

Title:

Multiple Parties, Single Sign On: How Kerberos Ties Everything Together

Abstract:

Lab2 consisted of getting familiar with the Kerberos system for Single-Sign-On, a feature that is prevalent in society today for logging into any computer and accessing a server without configuration of the machine. The lab allowed us to create a server-client relationship and authenticate a pseudo user using Kerberos with authentication tickets.

Aim:

The aim of lab 2 in Seng360 was to gain hands-on experience of how single-sign works and how Kerberos plays a main role in its function. Additionally, we were able to become familiar with authentication and how tickets are maintained within this system.

Intro and Background:

Single-Sign-On currently plays a massive role in our society. Numerous computers and servers rely on this system in order for users to be able to log in to servers from any computer they wish. It provides such flexibility that work can be completed without the need to configure a particular computer to log a user into a system. Kerberos is what allows this authentication for a user to sign into a computer they may have never seen before. It is due to their authentication that a user's data can remain safe while logging into a system. Therefore, it is important for us, the


software engineers, to understand how it works so that it can remain functional and provide this useful service for the foreseeable future. This lab dives deeper into the inner workings of Kerberos.

Method:

Lab 2 made use of two virtual machines running at the same time in order to show how Kerberos functions. As a result, VM cloud users, such as myself, had to pair up with another lab member to complete this assignment. The two VMs allowed for an admin server and example client to be initialized and analyzed.

Furthermore, after configuring the server we were able to add principals (unique user identities) and authenticate an example user (with domain user@SENG360.com) with the client-server relationship and sending tickets to the server.

Screenshot 1:



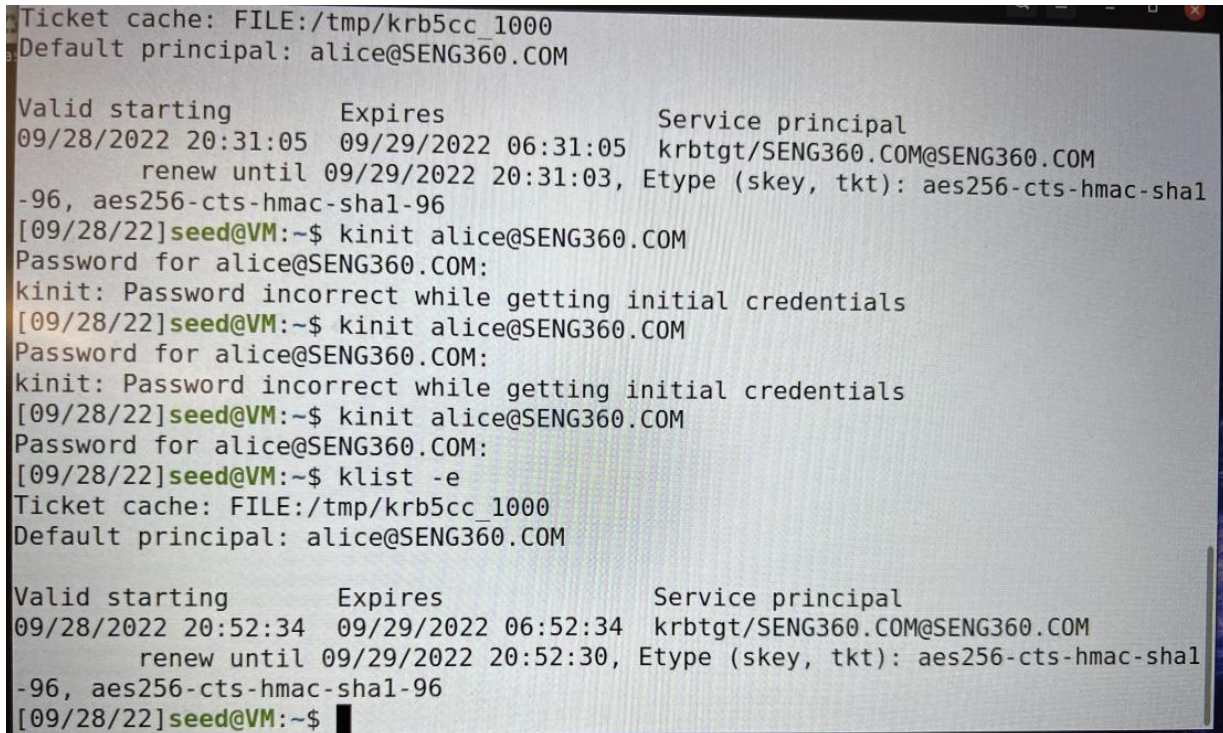
```
[09/28/22]seed@VM: ~/Desktop$ ifconfig -a
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:fe:6d:0b:ad txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.5 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::6e09:e03e:d07f:76e4 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:3d:39:83 txqueuelen 1000 (Ethernet)
    RX packets 31 bytes 4723 (4.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 85 bytes 9370 (9.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 140 bytes 12031 (12.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 140 bytes 12031 (12.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[09/28/22]seed@VM: ~/Desktop$
```

