective in decreasing reverse shell attacks. Service Hardening: It is necessary to verify that service executing in the target system is set and updated regularly that should reduce preference of those vulnerabilities. Network Segmentation: Implementing segmentation involves pulling some systems off the network since they are deemed unsafe from reverse shell attacks.

**Test 4: PDF Exploit**

**Vulnerability Being Tested**

In this scenario the PDF exploit that has been launched is a proven one and it attacks the PDF reader which usually is Adobe Acrobat and executes albatross code from a given PDF file on the victim’s machine. More specifically, this weakness is termed CVE-2010-0188; it results from a heap-based buffer-overflow error within Adobe Acrobat Reader software – specifically within the 9.3.0 build and older. The exploit take advantage of this vulnerability where the attack provides a PDF document to the victim through which once opened the program encounters the buffer overflow and the attacker’s pay load including reverse shell or remote code execution is launched. The PDF file is made to look like a regular file so that the user cannot tell it is a threat. As users tend to open attachments in PDF format received by email or download them from the Internet sites this type of attack fits the successful social engineering scheme. Malicious PDF files can be sent via phishing emails, or visitors may get to an infected website and download these files, thereby exploiting a high-risk vulnerability, if the target system has no security patches in place (Microsoft. 2008). This exploit was chosen because it is one of the typical class of attacks in real world cyber compromising. PDF files are used in almost all the business and personal communication and as such are often endangered by attackers. Getting access to one’s machine through a PDF reader gives the attacker control over the system or allow the attacker to get hold of sensitive data.

**Configuration of Devices**

In this test the attacker computer was a Kali Linux, which used Metasploit to launch the attack. The targeted machine was a Win 7 virtual machine with Adobe Acrobat Reader of an older version. The attack was performed in a realistic setting and the attacker created a specific PDF file for the purpose of the test and introduced it to the victim (Adobe, 2010). An important aspect with this configuration was setting connectivity between the Kali Linux machine and Windows 7 system to enable delivery of the exploit when the target is accessed without being hindered by firewalls and intrusion detection systems. For testing the purpose a Windows 7 machine was kept deliberately unpatched to make it vulnerable and outdated.

**Test Results**

The exploitation process started by running a target vulnerability using Metasploit’s integrated scanner for the specific vulnerability in question (CVE-2010-0188). Once the target was considered to be weak, the attacker customized an evil PDF file that was achieved by using the exploit/windows/fileformat/adobe\_pdf\_embedded\_exe which was in Metasploit. This module provides an opportunity to embed a payload in a PDF file.

Further, the unsafe object in this case was the PDF file from the attacker which was transferred to target machine. In this case, the attacker employed the use of social engineering where, for instance, he sent a fake email, a phishing email, that contained the file. After the target user opened the pdf file, the pop up occurred, the buffer overflow happened and the payload was executed to connect back to the attacker’s machine to create a reverse shell. Using the reverse shell, the attacker was fully in charge of the target system (Adobe, 2010). The attacker could now control the output of the commands on system, increase its access level as well as mine sensitive information from the system. This attack successfully exposes the weaknesses within social engineering approaches as well as an all-too-common software flaw.

**Mitigation Recommendations**

1. Patch Management: The only way to protect against such an attack is to ensure that the software is kept current on patches, particularly PDF reader software such as Adobe Acrobat. Bugs like CVE-2010-0188 are normally addressed in updates which mince the time available to an attacker.
2. Use of Sandboxing and Email Filtering: In order to radically minimize the effect of malicious PDF files on a system, the technologies that help to separate PDF readers from the rest of system should be employed. Besides, email filtering systems can be set regarding the capability to reject potentially unsafe file attachments, particularly any type of executables that might be encoded in PDF documents.
3. User Awareness Training: Due to the fact that most PDF exploits involve the aspect of social engineering, it is important for user to be trained on what should be done when they receive a message that contains an unknown attachment. There is a condition that users should be given an understanding on how to check the authenticity of an email and an attachment before opening them.

**Test 5: Aurora Exploit**

**Vulnerability Being Tested**

Aurora exploit was a state-level attack plan that started with taking advantage of a loophole in Internet Explorer to gain access to the company’s networks, including Google as the Aurora attack (Weidman, 2014). This exploit targets a vulnerability in the way that Internet Explorere processes object in memory, the MS10-002 vulnerability (CVE-2009-0672). The exploit is simply a form of a drive by download in which the attacker develops an evil web page, or posts an evil script on an existing website that compels the victim browser to run the exploit as soon as the victim launches the site.

