Bronze-Bit attack mitigation for old MIT Kerberos versions

Fixing CVE-2020-17049 for FreeIPA on CentOS 8 Stream and RHEL 8

Julien Rische

jrische@redhat.com

2024-04-18 SambaXP

Red Hat France

FreeIPA and MIT krb5

- FreeIPA is an authentication and identity management system
  - Relying on multiple projects
    - 389DS, MIT krb5, SSSD, Samba, ...
  - Use distribution's MIT krb5 package
- MIT KDC supports a range of **plugin interfaces**[1]
  - Preauth, ccache, password policy, realm mapping, KDC policy, KDB, ...
- FreeIPA has its own KDB plugin, using 389DS as a backend

The MS-SFU Kerberos extension

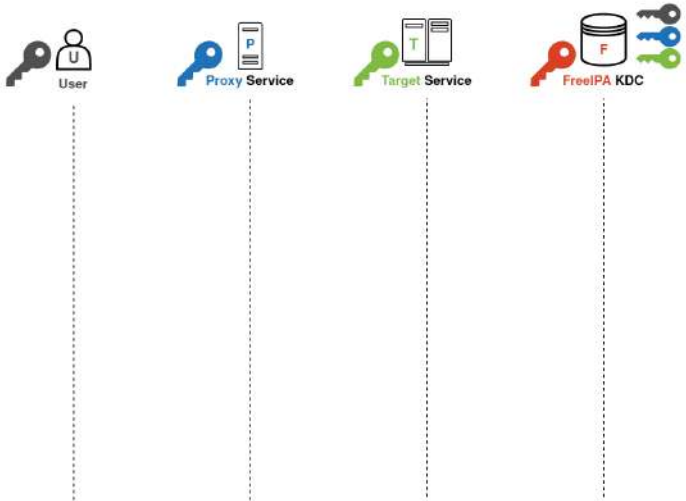
MS-SFU: Motivations

- Need to allow frontend **services to impersonate users**
  - Frontend: web service, ...
  - Backend: SQL database, distributed storage system, ...
- Historical solution: **TGT forwarding** (aka. *unconstrained delegation*)
  - Allow frontend service to access ANY service as the user
  - Bad solution from security perspective, more **granularity** required
- Microsoft implemented an extension called **MS-SFU**[2]
  - Introducing 2 new mechanisms
- Implemented in FreeIPA[3] using MIT krb5's KDB plugin interface

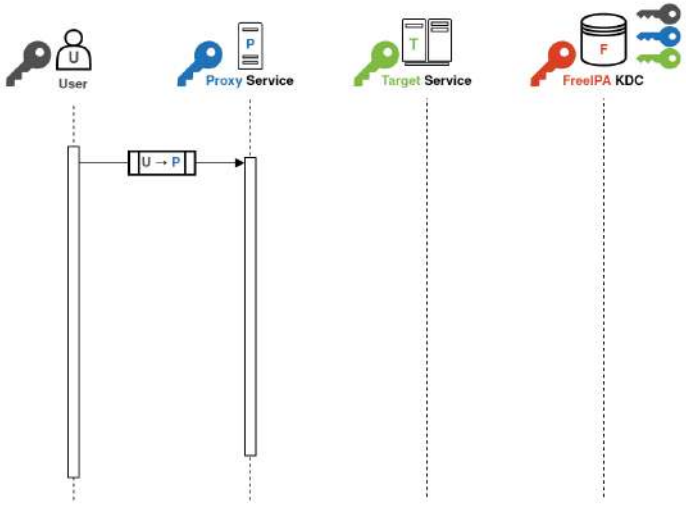
Constrained Delegation (S4U2Proxy)

- Allow a **proxy** service to impersonate a **user** against a specific **target** service[4,5,6,7]
- Configure service **delegation rules**
  - `ipa servicedelegation` commands
  - Specific administration permissions required to configure such rules
- At the condition of providing an **evicence ticket** to the **KDC**
  - Ticket for user-to-proxy service
  - With `forwarable` ticket flag set

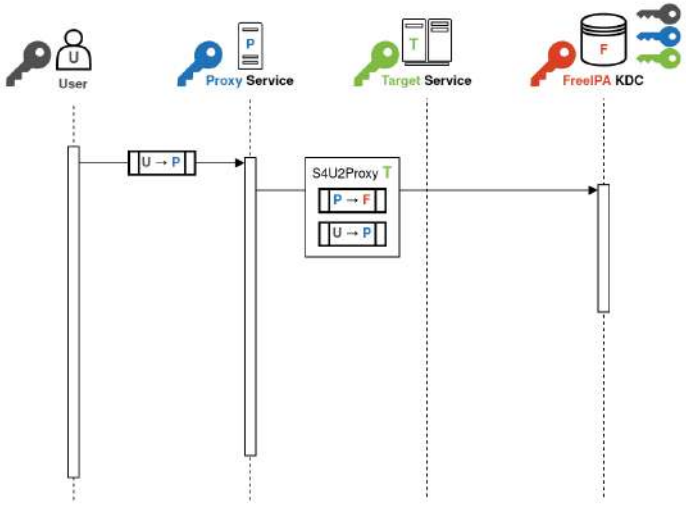
Constrained Delegation (S4U2Proxy)



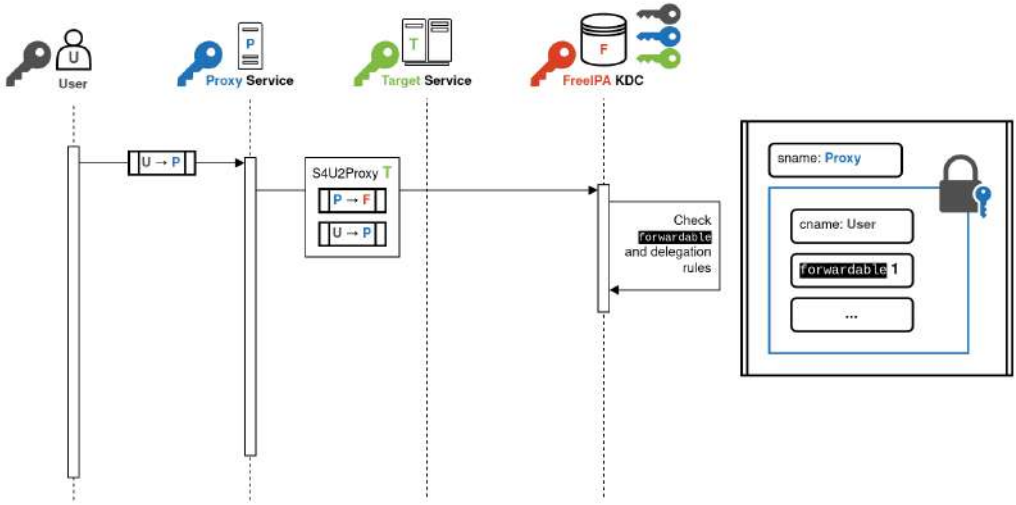
Constrained Delegation (S4U2Proxy)



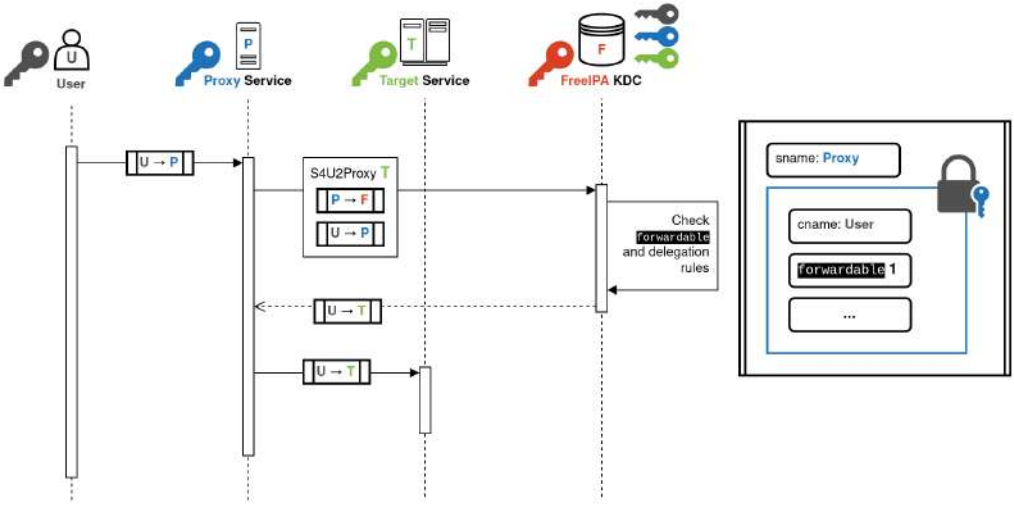
Constrained Delegation (S4U2Proxy)



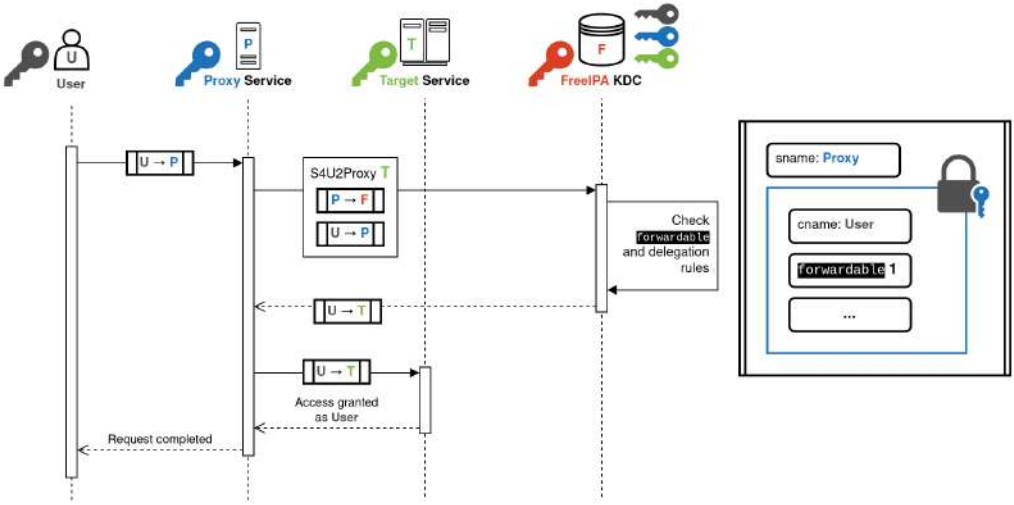
Constrained Delegation (S4U2Proxy)



Constrained Delegation (S4U2Proxy)



Constrained Delegation (S4U2Proxy)



Disclaimer

In **MS-SFU**, the naming is used the opposite way...

Disclaimer

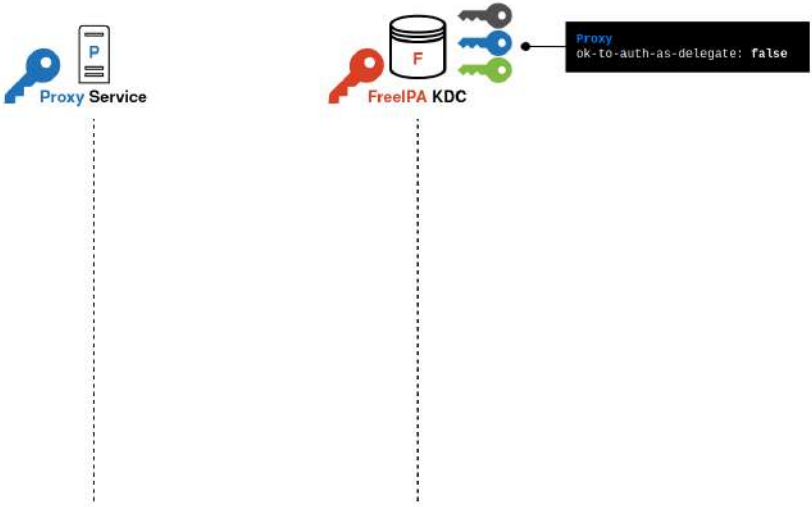
In **MS-SFU**, the naming is used the opposite way...

