foxglovesec / Potato

forked from breenmachine/Potato

Windows privilege escalation through NTLM Relay and NBNS Spoofing

README.md

Potato

Privilege Escalation on Windows 7,8,10, Server 2008, Server 2012

How it works

Potato takes advantage of known issues in Windows to gain local privilege escalation, namely NTLM relay (specifically HTTP->SMB relay) and NBNS spoofing.

Using the techniques outlined below, it is possible for an unprivileged user to gain "NT AUTHORITY\SYSYTEM" level access to a Windows host in default configurations.

The exploit consists of 3 main parts, all of which are somewhat configurable through command-line switches:

1. Local NBNS Spoofer

NBNS is a broadcast UDP protocol for name resolution commonly used in Windows environments. In penetration testing, we often sniff network traffic and respond to NBNS queries observed on a local network. For privilege escalation purposes, we can't assume that we are able to sniff network traffic, so how can we accomplish NBNS spoofing?

If we can know ahead of time which host a target machine (in this case our target is 127.0.0.1) will be sending an NBNS query for, we can craft a response and flood the target host with NBNS responses (since it is a UDP protocol). One complication is that a 2-byte field in the NBNS packet, the TXID, must match in the request and response. We can overcome this by flooding quickly and iterating over all 65536 possible values.

What if the host we are trying to spoof has a DNS record already? Well we can FORCE DNS lookups to fail in a funny way. Using a technique called "port exhaustion" we bind to every single UDP port. When you try to perform a DNS lookup it will fail because there will be no available source port for the DNS reply to come to.

In testing, this has proved to be 100% effective.

2. Fake WPAD Proxy Server

With the ability to spoof NBNS responses, we can target our NBNS spoofer at 127.0.0.1. We flood the target machine (our own machine) with NBNS response packets for the host "WPAD", or "WPAD.DOMAIN.TLD", and we say that the WPAD host has IP address 127.0.0.1.

At the same time, we run an HTTP server locally on 127.0.0.1. When it receives a request for "http://wpad/wpad.dat", it responds with something like the following:

```
FindProxyForURL(url,host){
    if (dnsDomainIs(host, "localhost")) return "DIRECT";
    return "PROXY 127.0.0.1:80";}
```

This will cause all HTTP traffic on the target to be redirected through our server running on 127.0.0.1.

Interestingly, this attack when performed by even a low privilege user will affect all users of the machine. This includes administrators, and system accounts. See the screenshots "egoldstein_spoofing.png" and "dade_spoofed.png" for an example.

3. HTTP -> SMB NTLM Relay

With all HTTP traffic now flowing through a server that we control, we can do things like request NTLM authentication...

In the Potato exploit, all requests are redirected with a 302 redirect to "http://localhost/GETHASHESxxxxx", where xxxxx is some unique identifier. Requests to "http://localhost/GETHASHESxxxxx" respond with a 401 request for NTLM authentication.

The NTLM credentials are relayed to the local SMB listener to create a new system service that runs a user-defined command. This command will run with "NT AUTHORITY\SYSTEM" privilege.

Using the Exploit

Usage is currently operating system dependant.

It is also a bit flaky sometimes, due to the quirks in how Windows handles proxy settings and the WPAD file. Often when the exploit doesn't work, it is required to leave it running and wait. When Windows already has a cached entry for WPAD, or is allowing direct internet access because no WPAD was found, it could take 30-60 minutes for it to refresh. It is necessary to leave the exploit running and try to trigger it again later, after this time has elapsed.

The techniques listed here are ordered from least to most complex. Any technique later in the list should work on all versions previous. Videos and screenshots are included for each.

Windows 7 - see https://www.youtube.com/watch?v=Nd6f5P3LSNM

Windows 7 can be fairly reliably exploited through the Windows Defender update mechanism.

Potato.exe has code to automatically trigger this. Simply run the following: `Potato.exe -ip <local ip> -cmd <command to run> -disable_exhaust true`

This will spin up the NBNS spoofer, spoof "WPAD" to 127.0.0.1, then check for Windows Defender updates.

If your network has a DNS entry for "WPAD" already, you can try "-disable_exhaust false". This should cause the DNS lookup to fail and it should fallback to NBNS. We've tested this a couple times and had it work

Windows Server 2008 - see https://www.youtube.com/watch?v=z_IGPWgL5SY

Since Windows Server doesn't come with Defender, we need an alternate method. Instead we'll simply check for Windows updates. The other caveat is that, at least on my domain, Server 2K8 wanted WPAD.DOMAIN.TLD instead of just WPAD. The following is an example usage:

```
Potato.exe -ip <local ip> -cmd <command to run> -disable_exhaust true -disable_defender true --spoof_host WPAD.EMC.LOCAL
```

After this runs successfully, simply check for Windows updates. If it doesn't trigger, wait about 30m with the exploit running and check again. If it still doesn't work, try actually downloading an update.

If your network has a DNS entry for "WPAD" already, you can try "-disable_exhaust false". This should cause the DNS lookup to fail and it should fallback to NBNS. We've tested this a couple times and had it work

Windows 8/10/Server 2012 - see https://www.youtube.com/watch?v=Kan58VeYpb8

In the newest versions of Windows, it appears that Windows Update may no longer respect the proxy settings set in "Internet Options", or check for WPAD. Instead proxy settings for Windows Update are controlled using "netsh winhttp proxy..."

Instead for these versions, we rely on a newer feature of Windows, the "automatic updater of untrusted certificates". Details can be found https://support.microsoft.com/en-us/kb/2677070 and https://technet.microsoft.com/en-us/library/dn265983.aspx

From the technet article "The Windows Server 2012 R2, Windows Server 2012, Windows 8.1, and Windows 8 operating systems include an automatic update mechanism that downloads certificate trust lists (CTLs) on a daily basis."

It appears that this part of Windows still uses WPAD, even when the winhttp proxy setting is set to direct.

In this case the usage of Potato is as follows: `Potato.exe -ip <local ip> -cmd <cmd to run> -disable_exhaust true -disable_defender true`

At this point, you will need to wait up to 24hrs or find another way to trigger this update.

If your network has a DNS entry for "WPAD" already, you can try "-disable_exhaust false". This should cause the DNS lookup to fail and it should fallback to NBNS. We've tested this a couple times and had it work

Mitigations

Enabling "Extended Protection for Authentication" in Windows should stop NTLM relay attacks.

SMB Signing may also mitigate this type of attack, however this would require some more research on my part to confirm.

Off Broadcast NBNS Spoofing

Using the same NBNS spoofing technique as the Potato exploit, we can perform NBNS spoofing against any host for which we can talk to UDP 137. We simply need to send UDP packets quickly enough to sneak in a valid reply before the NBNS request times out.

A demo video of this can be seen at https://www.youtube.com/watch?v=Mzn7ozkyG5g

The demo lab has the following setup:

PFSense firewall 10.0.0.0/24 -> Corporate LAN 10.0.1.0 /24 -> Server network

From the corporate network, we'll attack a machine on the server network.

Usage: `python Responder.py -I eth0 -spoof <target>:<spoof address>:<spoof host>`