Search

Active Directory Security

Active Directory & Enterprise Security, Methods to Secure Active Directory, Attack Methods & Effective Defenses, PowerShell, Tech Notes, & Geek Trivia…

Home | About | AD Resources | Attack Defense & Detection | Contact | Mimikatz | Presentations | Schema Versions | Security Resources | SPNs | Top Posts

Detecting Kerberoasting Activity

Sp4rkCon (2017) Talk Slides Posted – Active Directory Security: The Good, the Bad, & the UGLY

FEB
08
2017

Detecting Kerberoasting Activity Part 2 – Creating a Kerberoast Service Account Honeypot

By Sean Metcalf in ActiveDirectorySecurity, Microsoft Security, Technical Reference

In my previous post, "Detecting Kerberoasting Activity" I explain how Kerberoasting works and describe how to detect potential Kerberoasting activity. The trick to this is understanding what activity is normal in order to filter out the noise and increase the fidelity of the alert.

This post describes how to filter from millions of events to a single one to detect Kerberoasting activity.

Detecting Kerberoasting Activity
As I noted in the previous post, looking for TGS service tickets with RC4 encryption was a good method to discover Kerberoasting activity.

> *Windows added Kerberos AES (128 & 256) encryption starting with Windows Server 2008 and Windows Vista which means that*

most Kerberos requests will be AES encrypted with any modern Windows OS. Any Kerberos RC4 tickets requested should be the exception. There are systems that only support Kerberos RC4 by default (NetApp). Inter-forest Kerberos tickets also use RC4 unless configured for AES – ensure your forest trusts support AES and then enable AES over the trust.

Once all Domain Controllers are configured to log 4769 events, these events need to be filtered before sending the data into a SIEM/Splunk. Since we are only really interested in Kerberos TGS service tickets with RC4 encryption, it's possible to filter the events. As shown above, Kerberos events with AES encryption has Ticket Encryption Type set to 0x12.
Kerberos RC4 encrypted tickets have Ticket Encryption Type set to 0x17.

These events can be filtered using the following which greatly reduces the amount of events flowing into the SIEM/Splunk:

- Ticket Options: 0x40810000
- Ticket Encryption: 0x17

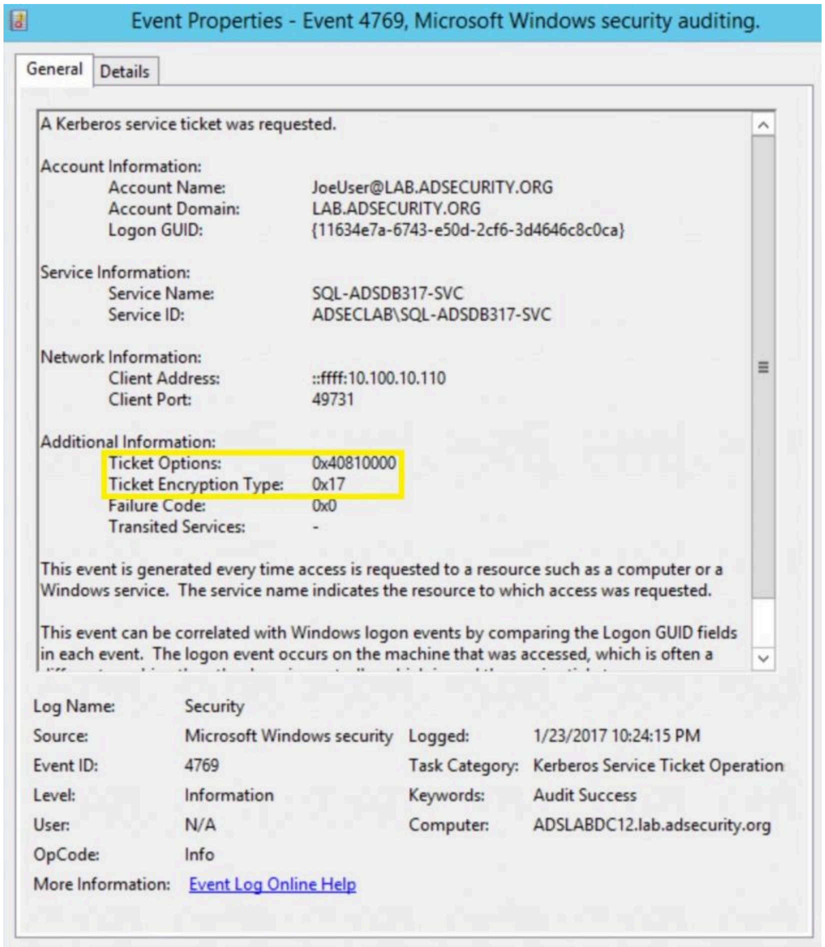
With this information, we can start investigating potential Kerberoasting activity and reduce the number of 4769 events.