- The **Target** service is called **Proxy** (or **Service 2**)
- The **Proxy** service is called **Service 1**

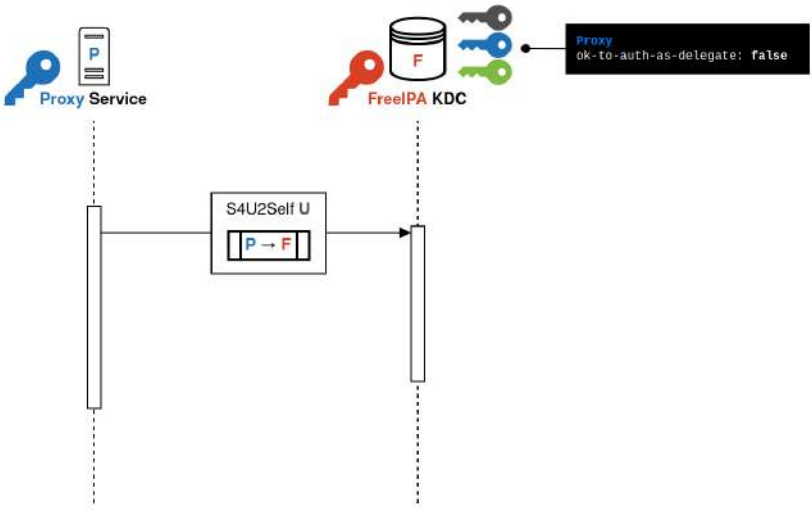
Protocol Transition (S4U2Self)

- Mean to:
  - Integrate services relying different authentication methods for users requests into the Kerberos authentication system
    - OIDC, SASL, ...
  - Obtain encrypted user authorization information
    - Use Kerberos as group membership provider
- Allow **any service with a valid TGT** to request a ticket from **any user to the service itself**
- Resulting ticket has `forwardable` flag set only if:
  - FreeIPA: principal configured with `ok-to-auth-as-delegate` privilege
  - AD: account configured with `TrustedToAuthForDelegation` privilege
  - (Or if no constrained delegation rules are set for the **proxy** service[8])

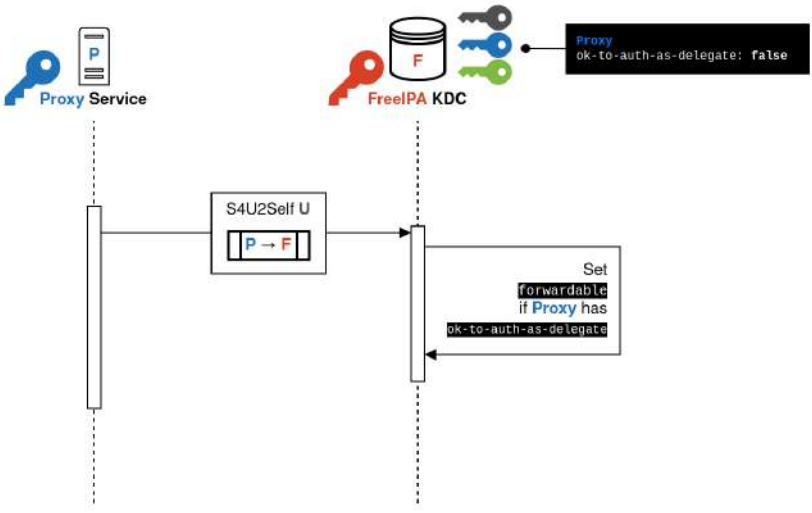
Protocol Transition (S4U2Self)



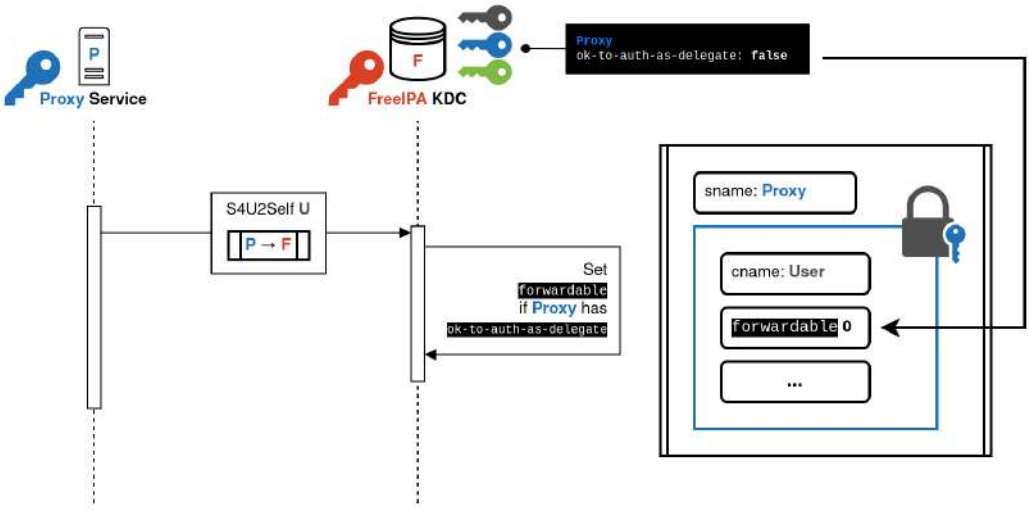
Protocol Transition (S4U2Self)



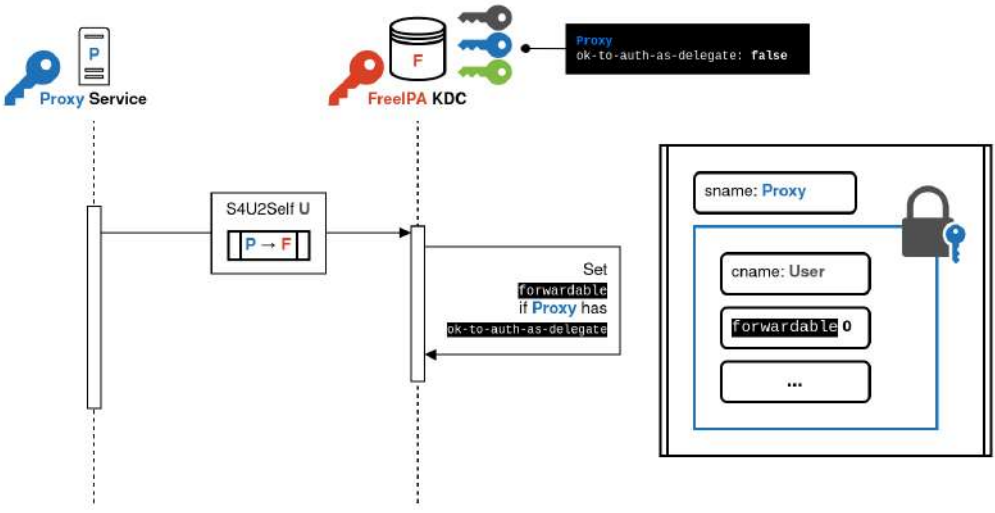
Protocol Transition (S4U2Self)



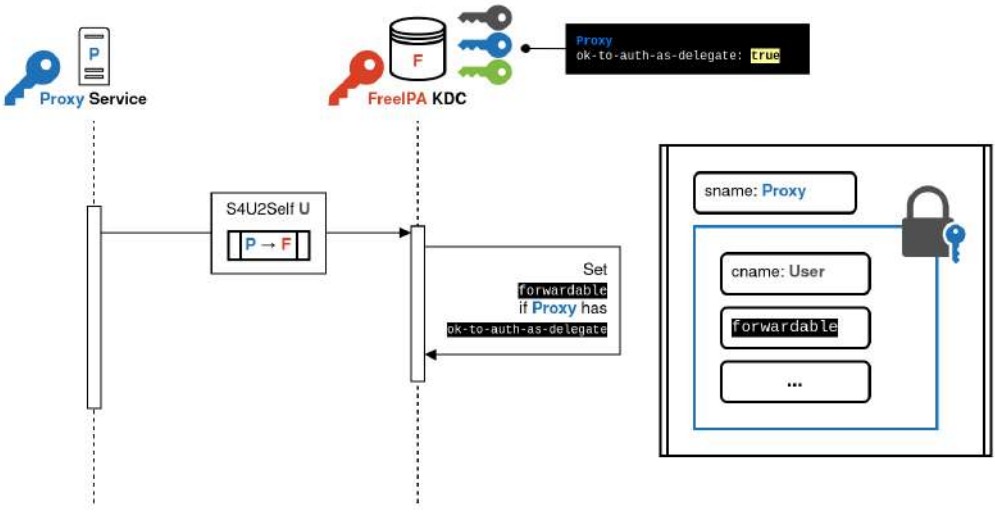
Protocol Transition (S4U2Self)



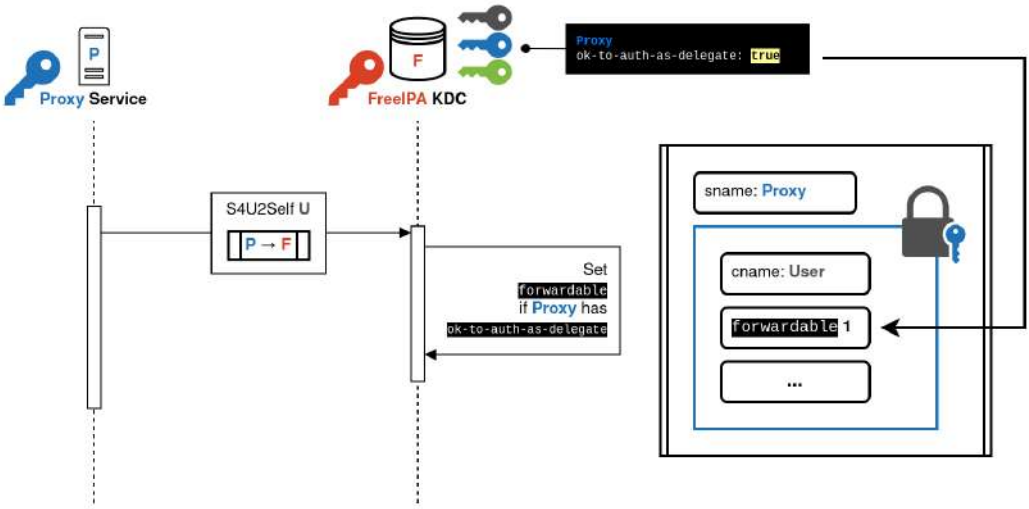
Protocol Transition (S4U2Self)



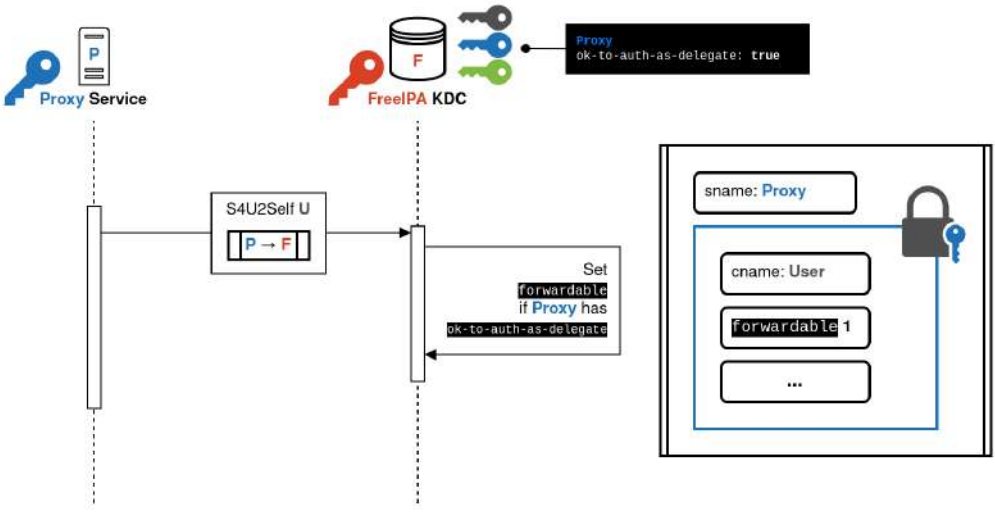
Protocol Transition (S4U2Self)