When the victim comes to the malicious or compromised site, the exploit sets up the memory corruption vulnerability in Internet Explorer through which an attacker can run arbitrary code on the victim PC. The payload may encompass anything from planting a virus, Trojan, Worm or Spyware, to copying data or commands to and from the computer system. Aurora exploit is was applied for the attacks on the specific organizations and it was one of the most advanced zero-day attacks.

Due to its practicality the Aurora exploit was chose for testing because it shows how attackers can compromise online browsers and gain access to a system bypassing perimeter security measures including firewalls and the intrusion detection system. As a result, the lesson demonstrates the dangers of having software that has not been updated and the importance of several tiers of protection.

**Configuration of Devices**

This was effectively executed in the attacker’s system which was using the Kali Linux with Metasploit Framework to conduct the tests. The target system was a Windows 7 virtual machine having pre installed Internet Explorer of an older version. The test was performed under controlled conditions, the attacker using the exploit/windows/browser/ms10\_002\_css from Metasploit tool to build a fake web site. This module takes advantage of the memory corruption vulnerability of IE when parsing CSS data.

The attacker’s machine and the targeted machine were in the same local network as a real environment was recreated for this test. Windows 7 target machine was not having any antivirus software or any security patches installed on it to make it look like an open target. In the case described in this article, the attacker set this exploit to change the address that is displayed in the target system’s web browser to the hyperlink’s location.

**Test Results**

The first stage in the attack operations was to establish the version of Internet Explorer installed on the target system. Once the version was identified as one that contains the vulnerability with CVE details number 2009-0672, the attacker launched the exploit from Kali Linux using Metasploit’s ms10\_002\_css module. The exploit was successfully yielded when the target user visits the malicious Web Site.

When the victim navigated to interact on the site, their browser received the malicious CSS data which caused the flaw. The sufferer pointed to their target of establishing that this exploit corrupted the memory target of the system in such a way that let the attacker tell it to do something (Weidman, 2014). The attacker then fired a reverse shell payload which granted him/her access to the target system. The reverse shell the attacker achieved was a complete root access on the victim’s system. The attacker could now perform commands, privilege level, and expand other malware or data exfiltration tools. This test effectively illustrated that a malicious website can use specific browser flaws to carry out a drive-by download assault on systems without involving the end user in any way.

**Mitigation Recommendations**

1. Regular Software Patching: The chief safeguard against the Aurora exploit is making certain that all systems are patched against the latest known vulnerabilities. The vulnerability used in this test has been known as CVE-2009-0672 and the Microsoft has released a patch for this since early 2010. Failure to update such browsers, operating systems and plugins exposes one to exploitation through the available vulnerabilities in their updated versions.
2. Use of Web Filtering and Sandboxing: Web filtering should be implemented by organization for purposes of denying traffic to the known risky sites while sandboxing should be used for containing any dangerous web content from spreading to the rest of the organization’s network.
3. Implementing a Secure Web Browser: Chrome and other default Web browsers have enhanced security that makes them less vulnerable to such exploits, so you could use them to be safe.
4. User Training and Awareness: Particularly, users who should be informed of the risks associated with opening untrusted Web sites and clicking suspicious links would significantly decrease the likelihood of getting infected by malware delivered through the drive-by download practices. Users should also be taught how to report anything peculiar or wrong that they come across when browsing the Web.

Conclusion

In conclusion, penetration testing can be considered as an important step toward ensuring that IT infrastructures are secure and cannot be compromised by intruders. With the help of this project various exploits were checked in a safe environment Meterpreter shell retrieves, Java vulnerabilities used, reverse TCP shell executes, PDF vulnerability used, and Aurora exploit used. In each test different types of attacks prevalent among hackers that allow for unauthorized access, privilege escalation, or persistence were shown. The results stress the respect of timely and proper patching, the security of systems’ configurations, and the efficiency of defense actions against such attack types. Knowing how these exploits work and adjust for it, organizations stand to gain on defending their systems from real world threats. This project has not only developed the understanding of how the penetration testing is conducted but also discovered the necessity of performing security tests constantly due to the existence of new threats and risks.

**References**

Weidman, G. (2014). Penetration testing: A hands-on introduction to hacking. No Starch Press.

Metasploit Project. (n.d.). Metasploit Framework Documentation. Retrieved from <https://www.metasploit.com/docs>

Adobe Systems. (2010). Security bulletin: Adobe Acrobat and Reader (CVE-2010-0188). Retrieved from <https://helpx.adobe.com/security/products/acrobat/apsb10-06.html>

Microsoft. (2008). Microsoft Security Bulletin MS08-067 – Critical. Retrieved from <https://www.microsoft.com/en-us/security/portal/mmpc/shared/2008/MS08-067.aspx>

Mandiant. (2010). APT1: Exposing One of China's Cyber Espionage Units. Retrieved from <https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pd>