
Offensive Security Certified Professional Exam Report

OSCP Exam Report

chase622@gmail.com, OSID: OS-564154

2023-05-04

Contents

1	Offensive Security OSCP Exam Report	1
1.1	Introduction	1
1.2	Objective	1
1.3	Requirements	1
2	High-Level Summary	2
2.1	Recommendations	2
3	Methodologies	3
3.1	Information Gathering	3
3.2	Penetration	3
3.2.1	Domain: oscp.exam	3
3.2.1.1	Active Directory Exploitation Chain	3
3.2.1.2	Service Enumeration	4
3.2.2	192.168.134.100 (DC01) LDAP Enumeration	8
3.2.3	192.168.134.102 (MS02) Initial Access	8
3.2.3.1	192.168.134.102 (MS02) Privilege Escalation	9
3.2.3.2	192.168.134.101 (MS01) Lateral Movement from 192.168.134.101 (MS02)	11
3.2.3.3	Privilege Escalation	11
3.2.3.4	192.168.134.100 (DC01) Lateral Movement from 192.168.134.101 (MS01)	12
3.2.4	Standalone System IP: 192.168.134.110	12
3.2.4.1	Service Enumeration	12
3.2.4.2	Privilege Escalation	16
3.2.5	Standalone System IP: 192.168.134.114	16
3.2.5.1	Service Enumeration	16
3.2.5.2	Privilege Escalation	25
3.2.6	Standalone System IP: 192.168.134.126	25
3.2.6.1	Service Enumeration	25
3.2.6.2	Privilege Escalation	33
3.3	Maintaining Access	34

4 Additional Items	35
4.1 Appendix - Proof and Local Contents:	35

1 Offensive Security OSCP Exam Report

1.1 Introduction

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam.

This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Offensive Security Exam network. The student is tasked with following methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report.

1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards Offensive Security Exam. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems - the OSCP.exam domain. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Offensive Security's network. When performing the attacks, I was able to gain access to all machines, primarily due to poor security configurations. During the testing, I had administrative level access to every system. All systems were successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 192.168.134.100 (DC01) - WinRM with hard-coded Domain Admin credentials in MS01 web application
- 192.168.134.101 (MS01) - Remote Desktop with credentials harvested from MS02
- 192.168.134.102 (MS02) - SSH using credentials discovered during DC01 LDAP enumeration
- 192.168.134.110 - Web application revealed setup scripts with hard-coded MySQL credentials
- 192.168.134.114 - FTP server hosted annual security report noting weak passwords
- 192.168.134.126 - Web application revealed login and password change logs

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

Exam Network

- 192.168.134.100
- 192.168.134.101
- 192.168.134.102
- 192.168.134.110
- 192.168.134.114
- 192.168.134.126

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **6** out of the **6** systems.

3.2.1 Domain: oscp.exam

3.2.1.1 Active Directory Exploitation Chain

In the oscp.exam domain, exploitation was made possible by enumeration of users and a default password from the Domain Controller DC01. Those credentials were used to gain low-privilege ac-

cess to MS02 via SSH, and privilege escalation to SYSTEM was made possible through weak service binary permissions. Credential harvesting on MS02 allowed for lateral movement to MS01, which contained plaintext credentials of a Domain Administrator account. This enabled pivoting to the Domain Controller to gain full control over the oscp.exam domain.

The next sections regarding the oscp.exam domain will first cover enumeration of all services, then run through the exploitation process in chronological order starting with LDAP enumeration on DC01 and ending with lateral movement from MS01 to DC01.

3.2.1.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Name	Ports Open
192.168.134.100	DC01	TCP: 53,88,135,139,389,445,464,593,636,3268,3269,5985

Nmap Scan Results:

Full port scan:

53/tcp	open	domain
88/tcp	open	kerberos-sec
135/tcp	open	msrpc
139/tcp	open	netbios-ssn
389/tcp	open	ldap
445/tcp	open	microsoft-ds
464/tcp	open	kpasswd5
593/tcp	open	http-rpc-epmap
636/tcp	open	ldaps
3268/tcp	open	globalcatLDAP
3269/tcp	open	globalcatLDAPssl
5985/tcp	open	wsman
9389/tcp	open	adws
49667/tcp	open	unknown

```
49673/tcp open  unknown
49674/tcp open  unknown
49676/tcp open  unknown
49691/tcp open  unknown
49770/tcp open  unknown
```

Script port scan:

```
53/tcp  open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_   bind
88/tcp  open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-
05-03 19:56:45Z)
135/tcp  open  msrpc          Microsoft Windows RPC
139/tcp  open  netbios-ssn    Microsoft Windows netbios-ssn
389/tcp  open  ldap           Microsoft Windows Active Directory LDAP (Domain: os
First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http     Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap           Microsoft Windows Active Directory LDAP (Domain: os
First-Site-Name)
3269/tcp open  tcpwrapped
5985/tcp open  http           Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp open  mc-nmf         .NET Message Framing
1 service unrecognized despite returning data. If you know the service/version,
bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.80%I=7%D=5/3%Time=6452BC81%P=x86_64-pc-linux-
gnu%r(DNSVe
SF:rsionBindReqTCP,20,","\x1e\x06\x81\x04\x01\x00\x00\x00\x07version\x
SF:04bind\x00\x10\x03");
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows
```


Host script results:

```
| smb2-security-mode:
|   2.02:
|_   Message signing enabled and required
| smb2-time:
|   date: 2023-05-03T19:59:01
|_   start_date: N/A
```

Server IP Address	Name	Ports Open
192.168.101.101	MS01	TCP: 445,3389,8080

Nmap Scan Results:

Full port scan:

```
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
8080/tcp open  http-proxy
```