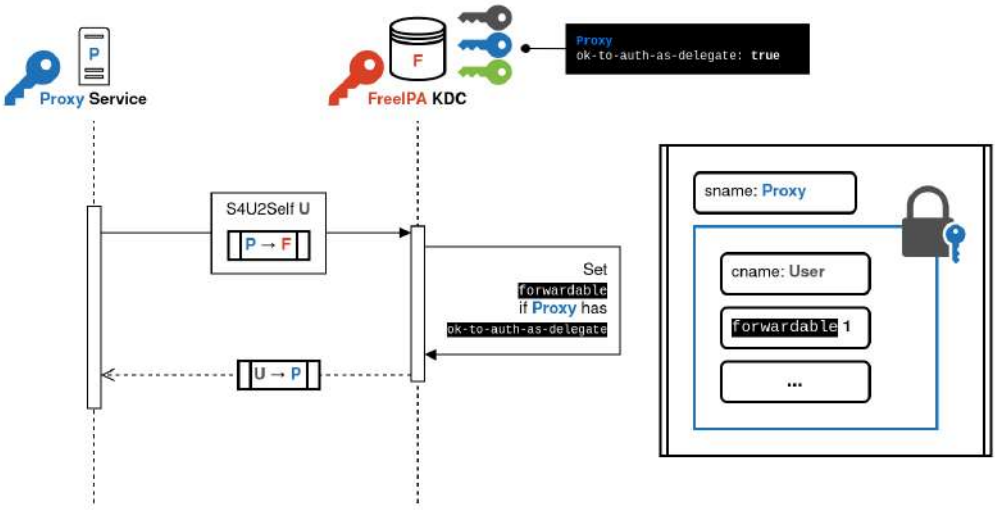
Protocol Transition (S4U2Self)



Protocol Transition (S4U2Self)



Protocol Transition (S4U2Self)



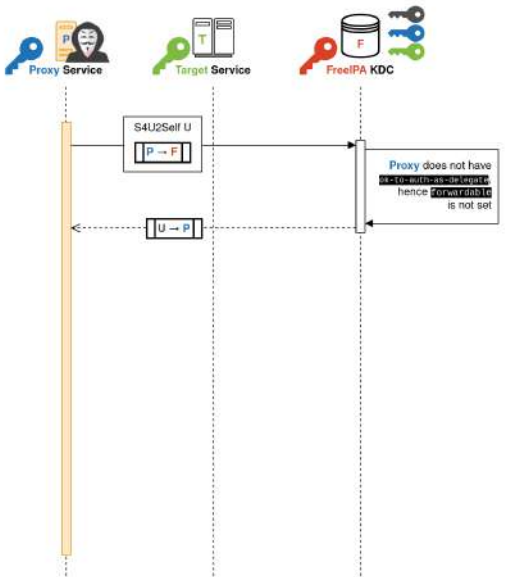
The Bronze-Bit exploit

The problem with MS-SFU

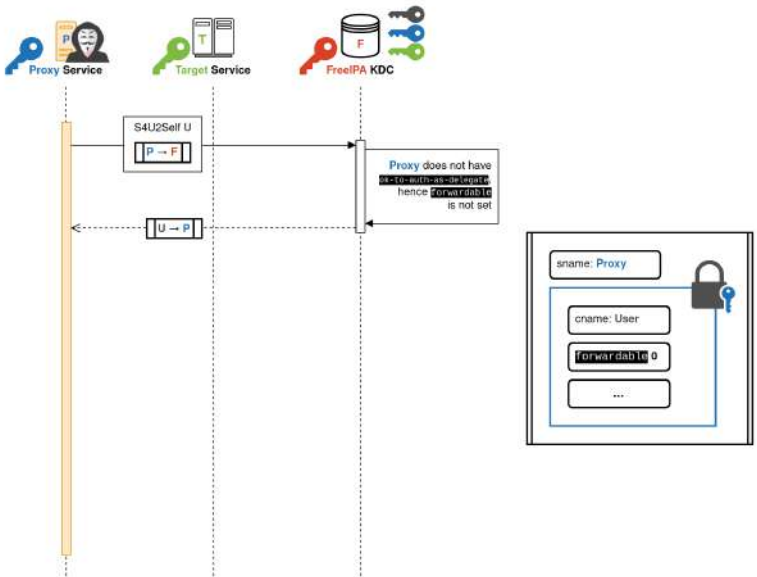
- A service with the `forwardable` S4U2Self ticket permission AND a constrained delegation rule can impersonate **any user** against the **target service** of this delegation rule
  - Including users with **administration privileges** for this service
- The `forwardable` flag is encrypted using the **proxy service** key
  - But nothing keeps the service from changing the value of this flag
- If the host running the proxy service is compromised, the attacker could use proxy service's credentials to **access the target service as an admin user**

