**EC Transcripts – Rahil Parikh**

**Course 6: Hands-on Password Attacks and Security**

**Chapter 1: Cracking Your First Password**

**1. Legal Considerations**

Hello world intended and welcome back to the Hands-On. Password cracking and security course, section one, tracking your first password.

We will start this section with some legal consideration. So we know what passwords we can use in this course or

elsewhere to crack or not. The most important thing to consider is do not attack systems passwords and

accounts without permission. Let's take a closer. Look at what we can attack my Facebook, Instagram, Twitter account password.

No. No, you cannot attack your own Facebook. Instagram, Twitter account. You cannot hack your own accounts because they are part

of the system of a company that you don't own. You're also not allowed to hack a stranger's email

account. Would you can attack and crack are self-made, hashed passwords and the passwords which will be provided from this course.

What about leaked password thumbs? Well, it's a bit of a gray area. But I'd recommend you to not attack linked

password thumps. What you also can attack is password thumbs that are part of your penetration test. Considering

that it's part of your clients scope on the percent of the passwords and hashes that I will provide in.

This course can be used to crack and that's all for the legal considerations. Durations just into anything illegal. And next

time, we will continue with the basics of entropy hashing and cryptography for password security.

**2. Basics of Entropy, Hashing, and Cryptography for Password Security**

Hello world. I'm Senate and welcome back to the Henson password cracking and security course. Today. We're going to take a look at basics of entropy hashing

and cryptography for password security. We will take a look at entropy what is entropy and how does entropy

relate to passwords? We're going to take a look at hashing and what one-way hash functions are and

what they aren't, and We are going to take a look at cryptography. How does it relate to passwords? And

we're also going to take a look at cryptography. What are the goals of cryptography and where do passwords fit in, in

this goats? Entropy entropy is basically a measure of information, which means the higher the entropy is, the

more possible combinations are possible. Entropy is measured in bit and displaced on probability. If all outcomes are equally

likely we can use lock 2 of n. To calculate the entropy. Experiments in the context of

entropy and probability is choosing an outcome. For example, if you throw a dice and you get a number from

the dice, this would be an experiment because the probability of a dice falling on that side has a

certain percentage entropy is additive. If the experiments are independent. What does that mean? If we go

back to our dice examples, if we throw our dice three times, the first outcome does not influence the second

outcome and the second outcome does not influence the third outcome. So the experiments are independent and the entropy is additive.

Good, but what does this have to do with passwords? Basically choosing a password, is the result of a random experiment. Let's

take a look at how entropy and passwords relate. We have a password consisting of six alphabetic character. A

to see, for example, f g as HTTP, and we're only using upper case letters. For this example, so we have 2

to the power of four point seven possibilities per character, which means our entropy for one character is 4.7 bits.

H is the symbol for entropy? Now. Since choosing a character is experiment and entropy is

additive and choosing. The first character does not influence, how we choose. The second character we can use the additive entropy

property and calculate, the total entropy of this password, which will be twenty eight point two bits.

Another example would be, if we have a constant string, hello world, which would mean every password used is this string. Since

there is only one password. The entropy is log 2 of 1 which is 0 B. There's no new information for us because it's always

the same password. Another example is picking a password from a list of 1024 words. 1024 equals 2 to

the power of 10 possible words that we can choose from since all passwords are equally, likely, we can choose one

from the list at random. Our entropy is calculated with log 2 of 2 to the power

of 10, which will be 10 bits. The entropy of a password stands in relation to the work factor, to correct that password.

So in short, it means that the time to crack a password is more or less relative to the entropy of a password. So

we will aim for our passwords to have a higher entropy. So it will take longer to crack them.

Examples to increase the entropy would be, instead of just using upper characters. Like, in the first example, we now can also

use lower characters. Numbers and a special character, which will increase the entropy. Next up, hashing. Let's

take a look at one-way hash functions. As the name, suggests one-way, hash functions are a one-way function, theorem mathematical

function, which cannot be reversed. Enter usually used to either store passwords or create check sums over

files. Daddy often used to store passwords, but they can also be used to create check sums over files or other things.

There is also not just one hash functions. There are multiple hash functions with different strengths and

there exists. Bad ones like md5, which is more or less cracked and broken and better ones like sha-2 or

shot three. Well interior shot. Three would be better than shot to not many browsers and companies and

tools have an implementation for sure tree. So we are stuck with shot to for the meant in theory. She'ii

would be better than shot too. But it doesn't have a large support right now, from bigger, companies and tools. So we're stuck with

chat to for the moment. A hash usually looks like some cheese fries and you cannot determine the output and

input from the hash alone. If I have to completely different word, like banana and test. This will also be completely different except for

the length, which will always be the same length. In theory, if you change one or two bits of the input that

has should be around fifty percent different than before. I say in theory because in hashes, like md5, there

are hash collisions, which means that two different words will create the same, hash you can think of hash functions.

I always like to bring up the example of hash browns when talking about hash functions. If you have the same potato, the

exact same potato two times and you make the exact same steps to prepare. The hash browns, you will get two hash

browns that are identical or even the same. If you use another potato, that is completely different. And to the same steps.

