WELCOME

bootCon

Presenter

James Nguyen
Info Safe
Pen Tester

Email: jnguyen@infosafe.com

bootCon

Presenter

Stephanie Ortega
Info Safe
Pen Tester

Email: sortega@infosafe.com

Researching the Inner Works of Password Cracking Tools

By: James Nguyen, Stephanie Ortega

Different Password Crackers & Techniques

- Brute Force
- Rainbow Table
- Dictionary
- Hybrid: Brute Force and Dictionary

Hydra

- Fast and flexible
- Primarily uses brute force dictionary-based attacks (Hybrid)
- Very effective against remote authentication services
- Most popular Operating Systems:
  - Windows, Linux/Unix, MacOs
- Supports a wide range of protocols:
  - TELNET, FTP, HTTP, HTTPS, SNMAP, IMAP, POP3

Cain and Abel

- Password recovery tool for Windows OS
- Can be used by:
  - Sniffing the network, deciphering passwords using Dictionary, brute force
- Also recovers wireless network keys

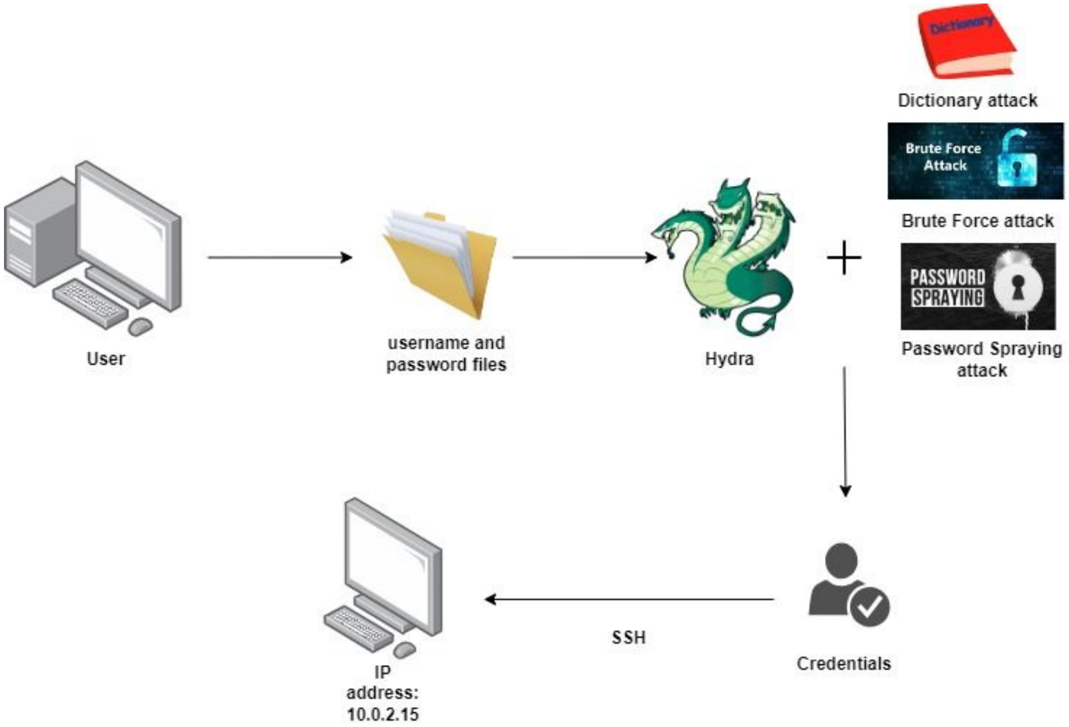
Rainbow Crack

- A computer program that creates rainbow tables to crack passwords
- Time consuming but still hundreds of times faster than brute forcing
- Operating systems:
  - Windows and Linux
- Offline attack