CVE-2020-17049: The Bronze-Bit exploit

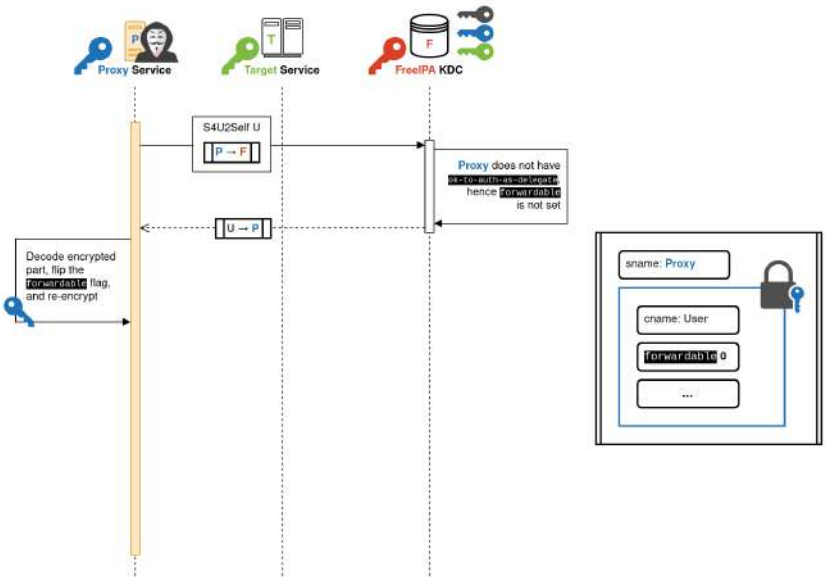
CVE-2020-17049: The Bronze-Bit exploit



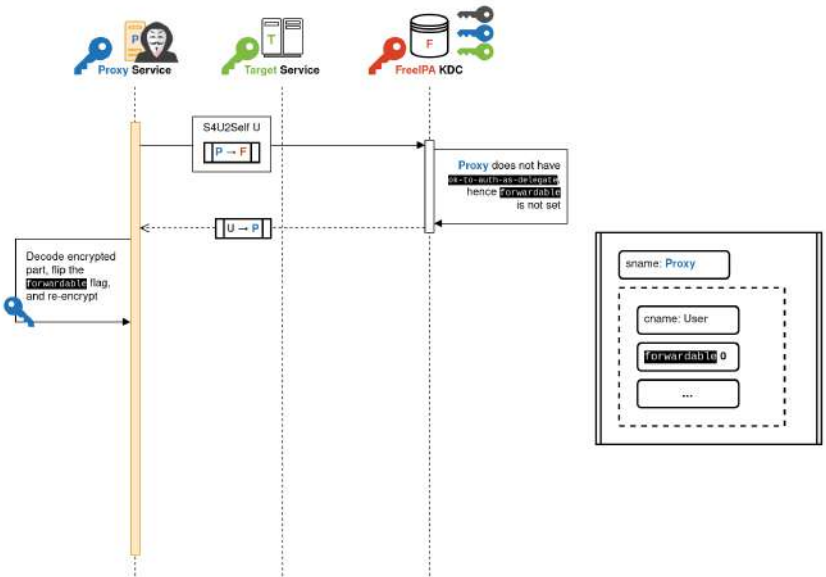
CVE-2020-17049: The Bronze-Bit exploit



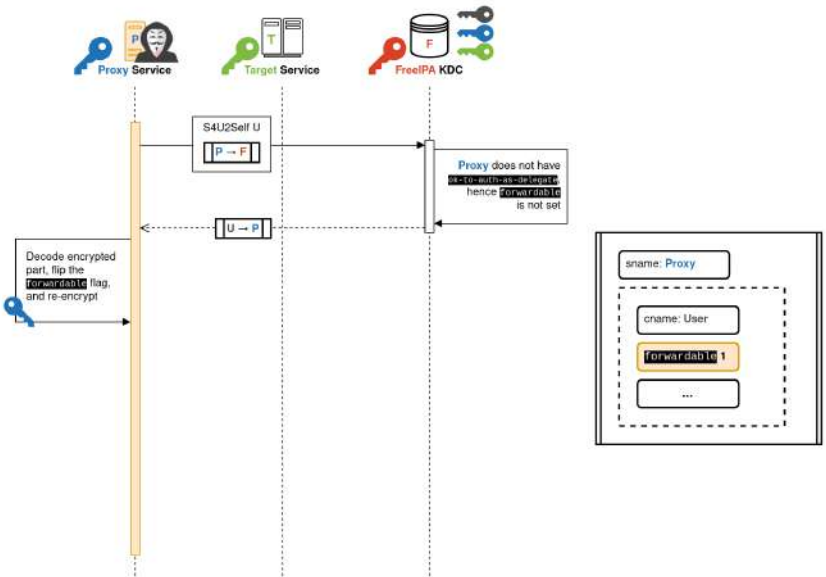
CVE-2020-17049: The Bronze-Bit exploit



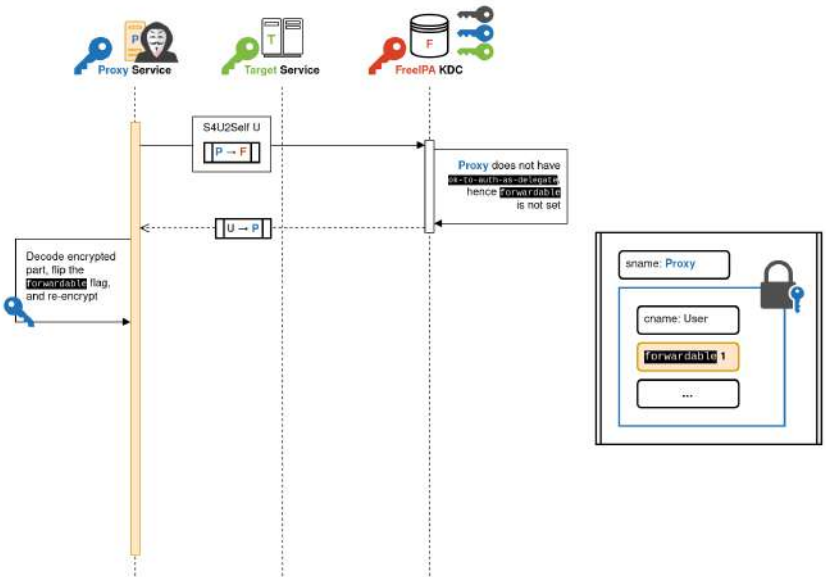
CVE-2020-17049: The Bronze-Bit exploit



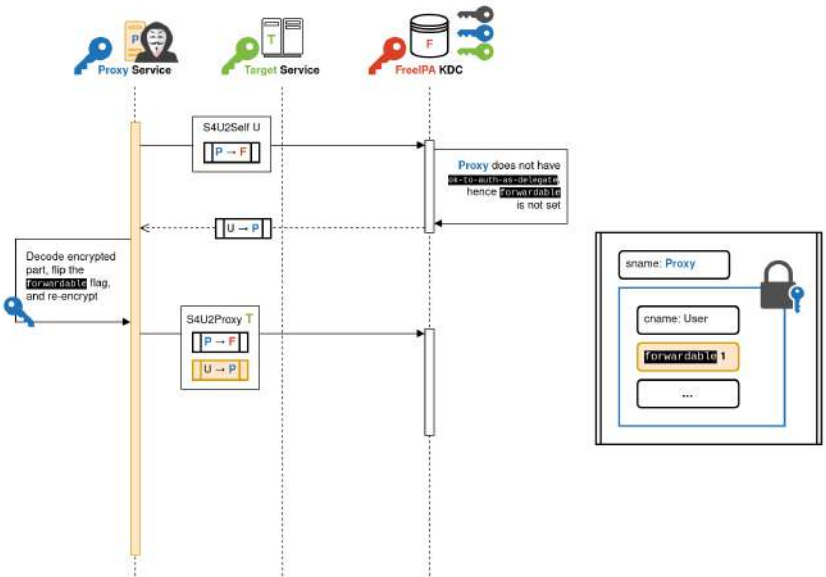
CVE-2020-17049: The Bronze-Bit exploit



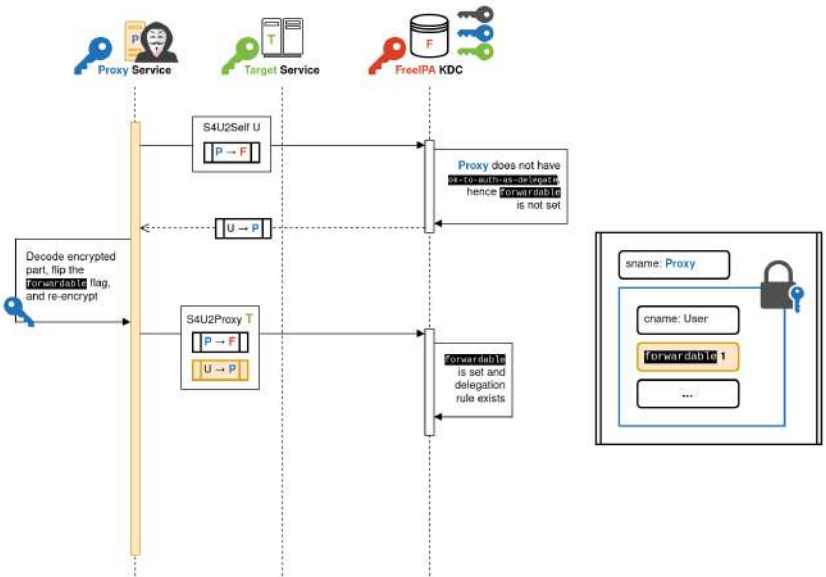
CVE-2020-17049: The Bronze-Bit exploit



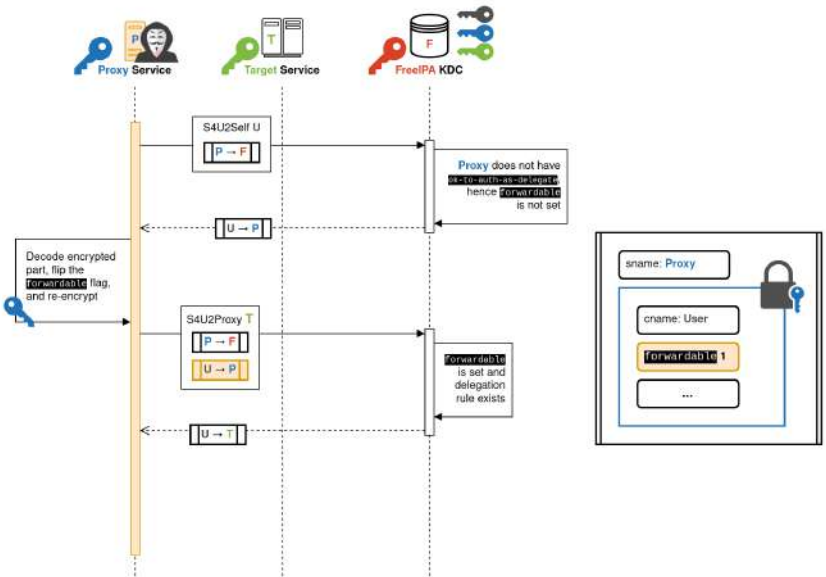
CVE-2020-17049: The Bronze-Bit exploit



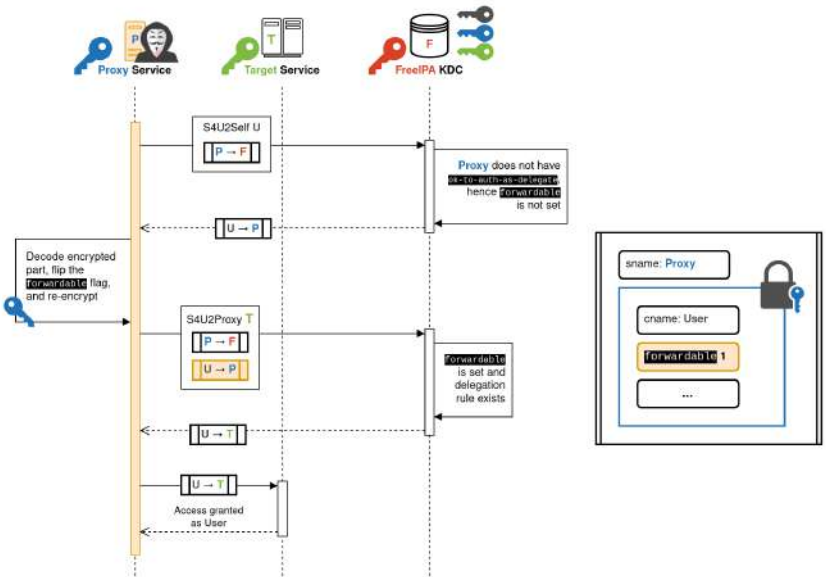
CVE-2020-17049: The Bronze-Bit exploit



CVE-2020-17049: The Bronze-Bit exploit



CVE-2020-17049: The Bronze-Bit exploit



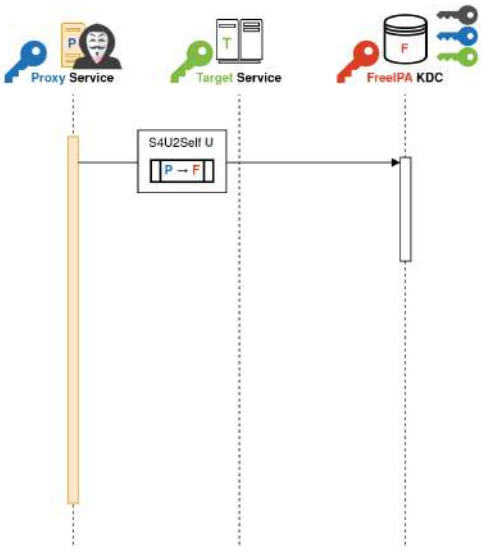
Reproducer for MIT Kerberos and FreeIPA

- All available reproducers designed for Active Directory
- None of them could work against FreeIPA, because they were missing support for:
  - `PA_S4U_X509_USER` ASN.1 sequence[9] (for S4U2Self)
  - AES HMAC-SHA2 encryption types family (from RFC8009[10])
- We implemented support for these 2 features in the **Impacket Python library**
  - `fortra/impacket#1684`[11]:
    Implement Kerberos encryption types from RFC8009 (AES HMAC-SHA2 family)
  - Will be needed when AD implements AES HMAC-SHA2 eventually[12]

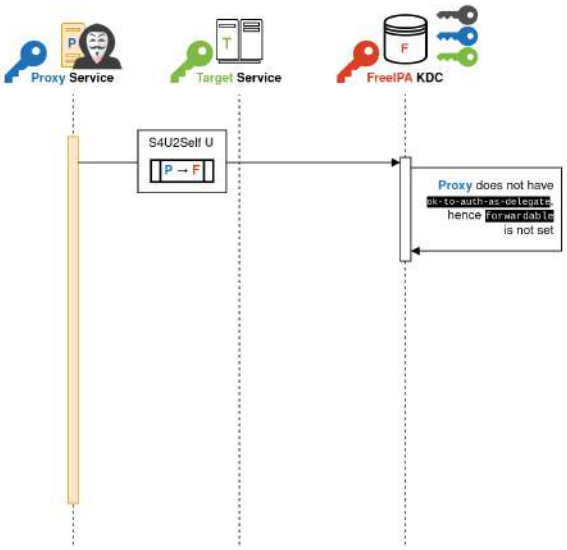
Fix: Ticket signature

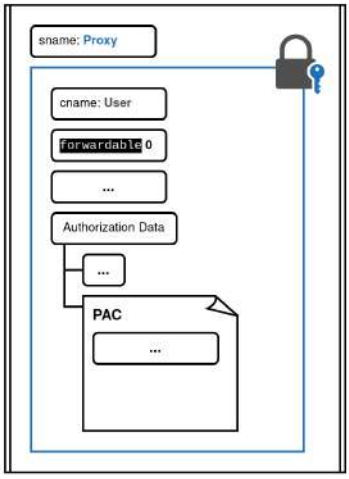
- Solution designed my Microsoft[13]
  - **Signature** actually means **keyed checksum** (RFC3961, RFC4120)
- Implemented by AD (KB4598347[14]) and MIT Kerberos 1.20[15]
- KDC signs the encrypted part of the ticket using the **TGS key**
  - KDC able to detect any modification of ticket's encrypted part
  - `forwardable` flag protected
- MS-PAC Kerberos extension
  - Add a **Privilege Attribute Certificate** (PAC) in the ticket

