

### MODULES

**standard** This is the main module of mimikatz, it contains quick commands to operate with the tool. For this particular one, no need to prefix command by the module name (but it works too), eg: exit is the same as standard::exit.

**privilege** This module provides some commands to manipulate privilege on mimikatz process.

**crypto** This module, one of the oldest, plays with CryptoAPI functions. Basically it's a little certutil that benefit of token impersonation, patch legacy CryptoAPI functions and patch CNG key isolation service.

**sekurlsa** This module extracts passwords, keys, pin codes, tickets from the memory of Lsass (Local Security Authority Subsystem Service)

**kerberos** This module can be used without any privilege. It permits to play with official Microsoft Kerberos API and to create offline 'Golden tickets', free, long duration TGT tickets for any users

**lsadump** This module interacts with the Windows Local Security Authority (LSA) to extract credentials. Most of these commands require either debug rights (privilege::debug) or local System. By default, the Administrators group has Debug rights. Debug still has to be "activated" by running "privilege::debug".

**vault** This module dumps passwords saved in the Windows Vault.

### MODULES (cont)

**token** This module deals with the Windows tokens (who does not really like elevating to NT AUTHORITY\SYSTEM).

**event** This module deals with the Windows Event logs (to clear footprints after compromise).

**ts** This module deals with the Terminal Services. It can be an alternative for getting clear-text passwords.

**process** This module deals with Windows processes. It can also be used for process injection and parent process spoofing.

**service** This module can interact with Windows services plus installing the mimikatzsvc service.

**net** some functionalities in this module are similar to the Windows net commands. Enumerating sessions and servers configured with different types of Kerberos delegations is also included.

**misc** This module is kind of a catch-all for commands that don't quite fit elsewhere. The most well known commands in this module are MISC::AddSID, MISC::MemSSP, and MISC::Skeleton.

### CRYPTO

**providers** This command list all providers: CryptoAPI, then CNG if available (NT 6).

**stores** This command lists logical store in a system store.

**sc** This command lists smartcard/token reader(s) on, or deported to, the system. When the CSP is available, it tries to list keys on the smartcard.

### CRYPTO (cont)

**scauth** This command creates a client certificate for smartcard authentication, signed by a Certificate Authority

**certificates** This command lists certificates and properties of theirs keys. It can export certificates too.

**keys** This command lists keys, by provider. It can export keys too.

**capi** This patch modify a CryptoAPI function, in the mimikatz process, in order to make unexportable keys, exportable (no specific right other than access to the private key is needed) This is only useful when the keys provider is one of: Microsoft Base Cryptographic Provider v1.0, Microsoft Enhanced Cryptographic Provider v1.0, Microsoft Enhanced RSA and AES Cryptographic Provider, Microsoft RSA SChannel Cryptographic Provider, Microsoft Strong Cryptographic Provider

**cng** This patch modify KeyIso service, in LSASS process, in order to make unexportable keys, exportable. This is only useful when the keys provider is Microsoft Software Key Storage Provider (you do not need to patch CNG for other providers).

### VAULT

**cred** Enumerates vault credentials

**list** Lists saved credentials in the Windows Vault such as scheduled tasks, RDP, Internet Explorer for the current user

### TS

<b>multirdp</b>	(experimental) Patch Terminal Server service to allow multiple users
<b>sessions</b>	List TS/RDP sessions.

### STANDARD

<b>exit</b>	Quits mimikatz, after cleaning routines.
<b>cls</b>	Clears screen, by filling the console window with spaces.
<b>answer</b>	Gives the Answer to the Ultimate Question of Life, the Universe, and Everything.
<b>coffee</b>	Because everyone deserves a good coffee.
<b>sleep</b>	Sleeps an amount of milliseconds (1000 ms by default).
<b>log</b>	Logs all outputs to a file (mimikatz.log by default).
<b>base64</b>	Switches from file writing on the disk, to Base64 output instead.
<b>version</b>	Displays versions of mimikatz and Windows
<b>cd</b>	Change or display current directory