John The Ripper

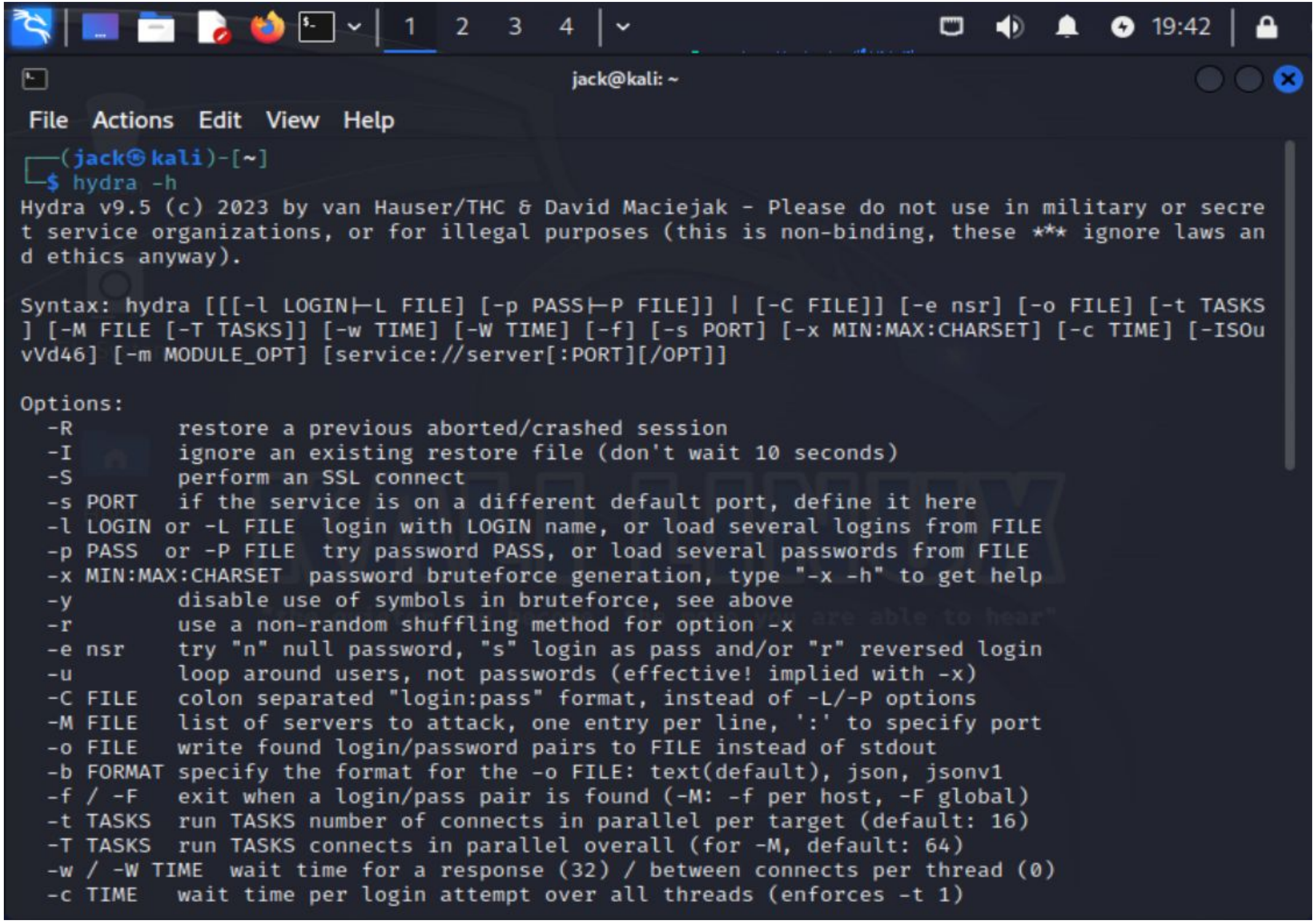
- Used to detect weak passwords, not designed to crack strong passwords
- Automatically detects password hash and can run against multiple encryptions
- Implements brute force strategy; time consuming
- Multiplatform