We can further reduce the number of 4769 events that flow into SIEM/Splunk:

- Filter out requests from service accounts (ads45service@lab.adsecurity.org)
- Filter on Audit Success
- Filter out requests for service names with a "$" which are typically for computer accounts (or trusts or Managed Service Accounts, all accounts where Windows automatically generates a long, complex password).

In limited testing, I've seen 4769 event totals reduced from millions to thousands and hundreds using these filtering techniques.

Here's a 4769 event that may potentially be from Kerberoasting activity:



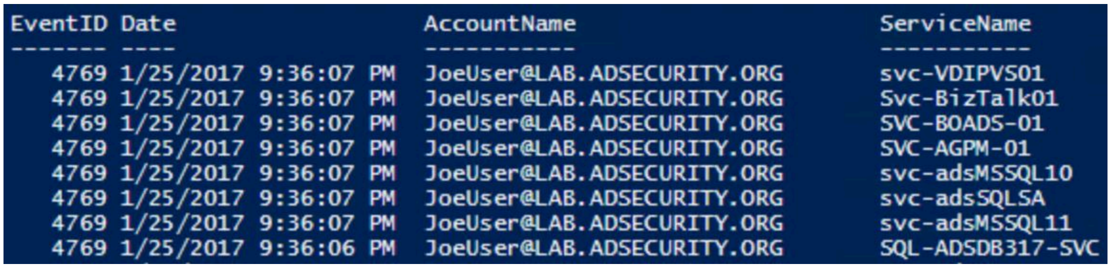
Event Properties - Event 4769, Microsoft Windows security auditing.

```
A Kerberos service ticket was requested.

Account Information:
        Account Name:            JoeUser@LAB.ADSECURITY.ORG
        Account Domain:          LAB.ADSECURITY.ORG
        Logon GUID:              {11634e7a-6743-e50d-2cf6-3d4646c8c0ca}

Service Information:
        Service Name:            SQL-ADSDB317-SVC
        Service ID:              ADSECLAB\SQL-ADSDB317-SVC

Network Information:
        Client Address:          ::ffff:10.100.10.110
        Client Port:             49731

Additional Information:
        Ticket Options:          0x40810000
        Ticket Encryption Type:  0x17
        Failure Code:            0x0
        Transited Services:      -

This event is generated every time access is requested to a resource such as a computer or a
Windows service.  The service name indicates the resource to which access was requested.

This event can be correlated with Windows logon events by comparing the Logon GUID fields
in each event.  The logon event occurs on the machine that was accessed, which is often a
```

```
Log Name:     Security
Source:       Microsoft Windows security   Logged:         1/23/2017 10:24:15 PM
Event ID:     4769                         Task Category:  Kerberos Service Ticket Operation
Level:        Information                  Keywords:       Audit Success
User:         N/A                          Computer:       ADSLABDC12.lab.adsecurity.org
OpCode:       Info
More Information:  Event Log Online Help
```

Following this line of thought, we can look at TGS ticket requests with specific ticket encryption & ticket options to identify potential Kerberoast activity.

Using the information regarding ticket encryption type and ticket options, we can use PowerShell to parse the DC's event log looking for 4769 events with this info.



```
EventID Date                  AccountName                  ServiceName
------- ----                  -----------                  -----------
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG   svc-VDIPVS01
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG   Svc-BizTalk01
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG   SVC-BOADS-01
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG   SVC-AGPM-01
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG   svc-adsMSSQL10
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG   svc-adsSQLSA
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG   svc-adsMSSQL11
   4769 1/25/2017 9:36:06 PM  JoeUser@LAB.ADSECURITY.ORG   SQL-ADSDB317-SVC
```

That looks really odd. Why would any account request several different service names (Citrix PVS, BizTalk, Business Objects, AGPM GPO management, and several SQL service accounts) within a second or two of each other?

That stands out and looks really suspicious and is very likely Kerberoasting activity. This provides great information on what users could be compromised and what activity on which computers should be scrutinized.

A single user requesting RC4 encrypted TGS tickets for several services, such as lots of SQL service principal names is suspicious and it's worth investigating the IP (client address) the requests came from. The same thing is true for multiple RC4 encrypted TGS requests over time for lots of service principal names. A pattern does emerge when there's one or two accounts that request a variety or RC4 TGS tickets.

Read the entire post for more information on how Kerberoasting works, etc.

