

David Thomsen

Deliverable 1. Provide screenshots similar to the ones above showing the last 3 entries in /etc/passwd and /etc/shadow.

```
[root@bios ~]# tail -n 3 /etc/shadow
gandalf.grey:$6$rounds=1000$LneEppAvGXMREfOV$vkOzEXBjXOD0XK3YJUgd5.n
fQVq/gM3BEbKbARZu/BNQNi6Uu3cie5Jv0IhkJ5A6mKGUIGKpUG3gFi4KE6xXW.:1914
3:0:99999:7:::
boromir:$6$rounds=1000$UvKLGAr/VWtqFGCE$DcfW0zRolV4T6GAB0U0FFXfg4lpm
D4mKriKX1n5sN3ugJSY3nnicjuGfbT9hgEeo.b6dpWSitnK3z3jjBQ2w//:19143:0:9
9999:7:::
galadriel:$6$rounds=1000$poPWvLT/CfA/sxS/$lHbu1oMqRV2aM18fkFPbJw25U2
.POqhonSmaUpbzPIPVKl2IxS86Qq8q9v3fYu5Y6qlWwbmqekbL3g1vtPmlQ/:19143:0
:99999:7:::
[root@bios ~]# █
```

```
(champuser@kali)-[~/SEC335/SEC-335/Week 5]
└─$ ssh peregrin.took@10.0.5.21
peregrin.took@10.0.5.21's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Mon Feb 20 18:05:08 2023 from 10.0.17.31
[peregrin.took@bios ~]$ sudo -i
[sudo] password for peregrin.took:
[root@bios ~]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfi
ed_r:unconfined_t:s0-s0:c0.c1023
[root@bios ~]# █
```

Deliverable 2. Research what hashing algorithm is being used on this server, one of the fields in /etc/shadow points to the format. Explain this.

Passwords starting with "\$5\$" or "\$6\$" are interpreted as hashed with Linux SHA256 or SHA512 password hashing, respectively.

<https://www.open.com.au> › radiator › ref › User-Password ⓘ

Section 7.1.1. User-Password, Password

```
[root@bios ~]# tail -n 3 /etc/shadow
gandalf.grey:$6$rounds=1000$LneEppAvG
i4KE6xXW.:19143:0:99999:7:::
boromir:$6$rounds=1000$UvKLGar/VWtqFG
2w//:19143:0:99999:7:::
galadriel:$6$rounds=1000$poPWvLT/CfA/
tPmlQ/:19143:0:99999:7:::
[root@bios ~]#
```

Each of the passwords in the /etc/shadow file begin with a \$6 indicating a SHA512 hash.

Deliverable 3. Examine user Galadriel's shadow entry.

- What is the salt?
- “When a user picks or is assigned a password, it is encoded with a randomly generated value called the salt. This means that any particular password could be stored in 4096 different ways. The salt value is then stored with the encoded password. When a user logs in and supplies a password, the salt is first retrieved from the stored encoded password.”
- Sources:
 - <https://superuser.com/questions/822079/etc-shadow-in-old-format-where-is-salt-stored>
 - <https://tldp.org/HOWTO/Shadow-Password-HOWTO-2.html>

What is the hashed salt+password?

<https://www.cyberciti.biz/faq/understanding-etcshadow-file/>

2. **Password** : Your encrypted password is in hash format. The password should be minimum 15-20 characters long including special characters, digits, lower case alphabetic and more. Usually password format is set to `idsalt$hashed`, The `$id` is the algorithm used On GNU/Linux as follows:

1. `1` is MD5
2. `$2a$` is Blowfish
3. `$2y$` is Blowfish
4. `5` is SHA-256
5. `6` is SHA-512

galadriel:`6rounds=1000$poPWvLT/CfA/sxS/$1HbulomQrV2aM18fkFPbJw25U2.P`
`OqhonSmaUpbzPIPVK12IxS86Qq8q9v3fYu5Y6qlWwbmqekbL3g1vtPmlQ/`:19143:0:999
99:7:::