Script port scan:

```
445/tcp  open  microsoft-ds?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: OSCP
|   NetBIOS_Domain_Name: OSCP
|   NetBIOS_Computer_Name: MS01
|   DNS_Domain_Name: oscp.exam
|   DNS_Computer_Name: ms01.oscp.exam
|   DNS_Tree_Name: oscp.exam
|   Product_Version: 10.0.17763
|_  System_Time: 2023-05-03T19:34:18+00:00
| ssl-cert: Subject: commonName=ms01.oscp.exam
| Not valid before: 2023-03-15T15:54:26
|_Not valid after:  2023-09-14T15:54:26
|_ssl-date: 2023-05-03T19:34:58+00:00; +1s from scanner time.
```

```
8080/tcp open  http           Microsoft IIS httpd 10.0
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Loading...
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Host script results:

```
| smb2-security-mode:
|   2.02:
|_   Message signing enabled but not required
| smb2-time:
|   date: 2023-05-03T19:34:21
|_   start_date: N/A
```

Passcore Web Application on Port 8080 [[Pasted image 20230504004018.png]]

Server IP Address	Name	Ports Open
192.168.134.102	MS02	TCP: 22,5040,7680

Nmap Scan Results:

Full port scan:

```
22/tcp    open  ssh
5040/tcp  open  unknown
7680/tcp  open  pando-pub
```

Script port scan:

```
22/tcp    open  ssh           OpenSSH for_Windows_8.1 (protocol 2.0)
|_ ssh-hostkey:
|   3072 22:dd:9e:24:c4:57:35:e7:91:47:ba:ae:ba:e2:d0:39 (RSA)
|   256 99:8f:59:d8:5c:7e:e0:60:03:b7:27:b2:54:77:04:b5 (ECDSA)
|_  256 67:70:80:14:41:ed:60:c9:02:f0:5b:93:99:58:97:2d (ED25519)
5040/tcp  open  unknown
7680/tcp  open  pando-pub?
```

3.2.2 192.168.134.100 (DC01) LDAP Enumeration

Vulnerability Exploited: LDAP is not properly configured on Domain Controller DC01, which allows for enumeration of objects in the domain. Using the `ldapsearch` command in Linux, we can discover many user accounts and a default password that we can use to spray the other machines with.

Vulnerability Explanation: The command to use to retrieve information is in the format of `ldapsearch -x -H ldap://<IP> -D '' -w '' -b "DC=<1_SUBDOMAIN>,DC=<TLD>".` The SUBDOMAIN and TLD fields correspond to the OSCP and exam fields of the OSCP.exam domain, respectively, and can be confirmed with the `ldapsearch -x -H ldap://<IP> -s base namingcontexts` command. The commands and their outputs should look like:

[[Pasted image 20230504001822.png]]

Below is a continuation of the `ldapsearch -x -H ldap://<IP> -D '' -w '' -b "DC=<1_SUBDOMAIN>,DC=<TLD>"` command output, focused in on one of the users. Here we can see many attributes of the Kevyn.Turk user, as well as what the default password was:

[[Pasted image 20230504001946.png]]

Compiling all the enumerated usernames into `users.txt` and the default password into `pass.txt`, these lists can be used to password spray other machines' protocols - such as SSH on MS02, where initial access to the domain was gained.

[[Pasted image 20230504001920.png]]

Vulnerability Fix: LDAP needs to be re-configured so that it hides sensitive information like user accounts and default passwords. This can be done by only allowing authenticated users to access such information, instead of all users who supply the correct TLD and subdomains.

Severity: This is quite a high level of severity. Although there is no loss of integrity or availability, anyone can probe the LDAP service for information and cause a potentially high amount of confidentiality loss. It reveals information about user accounts on the domain while highlighting bad security practices, such as keeping the default password configurations.

3.2.3 192.168.134.102 (MS02) Initial Access

Vulnerability Exploited: The previous section detailed domain object enumeration using `ldapsearch` on the Domain Controller. The enumerated user accounts and password can be tested on other systems in the domain.

Vulnerability Explanation: Armed with information gathered from DC01 - the `users.txt` user wordlist and the password stored in `pass.txt` - we can use `hydra` to conduct a password spray attack on the SSH protocol:

[[Pasted image 20230504002058.png]]

This reveals that a valid domain credential is `Ketty.Agan:ESMWaterP1p35!`, and that they have not changed their default password. We can gain a low privilege shell on MS02 by `ssh ketty.agan@192.168.134.102` and supplying their password.

Local.txt Proof: [[Pasted image 20230504002155.png]] Flag: `edd330f064ef3da6df495095a04e141f`

Vulnerability Fix: Securing LDAP configuration would have solved the root issue here. Remote access solutions can be understandably difficult to securely configure, based on the organization's remote work policies. However, SSH could have been configured more securely as to not let people outside the organization connect, whether through limiting IP ranges or using VPN connections. Finally, the `Ketty.Agan` user should have changed their default password to begin with.

Severity: This has a high severity rating because of the ease in which the information was enumerated and tested.

3.2.3.1 192.168.134.102 (MS02) Privilege Escalation

The `Ketty.Agan` user does not have very many privileges. `whoami /priv` reveals that we have basic privileges to remotely shut down the machine. We need to find ways to escalate.

To transfer tools and exploits over, we can use `impacket's smbserver.py` script to set up a local SMB share that other machines can connect to. We can enable SMB2 support to be safe and toss in authentication, with credentials being `chacei:asdf`:

[[Pasted image 20230504002532.png]]

With that, one of the first scripts we can run is `PrivescCheck.ps1`, from <https://github.com/itm4n/PrivescCheck>. It manages to find something very interesting for us:

[[Pasted image 20230504002547.png]]

[[Pasted image 20230504002623.png]]

Of note is that `PipesPrinting` service. The binary appears to be modifiable by us, even with our low privileges, due to its even lower permissions. However, we do not have the authority to start or stop the service or access the service control manager.

Vulnerability Exploited: Our goal here is to have a reverse shell payload impersonate `PipesPrinting.exe` and have it run by `SYSTEM` when it executes the service. Since it runs under `SYSTEM` context, we should have a `SYSTEM` shell callback. Since we cannot stop or start the service itself, we need to use our shutdown privileges to restart the machine and restart the service, where it will begin running again by executing the renamed reverse shell payload.

Vulnerability Explanation: First, we use msfvenom on our local machine to generate a reverse shell executable and name it to PipesPrinting.exe, as that is what is set for the services executable. We set up netcat listener on port 443 with `sudo nc -nlvp 443`:

[[Pasted image 20230504003002.png]]

Transferring over the reverse shell payload, we rename the original PipesPrinting.exe to PipesPrintingOLD.exe using the `move PipesPrinting.exe PipesPrintingOLD.exe` command. We then insert our own payload in the same directory and restart the computer.

[[Pasted image 20230504003129.png]]

After the 60 or so seconds where the computer is restarting, we should eventually get a callback on our listener once the service restarts.

Proof.txt Contents: [[Pasted image 20230504003333.png]] Flag: 4a96bdaf1d8a514623671f8b1df7ef0b

Vulnerability Fix: To fix this, the binary permissions for the PipesPrintingService need to be tightened up. Only authorized users and groups should be able to modify the permissions and the binaries.

Severity: This has a high severity rating because of the ease at which low-privileged users may elevate to SYSTEM privileges.

Post Exploitation: We can transfer over the SharpHound.exe data collector for Bloodhound from <https://github.com/BloodHoundAD/BloodHound> to map out the domain. We find that the passcore user is a Domain Administrator; this is likely our final target. The name also matches up with the Passcore web application hosted on MS01 port 8080, so it's likely we will find something in there.

[[Pasted image 20230504210704.png]]

With SYSTEM privileges, we can access cached credentials with mimikatz.exe and crack them. By running `sekurlsa::logonpasswords`, we find a particular user with stored creds on the machine - Liv.Ungley.

[[Pasted image 20230504003459.png]]

This NTLM hash is easily cracked using the hashcat hash cracking tool from <https://github.com/hashcat/hashcat>. `-m1000` denotes NTLM mode, `hashes.txt` stores the password hashes retrieved from mimikatz.exe, and `rockyou.txt` is the wordlist used to crack the hashes.

[[Pasted image 20230504003652.png]]

Liv.Ungley:RockYou! are the user credentials that may be used to access MS01.

3.2.3.2 192.168.134.101 (MS01) Lateral Movement from 192.168.134.101 (MS02)

To recap, access to MS01 was achieved through lateral movement from MS02 using Remote Desktop after harvesting Liv.Ungley user credentials.

Vulnerability Explanation: The method of access in MS01 was by harvesting and cracking Liv.Ungley's insecure password - RockYou! - after elevating to SYSTEM privileges on MS02. These were used with the xfreerdp tool to connect to Remote Desktop on port 3389 to gain a low-level shell on MS01.

Local.txt Contents: [[Pasted image 20230504003846.png]] Flag: 84cff16d270fab734043cf27ab014109

Vulnerability Fix: The Liv.Ungley user could have made their password more secure to begin with, but that would not have prevented a pass-the-hash login using the xfreerdp tool. Limiting the IP ranges allowed to connect with the machine would have been more effective at blocking this specific mode of initial access, or cutting down the users in the Remote Desktop Users group - this may have been unfeasible for this specific user account, as they were in the IT user group.

Severity: This rates at an overall medium to high severity score due to the full trust provided to the user as long as they had proper credentials, regardless of IP.

3.2.3.3 Privilege Escalation

While the security regarding insecure services, file paths, tokens, and such were exceptional on this machine, the permissions regarding the configuration files of the Passcore web application were not.

Vulnerability Exploited: The Passcore web application's directory was open and readable to all domain users. This included the appsettings.json file, which contained hard-coded LDAP credentials for the passcore Domain Administrator. These could then be used to gain complete control over the domain and authenticate to the Domain Controller over WinRM.

Vulnerability Explanation: We can see that the passcore directory is in the root directory of MS01, containing the code to run the Passcore web application below:

[[Pasted image 20230504004018.png]]

We are able to view the contents of the passcore directory as our current user Liv.Ungley:

[[Pasted image 20230504004058.png]]

The appsettings.json file is of particular note, and if we open it up then we can see that it is a configuration file of sorts for the passcore application.

[[Pasted image 20230504004225.png]]

And scrolling down, we can see the `passcore` user - the Domain Administrator - with their hard-coded credentials `passcore:G3x56wGq9fItu166` in the file.

[[Pasted image 20230504004254.png]]

Vulnerability Fix: Placing the plaintext credentials of the Domain Administrator in a file with such little permissions is not optimal for security. To start, enforcing file permissions on the configuration files would help tremendously - the rest of the machine is very well protected against privilege escalation vectors. Next, creating another user with lower permissions specifically for the LDAP communications would also help in the event that a similar event occurs.

Severity: This has an extremely high severity rating for revealing the Domain Administrators credentials in such a manner.

3.2.3.4 192.168.134.100 (DC01) Lateral Movement from 192.168.134.101 (MS01)

Vulnerability Exploited: Access to the Domain Controller is locked down tightly, unless one has a Domain Administrator credentials. It is a simple matter of using `evil-winrm` from <https://github.com/Hackplayers/evil-winrm> to gain Domain Administrator access to the Domain Controller with the `passcore` user's credentials.

Vulnerability Explanation: Although the initial LDAP misconfiguration is the root issue, WinRM could be configured to only allow connections from a certain IP range so as to not allow outside connections to be able to gain access.

Proof.txt Contents: [[Pasted image 20230504004643.png]] Flag: `d513b963ae8d75d5dbddeb04933d1a49`

Vulnerability Fix: IP ranges should certainly be limited in regards to the Domain Controller, either by limiting inbound connections through firewalls or implementing Access Control Lists.

Severity: This is high severity - the Domain Controller should certainly have more restrictions on IP ranges that could connect to it.

3.2.4 Standalone System IP: 192.168.134.110

3.2.4.1 Service Enumeration

Server IP Address	Ports Open
192.168.134.110	TCP: 21,22,80,3306,8080

Nmap Scan Results:

Full port scan:

```
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
3306/tcp  open  mysql
8080/tcp  open  http-proxy
```

Targeted port scan:

```
21/tcp    open  ftp      vsftpd 3.0.5
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))
|_http-generator: Nicepage 4.12.21, nicepage.com
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Home
3306/tcp  open  mysql    MySQL 5.7.38
| mysql-info:
|   Protocol: 10
|   Version: 5.7.38
|   Thread ID: 5
|   Capabilities flags: 65535
|   Some Capabilities: SupportsTransactions, InteractiveClient, Speaks41Protocol
|   Status: Autocommit
|   Salt: \x10t\x0FNE/h\x08aTAo^b\x01eVg\x0D\x01
|_ Auth Plugin Name: mysql_native_password
8080/tcp  open  http     Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

MySQL Script Scan:

```
3306/tcp  open  mysql    MySQL 5.7.38
| mysql-enum:
|   Valid usernames:
|     root:<empty> - Valid credentials
|     user:<empty> - Valid credentials
```



```
| netadmin:<empty> - Valid credentials
| guest:<empty> - Valid credentials
| web:<empty> - Valid credentials
| sysadmin:<empty> - Valid credentials
| administrator:<empty> - Valid credentials
| webadmin:<empty> - Valid credentials
| admin:<empty> - Valid credentials
| test:<empty> - Valid credentials
|_ Statistics: Performed 10 guesses in 1 seconds, average tps: 10.0
| mysql-info:
| Protocol: 10
| Version: 5.7.38
| Thread ID: 9
| Capabilities flags: 65535
| Some Capabilities: Support41Auth, DontAllowDatabaseTableColumn, InteractiveC
| Status: Autocommit
| Salt: \x03o?K\x01chq0y\x1E._?([UN?j
|_ Auth Plugin Name: mysql_native_password
```

Nikto Scan Results:

```
+ Server: Apache/2.4.52 (Ubuntu)
+ Server leaks inodes via ETags, header found with file /, fields: 0x10caa 0x5e1
+ The anti-clickjacking X-Frame-Options header is not present.
+ OSVDB-3268: /scripts/: Directory indexing found.
+ IP address found in the 'location' header. The IP is "127.0.1.1".
+ OSVDB-630: IIS may reveal its internal or real IP in the Location header via a
+ Allowed HTTP Methods: OPTIONS, HEAD, GET, POST
+ OSVDB-3092: /scripts/: This might be interesting... possibly a system shell fo
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3268: /images/?pattern=/etc/*&sort=name: Directory indexing found.
```

Gobuster Scan Results:

```
/images (Status: 301)
/blog (Status: 301)
/scripts (Status: 301)
/server-status (Status: 403)
```

Vulnerability Exploited: The webpage hosted on this machine has insecure permissions on the /scripts directory, where there are leftover setup scripts that have plaintext MySQL database credentials. Inside, there are user accounts with hashed credentials that may be cracked for initial access.

Vulnerability Explanation: In the initial scanning phase, a gobuster scan of the web page revealed a few directories:

[[Pasted image 20230503130934.png]]

The /scripts directory is enticing, and we can see things that we are definitely not supposed to be able to see as public users.

[[Pasted image 20230503131008.png]]

Most of these scripts are junk, but the very bottom wiki_setup.sh contains some database access creds:

[[Pasted image 20230503131027.png]]

Using the channel:Shinj16510 credentials and connecting remotely to the open 3306 MySQL port, we are able to authenticate as channel.

[[Pasted image 20230503131136.png]]

The mysql database generally contains interesting rows in the user table, so we can try getting information from there.

[[Pasted image 20230504005102.png]]

[[Pasted image 20230504005116.png]]

[[Pasted image 20230503131233.png]]

And indeed, we get 5 different hashes for users that we can try to crack. Dumping them into passhash.txt and running them through the hashcat tool under the -m300 mode for SQL4.1+ hashes, we are able to recover a password for the cristine user.

[[Pasted image 20230504005213.png]]

[[Pasted image 20230503131412.png]]

With the cristine:2ql4sql credentials, we can authenticate to SSH and grab the user flag.

Local.txt Contents [[Pasted image 20230503131534.png]] Flag: d700e9f4a877412386b37fbf9952b2f3

Vulnerability Fix: Clearing out the /scripts directory or making it un-readable to the public would secure an otherwise very secure machine. Limiting the MySQL database to connections only from localhost would also be a good fix, since there is usually no need for outside users to authenticate to the database. Last, cristine needs a less common password.

Severity: This is high severity due to the ease of access that attackers have to the open MySQL database and user credentials.

3.2.4.2 Privilege Escalation

One of the first checks run on machines is the command `sudo -l`, which shows what commands the current user may run as sudo. In this case, there was one command that created a critical privilege escalation vector.

Vulnerability Exploited: The `cristine` user is able to run `exiftool` as `sudo`, which can be abused to start a root shell.

Vulnerability Explanation: Running `sudo -l` gives the output:

[[Pasted image 20230503131627.png]]

According to <https://gtfobins.github.io/#>, the `exiftool` command may be used to access the file system, escalate or maintain privileged access. I followed the writeup from <https://exploit-notes.hdks.org/exploit/linux/privilege-escalation/sudo/sudo-exiftool-privilege-escalation/> that detailed how to create an exploit that would pop a root shell.

[[Pasted image 20230503131903.png]]

Proof.txt Contents [[Pasted image 20230503131930.png]] Flag: `c7f1cbc6e8de951f34f4e4ea2eda7d2d`

Vulnerability Fix: Not allowing users to run `exiftool` as `sudo` would remove this specific privilege escalation vector.

Severity: This is high severity, low-privilege users are easily able to get root access with tools already available on the machine, such as the `bzz` and `djvumake` commands.