Filtering the Noise to Find Malicious Activity

I note in that post that 4769 events on Domain Controllers are extremely numerous, some of the most numerous events on a network.

But, what if we only cared about 1 event?

That would reduce the number of 4769 events down to a single event that only occurs when something malicious is happening?

This post describes how to further narrow down to best detect Kerberoasting activity on a network: Creating a Service Account Honeypot to detect Kerberoasting.

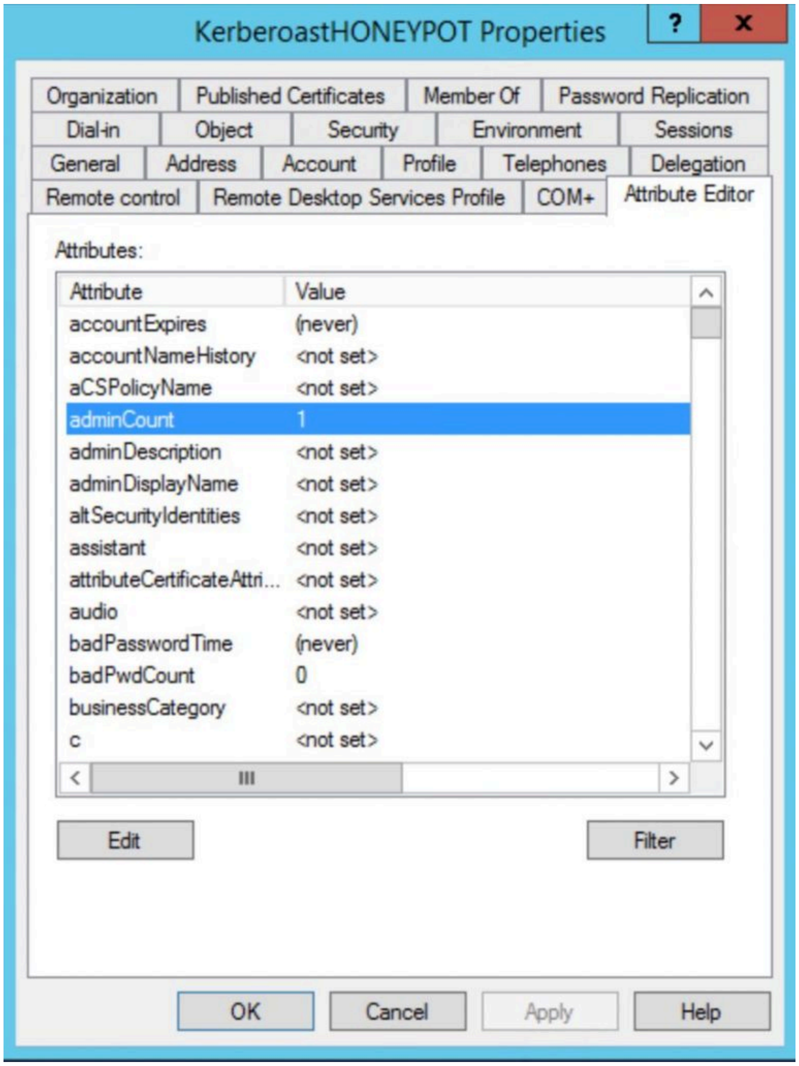
Note: I still recommend filtering 4769 event IDs on Domain Controllers and flowing them into SIEM/Splunk since this will provide information on resources users are accessing as well as help flag when a single user is requesting multiple service principal names (which is suspicious).

Creating a Kerberoast Service Account Honeypot

In order to create a Kerberoast Service Account Honeypot, we need to create a user account in AD and give it a fake service principal name (SPN). It has to be fake so we know that when it's requested, this request is not valid and therefore is malicious. It's also important to make this account look attractive by setting the "AdminCount" attribute to 1 as this flags the account as potentially having elevated AD rights. Adding this account to a bunch of fake groups made to look like it's providing additional elevated rights helps to add to the illusion.

Step 1: Create a new AD user account.

Step 2: Set the "AdminCount" attribute to 1.



Attribute	Value
accountExpires	(never)
accountNameHistory	<not set>
aCSPolicyName	<not set>
adminCount	1
adminDescription	<not set>
adminDisplayName	<not set>
altSecurityIdentities	<not set>
assistant	<not set>
attributeCertificateAttri...	<not set>
audio	<not set>
badPasswordTime	(never)
badPwdCount	0
businessCategory	<not set>
c	<not set>

Step 2: Add a Service Principal Name (SPN) to the account. This SPN needs to be unique, so it should not simply be copied from another system. SQL service accounts are pretty common, so that's not a bad one to use (MSSQLSvc/sql01.lab.adsecurity.org). Just don't reuse one that already exists.
The following example is a bit less… subtle. 😳

Step 3: It may also be useful to add the honeypot account to a fake group that looks like it might have admin rights.

