Shairoze Malik *et al.LGU (IJECI)* 2020

LGU (IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

LGU International Journal for
Electronic Crime Investigation

Research Article

Vol. 5 issue 4 Year 2021

The Secrets to MIMIKATZ - The Credential Dumper

Shairoze Malik and Erej Azeem
DFRSC - Digital Forensic Research and Service Centre,
Lahore Garrison University - Main Campus, Sector C DHA Phase 6,
Lahore – Pakistan
shairozemalik@lgu.edu.pk, erejazeem00@gmail.com

Abstract:

With the emergence of many credential dumping tools, Mimikatz has become an exceedingly dramatic tool against Windows users that allows the intruders to fetch plain text passwords. Moreover they also target memory to dump password hashes. Mimikatz capacity and potential will be briefly discussed throughout the paper. Several modules of Mimikatz to dump credentials will follow, and the paper will conclude with procedures and techniques that may be used as prevention against Mimikatz attacks that are performed.

Keywords: Mimikatz, hash dump, lsass, modules, kerberos, tickets, krbtg

1. Introduction

This well-known tool known to be as Mimikatz was initially derived in 2011 by Benjamin Delpy also known as *gentilkiwi* in, they proofed and confirmed how the protocols used for authentication of Microsoft were at severe risk to attack. Attackers make use of the vulnerability on Windows System to access internal storage. We can say that he, Benjamin Delpy- the creator of Mimikatz tool created one of the most widely are far apart used and downloaded hacker tools. There are a lot of modules provided by Mimikatz to gather and use Windows credential on targeted systemswhich basically includes recapture of passwords in clear text, LM Hashes (LAN Manager), NTLM Hashes (New Technology LAN Manager), Kerberos tickets [1] which include Golden Kerberos Ticket and Silver Kerberos Ticket. From Windows XP and forward, Mimikatz run on all Windows versions.

2. Concepts to be known

2.1 Active Directory

Active Directory that is also known as AD is a Microsoft's directory service. It is more like a database that runs on Windows server. The use of Active Directory is that is allows administrators to manage the access to resources I.e. network resources and also allows the administrators to manage the user permissions. Everything in Active Directory is stored as an object [2], whereas by object we mean any single element which could be a group, a user, an application or even a device. As said earlier, Active directory stores everything on network as objects allowing the information to be searched and used easier for the administrator.

2.2 NTDS.dit

We would not be wrong if we say that NTDS.dit fie is the heart of Active Directory that stores user accounts. It is basically a (.dit) file that is being used by AD. Because it stores the Active Directory data it is sometimes referred as a database. It comprises of information about the user-objects and multiple attributes are possessed by user objects, groups and also the group membership. Moreover, password hashes for all users in the domain are also stored in this NTDS.dit file [3].

Both LAN Manager (LM) hash and NT hash of the passwords are generated by Windows. Local Security Accounts Manager well known to be SAM database (C:\Windows\System32\config\SAM file) or in Active Directory AD (C:\Windows\NTDS\NTDS.dit file) [4] are basically the locations where these hashes are stored.

2.3 Kerberos

A well-known computer security protocol known as Kerberos authenticates the service requests across a network between two or more hosts, more likely to be across the Internet [5]. A secret key cryptography is used and also a trusted third party for client server authentication and the verification of user's identities. Kerberos is completely designed to avoid locally storing passwords and to send to send any passwords through the internet providing mutual authentication. By this we simply mean that both the server's and user's authenticity is being verified.

2.4 LSASS and LSA

LSASS is the abbreviation of Local Security Authority Server Service and LSA stands for Local Security Authority. Within Windows, the LSASS.exe is an executable for preserving and saving user credentials in memory both for internal (local) and domain users. LSASS is merely an implementation of Local Security Authority. "LSA" is the idea and a conception, whereas "lsass.exe" is a procedure and an action that is implementing many of the LSA functions.

2.5 KDC

A well-known mechanism in cryptography that is a key distribution center (KDC), is in charge for giving keys to users in a network that shares sensitive or confidential data. In a network whenever a connection is established each time, KDC is requested by both for distinctive password generation which is then used for verification by end system users. KDC (a.k.a) Key Distribution Centre is a type of symmetric encryption that allows the access of even more than two systems in a network by bringing about a distinctive ticket type key for shielded connection formation over which data and information is shared or spread out, moved and even transferred. Before the communication takes place, Key Distribution Centre is the main server which is called upon. Key Distribution Centre is typically used in compact grid or networks when link requests do not overwhelm the system due to its central architecture. Instead of standard key encryption, Key Distribution Centre is used because every time the connection is requested, the key is generated each time, which decreases the possibilities of attack.

