Agenda

» Kerberos 101
» Kerberoasting
» Silver Tickets
» Golden Tickets
» Wrapping up
Agenda

» Kerberos 101
» Kerberoasting
» Silver Tickets
» Golden Tickets
» Wrapping up
Kerberos 101 - Overview

- Authentication protocol for untrusted networks
- Initially Designed by MIT, adapted by Microsoft
- Default authentication protocol for Windows networks (Since Windows 2000)
- Requires valid DNS names (for example, `\\10.10.10.10\share` will fall back to NTLM)
- Kerberos relies on tickets for authentication
- Each ticket is stored in the credential cache on your local machine
Kerberos 101 – Service Principal Names

» A service principal name (SPN) is a unique identifier of a service instance. SPNs are used by Kerberos authentication to associate a service instance with a service logon account. (MSDN)

» Format: `<service class>/<host>:<port>/<service name>`
   » MSSQLSvc/sql.lab.local:1433/SQLEXPRESS
   » CIFS/files.lab.local

» List available SPNs in a domain:
   » setspn.exe -q */*

» Only show MSSQL SPNs
   » setspn.exe -q MSSQLSvc/*
Kerberos 101 - Components

» Client (Principal)
» Server
» Kerberos Distribution Center
  » Authentication Service
  » Ticket Granting Service

» Ticket Granting Tickets
» Service Ticket
Kerberos 101 – Authentication Workflow
Agenda

» Kerberos 101
» Kerberoasting
» Silver Tickets
» Golden Tickets
» Wrapping up
Kerberoasting - Overview

» Initially discovered and disclosed by Tim Medin in 2015 (see References for a link to the talk)
» Goal: Crack weak service passwords
» Cracked passwords can be used for
  » Lateral movement
  » Privilege escalation
  » Persistence
» Mitre ATT&CK [T1208]
Kerberoasting - Details

- Any domain user can request tickets for any service
  - No high privileges required
  - Service must not be active
- SPN scanning to discover service accounts
  - `setspn -q */*`
  - Find-PSServiceAccounts.ps1
- Request service account via powershell
  - Add-Type -AssemblyName System.IdentityModel
- Extract hashes with mimikatz and crack with johntheripper / hashcat
  - `mimikatz.exe "kerberos::list /export"` (convert with kirbi2john.py)
Kerberoasting – Details cont.

» Kerberoasting has since been automated: Invoke-Kerberoast and Rubeus.
» Rubeus: C# tool based off Benjamin Delpys kekeo
  » Written by @harmj0y from Specter Ops
  » Toolkit for attacking kerberos
» Rubeus.exe kerberoast /outfile:hashes.txt
  » yes, that's actually all you need
  » Will search AD for kerberoastable accounts (accounts with >= 1 SPN, Password never expires), request a ticket per account and dump it in hashcat format
» Crack ticket(s) with hashcat
  » ./hashcat64.bin -m 13100 -r hob064.rule hashes.txt rockyou.txt
Kerberoasting - Mitigation

» Set long and complex passwords for service accounts
  » Recommended length: >28 characters
» (Group) Managed Service Accounts
» Limit privileges of service accounts
  » Service accounts should NOT be part of the domain admin group!
» Use AES encryption instead of RC4 encryption
Kerberoasting - Detection

- No default way of detection Kerberoasting, custom detections/alerts are necessary
- Enable “Audit Kerberos Service Ticket Operations" on DC
- Kerberos event titled 4769 – „A Kerberos service ticket was requested.”
- Looking for TGS-REQ packets with RC4 encryption is probably the best method
- High rate of false positives
- Search for users with a high count of event 4769
Agenda

» Kerberos 101
» Kerberoasting
» **Silver Tickets**
» Golden Tickets
» Wrapping up
Silver Tickets - Overview

» Technique to maintain persistence in an already compromised domain
» Goal: Forge service ticket
» Knowledge of the service account or computer account hash required
» Stealthy persistence
» Server does not verify tickets with the KDC
» Mitre ATT&CK T1097 (Pass the ticket)
Silver Tickets - Details

- Password or NTLM hash of service account needed to forge a valid TGS ticket
  - Kerberoasting
  - Credential dumping with mimikatz
- Silver ticket is created directly on a compromised host
  - No TGT required (no AS-REQ / AS-REP)
  - No ticket is requested from the KDC (no TGS-REQ / TGS-REP)
  - Target server does not verify tickets with the KDC
- Create anywhere and used anywhere on the network, without elevated rights.
Silver Tickets – Details cont.

