- Images
- KrbRelayUp
- packages
- KrbRelayUp.sln
- README.md

README

KrbRelayUp

Simple wrapper around some of the features of Rubeus and KrbRelay (and a few other honorable mentions in the acknowledgements section) in order to streamline the abuse of the following attack primitive:

0. (Optional) New machine account creation (New-MachineAccount)
1. Local machine account auth coercion (KrbRelay)
2. Kerberos relay to LDAP (KrbRelay)

About

KrbRelayUp - a universal no-fix local privilege escalation in windows domain environments where LDAP signing is not enforced (the default settings).

Releases

No releases published

Packages

No packages published

Contributors 6

Languages

3. Add RBCD privs and obtain privileged ST to local machine (Rubeus)
4. Using said ST to authenticate to local Service Manager and create a new service as NT/SYSTEM. (SCMUACBypass)

This is essentially a universal no-fix local privilege escalation in windows domain environments where LDAP signing is not enforced (the default settings).

UPDATE: Here is an excellent writeup by @an0n_r0 on how to perform this attack manually (using the **original tools** for this attack path: PowerMad/SharpMad, KrbRelay, Rubeus and SCMUACBypass)

Update - Shadow Credentials Support

I added some features to support this attack primitive using shadow credentials. Note this eliminates the need for adding (or owning) another machine account.

Note: *this attack method bypasses the Protected Users (or 'Account is sensitive and cannot be delegated') mitigation due to the S4U2Self abuse.*

1. Local machine account auth coercion (KrbRelay)
2. Kerberos relay to LDAP (KrbRelay)
3. Generate new KeyCredential and add it to the local machine account's 'msDS-KeyCredentialLink' attribute. (Whisker and KrbRelay)
4. Using said KeyCredential to obtain a TGT for the local machine account via PKInit. (Rubeus)
5. Using the TGT to obtain privileged ST to local machine via S4U2Self and TGSSUB. (Rubeus)
6. Using said ST to authenticate to local Service Manager and create a new service as NT/SYSTEM. (SCMUACBypass)

UPDATE: Here is an excellent writeup by @icyguider on how to perform the ShadowCred method of this attack manually (using the **original tools** for this attack path: KrbRelay, Rubeus

and SCMUACBypass) along with the usage of NimCrypt2 to pack the various tools and evade some detection by defensive mechanism.

Update - ADCS Web Enrollment Support

I added support for relaying Machine KRB auth to ADCS Web Enrollment (instead of LDAP). This eliminates the requirement of LDAP Signing not to be enforced in the domain since this attack does not relay to LDAP.

Note: *this attack method bypasses the Protected Users (or 'Account is sensitive and cannot be delegated') mitigation due to the S4U2Self abuse.*

1. Local machine account auth coercion (KrbRelay)
2. Kerberos relay to ADCS (HTTP) (KrbRelay and ADCSPwn)
3. Generate certificate request on behalf of the local machine account, submit it to ADCS Web Enrollment and finally retreive the certificate for the local machine account (ADCSPwn)
4. Using said certificate to obtain a TGT for the local machine account via PKInit. (Rubeus)
5. Using the TGT to obtain privileged ST to local machine via S4U2Self and TGSSUB. (Rubeus)
6. Using said ST to authenticate to local Service Manager and create a new service as NT/SYSTEM. (SCMUACBypass)