PAC ticket signature

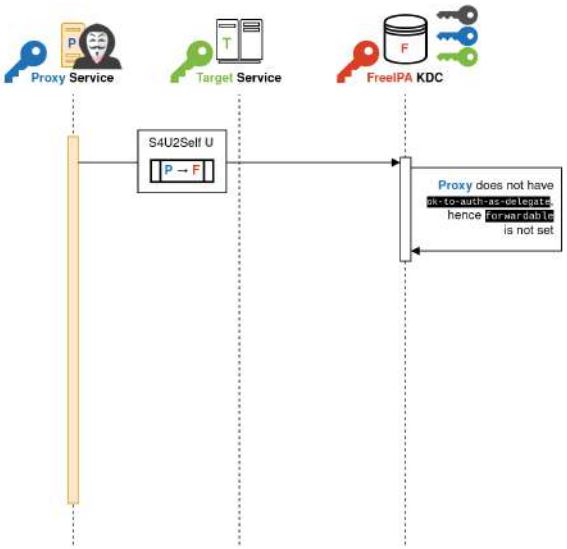


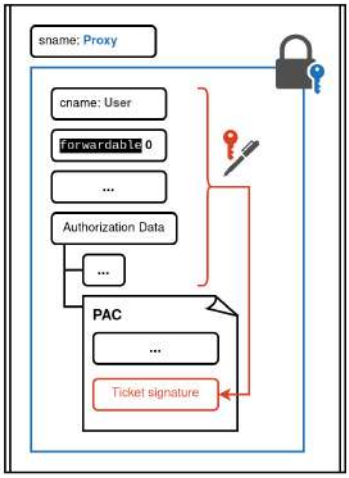
PAC ticket signature



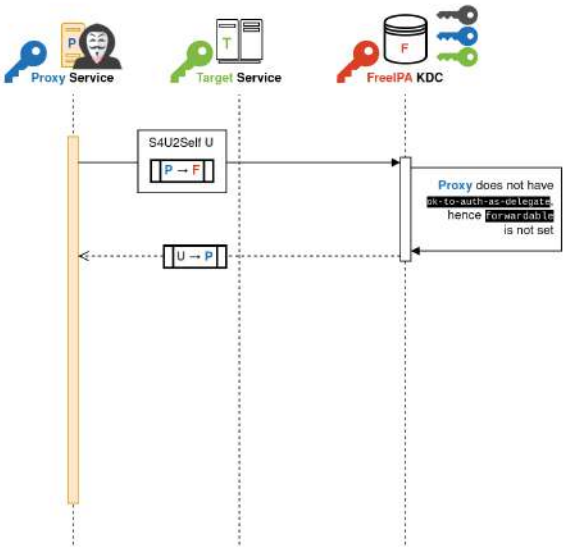


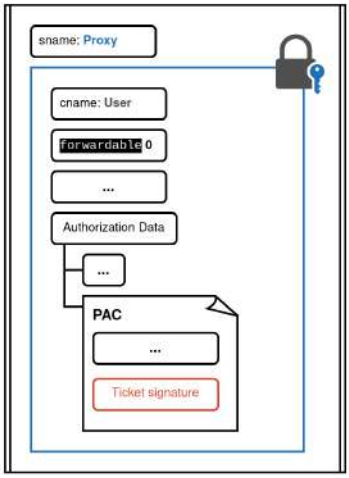
PAC ticket signature



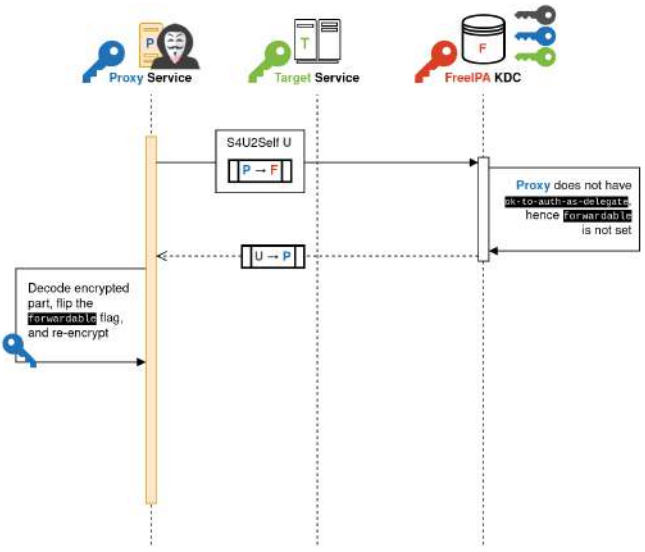


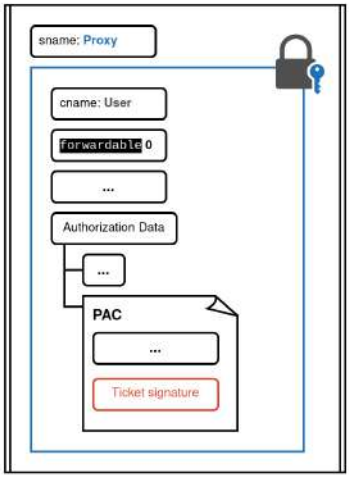
PAC ticket signature



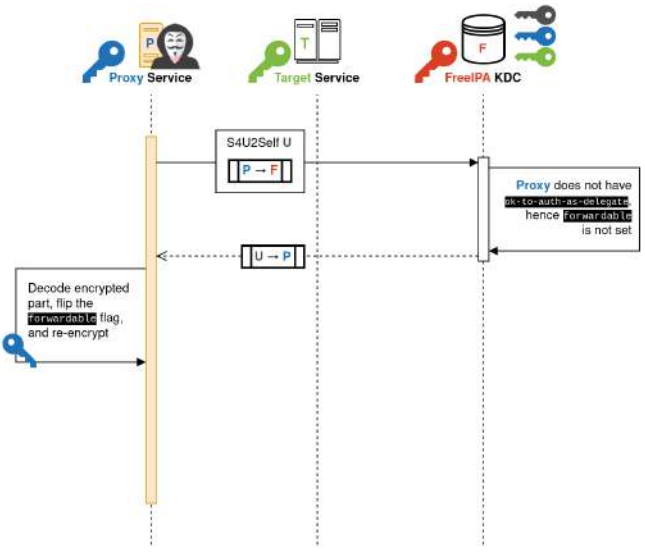


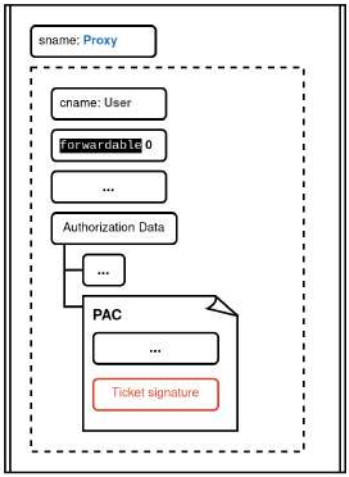
PAC ticket signature



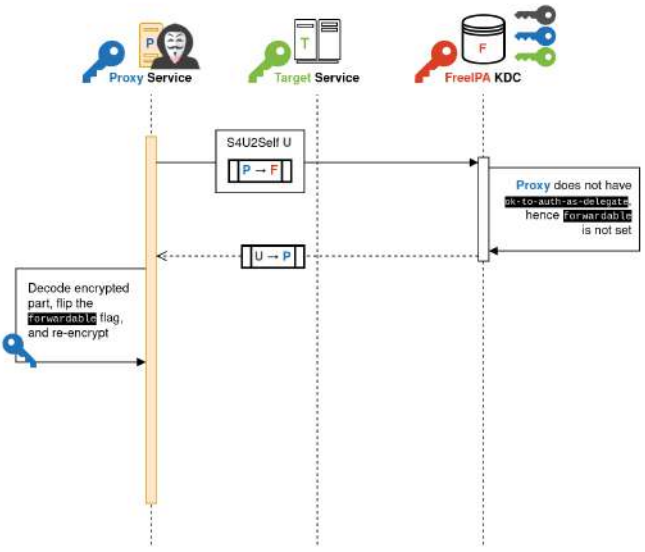


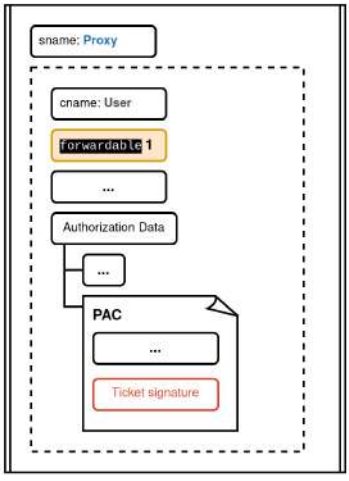
PAC ticket signature



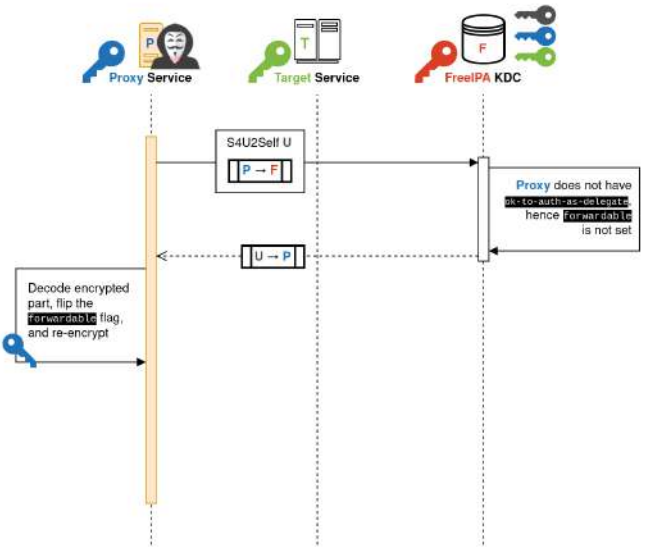


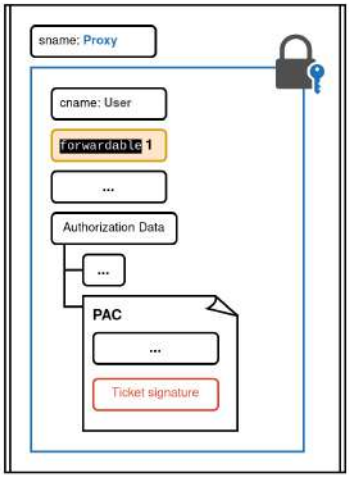
PAC ticket signature



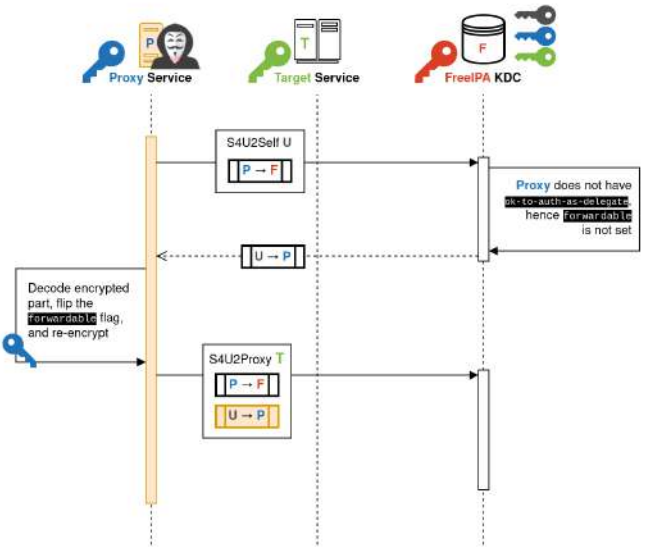


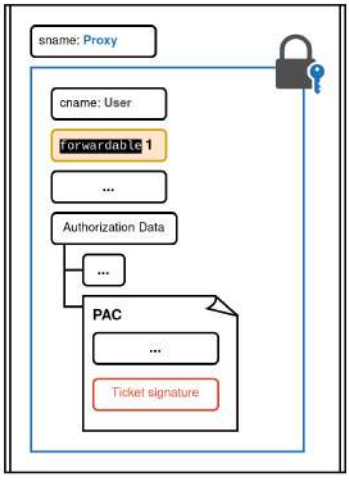
PAC ticket signature



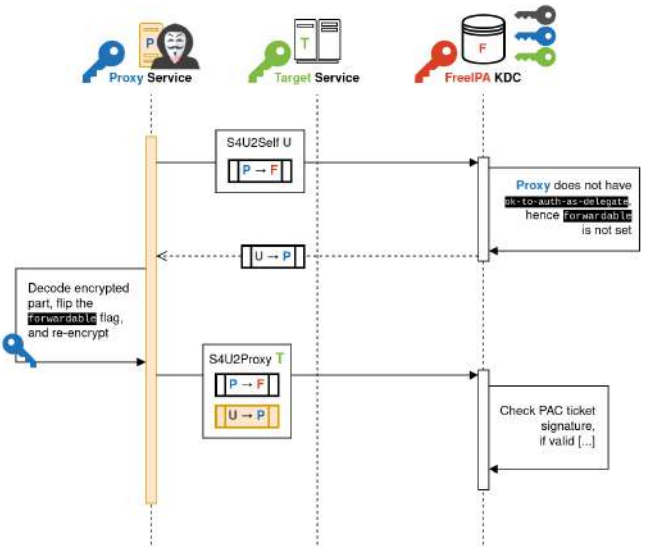


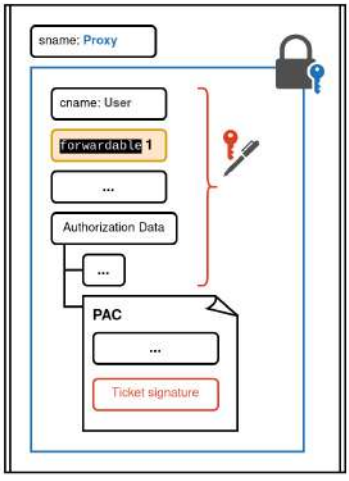
PAC ticket signature



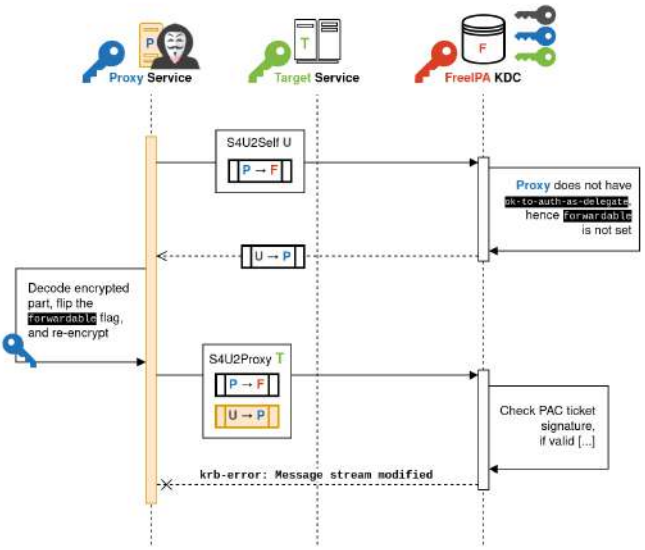


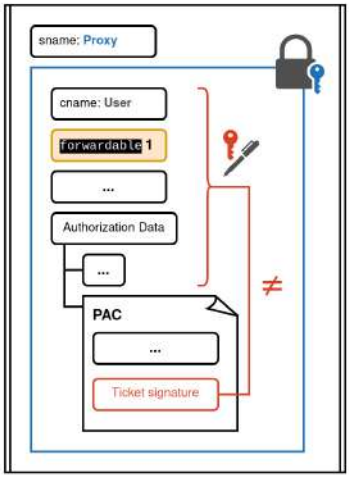
PAC ticket signature





PAC ticket signature





Fix for C8S and RHEL 8

C8S/RHEL8: Software constraints

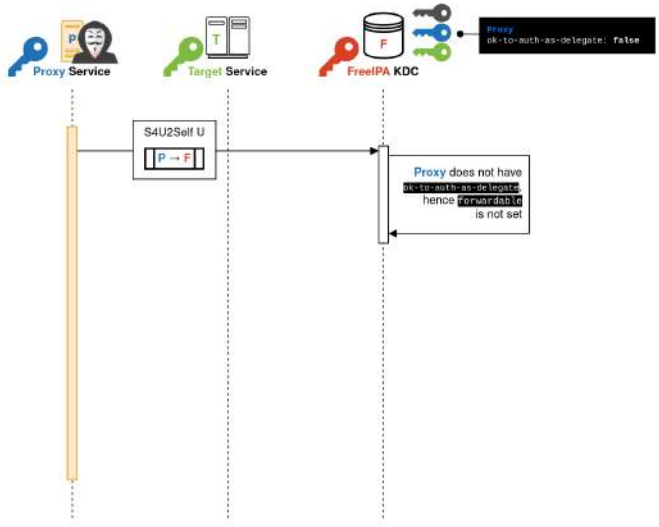
- Using MIT Kerberos 1.18
- PAC generation handled by IPA KDB plugin
- ABI compatibility within major release[16]
  - Update to MIT krb5 1.20 impossible
- PAC ticket signature not backportable[17]

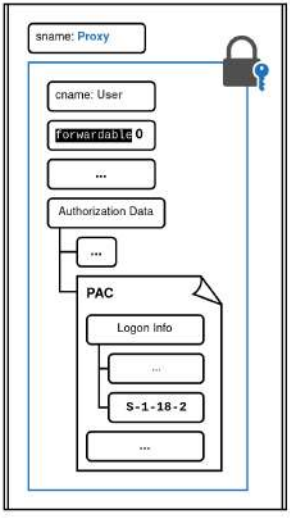