3. Mimikatz Attack Capabilities

3.1 Pass-the-Hash attack

A well-known tool for extracting hashes and passwords from memory; Mimikatz performs Pass-the-Hash attack from Active Directory user account. In other words, NTLM hash that is used by Windows to deliver passwords is obtained by Pass-the-Hash attack.

3.2 Pass-the-Ticket attack

For impersonating a user on Active Directory Domain there is a method that is well known that is called as Pass-the-Ticket. It fragments and cracks the Kerberos protocol. A Kerberos ticket is obtained for a user account and that can be used to login as that particular user on any other system.

3.3 Kerberos Golden Ticket attack

The encryption of authentication tickets is handled by a hidden root account known as KRBTGT. It is the default Microsoft Windows authentication protocol, Kerberos, which is implemented by Microsoft. The ticket for this hidden root account KRBTGT is obtained through a Kerberos Golden Ticket attack. In a Golden Ticket assault, hackers bypass the KDC and create their own TGTs to obtain access to various resources. A successful Golden Ticket assault grants the hacker near-unrestricted access to your domain's entire infrastructure, including all machines, files, directories, and domain controllers (DCs). They have the ability to impersonate anyone and perform almost any task.

3.4 Kerberos Silver Ticket attack

A faked service authentication ticket is known as a Silver Ticket. Silver Ticket is quite similar to Golden Ticket in that it involves manipulating the Kerberos protocol to gain the hacked system's credentials. An attacker using Silver Ticket can only falsify ticket-granting service (TGS) tickets for certain services. Silver Tickets are more difficult to detect than Golden Tickets because there is no connection between the service and the DC, and any logging is local to the targeted system. Hackers can use a pass-the-ticket technique to raise their access or utilize the service's privileges to gain greater access if they have a Silver Ticket.

3.5 Pass-the-Key attack

Because Pass the Hash is a combination of Pass the Hash and Pass the Ticket, it's also known as Over Pass the Hash Attack. Another variation of pass-the-hash, however this time a unique key is supplied to impersonate a user obtained from a domain controller. To put it another way, it gets a unique key that a user can use to authenticate to a domain controller. This key can be used by the attacker to impersonate the user.

3.6 Pass-the-Cache

A very well-known attack known as Pass-the-Cache attack is somewhat similar to the attack that is previously discussed in this paper called Pass-the-Ticket attack. In this case the login data that is saved, stored and encrypted is used on system that could be MAC, UNIX or even LINUX systems.

4. Mimikatz Modules

4.1 PROCESS Module:

This module of Mimikatz tool is used for dealing with the Windows processes. This module can also be used for process injection [6] and also for the parent process spoofing [7].

Export*:* This command in the Mimikatz tool lists all the functions exported from Dynamic Link Libraries, which are files that contain code for commonly used programs that each running process uses. If a process ID, often known as /PID, is not given, the results displayed are mimikatz.exe exports.

Import: this command is used for listing all the functions imported from the Dynamic Link Libraries each running process is using. If a process ID that is also known as /PID is not specified, then the results displayed are the imports of mimikatz.exe.

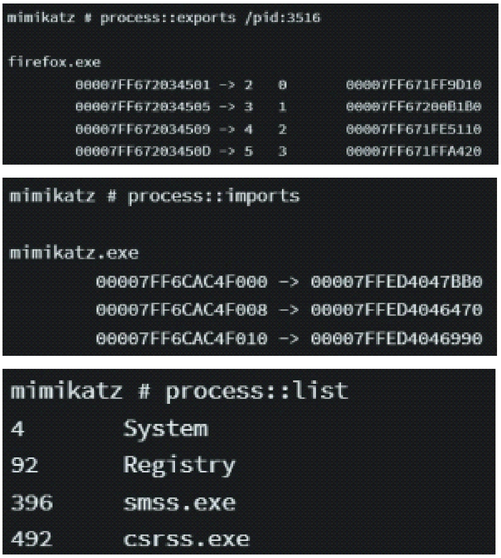
List: This command is used for listing all running processes.

Commands:

PROCESS::*export*

PROCESS::*import*

PROCESS::*list*



```
mimikatz # process::exports /pid:3516

firefox.exe
        00007FF672034501 -> 2   0       00007FF671FF9D10
        00007FF672034505 -> 3   1       00007FF67200B1B0
        00007FF672034509 -> 4   2       00007FF671FE5110
        00007FF67203450D -> 5   3       00007FF671FFA420
```

```
mimikatz # process::imports

mimikatz.exe
        00007FF6CAC4F000 -> 00007FFED4047BB0
        00007FF6CAC4F008 -> 00007FFED4046470
        00007FF6CAC4F010 -> 00007FFED4046990
```

```
mimikatz # process::list
4       System
92      Registry
396     smss.exe
492     csrss.exe
```

Figure 1: mimikatz tool showing the use of export, import and list commands.