Note:
To make things interesting, you could give this account an easy password that could be guessed, something like “Password1234” (or a keyboard combination). This way we can monitor if someone logs on with this account. However, the Kerberos TGS service ticket request is enough to know that Kerberoasting activity is occurring and we know from which computer it’s being done thanks to the event information (“Client Address”).

Detecting Kerberoast Service Account Honeypot “Access”

Once the honeypot account is created with a service principal name that doesn't do anything and therefore should never be requested or used. The reason this service principal name should never be requested is that we made it up and it isn't linked to any real application that would be requesting it. There is no reason for anyone to every request a Kerberos TGS service ticket for this since there is no actual associated service running for it. Therefore, if we see that someone requested a Kerberos TGS ticket, they are very likely attempting to Kerberoast this account.

An attacker gains access to the internal network and searches for accounts with service principal names and have admincount set to 1.

This shows our new honeypot account to the attacker who is now interested in this account and requests an RC4 encrypted Kerberos TGS ticket for the SPN. A klist shows the attacker got the TGS ticket in memory.

By looking for 4769 events on Domain Controllers with the ticket encryption option 0x12 (along with other filters I describe in the Kerberoast detection post), we can see that Joe User requested a Kerberos ticket for a SPN that doesn't exist and should never be requested!
The Account Name shows which account was used and Client Address provides the computer IP from where the attacker requested the honeypot Kerberos service account.

When we use our Kerberoast discovery PowerShell script against the Domain Controller event logs, we find that Joe User has requested a lot of Kerberos service tickets, including the one for our Honeypot (which again should never be requested since it doesn’t exist).

Using a service account honeypot, this changes this detection from “potential” Kerberoasting activity, to definite Kerberoasting activity.

Note that we can also configure an IDS rule that looks at TGS-REQ packets with the service name “KerberoastHoneyPot” (again, or something more boring and subtle).

Conclusion

Kerberoasting requires requesting Kerberos TGS service tickets with RC4 encryption which shouldn’t be regular activity on a network. Logging 4769 events on Domain Controllers, filtering these events by ticket encryption type (0x17), known service accounts (Account Name field) & computers (Service Name field) greatly reduces the number of events forwarded to the central logging and alerting system. Gathering and monitoring this data also creates a good baseline of what’s “normal” in order to more easily detect anomalous activity.

Detecting Kerberoasting activity is possible by logging the correct activity on Domain Controllers. Determining if this activity is malicious is not requires in-depth knowledge of how RC4 TGS tickets are used in the environment. Creating a service account honeypot with a SPN that doesn’t do anything, provides another data point. Any Kerberos ticket request involving the honeypot service account should be seen as malicious.

Kerberoasting References

- Detecting Kerberoasting Activity (part 1)
- Cracking Kerberos TGS Tickets Using Kerberoast – Exploiting Kerberos to Compromise the Active Directory Domain

- Attack Methods for Gaining Domain Admin Rights in Active Directory
- Sneaky Persistence Active Directory Trick #18: Dropping SPNs on Admin Accounts for Later Kerberoasting
- Targeted Kerberoasting (Harmj0y)
- Kerberoasting without Mimikatz (Harmj0y)
- Roasting AS REPs (Harmj0y)
- Sean Metcalf's Presentations on Active Directory Security
- Kerberoast (GitHub)
- Tim Medin's DerbyCon "Attacking Microsoft Kerberos Kicking the Guard Dog of Hades" presentation in 2014 (slides & video).

(Visited 53,832 times, 2 visits today)

AP-REQ, Audit Kerberos Service Ticket Operations, Detect Kerberoast Activity, Detecting Kerberoast activity, Event ID 4769, Kerberoast honeypot, Kerberoasting activity, Kerberos RC4 Encryption, Kerberos Service Ticket, Kerberos TGS, Kerberos TGS Ticket, NTLM Password, RC4_HMAC_MD5, service account honeypot, TGS-REP, TGS-REQ

Sean Metcalf

I improve security for enterprises around the world working for TrimarcSecurity.com
Read the About page (top left) for information about me. :)
https://adsecurity.org/?page_id=8

RECENT POSTS

BSides Dublin – The Current State of Microsoft Identity Security: Common Security Issues and Misconfigurations – Sean Metcalf