```
krb5_error_code
(*sign_authdata)(krb5_context kcontext,                  unsigned int flags,
                 krb5_const_principal client_princ,      krb5_const_principal server_princ,
                 krb5_db_entry *client,                  krb5_db_entry *server,
                 krb5_db_entry *header_server,           krb5_db_entry *local_tgt,
                 krb5_keyblock *client_key,              krb5_keyblock *server_key,
                 krb5_keyblock *header_key,              krb5_keyblock *local_tgt_key,
                 krb5_keyblock *session_key,             krb5_timestamp authtime,
                 krb5_authdata **tgt_auth_data,          void *ad_info,
                 krb5_data ***auth_indicators,           krb5_authdata ***signed_auth_data);
```

Attack detection

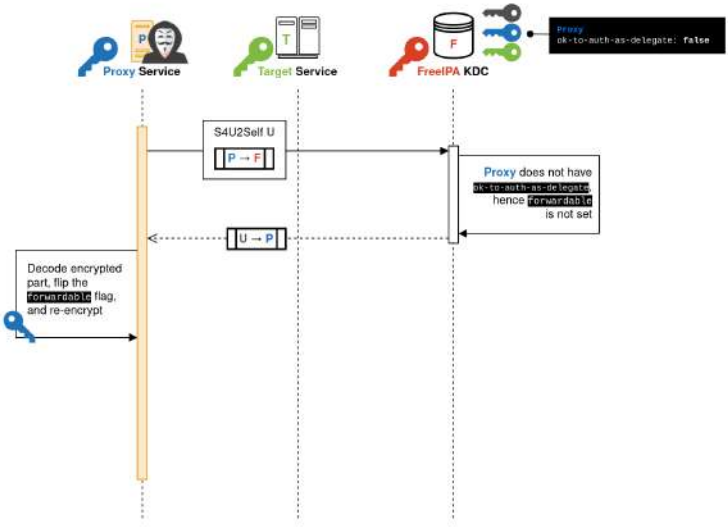
- If the ticket cannot be protected, maybe the KDC could detect the attack
- The PAC contains **additional authorization information**
  - List of SIDs
- *Security identifier* (SID)
  - Identifiers used in the AD world
  - Unique, except for some well-known ones[18]
- Well-known SIDs supported by FreeIPA:
  - **S-1-18-1**: *Authentication authority asserted identity*
    - Ticket obtained using normal user request
  - **S-1-18-2**: *Service asserted identity*[19]
    - Ticket obtained using S4U2Self