Visualization: Brute Force & Dictionary Attacks



Installation of each password cracker:

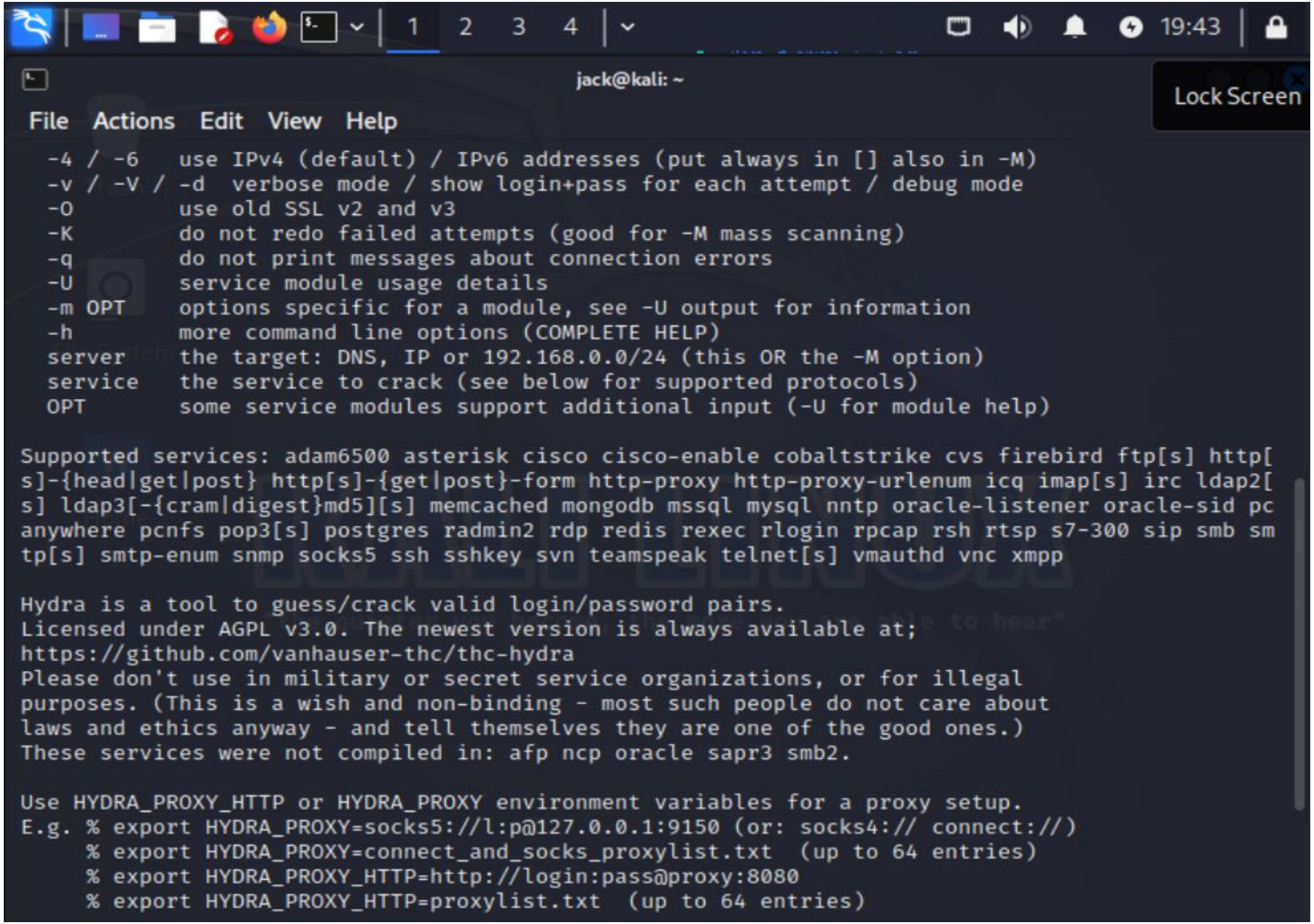
Hydra	`sudo apt install hydra`
Cain and Abel	http://www.oxid.it/cain.html
Rainbow Crack	`sudo apt install rainbowcrack`
John the Ripper	`sudo apt-get install john -y`



```
┌──(jack㉿kali)-[~]
└─$ hydra -h
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secre
t service organizations, or for illegal purposes (this is non-binding, these *** ignore laws an
d ethics anyway).

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS
] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOu
vVd46] [-m MODULE_OPT] [service://server[:PORT][/OPT]]

Options:
  -R        restore a previous aborted/crashed session
  -I        ignore an existing restore file (don't wait 10 seconds)
  -S        perform an SSL connect
  -s PORT   if the service is on a different default port, define it here
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -x MIN:MAX:CHARSET  password bruteforce generation, type "-x -h" to get help
  -y        disable use of symbols in bruteforce, see above
  -r        use a non-random shuffling method for option -x
  -e nsr    try "n" null password, "s" login as pass and/or "r" reversed login
  -u        loop around users, not passwords (effective! implied with -x)
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to attack, one entry per line, ':' to specify port
  -o FILE   write found login/password pairs to FILE instead of stdout
  -b FORMAT specify the format for the -o FILE: text(default), json, jsonv1
  -f / -F   exit when a login/pass pair is found (-M: -f per host, -F global)
  -t TASKS  run TASKS number of connects in parallel per target (default: 16)
  -T TASKS  run TASKS connects in parallel overall (for -M, default: 64)
  -w / -W TIME  wait time for a response (32) / between connects per thread (0)
  -c TIME   wait time per login attempt over all threads (enforces -t 1)
```