4.2 STANDARD Module:

In Mimikatz, the Standard module commands can be typed unaccompanied by the module name i.e. STANDARD. The purpose of Standard module do not need to be prefixed with "standard" [8]. They can easily be requested at first hand from the Mimikatz.

Commands:

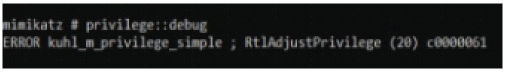
Log: This command is used for journaling steps and operations or we can even say actions and then just easily taking down the logs that is the time-stamped documentation.

Module privilege: This command contains some accommodation to work with privileges while functioning and running with Mimikatz.

Sleep: In Mimikatz, this command switches to sleep mode within the particular defined seconds.

base64: This command in the tool show or exchange the condition or state of input or output to the Base64.

Cd: Current directory is displayed or changed by this specific command.

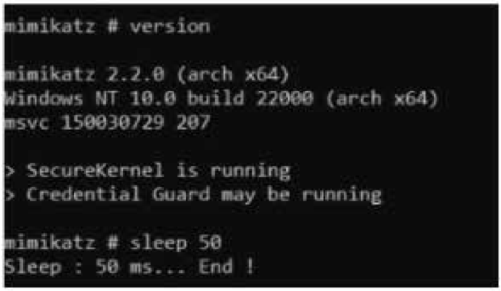


```
mimikatz # privilege::debug
ERROR kuhl_m_privilege_simple ; RtlAdjustPrivilege (20) c0000061
```

Figure 2: privilege::debug when not in root

ERROR:

kuhl_m_privilege_simple; RtlAdjustPrivilege (20) c0000061 this error exactly means that the client or user does not hold the required privileges that is mostly the case when you are not the administrator.



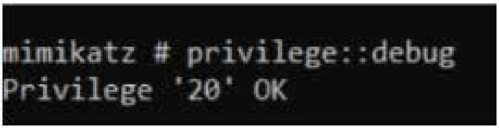
```
mimikatz # version

mimikatz 2.2.0 (arch x64)
Windows NT 10.0 build 22000 (arch x64)
msvc 150030729 207

> SecureKernel is running
> Credential Guard may be running

mimikatz # sleep 50
Sleep : 50 ms... End !
```

Figure 3 : Standard module commands

Running as Administrator:

Command: privilege::*debug*

We get the following output as shown in figure:



```
mimikatz # privilege::debug
Privilege '20' OK
```

Figure 4: privilege::debug when in root

4.3 NET Module:

NET module functionality are very much similar to the Windows net commands.

Alias: This command displays much of the information about the memberships (local groups) also including the Remote Desktop Users.

Group: This command of NET module displays the local or internal groups.

If*:* If lists the available hostnames and the local IP addresses.

Serverinfo: serverinfo command will exhibit data about the logged in server.

Session: lists the active running sessions.

Share: available shares are displayed by this share command of NET module

Commands:

NET::*group*

NET::*if*

NET::*serverinfo*

NET:*:session*

NET::*share*

```
mimikatz # NET::group
```

Figure 5: How to utilize NET module commands

4.4 CRYPTO Module:

The CRYPTO Mimikatz module comes up with modern potential and capacity to interface with Windows cryptography functions (CryptoAPI) [9] i.e. Cryptography Application Programming Interface.

Capi**:** This command patches Crypto API layer for the purpose of easy export.

Certificates: This command of Crypto module of Mimikatz tool is used to list and export certificate.

Hash: This command provides hash of a password with optional username that is being provided.

Keys: Key container are listed and exported by this key command.

Providers: providers of cryptography are listed by this command.

Stores: This command lists the cryptographic stores.