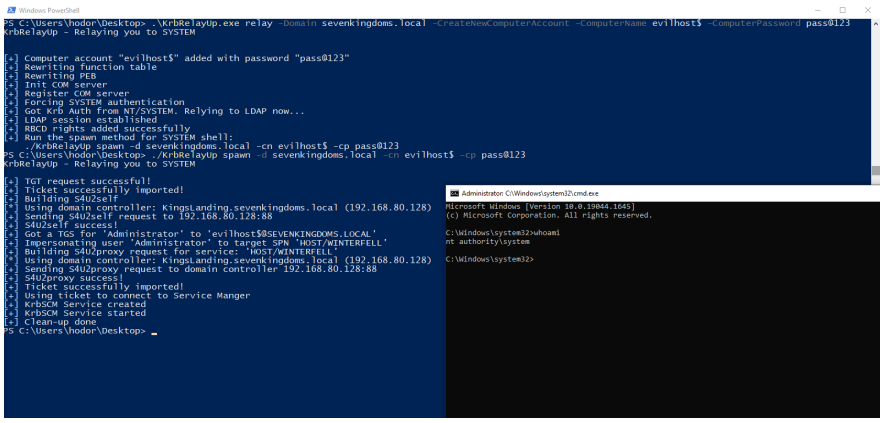
Usage

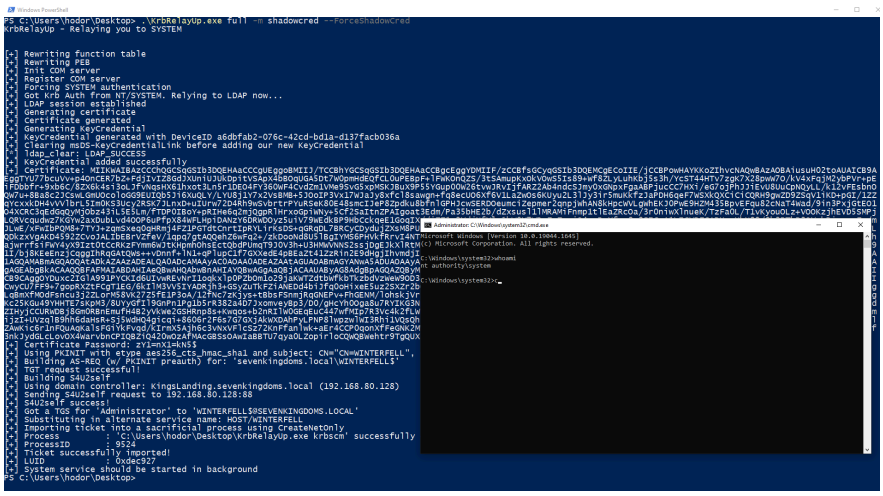
```
KrbRelayUp - Relaying you to SYSTEM

FULL: Perform full attack chain. Options are id

RELAY: First phase of the attack. Will Coerce K
Usage: KrbRelayUp.exe relay -d FQDN -cn COMPUTE

    -m   (--Method)                  Abuse met
```

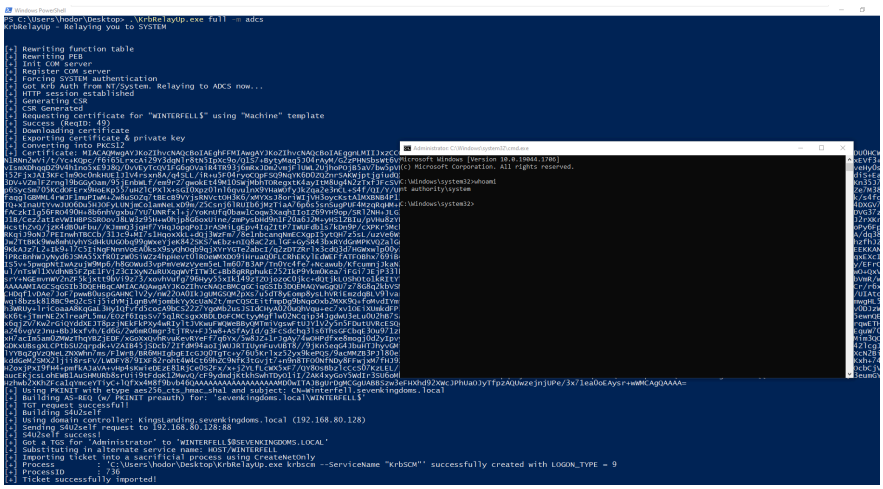
```
    -p   (--Port)                       Port for (
    -cls (--Clsid)                      CLSID to u

    # RBCD Method:
    -c   (--CreateNewComputerAccount) Create ne
    -cn  (--ComputerName)             Name of a
    -cp  (--ComputerPassword)         Password (

    # SHADOWCRED Method:
    -f   (--ForceShadowCred)          Clear the

    # ADCS Method:
    -ca  (--CAEndpoint)               CA endpoi
    -https                            Connect t
    -cet (--CertificateTemplate)      Certifica


SPAWN: Second phase of the attack. Will use the
Usage: KrbRelayUp.exe spawn -d FQDN -cn COMPUTE

    -m   (--Method)                   Abuse met
    -i   (--Impersonate)              User to i
    -s   (--ServiceName)              Name of t
    -sc  (--ServiceCommand)           Service c

    # RBCD Method:
    -cn  (--ComputerName)             Name of a
    -cp  (--ComputerPassword)         Password (
    -ch  (--ComputerPasswordHash)     Password

    # SHADOWCRED | ADCS Method:
    -ce  (--Certificate)              Base64 en
    -cep (--CertificatePassword)      Certifica


KRBSCM: Will use the currently loaded Kerberos
Usage: KrbRelayUp.exe krbscm <-s SERVICENAME> <

    -s  (--ServiceName)              Name of th
    -sc (--ServiceCommand)           Service co


General Options:
    -d  (--Domain)                   FQDN of do
    -dc (--DomainController)         FQDN of do
    -ssl                             Use LDAP o
```

```
    -n                                  Use Createl
    -v  (--Verbose)                     Show verbo
```