3.2.5 Standalone System IP: 192.168.134.114

3.2.5.1 Service Enumeration

Server IP Address	Ports Open
192.168.134.114	TCP: 21,22,80,139,445

Nmap Scan Results:

Full port scan:

```
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
139/tcp open netbios-ssn
445/tcp open  microsoft-ds
```

Targeted port scan:

```
21/tcp open  ftp          vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--    1 0          0          3557581 Nov 25  2021 2d5ef5a0f0c9579458c9
| -rw-r--r--    1 0          0          1258508 Nov 25  2021 4835e976619690ae006e
| -rw-r--r--    1 0          0          1617905 Nov 25  2021 4e8cce46d6abec9a9d9a
| -rw-r--r--    1 0          0          438095  Nov 25  2021 77cfe070405f6ca327a5
|_-rw-r--r--    1 0          0          841392  Nov 25  2021 c5237630ef40e2585d35
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to ::ffff:192.168.49.134
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 3
|   vsFTPD 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2)
80/tcp open  http          Apache httpd 2.4.41 ((Ubuntu))
|_http-generator: WordPress 6.0.2
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: The Stationery Warehouse &#8211; Just another WordPress site
|_http-trane-info: Problem with XML parsing of /evox/about
|_https-redirect: ERROR: Script execution failed (use -d to debug)
139/tcp open  netbios-ssn Samba smbd 4.6.2
```

445/tcp open netbios-ssn Samba smbd 4.6.2

Service Info: Host: the; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:

```
|_nbstat: NetBIOS name: OSCP, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (u
| smb2-security-mode:
| 2.02:
|_ Message signing enabled but not required
| smb2-time:
| date: 2023-05-03T11:22:24
|_ start_date: N/A
```

Nikto Scan Results:

```
+ Server: Apache/2.4.41 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ Uncommon header 'link' found, with contents: <http://192.168.134.114/>; rel=sh
+ Uncommon header 'x-redirect-by' found, with contents: WordPress
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ File/dir '/wp-admin/' in robots.txt returned a non-forbidden or redirect HTTP
+ Uncommon header 'x-robots-tag' found, with contents: noindex
+ "robots.txt" contains 2 entries which should be manually viewed.
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microso
us/library/e8z01xdh%28VS.80%29.aspx for details.
+ Server leaks inodes via ETags, header found with file /readme.html, fields: 0x
+ OSVDB-3092: /license.txt: License file found may identify site software.
+ /wp-app.log: Wordpress' wp-app.log may leak application/system details.
+ /wordpress/: A Wordpress installation was found.
+ Uncommon header 'x-frame-options' found, with contents: SAMEORIGIN
+ Cookie wordpress_test_cookie created without the httponly flag
```

enum4linux Scan Results:

=====(Target Information)=====

```
Target ..... 192.168.134.114
RID Range ..... 500-550,1000-1050
Username ..... ''
```

Password ''

Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

=====(Enumerating Workgroup/Domain on 192.168.134.114)==

[+] Got domain/workgroup name: WORKGROUP

=====(Nbtstat Information for 192.168.134.114)=====

Looking up status of 192.168.134.114

OSCP	<00>	-	B	<ACTIVE>	Workstation Service
OSCP	<03>	-	B	<ACTIVE>	Messenger Service
OSCP	<20>	-	B	<ACTIVE>	File Server Service
..__MSBROWSE__.	<01>	-	<GROUP>	B	<ACTIVE> Master Browser
WORKGROUP	<00>	-	<GROUP>	B	<ACTIVE> Domain/Workgroup Name
WORKGROUP	<1d>	-	B	<ACTIVE>	Master Browser
WORKGROUP	<1e>	-	<GROUP>	B	<ACTIVE> Browser Service Elections

MAC Address = 00-00-00-00-00-00

=====(Session Check on 192.168.134.114)=====

[+] Server 192.168.134.114 allows sessions using username '', password ''

=====(Getting domain SID for 192.168.134.114)=====

Domain Name: WORKGROUP

Domain Sid: (NULL SID)

[+] Can't determine if host is part of domain or part of a workgroup

=====(OS information on 192.168.134.114)=====

[E] Can't get OS info with smbclient

[+] Got OS info for 192.168.134.114 from srvinfo:

```
OSCP          Wk Sv PrQ Unx NT SNT Samba 4.13.14-Ubuntu
platform_id   :    500
os version    :    6.1
server type   :    0x809a03
```

======(Users on 192.168.134.114)=====

Use of uninitialized value \$users in print at ./enum4linux.pl line 978.

Use of uninitialized value \$users in pattern match (m//) at ./enum4linux.pl line

Use of uninitialized value \$users in print at ./enum4linux.pl line 996.

Use of uninitialized value \$users in pattern match (m//) at ./enum4linux.pl line

======(Share Enumeration on 192.168.134.114)=====

smbXcli_negprot_smb1_done: No compatible protocol selected by server.

Sharename	Type	Comment
-----	----	-----
Developer	Disk	Developer Files
IPC\$	IPC	IPC Service (Samba 4.13.14-Ubuntu)

Reconnecting with SMB1 for workgroup listing.

protocol negotiation failed: NT_STATUS_INVALID_NETWORK_RESPONSE

Unable to connect with SMB1 -- no workgroup available

[+] Attempting to map shares on 192.168.134.114

//192.168.134.114/Developer Mapping: OK Listing: OK Writing: N/A

[E] Can't understand response:

NT_STATUS_OBJECT_NAME_NOT_FOUND listing *

//192.168.134.114/IPC\$ Mapping: N/A Listing: N/A Writing: N/A

=====(Password Policy Information for 192.168.134.114)==

[+] Attaching to 192.168.134.114 using a NULL share

[+] Trying protocol 139/SMB...

