CVE-2020-17049: Kerberos Bronze Bit Attack – Overview

With the release of Microsoft's patch to fix CVE-2020-17049, I'm excited to share details about this vulnerability and how it could be exploited. This post is only a very high-level overview, and I strongly encourage readers who are interested to check out my follow-up posts which provide much more depth:

- To learn about Kerberos, Kerberos delegation, and the vulnerability behind this attack, please see CVE-2020-17049: Kerberos Bronze Bit Attack – Theory.
- To learn about when the attack could be used, its implementation and to see practical exploit scenarios, please see CVE-2020-17049: Kerberos Bronze Bit Attack – Practical Exploitation.

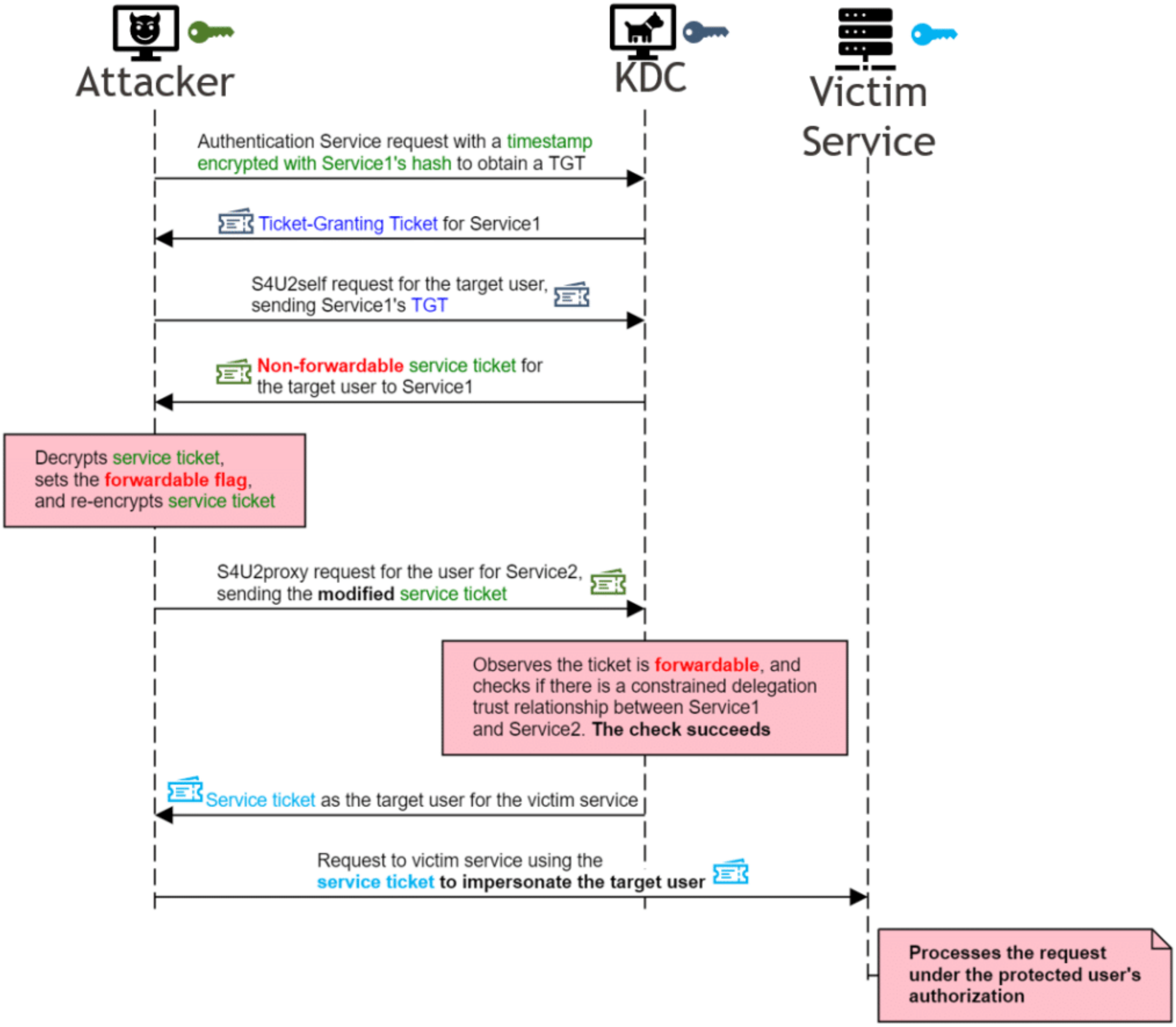
From the attacker's perspective, the exploit requires a few prerequisites:

1. A foothold in the target environment to launch the attack.
2. The password hash of a service account.
3. That service account must be allowed to perform constrained delegation to another service.
    1. This could be classic constrained delegation (with either the "– Use Kerberos only" or the "- Use any authentication protocol" setting).
    2. This could also be resource-based constrained delegation.

With these prerequisites met, the attacker can authenticate to the second service as any user. **This includes members of the Protected Users group and any other users explicitly configured as "sensitive and cannot be delegated."** The second service will accept and process the attacker's requests as if they came from the impersonated user.

This attack uses the S4U2self and S4U2proxy protocols introduced by Microsoft as extensions to the Kerberos protocol used by Active Directory. The attack uses the S4U2self protocol to obtain a service ticket for a targeted user to the compromised service, using the service's password hash.

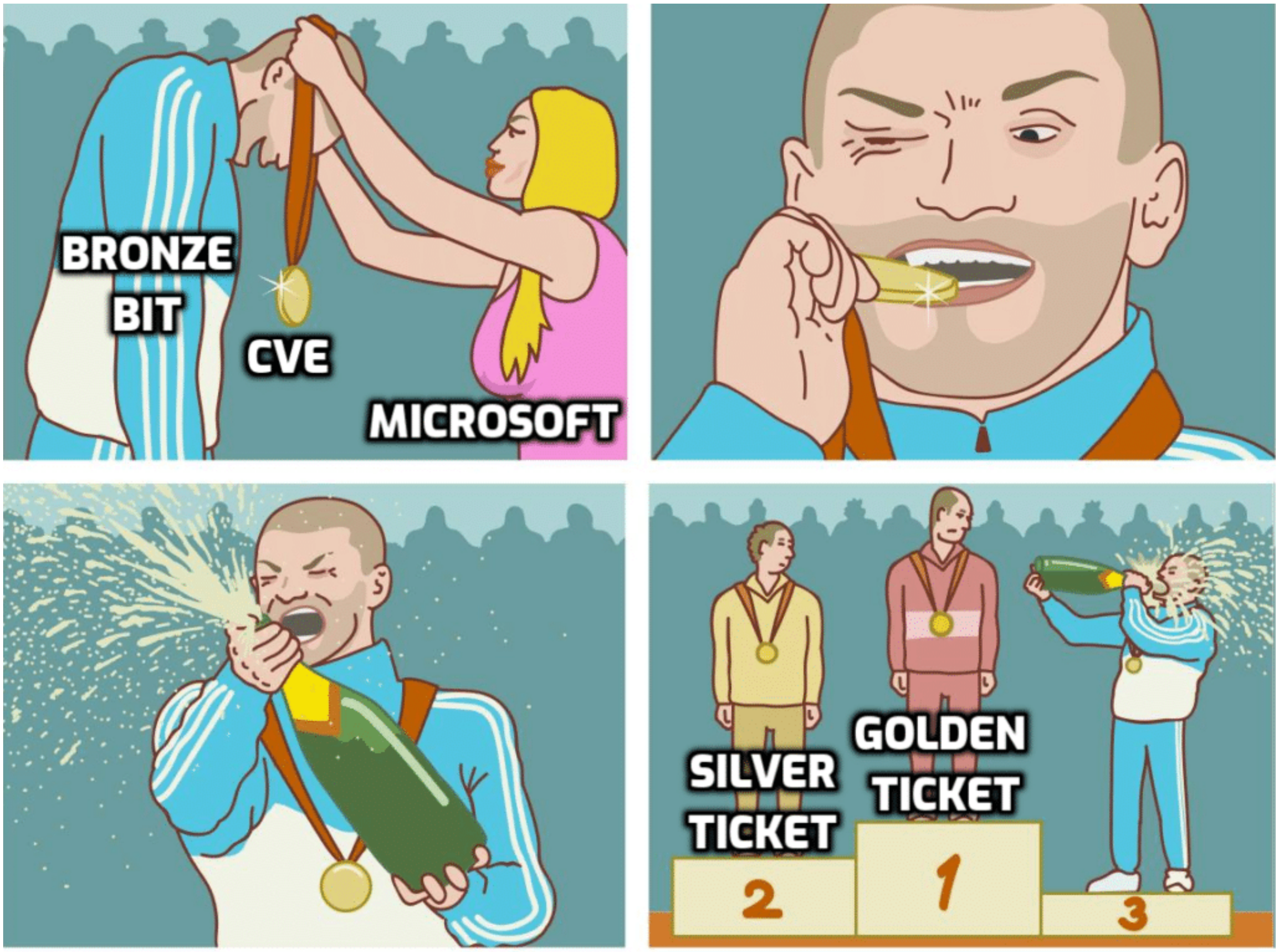
The attack then manipulates this service ticket by ensuring its forwardable flag is set (flipping the "Forwardable" bit to 1). The tampered service ticket is then used in the S4U2proxy protocol to obtain a service ticket for the targeted user to the targeted service. With this final service ticket in hand, the attacker can impersonate the targeted user, send requests to the targeted service, and the requests will be processed under the targeted user's authority.