### SEKURLSA

<b>logonpasswords</b>	Lists all available provider credentials. This usually shows recently logged on user and computer credentials.
<b>pth</b>	Pass-the-Hash and Over-Pass-the-Hash (aka pass the key).
<b>tickets</b>	List and export Kerberos tickets of all sessions.
<b>ekeys</b>	List Kerberos encryption keys
<b>dpapi</b>	Read masterkeys from memory
<b>minidump</b>	Switch to LSASS minidump process context
<b>process</b>	Switches (or reinits) to LSASS process context

### SEKURLSA (cont)

<b>searchpasswords</b>	Present username and passwords available in memory
<b>msv</b>	Responsible for collecting password hashes from the LSASS address space
<b>wdigest</b>	List WDigest credentials
<b>kerberos</b>	List Kerberos credentials for all authenticated users (including services and computer account)
<b>tspkg</b>	Used for Terminal Server authentication
<b>krbtgt</b>	Get Domain Kerberos service account (KRBtgt)password data
<b>ssp</b>	Lists Security Support Provider credentials
<b>credman</b>	List Credentials Manager

### EVENT

<b>clear</b>	Clear an event log
<b>drop</b>	(experimental) Patch Events service to avoid new events
Run privilege::debug then event::drop to patch the event log. Then run Event::Clear to clear the event log without any log cleared event (1102) being logged.	

### SERVICE

<b>+ (plus sign)</b>	Install Mimikatz service ('mimikatzsvc')
<b>- (minus sign)</b>	Uninstall Mimikatz service ('mimikatzsvc')
<b>list</b>	List Services
<b>preshutdown</b>	Pre-shuts down a specified service by sending a SERVICE_CONTROL_PRESHUTDOWN signal
<b>remove</b>	Removes the specified service (It must be used with caution)

### SERVICE (cont)

<b>resume</b>	Resumes a specified service, after successful suspending, by sending a SERVICE_CONTROL_CONTINUE signal
<b>shutdown</b>	Shuts down a specified service by sending a SERVICE_CONTROL_SHUTDOWN signal
<b>start</b>	Start a service
<b>stop</b>	Stops a specified service by sending a SERVICE_CONTROL_STOP signal
<b>suspend</b>	Suspend the service. It sends a SERVICE_CONTROL_PAUSE signal

### MISC

<b>aadcookie</b>	Can be used to dump the Azure Panel's session cookie from login.microsoftonline.com
<b>clip</b>	Monitors clipboard. CTRL+C stops the monitoring
<b>cmd</b>	Launches the command prompt
<b>compress</b>	Performs a self compression of mimikatz
<b>detours</b>	(Experimental) Tries to enumerate all modules with Detours-like hooks
<b>efs</b>	Mimikatz's implementation of the MS-EFSR abuse (Petit-Potam), an authentication coercion technique
<b>lock</b>	Locks the screen. It can come in handy with misc::memssp
<b>memssp</b>	Patches LSASS by injecting a new Security Support Provider (a DLL is registered)

### MISC (cont)

<b>mfit</b>	Identifies Windows minifilters inside mimikatz, without using fltmc.exe. It can also assist in fingerprinting security products, by altitude too (Gathers details on loaded drivers, including driver altitude)
<b>ncroutemon</b>	Displays Juniper network connect (without route monitoring)
<b>ngcsign</b>	Can be used to dump the NGC key (Windows Hello keys) signed with the symmetric pop key.
<b>printrightmare</b>	Can be used to exploit the PrintNightMare vulnerability in both [MS-RPRN RpcAddPrinterDriverEx] and [MS-PAR AddPrinterDriverEx].
<b>regedit</b>	Launches the registry editor
<b>sccm</b>	Decrypts the password field in the SC_UserAccount table in the SCCM database
<b>shadow-copies</b>	Used to list the available shadow copies on the system
<b>skeleton</b>	Injects a "Skeleton Key" into the LSASS process on the domain controller
<b>spooler</b>	Mimikatz's implementation of the MS-RPRN abuse (PrinterBug), an authentication coercion technique
<b>taskmgr</b>	Launches the task manager
<b>wp</b>	Sets up a wallpaper
<b>xor</b>	Performs XOR decoding/-encoding on a provided file with 0x42 default key

### PRIVILEGE

<b>debug</b>	Ask for debug privilege for mimikatz process.
The debug privilege allows someone to debug a process that they wouldn't otherwise have access to. For example, a process running as a user with the debug privilege enabled on its token can debug a service running as local system. Remark: ERROR kuhl_m_privilege_simple ; RtlAdjustPrivilege (20) c0000061 means that the required privilege is not held by the client	