[+] Found domain(s):

[+] OSCP

[+] Builtin

[+] Password Info for Domain: OSCP

[+] Minimum password length: 5

[+] Password history length: None

[+] Maximum password age: 37 days 6 hours 21 minutes

[+] Password Complexity Flags: 000000

[+] Domain Refuse Password Change: 0

[+] Domain Password Store Cleartext: 0

[+] Domain Password Lockout Admins: 0

[+] Domain Password No Clear Change: 0

[+] Domain Password No Anon Change: 0

[+] Domain Password Complex: 0

[+] Minimum password age: None

[+] Reset Account Lockout Counter: 30 minutes

[+] Locked Account Duration: 30 minutes

[+] Account Lockout Threshold: None

[+] Forced Log off Time: 37 days 6 hours 21 minutes

[+] Retrieved partial password policy with rpcclient:

Password Complexity: Disabled

Minimum Password Length: 5

=====(Groups on 192.168.134.114)=====

[+] Getting builtin groups:

[+] Getting builtin group memberships:

[+] Getting local groups:

[+] Getting local group memberships:

[+] Getting domain groups:

[+] Getting domain group memberships:

=====(Users on 192.168.134.114 via RID cycling (RIDS: 500-550,1000-1050))=====

[I] Found new SID:

S-1-22-1

[I] Found new SID:

S-1-5-32

[I] Found new SID:
S-1-5-32

[I] Found new SID:
S-1-5-32

[I] Found new SID:
S-1-5-32

[+] Enumerating users using SID S-1-5-32 and logon username '', password ''

S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)

[+] Enumerating users using SID S-1-22-1 and logon username '', password ''

S-1-22-1-1000 Unix User\sarah (Local User)
S-1-22-1-1001 Unix User\nick (Local User)
S-1-22-1-1002 Unix User\paul (Local User)
S-1-22-1-1003 Unix User\linda (Local User)
S-1-22-1-1004 Unix User\joe (Local User)

[+] Enumerating users using SID S-1-5-21-2501119518-2288645244-739936310 and logon username '', password ''

S-1-5-21-2501119518-2288645244-739936310-501 OSCP\nobody (Local User)
S-1-5-21-2501119518-2288645244-739936310-513 OSCP\None (Domain Group)

=====(Getting printer info for 192.168.134.114)=====

No printers returned.

Vulnerability Exploited: The anonymous login ability on FTP enabled us to pull 5 PDF documents

off the FTP server. One of these was an Annual Report on the security posture of the organization, detailing several commonly used and insecure passwords. Combined with the users enumerated from enum4linux and the knowledge that there is no password policy, we can conduct a dictionary attack on the machine's authentication protocols and gain access.

Vulnerability Explanation: The initial nmap scans showed that this machine had anonymous FTP logins enabled.

[[Pasted image 20230503085506.png]]

Logging in, we see 5 files and can pull them off to see what they are.

[[Pasted image 20230504005412.png]]

[[Pasted image 20230504005442.png]]

Upon further examination, these are all PDFs. One of these stands out from the rest:

[[Pasted image 20230503085616.png]]

A security audit's annual findings is among the other PDFs. Inside, we can see that they have audited passwords as well, and have noted the weaker and more common passwords for us.

[[Pasted image 20230503085704.png]]

Back in the initial scan, port 139 and 445 were open, which prompted an enum4linux scan of the host. In here, we discover that it is poorly configured, that there is no password lockout threshold, and that certain user accounts exist.

[[Pasted image 20230503085850.png]]

[[Pasted image 20230503085911.png]]

[[Pasted image 20230503085925.png]]

We have users, we have possible passwords, and we know there is no password policy. We are free to conduct dictionary attacks using hydra on FTP or SSH - FTP was chosen first.

[[Pasted image 20230504005544.png]]

[[Pasted image 20230503090111.png]]

Hydra returns the credentials sarah : ! Password-Reset0000, which enables us to authenticate to SSH as the sarah user.

Local.txt Contents: [[Pasted image 20230503090210.png]] Flag: 6b4a79fa6bf204b8ca285a808603a6fb

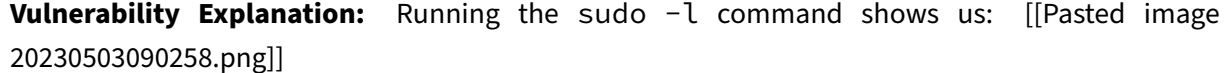
Vulnerability Fix: There are a couple of things to patch these up. First, removing anonymous FTP access, or at the very least removing the Annual Report, is essential. Next, SMB needs to be properly configured so that user accounts and other details, like password policy, are not leaked in scans. Last,

the organization must enforce password complexity - especially after the sarah user was one who was audited for their password being too weak.

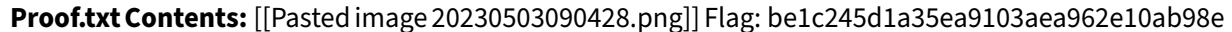
Severity: This is high severity due to the amount of information that is being leaked here and the lack of security measures.

3.2.5.2 Privilege Escalation

Vulnerability Exploited: The sarah user is able to run mawk as sudo, which can be used to gain a root shell according to <https://gtfobins.github.io/>.

Vulnerability Explanation: Running the `sudo -l` command shows us: 

We are able to run mawk as sudo. Using the above gtfobins link as a reference, it is trivial to start a root shell with one command.

Proof.txt Contents:  Flag: be1c245d1a35ea9103aea962e10ab98e

Vulnerability Fix: Removing the ability to run mawk as sudo would immediately remove this privilege escalation vector.

Severity: This is high severity for the trivial amount of work required to gain so much privilege.

3.2.6 Standalone System IP: 192.168.134.126

3.2.6.1 Service Enumeration

Server IP Address	Ports Open
192.168.134.126	TCP: 21,22,80,139,445

Nmap Scan Results:

Full port scan:

```
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
139/tcp open netbios-ssn
445/tcp open microsoft-ds
```