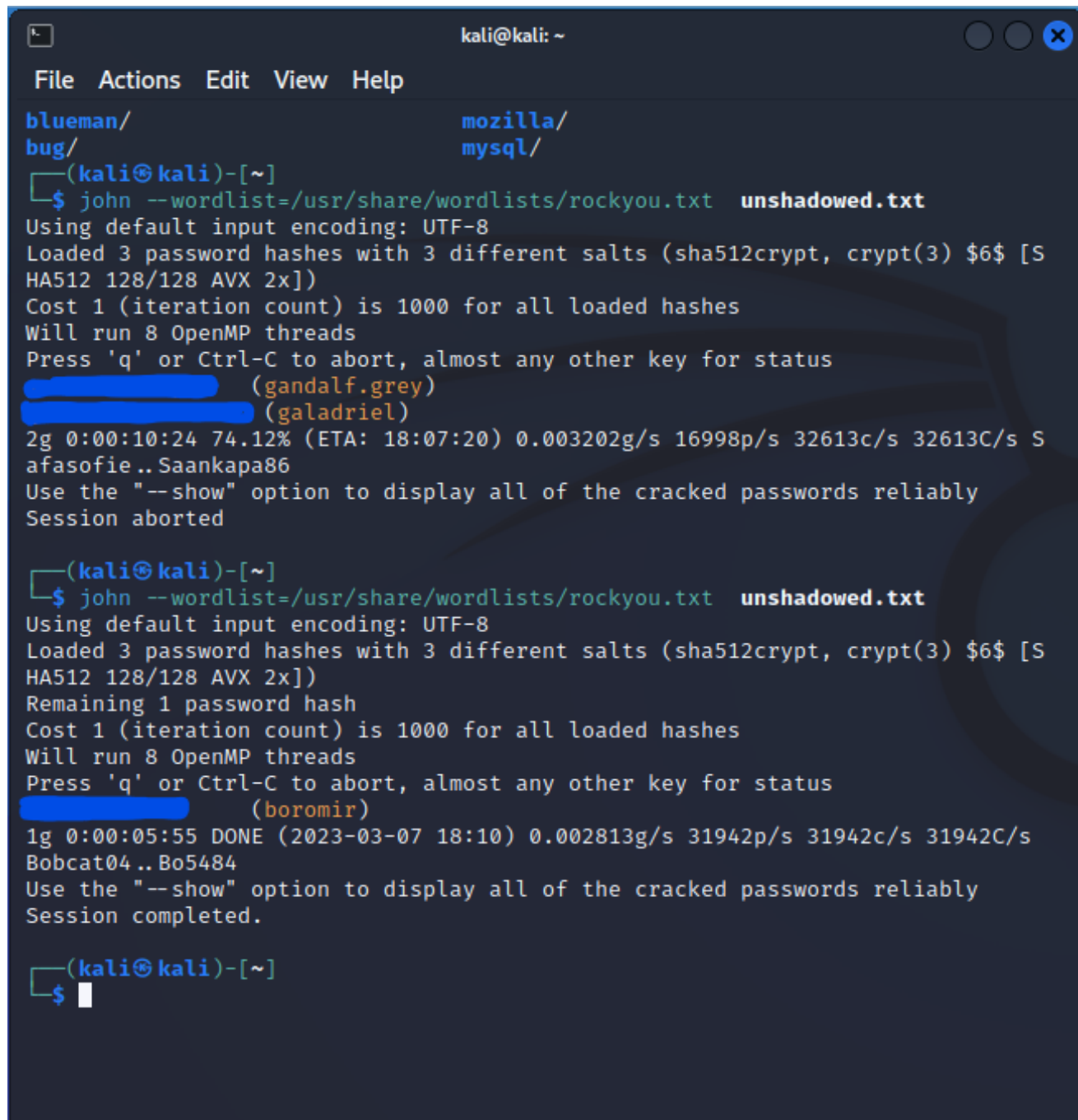
Blue:ID

Green: Rounds of Hash

Pink:Salt

Purple: Hashed Pass

Deliverable 4. Figure out how to use the unshadow utility to create a file usable by John the Ripper(JtR) and then crack the unshadowed files hashes using JtR. Provide a screenshot showing your results.