Screenshot 2:



```
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: alice@SENG360.COM

Valid starting      Expires           Service principal
09/28/2022 20:31:05 09/29/2022 06:31:05 krbtgt/SENG360.COM@SENG360.COM
    renew until 09/29/2022 20:31:03, Etype (skey, tkt): aes256-cts-hmac-sha1
-96, aes256-cts-hmac-sha1-96
[09/28/22]seed@VM:~$ kinit alice@SENG360.COM
Password for alice@SENG360.COM:
kinit: Password incorrect while getting initial credentials
[09/28/22]seed@VM:~$ kinit alice@SENG360.COM
Password for alice@SENG360.COM:
kinit: Password incorrect while getting initial credentials
[09/28/22]seed@VM:~$ kinit alice@SENG360.COM
Password for alice@SENG360.COM:
[09/28/22]seed@VM:~$ klist -e
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: alice@SENG360.COM

Valid starting      Expires           Service principal
09/28/2022 20:52:34 09/29/2022 06:52:34 krbtgt/SENG360.COM@SENG360.COM
    renew until 09/29/2022 20:52:30, Etype (skey, tkt): aes256-cts-hmac-sha1
-96, aes256-cts-hmac-sha1-96
[09/28/22]seed@VM:~$
```

Results:

We were able to authenticate our created username to the server by using Kerberos single sign on and establishing the server and client machines. While exploring the server-client relationship with more VMs was not possible, since our laptop could not handle the production of over 2 VMs in our environment. However, it was clear to see how authentication would function with the creation of multiple tickets and users with multiple machines. The 3 questions were answered below.

Q1 What is the purpose of the host file?

The host file can alter who and how a user is accepted for authentication by domain name. Therefore, if an attacker were to gain access to these host files, they could know the user's data when they attempt to authenticate themselves and could also change the rules for authentication. Additionally, they may be able to sign in to the Kerberos server which could cause a data breach in the system.

Q2 What does the max_life and max_renewable_life mean?

From the kadmin man pages: the max_life determines the maximum ticket life for the principal and the max_renewable_life determines the maximum amount of time of tickets for the principal. Tickets have these two lifetimes, after the max_life runs out, the ticket can no longer be used, however, one could use the renewable_life to extend this ticket lifetime. Therefore, one could extend the time for allowed authentication when using altering and using these fields. This is good for freshness, since continually creating new tickets may increase security, however users may want to renew their ticket when accessing the server for longer amounts of time. [1]

Q3 How does the ticket start and expiration time relate to the settings you observed earlier?

The ticket start and expiration time determines when one will have to re-enter their credentials. Therefore, if one would want to stay logged on after the expiration time, they could renew their ticket life. This could be harmful in the hands of an attacker, who could continually renew a user's ticket to access their credentials without having to re-enter their username and password, which they may not have access to.

References

[1] tickets, L. and Mishra, A., 2022. *Lifetime of Kerberos tickets*. [online] Stack Overflow. Available at:

<<https://stackoverflow.com/questions/14682153/lifetime-of-kerberos-tickets>>

[Accessed 30 September 2022].

[2] 2022. [online] Available at:

<<https://www.simplilearn.com/what-is-kerberos-article>> [Accessed 30 September 2022].