Targeted port scan:

```
21/tcp open  ftp          vsftpd 3.0.5
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--    1 113      121          65885 Sep 05 2022 backup1.zip
| _-rw-r--r--    1 113      121          40689 Sep 05 2022 backup2.zip
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to ::ffff:192.168.49.134
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 4
|   vsFTPD 3.0.5 - secure, fast, stable
|_End of status
22/tcp open  ssh          OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http        Apache httpd 2.4.52 ((Ubuntu))
|_http-generator: Nicepage 4.17.10, nicepage.com
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Home
139/tcp open  netbios-ssn Samba smbd 4.6.2
445/tcp open  netbios-ssn Samba smbd 4.6.2
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Nikto Scan Results:

- + Server: Apache/2.4.52 (Ubuntu)
- + Server leaks inodes via ETags, header found with file /, fields: 0xbba1 0x5ed6
- + The anti-clickjacking X-Frame-Options header is not present.
- + No CGI Directories found (use '-C all' to force check all possible dirs)
- + IP address found in the 'location' header. The IP is "127.0.1.1".
- + OSVDB-630: IIS may reveal its internal or real IP in the Location header via a
- + Allowed HTTP Methods: OPTIONS, HEAD, GET, POST
- + OSVDB-3092: /log.txt: This might be interesting...

- + OSVDB-3268: /images/: Directory indexing found.
- + OSVDB-3268: /images/?pattern=/etc/*&sort=name: Directory indexing found.

gobuster Scan Results:

```
/images (Status: 301)
/server-status (Status: 403)
```

enum4linux Scan Results:

```
=====( Target Information )=====

Target ..... 192.168.134.126
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

=====( Enumerating Workgroup/Domain on 192.168.134.126 )=====

[+] Got domain/workgroup name: WORKGROUP

=====( Nbtstat Information for 192.168.134.126 )=====

Looking up status of 192.168.134.126
  OSCP          <00> -          B <ACTIVE>  Workstation Service
  OSCP          <03> -          B <ACTIVE>  Messenger Service
  OSCP          <20> -          B <ACTIVE>  File Server Service
  ..__MSBROWSE__.. <01> - <GROUP> B <ACTIVE>  Master Browser
  WORKGROUP     <00> - <GROUP> B <ACTIVE>  Domain/Workgroup Name
  WORKGROUP     <1d> -          B <ACTIVE>  Master Browser
  WORKGROUP     <1e> - <GROUP> B <ACTIVE>  Browser Service Elections

  MAC Address = 00-00-00-00-00-00
```

=====(Session Check on 192.168.134.126)=====

[+] Server 192.168.134.126 allows sessions using username '', password ''

=====(Getting domain SID for 192.168.134.126)=====

Domain Name: WORKGROUP

Domain Sid: (NULL SID)

[+] Can't determine if host is part of domain or part of a workgroup

=====(OS information on 192.168.134.126)=====

[E] Can't get OS info with smbclient

[+] Got OS info for 192.168.134.126 from srvinfo:

OSCP	Wk Sv PrQ Unx NT SNT oscp server (Samba, Ubuntu)
platform_id	: 500
os version	: 6.1
server type	: 0x809a03

=====(Users on 192.168.134.126)=====

Use of uninitialized value \$users in print at ./enum4linux.pl line 978.

Use of uninitialized value \$users in pattern match (m//) at ./enum4linux.pl line

Use of uninitialized value \$users in print at ./enum4linux.pl line 996.

Use of uninitialized value \$users in pattern match (m//) at ./enum4linux.pl line

=====(Share Enumeration on 192.168.134.126)=====

smbXcli_negprot_smb1_done: No compatible protocol selected by server.

Sharename	Type	Comment
-----	----	-----
print\$	Disk	Printer Drivers
IPC\$	IPC	IPC Service (oscp server (Samba, Ubuntu))

Reconnecting with SMB1 for workgroup listing.

protocol negotiation failed: NT_STATUS_INVALID_NETWORK_RESPONSE

Unable to connect with SMB1 -- no workgroup available

[+] Attempting to map shares on 192.168.134.126

//192.168.134.126/print\$ Mapping: DENIED Listing: N/A Writing: N/A

[E] Can't understand response:

NT_STATUS_OBJECT_NAME_NOT_FOUND listing *

//192.168.134.126/IPC\$ Mapping: N/A Listing: N/A Writing: N/A

=====(Password Policy Information for 192.168.134.126)==

[+] Attaching to 192.168.134.126 using a NULL share

[+] Trying protocol 139/SMB...

[+] Found domain(s):

[+] OSCP

[+] Builtin

[+] Password Info for Domain: OSCP

[+] Minimum password length: 5

[+] Password history length: None

[+] Maximum password age: 37 days 6 hours 21 minutes

[+] Password Complexity Flags: 000000

[+] Domain Refuse Password Change: 0
[+] Domain Password Store Cleartext: 0
[+] Domain Password Lockout Admins: 0
[+] Domain Password No Clear Change: 0
[+] Domain Password No Anon Change: 0
[+] Domain Password Complex: 0

[+] Minimum password age: None
[+] Reset Account Lockout Counter: 30 minutes
[+] Locked Account Duration: 30 minutes
[+] Account Lockout Threshold: None
[+] Forced Log off Time: 37 days 6 hours 21 minutes

[+] Retrieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 5

======(Groups on 192.168.134.126)=====

[+] Getting builtin groups:

[+] Getting builtin group memberships:

[+] Getting local groups:

[+] Getting local group memberships:

[+] Getting domain groups:

[+] Getting domain group memberships:

```
===== ( Users on 192.168.134.126 via RID cycling (RIDS: 500-550,1000-1050) )=====
```

[I] Found new SID:
S-1-22-1

[I] Found new SID:
S-1-5-32

[I] Found new SID:
S-1-5-32

[I] Found new SID:
S-1-5-32

[I] Found new SID:
S-1-5-32

[+] Enumerating users using SID S-1-22-1 and logon username '', password ''

```
S-1-22-1-1000 Unix User\rowan (Local User)
S-1-22-1-1010 Unix User\douglas (Local User)
S-1-22-1-1011 Unix User\thomas (Local User)
S-1-22-1-1012 Unix User\alice (Local User)
S-1-22-1-1013 Unix User\arlene (Local User)
S-1-22-1-1014 Unix User\megan (Local User)
S-1-22-1-1015 Unix User\kim (Local User)
S-1-22-1-1016 Unix User\timothy (Local User)
S-1-22-1-1017 Unix User\mark (Local User)
S-1-22-1-1018 Unix User\norman (Local User)
S-1-22-1-1019 Unix User\craig (Local User)
S-1-22-1-1020 Unix User\bradley (Local User)
```