Examples







TODO

- [x] Code refactoring and cleanup!!!
- [x] Add ShadowCred attack as a RELAY method
- [ ] Add TGTDELEG attack in SPAWN method to be used in Network Service->SYSTEM scenarios (potatoes alternative)
- [x] Fix the issue I'm having trying to combine the RELAY and SPAWN methods into one run so it can be used as one complete command. Probably has something to do with the fact that both RELAY and SPAWN functionalities rely on hooks during the initialization of the COM Server (Once RELAY initializes its COM Server the SPAWN can't re-initialize it to place its hooks as well)

Mitigation & Detection

- Enforce LDAP Signing and LDAP Channel Binding to mitigate the relay of the machine account KRB auth to LDAP. This can be configured via the "Domain Controller: LDAP server signing requirements" GPO. (Thank you Will Dormann for your tweet on this matter)
- Make the attack requirements harder to obtain by setting the MS-DS-Machine-Account-Quota attribute in AD to **0**, thus removing the ability of any user to add a new machine account to the domain. This is a dangerous default setting in AD - make sure you change it.
- Setting the flag "Account is sensitive and cannot be delegated" on all admin accounts (or put them in protected users) would make it so there is no account with the privileges required that can be delegated in order to complete the attack path. (Thanks to Christoph Falta for this tweet)
- Mitigation for ADCS Relay - Enforcing the use of TLS on the ***certsrv*** site and enabling ***Extended Protection for Authentication (EPA)*** in IIS will prevent the relay to ADCS. (Thanks to Will Dormann for pointing that out in his tweet,

this also was mentioned in Dirk-jan Mollema post on Relaying Kerberos over DNS using krbrelayx and mitm6)

- Resources for possible monitoring and detection rules:
  i. https://github.com/tsale/Sigma_rules/blob/main/windows_exploitation/KrbRelayUp.yml (@Kostastsale)
  ii. https://twitter.com/SBousseaden/status/1518976397364056071 (@SBousseaden). Mainly the rule about authentication to Service Manager via Kerberos from 127.0.0.1, Great Work!.
  iii. https://www.linkedin.com/posts/john-dwyer-xforce_threathunting-threatdetection-blueteam-activity-6924739962131140608-py45/ (John Dwyer @TactiKoolSec)
  iv. https://twitter.com/cyb3rops/status/1519241598311321601 (@cyb3rops)

Acknowledgements

- James Forshaw for his research on Kerberos relaying and for figuring out how to use Kerberos Service Tickets for LOCAL authentication to Service Manager which was the missing piece of the puzzle in order to make this attack primitive **local only** (before that, we had to export the ST to a remote machine in order to use it and gain privileged access to our target machine). Also for his New-MachineAccount functionality which was used in this project.
- Cube0x0 This project wouldn't exist without his amazing work on KrbRelay - a lot of code was taken from there and it made me gain a deeper understanding on how Kerberos Relaying works (I really recommend going through the code for anyone who wish to understand the concept better).
- Elad Shamir for his research on Shadow Credentials and his awsome tool Whisker - parts of his code (and of course

cube0x0's KrbRelay code) was used to add support for the Shadow Credentials attack in this tool.

- Will Schroeder and everyone who contributed to Rubeus which we all know and love. Basically all the RBCD-S4U functionality was taken from there. Also, for Certify and the Certified Pre-Owned whitepaper (credits goes to Lee Christensen as well) which was used when adding the ADCS Web Enrollment Relay option.
- batsec and everyone who contributed to ADCSPwn. A lot of code related to the ADCS Web Enrollment Relay option was taken from this awesome tool.
- Michael Grafnetter for his tool DSInternals which was used