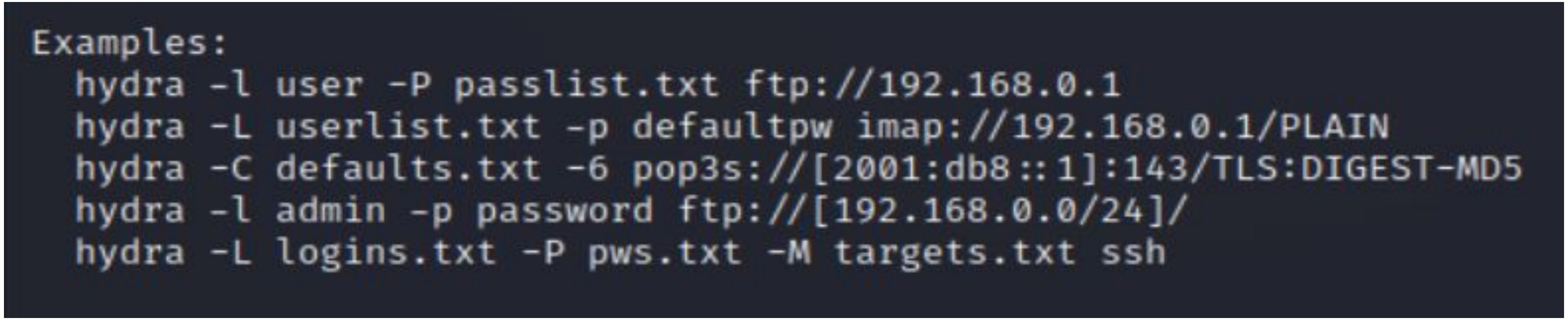
```
  -4 / -6   use IPv4 (default) / IPv6 addresses (put always in [] also in -M)
  -v / -V / -d  verbose mode / show login+pass for each attempt / debug mode
  -O        use old SSL v2 and v3
  -K        do not redo failed attempts (good for -M mass scanning)
  -q        do not print messages about connection errors
  -U        service module usage details
  -m OPT    options specific for a module, see -U output for information
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s] http[
s]-{head|get|post} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[
s] ldap3[-{cram|digest}md5][s] memcached mongodb mssql mysql nntp oracle-listener oracle-sid pc
anywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb sm
tp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)
These services were not compiled in: afp ncp oracle sapr3 smb2.

Use HYDRA_PROXY_HTTP or HYDRA_PROXY environment variables for a proxy setup.
E.g. % export HYDRA_PROXY=socks5://l:p@127.0.0.1:9150 (or: socks4:// connect://)
     % export HYDRA_PROXY=connect_and_socks_proxylist.txt  (up to 64 entries)
     % export HYDRA_PROXY_HTTP=http://login:pass@proxy:8080
     % export HYDRA_PROXY_HTTP=proxylist.txt  (up to 64 entries)
```