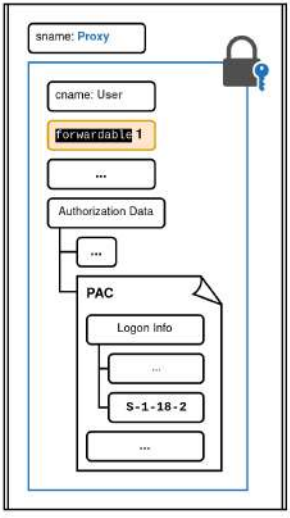
Bronze-Bit attack detection



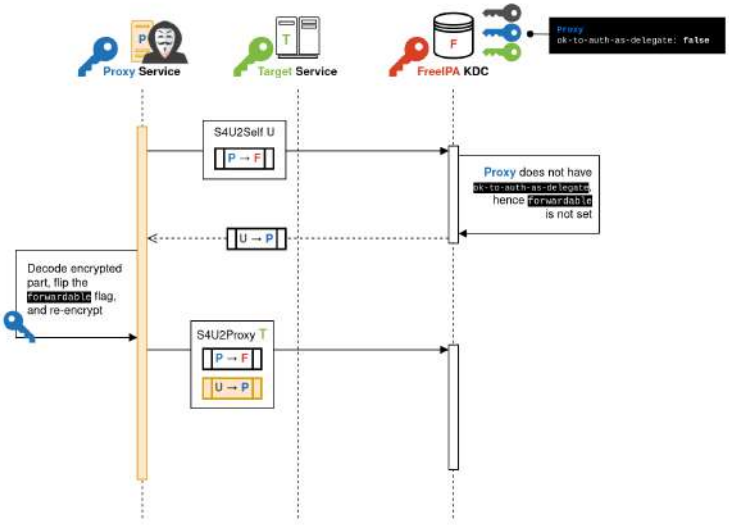


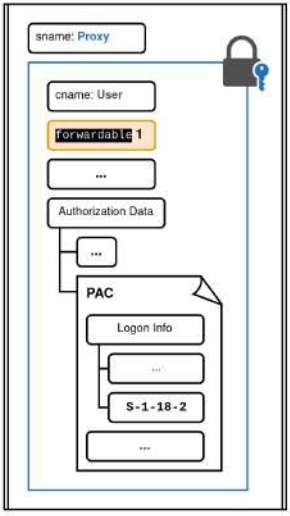
Bronze-Bit attack detection



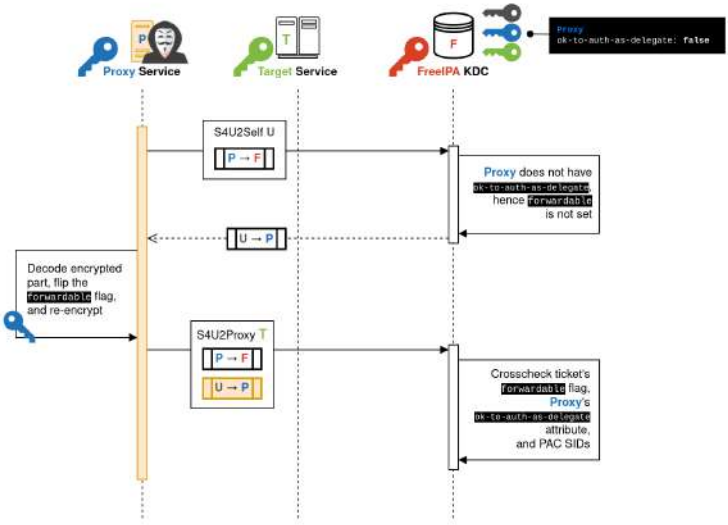


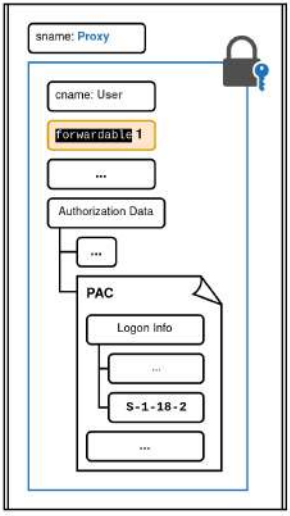
Bronze-Bit attack detection



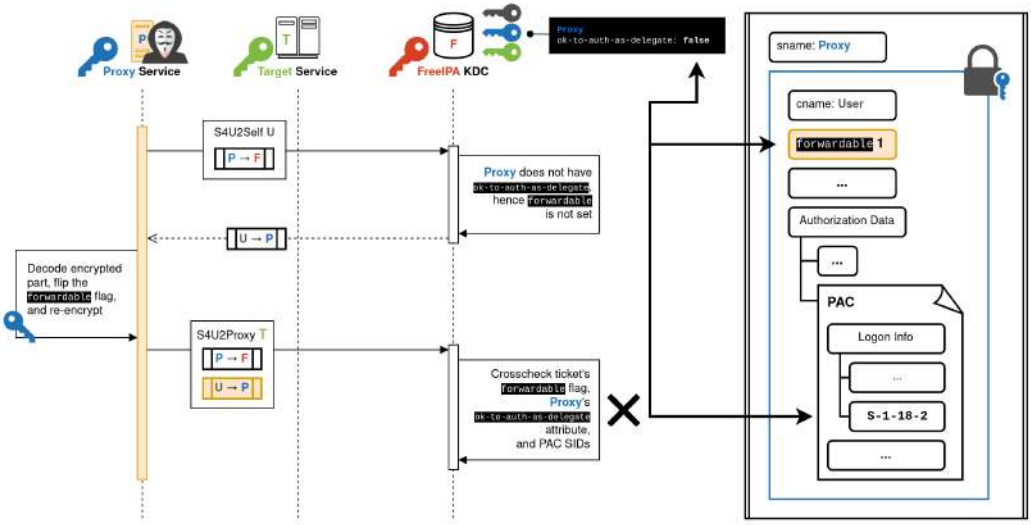


Bronze-Bit attack detection

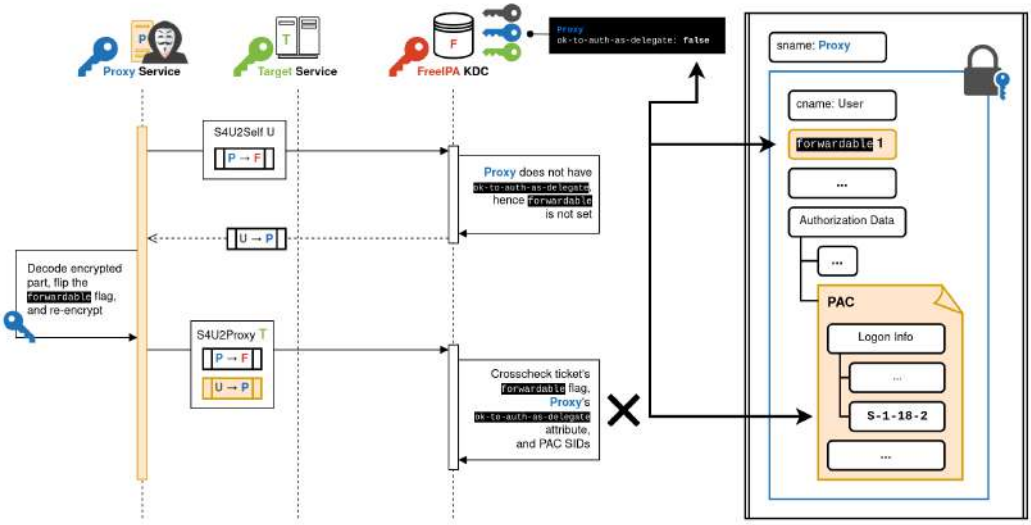




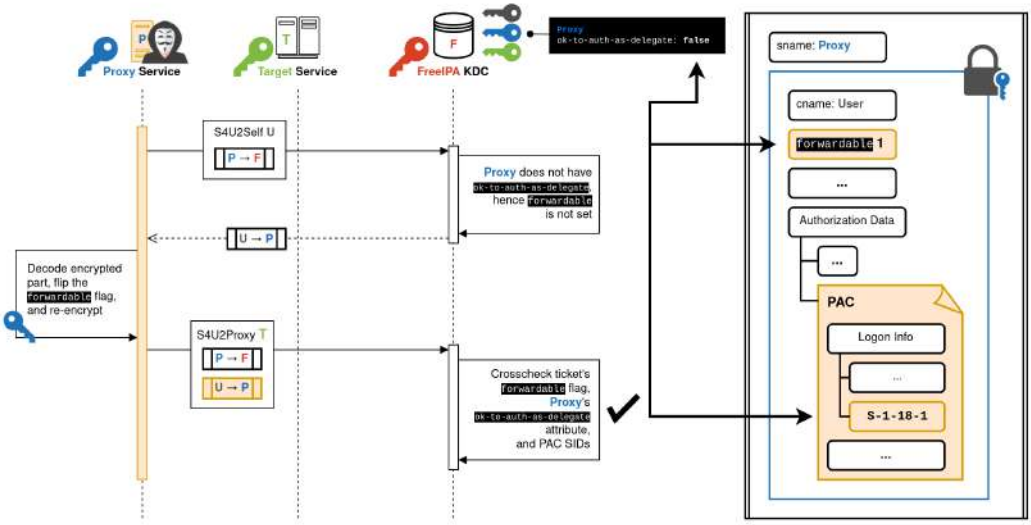
Bronze-Bit attack detection



Bronze-Bit attack detection



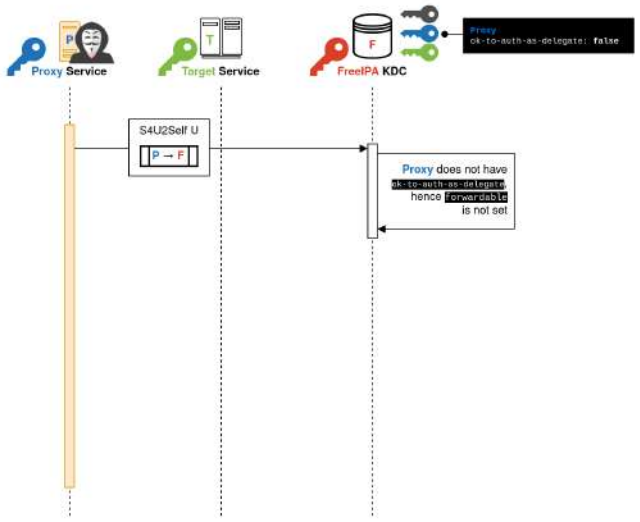
Bronze-Bit attack detection

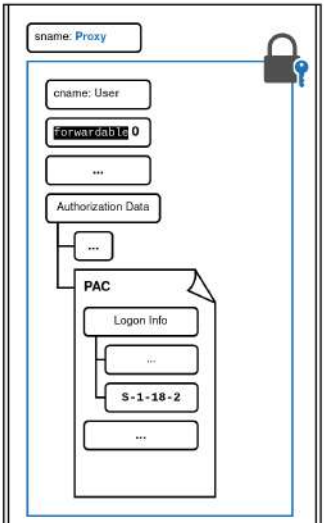


CVE-2022-37967

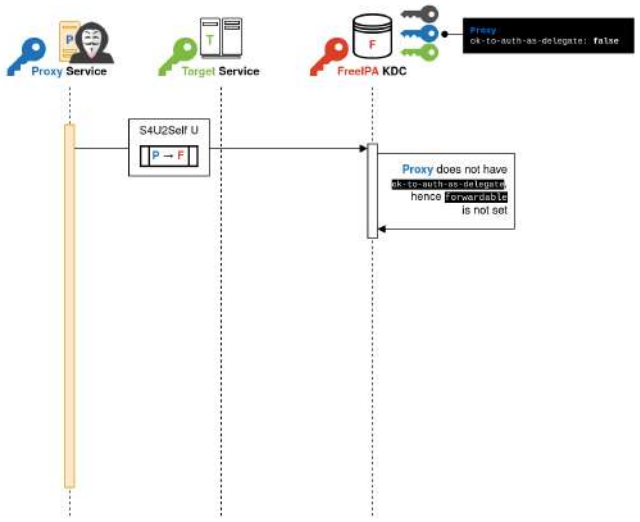
- **PAC spoofing**
  - Authorization information can be modified[20]
- MS-PAC updated to add the **extended KDC signature**[21]
  - Implemented in MIT krb5 as **“full PAC checksum”**[22]

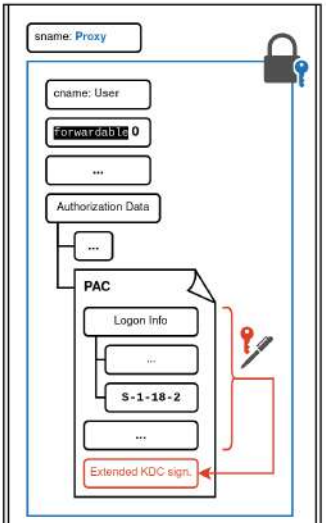
Bronze-Bit attack detection with PAC extended KDC signature



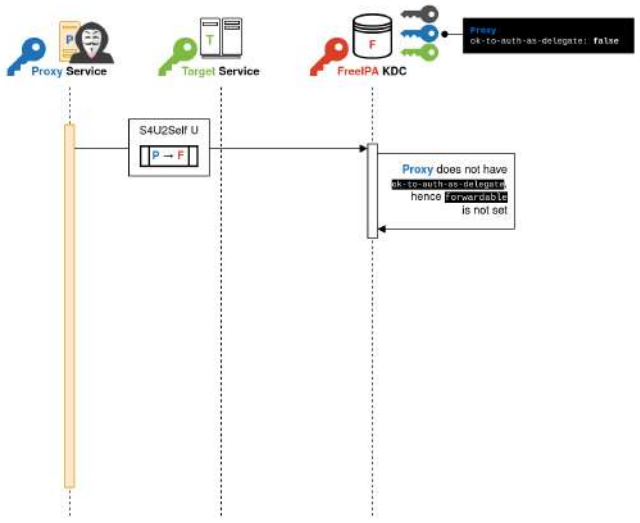


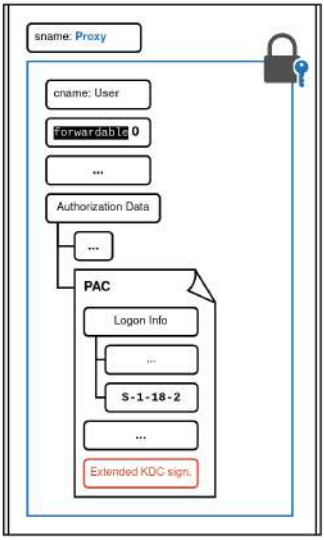
Bronze-Bit attack detection with PAC extended KDC signature



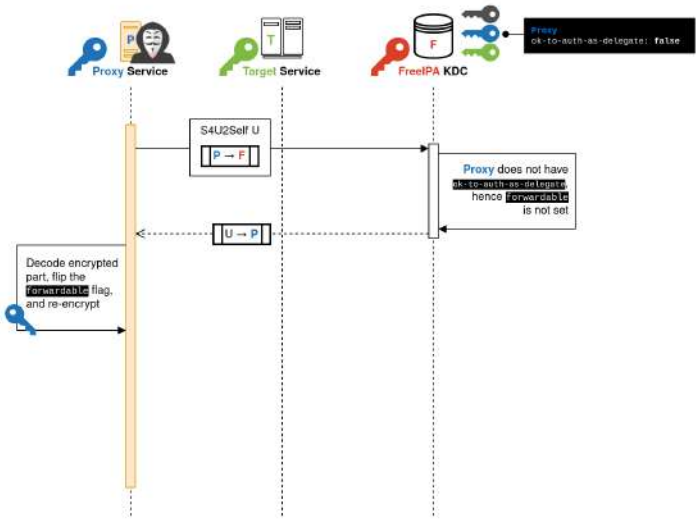


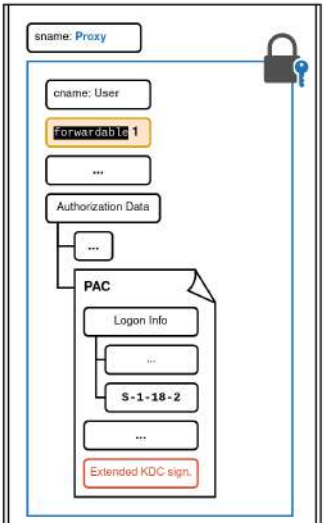
Bronze-Bit attack detection with PAC extended KDC signature



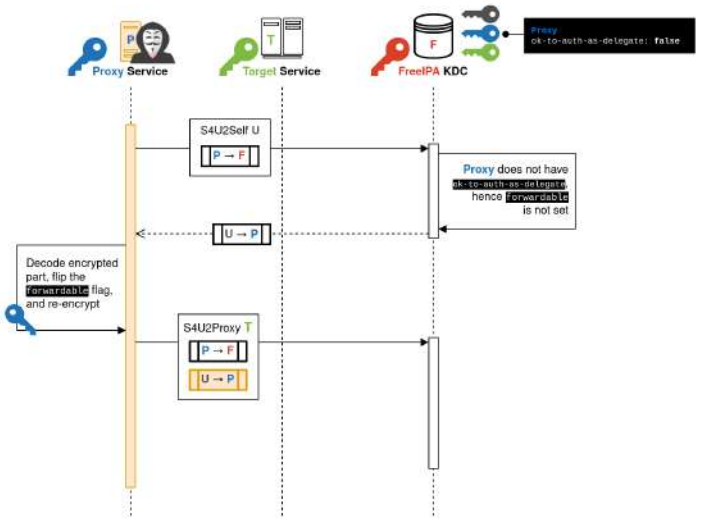


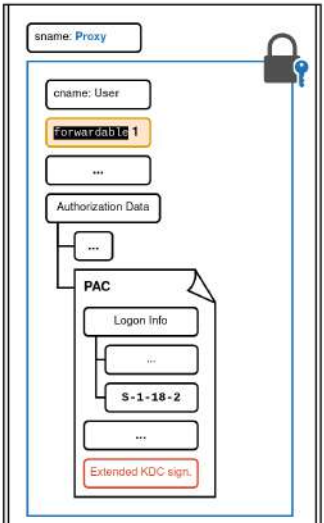
Bronze-Bit attack detection with PAC extended KDC signature



