

Name: Zak Rowland  
Embedded Security  
CST 466  
Assignment 2: John the ripper  
Due date: April 16<sup>th</sup>, 2021

Important: Use the jumbo version of Johntheripper from

<http://www.openwall.com/john/>

```
john --format=raw-SHA1 passwords.txt // Specify raw SHA1 format,  
use passwords.txt
```

```
john passwords.txt --show //Shows passwords that were cracked
```

1. Run a benchmark on your computer using John, by calling

```
john -test
```

Provide the benchmark for sha256crypt and md5crypt.

sha256crypt: 18,646 c/s real, 1,605 c/s virtual

```
Benchmarking: sha256crypt, crypt(3) $5$ (rounds=5000) [SHA256 256/256 AVX2 8x]... (12xOMP) DONE  
Speed for cost 1 (iteration count) of 5000  
Raw:      18646 c/s real, 1605 c/s virtual
```

md5crypt: Many salts – 675,857 c/s real, 58,949 c/s virtual

Only one salt – 652,183 c/s real, 55,747 c/s virtual

```
Benchmarking: md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3]... (12xOMP) DONE  
Many salts:      675857 c/s real, 58949 c/s virtual  
Only one salt:   652183 c/s real, 55747 c/s virtual  
  
Benchmarking: md5crypt-long, crypt(3) $1$ (and variants) [MD5 32/64]... (12xOMP) DONE  
Raw:      60809 c/s real, 5181 c/s virtual
```

2. What does c/s in the previous benchmarks mean?

From [openwall.com/john/doc/FAQ.shtml](http://openwall.com/john/doc/FAQ.shtml) :

“... c/s is "crypts" (password hash or cipher computations) per second ...”

3. Crack the raw-SHA1 hashed passwords using ANY method you want. (john --format=raw-SHA1 password.txt) Where password.txt is the file you dumped the three passwords below. Run it for a maximum of 15 minutes.

Name: Zak Rowland  
Embedded Security  
CST 466  
Assignment 2: John the ripper  
Due date: April 16<sup>th</sup>, 2021

pw1:ffb4761cba839470133bee36aeb139f58d7dbaa9

pw2:5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8

pw3:709d4cb6ef82fd0dc7260263fc07b94711b6a4ad

What are the passwords?

1. password

2. kevin

3. Uncracked after 15 minutes. Intel Core i7-10750H, single thread boosted to 4.7GHz. Still uncracked after 5 more minutes using all 12 threads boosted to 3.7GHz. A SHA-1 decrypting website returns “saltedpassword” in 100 milliseconds ... my laptop is slow I guess.

You probably won't be able to get the third password. Why do you think, even though they are hashed using the same algorithm that the third password is so much harder to get?

The first two passwords are common phrases which makes a dictionary attack trivial. The last password is salted and requires a brute force attack.

What type of attack(s) are we performing? Dictionary? Brute force?

John first tests list of common words, names, and passwords, which is a dictionary attack. This is why “password” and “kevin” were cracked almost instantly. John then moves on to a brute force attack, trying every possible combination of ASCII characters until a match is found.

4. Hashes do not encrypt the password. What's the point of hashing password(s) before storing it?

Although hashing does not encrypt the password, it is still infinitely better than storing the passwords in plaintext. Storing them in plaintext just makes a potential attackers job much easier. The password should be salted before it is hashed, further elevating security.

Name: Zak Rowland  
Embedded Security  
CST 466  
Assignment 2: John the ripper  
Due date: April 16<sup>th</sup>, 2021

5. Before we hash a password, we salt it.

Give two reasons we salt the password.

One reason passwords are salted is because it makes every password unique, so even if two users use the same password the hashed password will be different. Another reason to salt passwords is they defend against hash table attacks because attackers must compute them for each password.

How is the password salted?

Passwords are salted by feeding the password and some random data (the salt) into a function that hashes it.

Should the salt be the same for all user passwords? Why or why not?

A new salt should be randomly generated for each password, otherwise the attacker could crack one password and know the salt for the rest of them.

Does using salt increase the security for an individual password?

The salt doesn't make the password "stronger" in the same sense that adding symbols, numbers, etc., but it does increase security for when the password is stored or moved around. For example, if someone's password was "password," it is extremely insecure on its own, but once it is salted and hashed it is much harder to crack.

6. Thinking in terms of an embedded system, what could be the problems with salting a password or using a higher order encryption or hashing scheme (AES128 vs AES256)?