Commands:

CRYPTO::*CAPI*

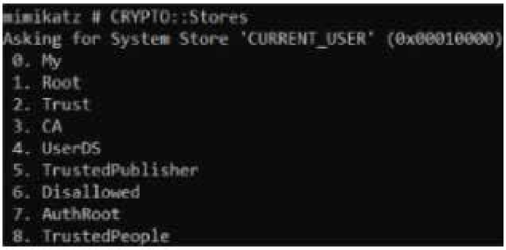
CRYPTO::*Certificates*

CRYPTO::*Hash*

CRYPTO::*Providers*

CRYPTO::*Stores*

CRYPTO::*Keys*



```
mimikatz # CRYPTO::Stores
Asking for System Store 'CURRENT_USER' (0x00010000)
 0. My
 1. Root
 2. Trust
 3. CA
 4. UserDS
 5. TrustedPublisher
 6. Disallowed
 7. AuthRoot
 8. TrustedPeople
```

Figure 6: Listing down the cryptography stores

4.5 DPAPI Module:

The (DPAPI) that stands for Data Protection API that basically helps in data protection. The DPAPI, Data Protection API Mimikatz module comes up with the proficiency for pulling out

Windows stored and saved (and even protected) credential data using DPAPI module. DPAPI is the Windows official method to preserve or encrypt local or internal data that are usually the passwords. A DPAPI blob [10] is a non-transparent binary structure, containing the private data of the applications that are encrypted using DPAPI.

Blob**:** This specific command is used to unprotect a DPAPI blob with a master key or application programming interface.

Chrome/in:"%localappdata%\Google\Chrome\UserData\Default\LoginData" /unprotect*:* This command of DPAPI module dumps credentials that are stored and also the cookies from the Chrome browser. ***Protect*** – This command protects data using DPAPI.

Commands:

DPAPI::*blob*

DPAPI::*chrome/in:"%localappdata%\Google\Chrome\UserData\Default\LoginData"/unprotect*

DPAPI::*protect*

```
mimikatz # DPAPI::Blob
mimikatz # DPAPI::Protect
```

Figure 7: DPAPI blob and protect commands

4.6 SEKURLSA Module:

Dumping the passwords from the memory is possible using the SEKURLSA module of Mimikatz tool. To use the commands of the SEKURLSA module we need to have root permissions or else we will face errors.

Logonpasswords*:* This command displays a list of all accessible provider credentials.

Krbtgt**:** This command helps in getting the password data of Domain Kerberos service account (KRBTGT).

Kerberos: For all the authenticated and verified users. this command lists Kerberos credentials.

Tickets /export: Kerberos tickets of all sessions are exported and listed by this command.

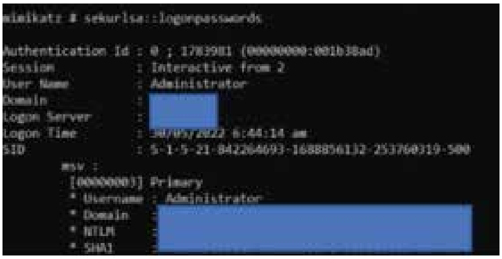
Commands:

SEKURLSA::*Logonpasswords*

SEKURLSA::*krbtgt*

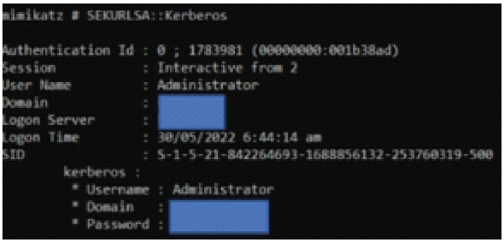
SEKURLSA::*kerberos*

SEKURLSA::*tickets /export*



```
imikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 1783981 (00000000:001b38ad)
Session           : Interactive from 2
User Name         : Administrator
Domain            : [redacted]
Logon Server      :
Logon Time        : 30/05/2022 6:44:14 am
SID               : S-1-5-21-842264693-1688856132-253760319-500
        msv :
         [00000003] Primary
         * Username : Administrator
         * Domain   : [redacted]
         * NTLM     :
         * SHA1     :
```

Figure 8: Recently logged on user credentials



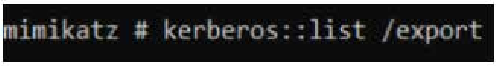
```
imikatz # SEKURLSA::Kerberos

Authentication Id : 0 ; 1783981 (00000000:001b38ad)
Session           : Interactive from 2
User Name         : Administrator
Domain            : [redacted]
Logon Server      :
Logon Time        : 30/05/2022 6:44:14 am
SID               : S-1-5-21-842264693-1688856132-253760319-500
        kerberos :
         * Username : Administrator
         * Domain   : [redacted]
         * Password :
```

Figure 9: Listing kerberos credentials

4.7 KERBEROS Module:

It is of great interest that he Kerberos module is capable of being used without the need of any privilege or Admin rights. Microsoft Kerberos API is involved and produces and generates the Golden tickets.