DEFCON 2017: Transcript – Hacking the Cloud

Detecting the Elusive: Active Directory Threat Hunting

Detecting Kerberoasting Activity

Detecting Password Spraying with Security Event Auditing

POPULAR POSTS

PowerShell Encoding & Decoding (Base64)

Attack Methods for Gaining Domain Admin Rights in…

Kerberos & KRBTGT: Active Directory’s…

Finding Passwords in SYSVOL & Exploiting Group…

Securing Domain Controllers to Improve Active…

Securing Windows Workstations: Developing a Secure Baseline

Detecting Kerberoasting Activity

Mimikatz DCSync Usage, Exploitation, and Detection

Scanning for Active Directory Privileges &…

Microsoft LAPS Security & Active Directory LAPS…

CATEGORIES

- ActiveDirectorySecurity
- Apple Security
- Cloud Security
- Continuing Education
- Entertainment
- Exploit
- Hacking
- Hardware Security
- Hypervisor Security
- Linux/Unix Security
- Malware
- Microsoft Security
- Mitigation
- Network/System Security
- PowerShell
- RealWorld
- Security
- Security Conference Presentation/Video
- Security Recommendation
- Technical Article
- Technical Reading
- Technical Reference

TheCloud

Vulnerability

TAGS

ActiveDirectory Active Directory Active Directory Security ActiveDirectorySecurity ADReading AD Security ADSecurity Azure AzureAD DCSync DomainController GoldenTicket GroupPolicy HyperV Invoke-Mimikatz KB3011780 KDC Kerberos KerberosHacking KRBTGT LAPS LSASS MCM MicrosoftEMET MicrosoftWindows mimikatz MS14068 PassTheHash PowerShell PowerShellCode PowerShellHacking PowerShellv5 PowerSploit Presentation Security SilverTicket SneakyADPersistence SPN TGS TGT Windows7 Windows10 WindowsServer2008R2 WindowsServer2012 WindowsServer2012R2

Search

RECENT POSTS

BSides Dublin – The Current State of Microsoft Identity Security: Common Security Issues and Misconfigurations – Sean Metcalf

DEFCON 2017: Transcript – Hacking the Cloud

Detecting the Elusive: Active Directory Threat Hunting

Detecting Kerberoasting Activity

Detecting Password Spraying with Security Event Auditing

RECENT COMMENTS

Derek on Attacking Read-Only Domain Controllers (RODCs) to Own Active Directory

Sean Metcalf on Securing Microsoft Active Directory Federation Server (ADFS)

Brad on Securing Microsoft Active Directory Federation Server (ADFS)

Joonas on Gathering AD Data with the Active Directory PowerShell Module

Sean Metcalf on Gathering AD Data with the Active Directory PowerShell Module

ARCHIVES

June 2024

May 2024

May 2020

January 2020

August 2019

March 2019

February 2019

October 2018

August 2018

May 2018

January 2018

November 2017

August 2017

June 2017

May 2017

February 2017

January 2017

November 2016

October 2016

September 2016

August 2016

July 2016

June 2016

April 2016

March 2016

February 2016

January 2016

December 2015

November 2015

October 2015

September 2015

August 2015

July 2015

June 2015

May 2015

April 2015

March 2015

February 2015

January 2015

December 2014

November 2014

October 2014

September 2014

August 2014

July 2014

June 2014

May 2014

April 2014

March 2014

February 2014

July 2013

November 2012

March 2012

February 2012

CATEGORIES

ActiveDirectorySecurity

Apple Security

Cloud Security

Continuing Education

Entertainment

Exploit

Hacking

Hardware Security

Hypervisor Security

Linux/Unix Security

Malware

Microsoft Security

Mitigation

Network/System Security

PowerShell

RealWorld

Security

Security Conference Presentation/Video

Security Recommendation

Technical Article

Technical Reading

Technical Reference

TheCloud

Vulnerability

META

Log in

Entries feed

Comments feed

WordPress.org

COPYRIGHT

Content Disclaimer: This blog and its contents are provided "AS IS" with no warranties, and they confer no rights. Script samples are provided for informational purposes only and no guarantee is provided as to functionality or suitability. The views shared on this blog reflect those of the authors and do not represent the views of any companies mentioned. Content Ownership: All content posted here is intellectual work and under the current law, the poster owns the copyright of the article. Terms of Use Copyright © 2011 - 2020.

Content Disclaimer: This blog and its contents are provided "AS IS" with no warranties, and they confer no rights. Script samples are provided for informational purposes only and no guarantee is provided as to functionality or suitability. The views shared on this blog reflect those of the authors and do not represent the views of any companies mentioned.