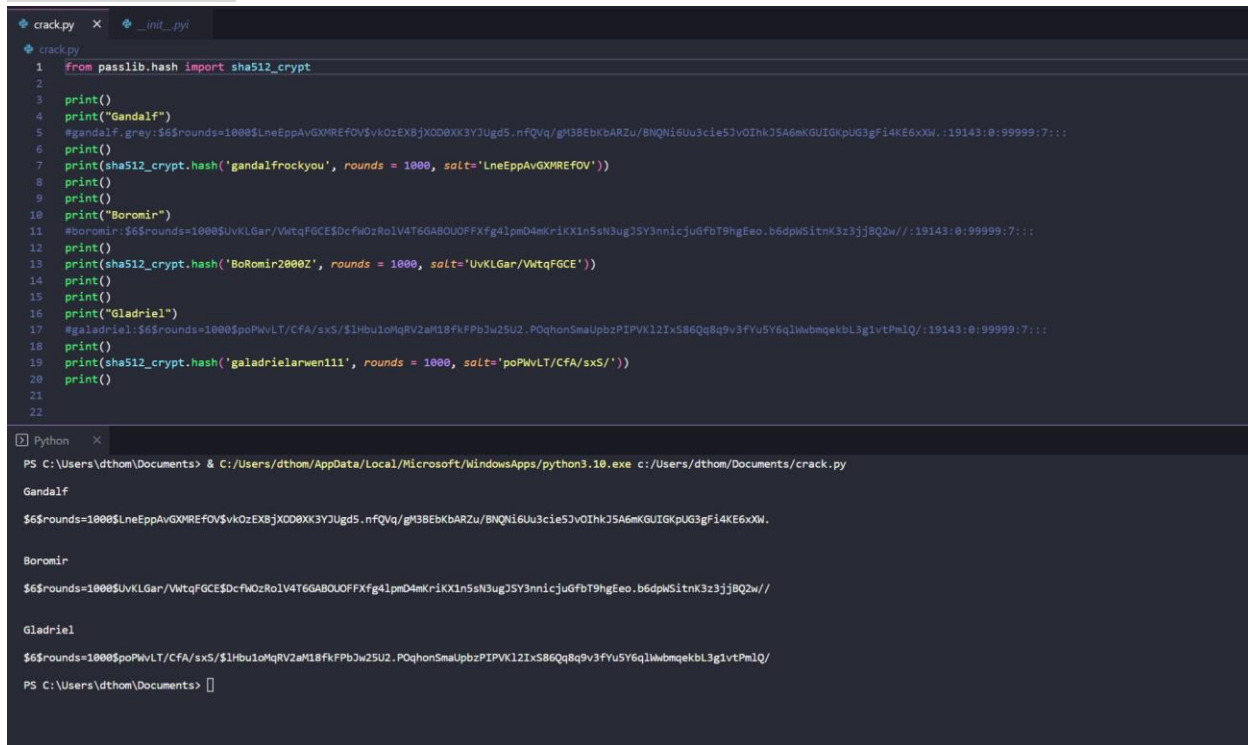


```
kali@kali: ~
File Actions Edit View Help
blueman/      mozilla/
bug/          mysql/
(kali@kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt unshadowed.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (sha512crypt, crypt(3) $6$ [S
HA512 128/128 AVX 2x])
Cost 1 (iteration count) is 1000 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
(gandalf.grey)
(galadriel)
2g 0:00:10:24 74.12% (ETA: 18:07:20) 0.003202g/s 16998p/s 32613c/s 32613C/s S
afasofie..Saankapa86
Use the "--show" option to display all of the cracked passwords reliably
Session aborted

(kali@kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt unshadowed.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (sha512crypt, crypt(3) $6$ [S
HA512 128/128 AVX 2x])
Remaining 1 password hash
Cost 1 (iteration count) is 1000 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
(boromir)
1g 0:00:05:55 DONE (2023-03-07 18:10) 0.002813g/s 31942p/s 31942c/s 31942C/s
Bobcat04..Bo5484
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali@kali)-[~]
$
```