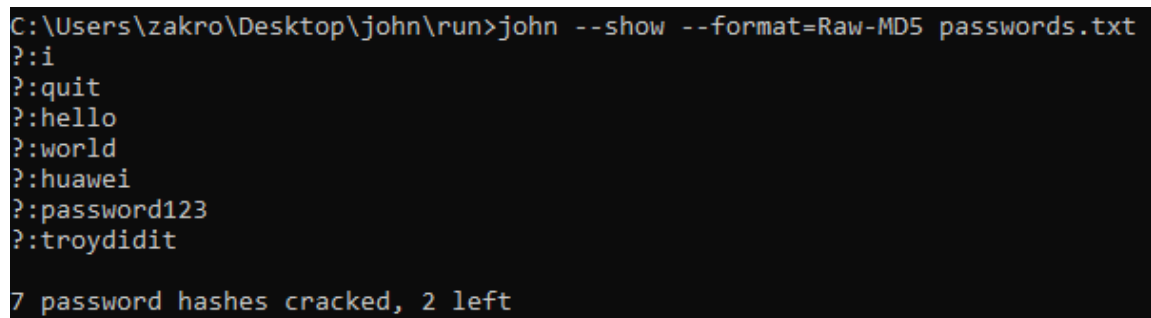
The embedded system may not have the processing power required to do these computations in a timely manner. If it does have the power, it may require a cooling solution, or could be taxing on the battery if it uses one.

Name: Zak Rowland  
Embedded Security  
CST 466  
Assignment 2: John the ripper  
Due date: April 16<sup>th</sup>, 2021

7. Thinking in terms of an embedded system, why are embedded systems more vulnerable to password attacks than the typical desktop or laptop computer?

Embedded systems are more vulnerable to attacks for a few reasons; as mentioned earlier, they just don't have the same processing power that a typical computer. The system may have to use a weaker encryption or hashing scheme that can be computed quickly. The embedded systems also do not receive updates the same way computers do. If a vulnerability is found in Windows, Microsoft could push out an update to a billion machines the same day. The same cannot be said for most embedded systems in use.

8. Crack as many of the following MD5 hashes as you can. Try to run it for 24 hours.



```
C:\Users\zakro\Desktop\john\run>john --show --format=Raw-MD5 passwords.txt
?:i
?:quit
?:hello
?:world
?:huawei
?:password123
?:troydidit

7 password hashes cracked, 2 left
```

pw0:865c0c0b4ab0e063e5caa3387c1a8741

i

pw01:dbd73c2b545209688ed794c0d5413d5a

quit

pw1:5d41402abc4b2a76b9719d911017c592

hello

pw2:7d793037a0760186574b0282f2f435e7

world

pw3:c1aafc7e23f24ba11aae492f5caa2d97

huawei

Name: Zak Rowland  
Embedded Security  
CST 466  
Assignment 2: John the ripper  
Due date: April 16<sup>th</sup>, 2021

pw4:482c811da5d5b4bc6d497ffa98491e38

password123

pw5:fc5e038d38a57032085441e7fe7010b0

**Not cracked.**

pw6:77e220f89075b0c63610bb174b603646

troydedit

pw7:13fd2ac2706b1294ae8206d1262d51d0

**Not cracked.**

9. Now run them through crackstation.net. are you able to get additional values?

Hash	Type	Result
865c0c0b4ab0e063e5caa3387c1a8741	md5	i
dbd73c2b545209688ed794c0d5413d5a	md5	quit
5d41402abc4b2a76b9719d911017c592	md5	hello
7d793037a0760186574b0282f2f435e7	md5	world
c1aafc7e23f24ba11aae492f5caa2d97	md5	huawei
482c811da5d5b4bc6d497ffa98491e38	md5	password123
fc5e038d38a57032085441e7fe7010b0	md5	helloworld
77e220f89075b0c63610bb174b603646	Unknown	Not found.
13fd2ac2706b1294ae8206d1262d51d0	Unknown	Not found.

Color Codes: **Green** Exact match, **Yellow** Partial match, **Red** Not found.

pw5: helloworld

pw7: **Still not cracked.**

Name: Zak Rowland  
Embedded Security  
CST 466  
Assignment 2: John the ripper  
Due date: April 16<sup>th</sup>, 2021

10. From the previous MD5 hash cracking, what was the most effective way to make cracking take longer? In other words, in what order is John operating?

The most effective way seems to be length. With 10+ characters, brute force attacks take much longer. John checks against a list of common passwords and words first, so as long as the password is 10+ characters and isn't a common word, crack time increases significantly.

11. Can you describe the relationship between the size of the password vs how easy it is to crack? What would you say about password length requirements?

As the length of the password increases, the security increases exponentially. As long the password string isn't a common phrase or word, even just one symbol and number added can make it near impossible to crack with modern computers. I would require a password to be at minimum 12 characters, anything less can be cracked too fast even if it a mix of random characters.

12. Does password expiration (Rule where you have to change your password every X days) help or hinder security?

Forced password changes tend to hinder security more than help. If a user is constantly having to remember new passwords, they are going to make it easy to remember which usually means easy to crack. It is better to require one extremely strong password than change them constantly. This explains why password managers like Bitwarden or LastPass are becoming increasingly popular.