Examples of different Hydra commands

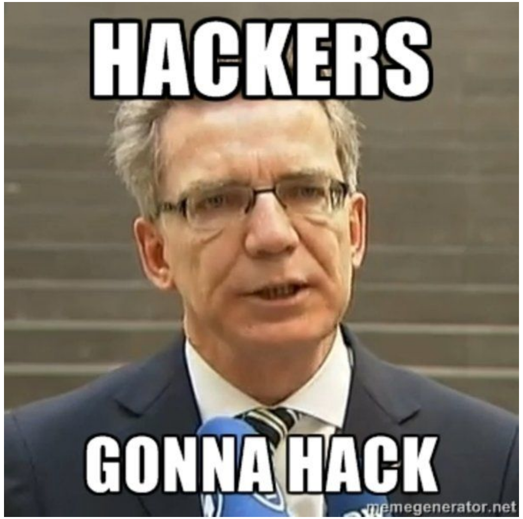


```
Examples:
  hydra -l user -P passlist.txt ftp://192.168.0.1
  hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN
  hydra -C defaults.txt -6 pop3s://[2001:db8::1]:143/TLS:DIGEST-MD5
  hydra -l admin -p password ftp://[192.168.0.0/24]/
  hydra -L logins.txt -P pws.txt -M targets.txt ssh
```

Hydra Kali Linux Demo

Effects of the Attack

- We were able to obtain the credentials of the admin and log in via ssh
- From here we could create/delete files
- Maintain access in the system
- We could elevate root privileges
- Change passwords and lock out users
- Information could be gathered and leaked/sold



Mitigating Brute Force & Dictionary Attacks

- Strong passwords!
- Multi-Factor Authentication
- Account Lockout policies
- Rate Limiting
- Password Hashing and Salting
- Web application firewalls (WAF)
- Software updates and patching

The End

References

- http://repository.futminna.edu.ng:8080/jspui/bitstream/123456789/9652/1/A%20Review%20of%20Top%20Open%20Source%20Password%20Cracking%20Tools.pdf
- https://resources.infosecinstitute.com/topics/hacking/password-cracking-using-cain-abel/
- https://www.kali.org/tools/hydra/
- https://www.kali.org/tools/rainbowcrack/
- https://www.freecodecamp.org/news/crack-passwords-using-john-the-ripper-pentesting-tutorial/
- https://www.techtarget.com/searchsecurity/tutorial/How-to-use-the-Hydra-password-cracking-tool
- https://capec.mitre.org/data/definitions/55.html
-

References

https://www.youtube.com/watch?v=RyQL9AdxHqY

https://resources.infosecinstitute.com/topics/hacking/10-popular-password-cracking-tools/

https://resources.infosecinstitute.com/topics/hacking/popular-tools-for-brute-force-attacks/

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://bpb-us-e1.wpmucdn.com/sites.psu.edu/dist/9/24816/files/2016/06/PasswordCracking.pdf

https://youtu.be/Y2fhWtZedTQ?si=4ftHjlb4lPkG6sPU