BUY NOW

Kerberoast Attack Techniques

In this blog we will focus on Kerberoast attack techniques (Old Technique and New Technique).

NOV 1, 2017 EST READ TIME: 5 MIN

COBALT

In this blog we will focus on Kerberoast attack techniques (Old Technique and New Technique). I will try to cover the basics about Kerberos protocol and then we will see the attacking techniques from a penetration testing perspective.

What is Kerberos?

Kerberos is designed to provide authentication of user identity in a networked computing environment consisting of workstations and servers.

Kerberos Defined

Kerberos is an exploitation attack that extracts service account credentials with a combination of weak encryption and poor service account passwords.

Kerberos in a Nutshell

The Kerberos authentication system is built on top of tickets served by KDC. The core idea behind Kerberos is that the users don't share acc[BUY NOW]ords to each service they want to use. Instead, they share a ticket which they get from KDC.

When a user first starts using the system, they will use their password to get a master ticket called a TGT (**ticket-granting ticket**). This master ticket expires in 25 hours, after which, the user will need to enter the password again to get another one.

When the user needs service access, that uses Kerberos, they will show that master ticket (TGT) to the Kerberos server and get a ticket specifically for that service. Then, shows the ticket just for that service to the service to prove who you are.

Steps in Kerberos Authentication

- Password converted to **NTLM hash**, a timestamp is encrypted with the hash and sent to the KDC as an authenticator in the authentication ticket (TGT) request (AS-Request).
- The **Domain Controller** (KDC) checks user information & creates **Ticket-Granting Ticket** (TGT).
- The TGT is encrypted, signed, & delivered to the user (AS-Reply). Only the Kerberos service (**KRBTGT**) in the domain can open and read TGT data.
- The User presents the TGT to the DC when requesting a **Ticket Granting Service** (TGS) ticket (TGS-Request). The data in the TGT is effectively copied to create the TGS ticket.

- The TGS is encrypted using the target service accounts' NTLM password hash and sent to the user (TGS-Reply).
- The user connects to the server hosting the service on the appropriate port & presents the TGS. The service opens the TGS ticket using its NTLM password hash.

Kerberos Attacks:

There are several different types of Kerberos attacks ranging from recon (SPN Scanning), to offline service account password cracking (Kerberoast), to persistence (Silver & Golden Tickets).

Here are the most popular AD Kerberos attacks:

1. ***SPN Scanning*** *– finding services by requesting service principal names of a specific SPN class/type.*
2. ***Silver Ticket*** *– forged Kerberos TGS service ticket*
3. ***Golden Ticket*** *– forged Kerberos TGT authentication ticket*
4. ***MS14–068 Forged PAC Exploit*** *– exploitation of the Kerberos vulnerability on Domain Controllers.*

Now, let's see how we can leverage the Kerberos implementation to our advantage.

Old Technique

We will see and understand the old technique first (i.e. SPN Scanning and then cracking the tickets).

In general, we follow the process below:

- Enumerate the domain accounts with SPNs set- either with GetUserSPNS.ps1 script from PowerView's or Impacket's "GetUserSPN.py".
- Request TGSs for these specific SPNs with the built-in Windows tool setspn.exe.
- Extract these tickets from memory by invoking the kerberos::list /export Mimikatz command, with the optional base64 export format set first. The tickets were then downloaded, or the base64-encoded versions pulled down to the attacker's machine and decoded. (Note: You don't need admin rights to execute the command :))
- Begin offline password cracking with "tgsrepcrack.py", or John whit the help for kirbi2john.py.

Let's see the Demo :)

We can scan the services with windows built-in utility. I have used in-built utility (i.e setspn.exe).

"setspn.exe" output

Now, if you notice we have "CN= Computers" and "CN=Users" for listed service accounts. We will be focusing on "CN=Users" as these are user generated and so we can try to crack :).

Now that we know the service accounts which we will be cracking or trying to crack, let's go ahead and request Kerberos tickets for specific service accounts.

Command: *Add-Type -AssemblyName System.IdentityModel New-ObjectSystem.IdentityModel.Tokens.KerberosRequestorSecurityToken –ArgumentList "SPN Name"*

Powershell Command (Non Admin User)

Now, we have tickets in memory. We will use Mimikatz to export the tickets from memory. This is one of the down side of this method as you are running Mimikatz this might trigger Alert or this can be detected by AV's.

Note: You can also load Mimikatz into memory using PowerShell "IEX (New-Object Net.WebClient).DownloadString" feature)

Command: *Invoke-Pwc -Command '"kerberos::list /export" exit"'*

Export Ticket from Memory

Extracted Tickets

We have successfully extracted the tickets from memory. Can we crack these tickets?? There are multiple ways to try this. Let's see how we can leverage tgsrepcrack.py form Kerberoast toolkit.

1 > Using Kerberosast: Tgsrepcrack.py

We have provided the wordlist to crack the kirbi file

Command: *C:\Users\pratik\Desktop\kerberoast>python tgsrepcrack.py dict.txt "Ticket.kirbi"*

Cracked Ticket

:) Cracked

2> Convert .kirbi file to John the Ripper format

Now, we will use John the Ripper to crack the tickets. We know that tickets are in kirbi format so first we will convert the ticket to John the Ripper format. We can use Kerberoast (kirbi2john.py) for the same.

John the Ripper format

***Command**:./john –format=krb5tgs crack_file — wordlist=dict.txt*

Cracked using John the Ripper

Cracked :)

New Technique

HarmJ0y has written a good blog on kerberoasting without Mimikatz. This technique is pretty straight forward and simpler than the old technique :)

What you need is “Invoke-Kerberoast.ps1” and then you are good to go :) To crack the tickets, first import “.ps1” module.

This will request the associated TGS Tickets in john or hashcat crackable format :)

Invoke-Kerberoast

Crack the tickets using John the Ripper

Cracked using John the Ripper

Back to Blog

About Cobalt

Cobalt combines talent and technology to provide end-to-end offensive security solutions that enable organizations to remediate risk across a dynamically changing attack surface. As the innovators of Pentest as a Service (PtaaS), Cobalt empowers businesses to optimize their existing resources, access an on-demand community of trusted security experts, expedite remediation cycles, and share real-time updates and progress with internal teams to mitigate future risk.

MORE BY COBALT →

Related readings

Blog

A Pentester's Guide to Code Injection

READ MORE →

Blog

4 Simple Steps to Protect Your Organization from Ransomware Attacks

READ MORE →

Blog

Dangers of Ransomware through File-Sharing Software

READ MORE →

SEE MORE

BUY NOW

SCHEDULE A DEMO

CONTACT

PLATFORM

- Cobalt Platform
- Offensive Security
- PtaaS
- Pricing

SERVICES

- Application Security
- Application Pentest
- Network Security
- Cloud Security
- Brand Protection
- Device Security

COMPANY

- About
- Leadership
- Core Community
- Careers
- Partners

HELPFUL LINKS

- Product Documentation
- Resource Library
- Blog
- Events & Webinars
- Vulnerability Wiki
- Trust Center