```
S-1-22-1-1021 Unix User\gilbert (Local User)
S-1-22-1-1022 Unix User\louise (Local User)
S-1-22-1-1023 Unix User\liz (Local User)
S-1-22-1-1024 Unix User\nicola (Local User)
S-1-22-1-1025 Unix User\david (Local User)
S-1-22-1-1026 Unix User\robert (Local User)
S-1-22-1-1027 Unix User\lee (Local User)
S-1-22-1-1028 Unix User\brendan (Local User)
```

[+] Enumerating users using SID S-1-5-21-2284790533-161500423-4153759225 and log

```
S-1-5-21-2284790533-161500423-4153759225-501 OSCP\nobody (Local User)
S-1-5-21-2284790533-161500423-4153759225-513 OSCP\None (Domain Group)
```

[+] Enumerating users using SID S-1-5-32 and logon username '', password ''

```
S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)
```

```
=====( Getting printer info for 192.168.134.126 )=====
```

No printers returned.

Vulnerability Exploited: An authentication log on the webpage shows changed password hashes, which can be cracked and combined with enumerated users from enum4linux to dictionary-attack FTP/SSH and obtain valid credentials.

Vulnerability Explained: The initial Nikto scans direct an interesting /log.txt page to our attention:

[[Pasted image 20230503144829.png]]

[[Pasted image 20230503144850.png]]

The output of /log.txt seems to show logons, logoffs, and password changes. The 3 password change entries actually give the changed password's MD5 hash, which we can try cracking with hashcat:

[[Pasted image 20230504005704.png]]

[[Pasted image 20230503145104.png]]

This gives us 2 possible passwords. During the initial scans, port 139 and 445 were open, which prompted a scan of SMB using enum4linux:

[[Pasted image 20230503145218.png]]

[[Pasted image 20230503145359.png]]

[[Pasted image 20230503145327.png]]

We can see that there are a good number of users available, with two potential passwords and no password policy - not that trying 2 passwords per user would really trigger too much. Creating a custom user .txt and pass .txt, we can use hydra and try to validate a set of credentials.

[[Pasted image 20230504005734.png]]

[[Pasted image 20230503145506.png]]

hydra yields the credentials rowan:1ntr0spect, and we can authenticate to SSH and log in.

Local.txt Contents: [[Pasted image 20230503145928.png]] Flag: d11f162102e5a4be16fc071057a1d352

Vulnerability Fix: The /log.txt should not be visible to outside attackers. I was not able to find what it was for, but I'm sure it wasn't necessary for the public to view. In addition, the rowan user should change their password to something more secure.

Severity: This is a high severity vulnerability because it is simple and the rowan user is in the admin group on the host.

3.2.6.2 Privilege Escalation

Privilege escalation is not immediately obvious here. rowan is unable to run commands as sudo, and there are no SUID binaries. There is a suspect file in the /opt directory called backup.sh, which seems to be just a one-liner that executes the rsync command and makes a backup.

[[Pasted image 20230503145739.png]]

Transferring over tools would assist greatly in our endeavors here. To transfer, we can run `scp lin-peas.sh rowan@192.168.134.126:/home/rowan` to transfer over our linpeas.sh script from <https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS> that can point out privilege escalation vectors. [[Pasted image 20230503145622.png]]

Vulnerability Exploited: The rsync command has the `cap_dac_override=ep` capability, which enables it sync (copy) files regardless of permissions. Although we have root-level privileges and

could just copy over the flag and read it, we need a shell. We can `rsync` the `/etc/passwd` file to our home directory, add a new root user to it, and `rsync` it straight back so that it overwrites the old `/etc/passwd`.

Vulnerability Explanation: Running the afore-mentioned `linpeas.sh` script, it highlights an interesting command that we've seen in an interesting scripts:

[[Pasted image 20230503145649.png]]

The `cap_dac_override=ep` capability gives `rsync` the power to operate while ignoring file permissions. This is probably due to the `backup.sh` script that it is used in. We can give it a test run by seeing what happens when we `rsync /etc/passwd /home/rowan` and sync over `/etc/passwd`:

[[Pasted image 20230503150055.png]]

It copies over `/etc/passwd`, but under our `rowan` user's ownership. Since it ignores file permissions, we can probably rewrite the old `/etc/passwd` as well. To test that out, let's modify the `passwd` file in our possession with a new root user with credentials `chacei:chacei`.

[[Pasted image 20230503150412.png]]

[[Pasted image 20230503150349.png]]

Hashing the password with the built-in `openssl` command and giving our `chacei` user root permissions of `0:0`, this is what the final product looks like when we `rsync` it over:

[[Pasted image 20230503150456.png]]

All that is left is to `su chacei`.

Proof.txt Contents: [[Pasted image 20230503150632.png]] Flag: 507a3fb2ad3a408faf4483622bc6056c

Vulnerability Fix: Although the `cap_dac_override=ep` capability was probably enabled for the `backup.sh` script that modified the `backup.txt` in `nicola`'s directory, changing around file permissions is probably a safer bet than giving `rsync` that sort of absolute power.

Severity: Complete file read/write and escalation to root gives this a high level of severity.

3.3 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

4 Additional Items

4.1 Appendix - Proof and Local Contents:

IP	Local.txt Contents	Proof.txt Contents
192.168.134.100	N/A	d513b963ae8d75d5dbddeb04933d1a49
192.168.134.101	84cff16d270fab734043cf27ab014109	N/A
192.168.134.102	edd330f064ef3da6df495095a04e141f	4a96bdaf1d8a514623671f8b1df7ef0b
192.168.134.110	d700e9f4a877412386b37fbf9952b2f3	c7f1cbc6e8de951f34f4e4ea2eda7d2d
192.168.134.114	6b4a79fa6bf204b8ca285a808603a6fbbe1c245d1a35ea9103aea962e10ab98e	
192.168.134.126	d11f162102e5a4be16fc071057a1d352507a3fb2ad3a408faf4483622bc6056c	