### LSADUMP

<b>sam</b>	This command dumps the Security Account Managers (SAM) database. It contains NTLM, and sometimes LM hash, of users passwords.
<b>secrets</b>	Get the SysKey to decrypt SECRETS entries (from registry or hives).
<b>setntlm</b>	Used to perform a password reset without knowing the user's current password. It can be useful during an active directory Access Control (ACL) abuse scenario
<b>lsa</b>	Ask LSA Server to retrieve SAM/AD enterprise (normal, patch on the fly or inject). Use to dump all Active Directory domain credentials from a Domain Controller or lsass.dmp dump file. Also used to get specific account credential such as krbtgt with the parameter /name: "/name:krbtgt"
<b>dcsync</b>	Ask a DC to synchronize an object (get password data for account). No need to run code on DC.
<b>trust</b>	Ask LSA Server to retrieve Trust Auth Information (normal or patch on the fly). Dumps trust keys (passwords) for all associated trusts (domain/forest).

### LSADUMP (cont)

<b>backupkeys</b>	Dumps the DPAPI backup keys from the Domain Controller
<b>cache</b>	Can be used to enumerate Domain Cached Credentials from registry. It does so by acquiring the SysKey to decrypt NL\$KM (binary protected value) and then MSCache(v1/v2)
<b>changentim</b>	Used to change the password of a user
<b>zerologon</b>	Detects and exploits the ZeroLogon vulnerability

### KERBEROS

<b>ptt</b>	Pass-the-Ticket. Typically used to inject a stolen or forged Kerberos ticket (golden/silver/trust).
<b>golden / silver</b>	This command create Kerberos ticket, a TGT or a TGS with arbitrary data, for any user you want, in groups you want
<b>tgt</b>	Displays informations about the TGT of the current session.
<b>list</b>	Lists and export Kerberos tickets (TGT and TGS) of the current session.
<b>purge</b>	Purges all tickets of the current session.

### TOKEN

<b>elevate</b>	Used to impersonate a token. Used to elevate permissions to SYSTEM (default) or find a domain admin token on the box using the Windows API.
<b>list</b>	List all tokens of the system
<b>revert</b>	Revert to previous token
<b>run</b>	Executes a process with its token
<b>whoami</b>	Display current identity

PROCESS		NET	
<b>exports</b>	Lists all the exported functions from the DLLs each running process is using. If a /pid is not specified, then exports for mimikatz.exe will be displayed	<b>alias</b>	Displays more information about the local group memberships including Remote Desktop Users, Distributed COM Users, etc
<b>imports</b>	Lists all the imported functions from the DLLs each running process is using. If a /pid is not specified, then imports for mimikatz.exe will be displayed	<b>deleg</b>	Checks for Kerberos delegations
<b>list</b>	Lists all running processes. It uses the NtQuerySystemInformation Windows Native API function	<b>group</b>	Displays the local groups
<b>resume</b>	Resumes a suspended process by using the NtResumeProcess Windows Native API function	<b>if</b>	Displays the available local IP addresses and the hostname
<b>start</b>	Starts a process by using the CreateProcess Win32 API function. The PID of the process is also displayed	<b>serverinfo</b>	Displays information about the logged in server
<b>stop</b>	Terminates a process by using the NtTerminateProcess Windows Native API function. The Win32 API equal one is TerminateProcess	<b>session</b>	Displays the active sessions through NetSessionEnum() Win32 API function
<b>suspend</b>	Suspends a process by using the NtSuspendProcess Windows Native API function	<b>share</b>	Displays the available shares
<b>run</b>	Creates a process by using the CreateProcessAsUser Win32 API function. The CreateEnvironmentBlock is also utilized	<b>stats</b>	Displays when the target was booted
<b>runp</b>	Runs a subprocess under a parent process (Default parent process is LSASS.exe). It can also be used for lateral movement and process spoofing	<b>tod</b>	Displays the current time
		<b>trust</b>	Displays information for the active directory forest trust(s)
		<b>user</b>	Displays the local users
		<b>wsession</b>	Displays the active sessions through NetWkstaUserEnum() Win32 API function