Attacker
KDC
Victim Service
Authentication Service request with a timestamp encrypted with Service1's hash to obtain a TGT
Ticket-Granting Ticket for Service1
S4U2self request for the target user, sending Service1's TGT
Non-forwardable service ticket for the target user to Service1
Decrypts service ticket, sets the forwardable flag, and re-encrypts service ticket
S4U2proxy request for the user for Service2, sending the modified service ticket
Observes the ticket is forwardable, and checks if there is a constrained delegation trust relationship between Service1 and Service2. The check succeeds
Service ticket as the target user for the victim service
Request to victim service using the service ticket to impersonate the target user
Processes the request under the protected user's authorization

This attack is made possible because the forwardable flag is only protected by encrypting the service ticket with the first service’s password hash. Having already obtained the hash, the attacker is free to decrypt the service ticket, flip the bit to set the forwardable flag, and then re-encrypt the ticket. Unlike the PAC, targeted in the MS14-068 attack, there is no signature in this portion of the ticket to detect tampering.

This exploit bypasses 2 existing protections for Kerberos delegation, and provides an opportunity for impersonation, lateral movement, and privilege escalation. Because this is accomplished by flipping a single bit, and in the spirit of the Golden Ticket and Silver Ticket attacks, I’ve dubbed this the Bronze Bit attack.



If you'd like try the exploit in your own environment, it has been implemented as addition to the Impacket framework with a pull request pending. Of course, this will have to be tested in a controlled environment with an unpatched domain controller. I recommend checking out the Practical Exploitation post for further details on how the exploit can be used.

Of course, any new research is built on the great work of many others. I'd like to thank the following individuals in particular for publishing their own research and for helping me with this finding:

- Benjamin Delpy
- Karl Fosaaen
- Tim Medin
- Sean Metcalf
- Kevin Robertson
- Will Schroeder
- Elad Shamir
- Alberto Solino
- Scott Sutherland