» Creating a silver ticket:
  » mimikatz.exe ,,kerberos::golden /admin:admkevin /id:1107
     /domain:windomain.local /sid:S-1-5-21-539236762-368423896-1554642573
     /target:dc.windomain.local /rc4:4fb8848a7509c605673bc4021c05e74f
     /service:cifs /ptt; exit"
Silver Tickets - Mitigation

- No direct mitigation available
- Protect assets (especially the domain controller)
- Same mitigations as for kerberoastig apply
Silver Tickets - Detection

» Indicators
  » The Account Domain field is blank when it should be DOMAIN.
  » The Account Domain field is DOMAIN FQDN when it should be DOMAIN.

» Events:
  » 4624 Account Logon
  » 4634 Account Logoff
  » 4672 Admin Logon

» Disclaimer: Not a blue teamer. If I overlooked something, let me know!
Agenda

» Kerberos 101
» Kerberoasting
» Silver Tickets
» **Golden Tickets**
» Wrapping up
Golden Tickets - Overview

» Golden Tickets are forged Ticket-Granting Tickets (TGT)
» Require knowledge of the krbtgt password hash
» Mitre ATT&CK T1097 (Pass the ticket)
Golden Tickets - Details

- Golden Ticket requires the KRBTGT password hash.
- Create anywhere and user anywhere on the network, without elevated rights.
- No AS-REQ or AS-REP (steps 1 & 2) communication with the domain controller (KDC)
- Golden ticket is a valid TGT Kerberos ticket (signed with krbtgt password hash)
- Requirements (for mimikatz)
  - Domain Name [AD PowerShell module: (Get-ADDomain).DNSRoot]
  - Domain SID [AD PowerShell module: (Get-ADDomain).DomainSID.Value]
  - Domain KRBTGT Account NTLM password hash
  - UserID for impersonation.
Golden Tickets – Details cont.

» Creating a golden ticket
  » `\mimikatz.exe “kerberos::golden /user:kevin /domain:windomain.local /sid:S-1-5-21-539236762-368423896-1554642573 /krbtgt:4fb8848a7509c605673bc4021c05e74f /ptt” exit`

» The user is added to the domain admin group

» The ticket is automatically added to the local credential cache with the /ptt flag

» To get rid of the golden ticket, the krbtgt account password must be changed twice. Once is not enough as the last two passwords are cached on the DC.
Golden Tickets - Mitigation

- Behavior is working as intended
- No real “fix"
- Protect domain controller and domain admin accounts
- Protect the domain controller and Domain admin account
- The KRBTGT account password is never changed* and the attacker can create Golden Tickets until the KRBTGT password is changed (twice)
- It’s advisable to regularly change the KRBTGT password (
Golden Tickets - Detection

» Hard to detect (ticket expiration is not logged by default)
» MS ATA is able to detect golden tickets
   » Only when actively used!
» Indicators:
   » The Account Domain field is blank when it should be DOMAIN
   » The Account Domain field is DOMAIN FQDN when it should be DOMAIN
» Events
   » 4624 Account Logon
   » 4672 Admin Logon

» Disclaimer: Not a blue teamer. If I overlooked something, let me know!
Agenda

» Kerberos 101
» Kerberoasting
» Silver Tickets
» Golden Tickets
» Wrapping up
Wrappig Up

» Kerberoasting exploits weak passwords and overprovisioned service accounts
» Silver and Golden tickets are stealthy persistence techniques for an already compromised domain
» To mitigate those attacks
  » Service account passwords > 28 characters
  » Minimal privileges for service accounts
  » Protect domain controllers and domain admin accounts
Questions?
THANKS!

Twitter: @kevin0x90
GITHUB: github.com/shellhunter
Sources / References

» https://adsecurity.org/?p=1515
» https://adsecurity.org/?page_id=1821
» https://blogs.technet.microsoft.com/askds/2008/03/06/kerberos-for-the-busy-admin/
» https://www.roquelynn.com/words/explain-like-im-5-kerberos/
» https://www.varonis.com/blog/kerberos-attack-silver-ticket/
» https://www.varonis.com/blog/kerberos-how-to-stop-golden-tickets/
» https://blog.stealthbits.com/extracting-service-account-passwords-with-kerberoasting/
Tools

» Rubeus: https://github.com/GhostPack/Rubeus
» Powersploit: https://github.com/PowerShellMafia/PowerSploit
» Mimikatz: https://github.com/gentilkiwi/mimikatz
» Powershell Empire: https://github.com/EmpireProject/Empire