Deliverable 5. Let's see if you can reverse engineer the shadow file using python. We will use peregrin.took's bios hash. The grayed out area has the plaintext password for gandalf. Provide a screenshot similar to the one below. Use Boromir or Galadriel's shadow entry.



```
crack.py x  _init_.pyl
crack.py
1 from passlib.hash import sha512_crypt
2
3 print()
4 print("Gandalf")
5 #gandalf:grey:$6$rounds=1000$LneEppAvGXMRfOV$vkOzEXBjXOD0XK3YJUGd5.nfQVq/gM3BEbKbARZu/BNQNi6Uu3cie5JvOIhkJ5A6mKGUIGKpUG3gFi4KE6xXW.:19143:0:99999:7:::
6 print()
7 print(sha512_crypt.hash('gandalfrockyou', rounds = 1000, salt='LneEppAvGXMRfOV'))
8 print()
9 print()
10 print("Boromir")
11 #boromir:$6$rounds=1000$UvKLGar/VwtqFGCE$DcfW0zRoLV4T6GABOUOFFXfg41pmD4mKr1KX1n5sN3ugJSY3nnicjuGfbT9hgEeo.b6dpwSitrK3z3jj8Q2w//:19143:0:99999:7:::
12 print()
13 print(sha512_crypt.hash('BoRomir2000Z', rounds = 1000, salt='UvKLGar/VwtqFGCE'))
14 print()
15 print()
16 print("Galadriel")
17 #galadriel:$6$rounds=1000$poPWvLT/CfA/sxS/$lHbu1oMqRV2aM18fkFPbJw25U2.PQqhonSmaUpbzPIPVK12IxS86Qq8q9v3fYu5Y6qLWbmqekbL3g1vtPmLQ/:19143:0:99999:7:::
18 print()
19 print(sha512_crypt.hash('galadrielarwen111', rounds = 1000, salt='poPWvLT/CfA/sxS/'))
20 print()
21
22

Python x
PS C:\Users\dthom\Documents> & C:/Users/dthom/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/dthom/Documents/crack.py

Gandalf

$6$rounds=1000$LneEppAvGXMRfOV$vkOzEXBjXOD0XK3YJUGd5.nfQVq/gM3BEbKbARZu/BNQNi6Uu3cie5JvOIhkJ5A6mKGUIGKpUG3gFi4KE6xXW.:19143:0:99999:7:::

Boromir

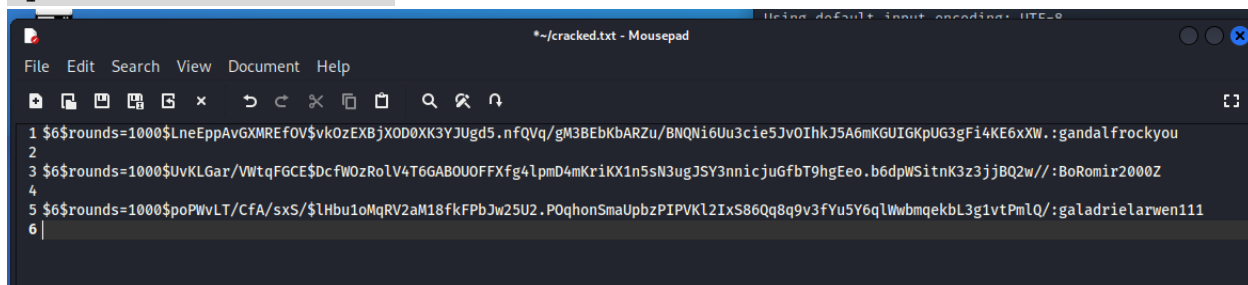
$6$rounds=1000$UvKLGar/VwtqFGCE$DcfW0zRoLV4T6GABOUOFFXfg41pmD4mKr1KX1n5sN3ugJSY3nnicjuGfbT9hgEeo.b6dpwSitrK3z3jj8Q2w//:19143:0:99999:7:::

Galadriel

$6$rounds=1000$poPWvLT/CfA/sxS/$lHbu1oMqRV2aM18fkFPbJw25U2.PQqhonSmaUpbzPIPVK12IxS86Qq8q9v3fYu5Y6qLWbmqekbL3g1vtPmLQ/:19143:0:99999:7:::

PS C:\Users\dthom\Documents>
```

Deliverable 6. crack at least one of the hashes using hashcat and show the result in a screenshot similar to the one below (hit s for status). Again, consider leveraging humpty or your own physical system for this crack.



```
*-/cracked.txt - Mousepad
File Edit Search View Document Help
1 $6$rounds=1000$LneEppAvGXMRfOV$vkOzEXBjXOD0XK3YJUGd5.nfQVq/gM3BEbKbARZu/BNQNi6Uu3cie5JvOIhkJ5A6mKGUIGKpUG3gFi4KE6xXW.:gandalfrockyou
2
3 $6$rounds=1000$UvKLGar/VwtqFGCE$DcfW0zRoLV4T6GABOUOFFXfg41pmD4mKr1KX1n5sN3ugJSY3nnicjuGfbT9hgEeo.b6dpwSitrK3z3jj8Q2w//:BoRomir2000Z
4
5 $6$rounds=1000$poPWvLT/CfA/sxS/$lHbu1oMqRV2aM18fkFPbJw25U2.PQqhonSmaUpbzPIPVK12IxS86Qq8q9v3fYu5Y6qLWbmqekbL3g1vtPmLQ/:galadrielarwen111
6
```

Deliverable 7. Start a text or csv or markdown file similar to the one below. Include your successful guesses from Week 5 as well as the cracks from this week. We will need this data in our future adventures. a listing or screenshot of all your acquired passwords. This type of material is normally called "loot" in hacker parlance. Documenting uncracked hashes is also a great idea. You may have better luck cracking them as you learn more about your target or decide to crack on a real workstation instead of a kali vm.

J21:J22 ▾ *fx* |

	A	B	C	D
1	SSH		HTTP	
2	Username	Password	Username	Password
3	samwise	SamwiseGamgee19	frodo	1Brandywine
4	bilbo.baggins	Frodo2013	pippin	adminPippin
5	frodo.baggins	Strider2020	samwise	RosieRosie
6	samwise.gamgee	Mallorn79		
7	peregrin.took (sudoer)	28Peregrin		
8	gandalf.grey	gandalfrockyou		
9	bromir	BoRomir2000z		
10	galadriel	galadrielawren111		
11				
12				