Tgt: Information about the current session is displayed by this command of Kerberos module of Mimikatz tool. ***List /export:*** The Kerberos Tickets are listed by this command.



```
mimikatz # kerberos::list /export
```

Figure 10: Listing Kerberos Tickets

5. Famous Attacks Performed Using Mimikatz

5.1 Notpetya Cyber Attack

Notpetya a well-known malware whose prime target was Ukraine left multinational companies with many knocked down computer systems. In Windows devices the remaining credentials that are stored in RAM were the target of this malware called Notpetya [11]. This attack was only possible when accounts were logged on at the run time because that is when LSASS memory loads the credentials which are then hijacked by the Mimikatz tool. Notpetya spread escalated through the patched and not patched devices both without mercy with the union of Mimikatz and Eternal Blue [12]. Credentials from patched machines were withdrawn with Mimikatz whereas credentials from not patched devices were pulled out with Eternal Blue.

5.2 Bad Rabbit Ransomware

A type of Crypto Virus known as Bad Rabbit was mainly designed for a specific purpose that is to lock or encrypt files through Drive-by-Attack [13] where not secure web pages are compromised. Using the JavaScript, the Bad Rabbit ransomware is inoculated into the websites HTML code. The infected file is somehow downloaded and looks safe. It does not infects the computer system unless and until the file is opened of executed [14]. It will lock the computer showing its ransomware capacity once it is clicked. This ransomware will then gather credentials using the credential dumper well known tool called Mimikatz.

6. Defensive Approach Against Mimikatz Attacks

6.1 Updating Your Operating System On Your Windows Server

Updating your operating system on your windows server will provide more security against the Mimikatz attacks. So this prevention suggested should be in the to-do list. Many corporations security features are not available unless and until the functional level of Active Directory a.k.a AD is not updated to some present-day category version.

6.2 Debug Rights For Local Administrators To Be Disabled On All Servers And Workstations

To bypass many of the local protections, Windows has a mode known as “Debug Mode”. The main purpose of this Debug Mode relates to the troubleshooting process mainly to be named as troubleshooting the device drivers and many more. In newer Operating Systems, this is a default setting and to simply disable this particular setting mode in MS Configuration, navigate to the boot/Advanced options and disable “Debug”.

6.3 In Active Directory Disable The Storage Of Plain Text Passwords And Password Caching

Password decryption was back in time decided to be enabled for users which maybe sadly still remains with us. In addition, if no Domain Controller [15] is accessible, Windows will cache the up to the minute number of authentications, which includes the password hashes.

6.4 Change The KRBTGT Account Password

Administrators of Active Directory neglect changing passwords of the KRBTGT account that provides the Golden Ticket to the attackers providing the admin rights. There is one more thing to take in consideration about the Active Directory that it preserver the past and current passwords both so therefore, changing the passwords twice would be a best consideration, ensuring that the two password resets are completely synchronized in Active Directory AD.

7. Conclusion

The exceptionally impressive tool mainly invented for penetration testers now being also used by the Cyber criminals for malicious motives - Mimikatz, attacks pertinent Windows systems to gain privilege and dump credentials from the memory. This paper has covered many sections including the attack capabilities and modules of Mimikatz followed by the Prevention steps to be taken against these attacks. The Mimikatz tool keeps on updating its functionality so the defenders must take appropriate steps to protect against the attacks performed using Mimikatz.

8. References

[1] https://www.techtarget.com/searchsecurity/definition/Kerberos

[2] https://networkencyclopedia.com/object-in-active-directory/

[3] https://www.windowstechno.com/what-is-ntds-dit-and-where-its-held-what-other-folders-are-related-to-ad/

[4] https://www.elcomsoft.com/help/en/esr/esr_system.html

[5] https://web.mit.edu/kerberos/

[6] https://medium.com/csg-govtech/process-injection-techniques-used-by-malware-1a34c078612c

[7] https://www.ired.team/offensive-security/defense- evasion/parent-process-id-ppid-spoofing

[8] https://redteam.wiki/postexploitation/mimikatz/standard

[9] https://networkencyclopedia.com/cryptoapi/

[10]https://www.insecurity.be/blog/2020/12/24/dpapi-in-depth-with-tooling-standalone-dpapi/

[11] https://www.trellix.com/en-us/security awareness/ransomware/petya.html

[12] https://www.sentinelone.com/blog/eternalblue nsa-developed-exploit-just-wont-die/

[13] https://encyclopedia.kaspersky.com/glossary/drive-by-attack/

[14] https://www.proofpoint.com/us/threat reference/bad-rabbit

[15] https://www.varonis.com/blog/domain-controlle