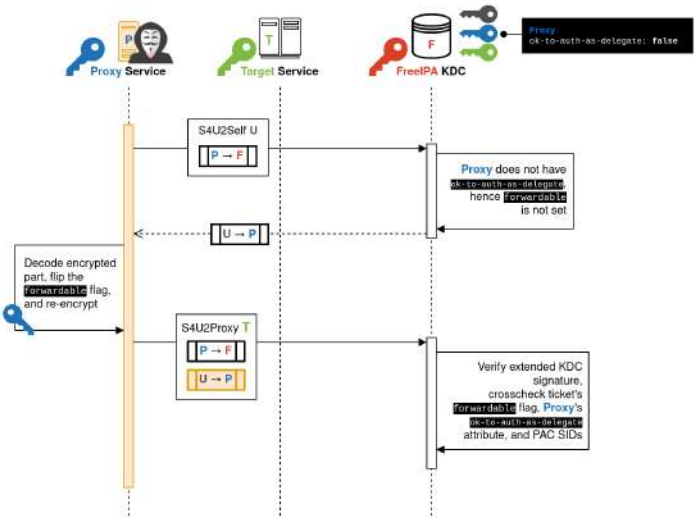


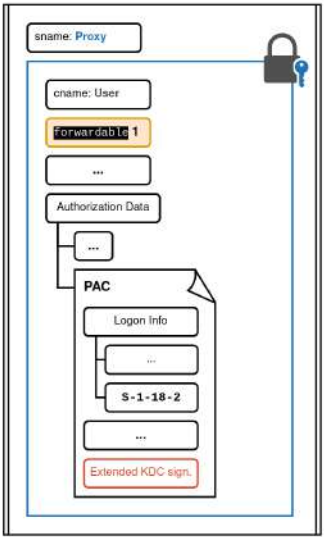
Bronze-Bit attack detection with PAC extended KDC signature



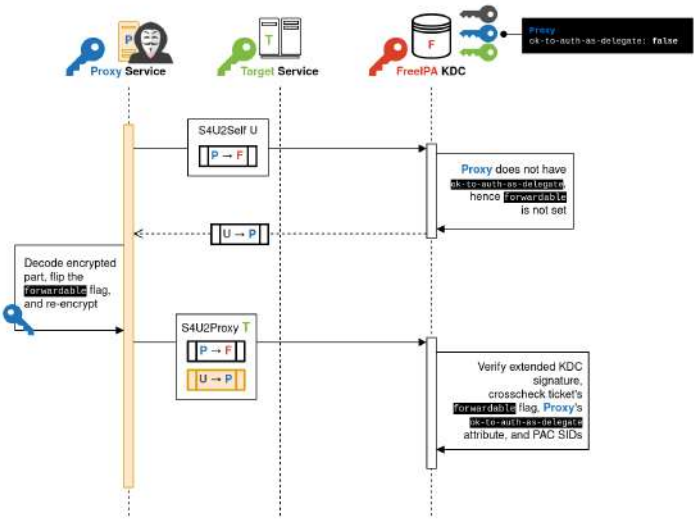


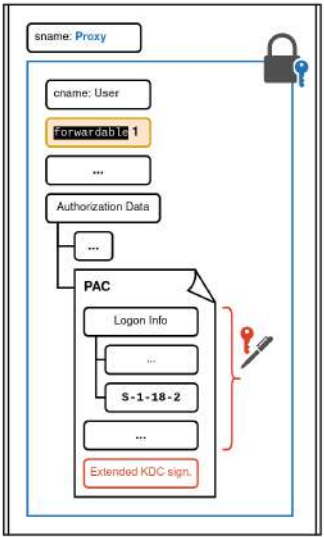
Bronze-Bit attack detection with PAC extended KDC signature



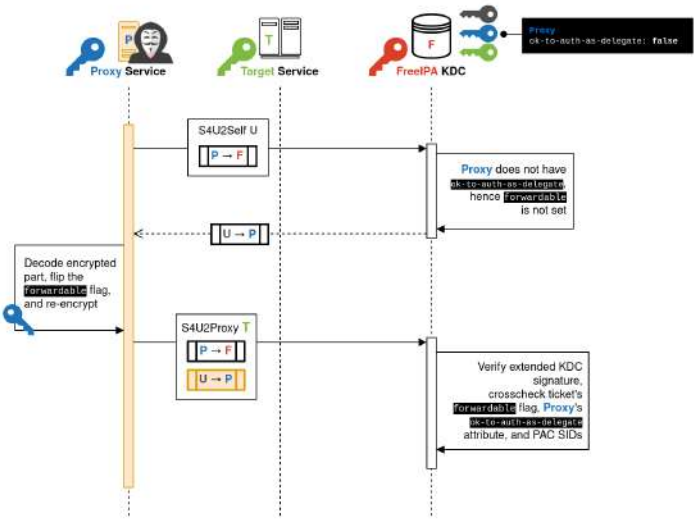


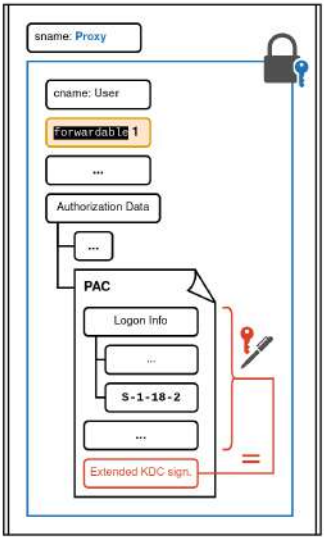
Bronze-Bit attack detection with PAC extended KDC signature



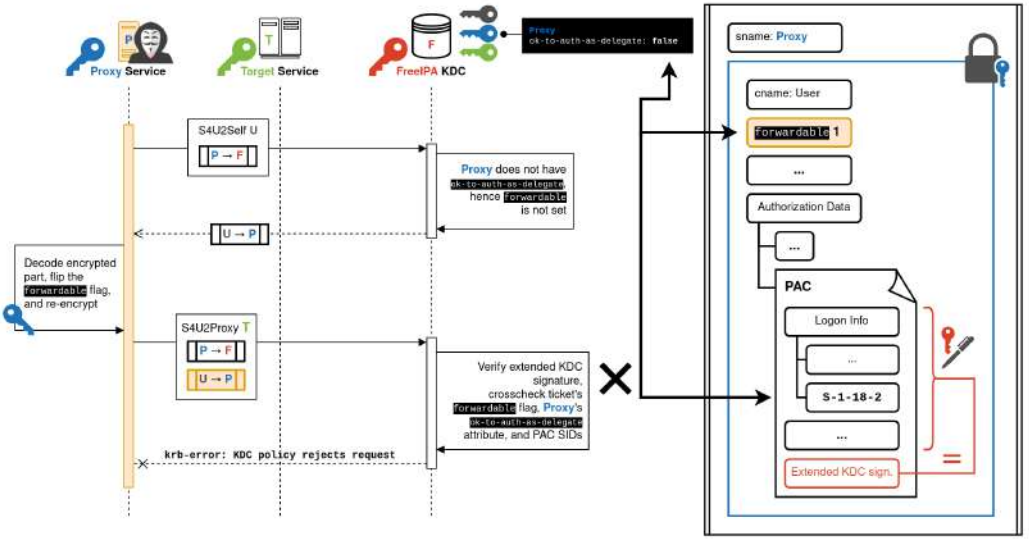


Bronze-Bit attack detection with PAC extended KDC signature





Bronze-Bit attack detection with PAC extended KDC signature



Conclusion

Conclusion

- C8S/RHEL8
  - MIT krb5: **extended KDC signature** support backported[23]
  - FreeIPA: **Bronze-Bit attack detection mechanism** released[24,25,26]
- Limitation: not compatible with cross-realm constrained delegation
  - But RBCD (not supported on RHEL8) required by AD in this case[27]
- Good example of the typical tribulations of **long-term support**
  - Especially for security-related network protocols
- MS-SFU is the continuation of Kerberos' gradual shift
  - From authentication only to **authentication and authorization**

References

1. MIT krb5 plugin modules
   https://web.mit.edu/kerberos/krb5-1.21/doc/plugindev/index.html
2. MS-SFU: Service for User and Constrained Delegation Protocol
   https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-sfu/
3. FreeIPA general constrained delegation
   https://freeipa.readthedocs.io/en/ipa-4-10/designs/rbcd.html#general-constrained-delegation-design
4. [Blog] Kerberos: How does delegation work? (Tarlogic)
   https://www.tarlogic.com/blog/kerberos-iii-how-does-delegation-work/
5. [Blog] Kerberos constrained delegation with protocol transition (Phackt)
   https://phackt.com/en-kerberos-constrained-delegation-with-protocol-transition
6. [Blog] Kerberos Delegation (Hacknodo)
   https://en.hacknodo.com/constrained-unconstrained-delegation/
7. [Blog] Kerberos Constrained Delegation (ired.team)
   https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse/abusing-kerberos-constrained-delegation
8. MS-SFU 3.2.5.1.2: KDC Replies with Service Ticket
   https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-sfu/ad98268b-f75b-42c3-b09b-959282770642
9. MS-SFU 2.2.2: `PA_S4U_X509_USER`
   https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-sfu/cd9d5ca7-ce20-4693-872b-2f5dd41cbff6
10. RFC8009: AES Encryption with HMAC-SHA2 for Kerberos 5
    https://datatracker.ietf.org/doc/html/rfc8009
11. impacket#1684: Implement Kerberos encryption types from RFC8009 (AES HMAC-SHA2 familly)
    https://github.com/fortra/impacket/pull/1684
12. [Blog] Improvements in Windows Kerberos Architecture (Steve Syfuhs)
    https://syfuhs.net/improvements-in-windows-kerberos-architecture
13. MS-PAC 2.8.3: ticket signature
    https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-pac/76c10ef5-de76-44bf-b208-0d8750fc2edd
14. Microsoft KB4598347 update
    https://support.microsoft.com/en-us/topic/kb4598347-managing-deployment-of-kerberos-s4u-changes-for-cve-2020-17049-569d60b7-3267-e2b0-7d9b-e46d770332ab
15. MIT Kerberos upstream pull request for PAC ticket signature
    https://github.com/krb5/krb5/pull/1225
16. RHEL8 Compatibility Levels
    https://access.redhat.com/articles/rhel8-abi-compatibility
17. MIT Kerbreos 1.18.2 KDB plugin API
    https://github.com/krb5/krb5/blob/krb5-1.18.2-final/src/include/krb5/kdcpolicy_plugin.h#L120-L126
18. AD special identity groups
    https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-special-identities-groups
19. Service Asserted Identity SID set by FreeIPA for S4U2Self
    https://github.com/freeipa/freeipa/blob/release-4-9-12/daemons/ipa-kdb/ipa_kdb_mspac.c#L386-L390
20. Kerberos' RC4-HMAC broken in practice: spoofing PACs with MD5 collisions
    https://i.blackhat.com/EU-22/Thursday-Briefings/EU-22-Tervoort-Breaking-Kerberos-RC4-Cipher-and-Spoofing-Windows-PACs-wp.pdf
21. MS-PAC 2.8.4: exteneded KDC signature
    https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-pac/9cf6f6ad-6b76-44b3-aefa-901a1ff5a08
22. MIT Kerberos upstream pull request for PAC extended KDC signature (aka. PAC full checksum)
    https://github.com/krb5/krb5/pull/1284
23. Backport of PAC extended KDC signature support to CentOS 8 Stream
    https://gitlab.com/redhat/centos-stream/rpms/krb5/-/merge_requests/38
24. Bronze-Bit attack detection for FreeIPA
    https://github.com/freeipa/freeipa/commit/a847e2483b4c4832ee5129901da169f4eb0d1392
25. Build conditions for Bronze-Bit attack detection in FreeIPA
    https://github.com/freeipa/freeipa/commit/67ca47ba4092811029eec02f8af9c34ba7662924
26. Bronze-Bit attack detection patch for CentOS 8 Stream
    https://gitlab.com/redhat/centos-stream/rpms/ipa/-/merge_requests/58/
27. FreeIPA constrained delegation use cases
    https://freeipa.readthedocs.io/en/ipa-4-10/designs/rbcd.html#use-cases

Questions?

Thank you!