You will get the completely different hash browns. Also, you cannot convert your hash browns back into regular potatoes. Let's

take a look at cryptography. Now, cryptography has three main goals which make the acronym CIA confidential.

Confidentiality. Only for the ride eyes. Integrity. The message I sent is the same, you will receive no attacker has edited

the message, I sent you. Authenticity, if a message tells that I sent it, I should actually be the real Center

of that message. Now, what does cryptography have to do with passwords first and foremost passwords are never encrypted

or never should be passwords are hashed. And ashes are not encryption because any friction can usually be reversed

and hash not. But passwords are used for encryption. Like, if you have ending password-protected sip, it's basically just an encrypted

sip. Password-based encryption will not be the topic of this course, but with the methods and strategies and

tools from this course, you can also crack encryption which is password based something like a password protected. Zip file can be cracked with

the tools and methods, learn in this course. Now, let's do a quick recap. We took a look at entropy, which is a measure of information, the

higher, the entropy, the more Cure the password. So our goal is to make passwords that I have a high entropy to an

attacker needs more time to crack it. We have taken a look at hashing. And what hashes are.

Usually a one-way function to scramble passwords and store them in a safe way. Cryptography, cryptography should not be used

to store password. You do not encrypted passwords and store them in databases. You hash them store them in databases and

when a user tries to log in your hash, what he enters in the password box and compare the two

hashes. Encryption and decryption will not be the topic of this course, we will focus on passwords and

how to exploit them and how you can prevent that and that's it for this video. Thank you for watching. In the next

one will be setting up the environment.

**3. Setting up the Environment**

Hello World. I'm Janet and welcome back to the Hands-on Password Cracking and Security course. In this video, I will show you how to set up the environment for the rest of the course.

First of all, make sure you have enough power, memory, and storage on your machine. Otherwise, you won't have enough disc space for the rainbow tables VMs,

and not enough power to power up the vm. You can do it with four cores and four gigabytes of ram,

but it'll be extremely slow, especially the cracking, which needs a lot of power will be extremely slow. Now you can go download and install the virtual box software,

which you can found on virtual box.org. They also have guides on how to install them on different oss, and after you're done with that,

you can continue this video and we can download and install the Cali VM together. Then clone the GI repository with Tagable passwords and we are ready to crack

our first password. You can download the Kali Linux images from their official website, which is cali.org/downloads. Once you have downloaded the EOP file,

you can create a new machine in virtual box. Name it Cali. Choose Linux. And it doesn't really matter what we pick here,

but I think we will go with up on two. Go on to the next step. We will give the machine two to six gigs of ram.

Depending on your configuration, I'm giving my machine eight gigabytes of rem. Also, we want to create a virtual heart disk,

a vdi, and give them a fixed size. Now the size of the disks depends if you want to have the rainbow tables

in the machine or outside of the machine. I will give it around 30 gigs because I will have the rainbow tables on my local

machine instead of my virtual machine. Once the machine is created, we can start it up by double clicking it,

and then it asks us to select an disk. We will add our Cali ISO as optical disk and choose it.

Once we edit the ISO as an optical disc, we will be able to choose it from this list and click start,

which will start the Cali installation. You can choose whichever installer you want from here on, you just need to follow the Cali installation.

You can pick the language and click continue. Select where you're from, what key map to use, and the rest will be done by the installation wizard.

Once the wizard is done with the setup, we can configure our host name. I'll just pick calling for this one and continue the main name.

We can live empty. Then we can set up a new user, which I will call Senate, but you can choose your own username

and choose a password. Make sure it's one with a high entropy. Next, we configure the clock and let the wizard work again.

Once the wizard has found our virtual discs, we can just use guided, use the entire disk install, continue, continue, continue,

right, change to disk. Yes, and continue. If you have a proxy, enter your proxy. If not, just continue. Then a little bit of more waiting

on this screen, you can just select all the that's default and continue picks. When installing the group boot loader,

select this only device. Continue. Once the installation is done, you can log in with the credentials that you set in the installation.

After the installation, reboot the machine and log in with the credentials that you set. During the installation, open a terminal and make a directory with MK d r

and call it passwords. Change into that directory with cd. Then clone the repository from the course page into this directory,

and you should have some password files. And that's it for the setup. In the next video, we will crack our first password. See you next time.

**4. Cracking Your First Password**

Hello world. I'm sending it and welcome back to Hands-On password cracking and Security. Today. We cracking our first password to

crack our password. We need John, the Ripper, the rapport with downloaded during the setup last time, then we

can start tracking. What is John, the Ripper? John the Ripper is an open-source password cracking tool.

We will use this tool for various reason. One thing that it's already installed on Carly. It can do the major

attacks like dictionary and Brute Force attacks. And it can crack many different password, hashes and types. The tool itself

is used in the console and you call it with John some options. And then password file.

The password file is the file that stores. The hashes that we're trying to crack. John, the Ripper features multiple different

cracking modes. One is the single crack mode. There's the word list mode, the incremental mode, and the external mode.

We will be mostly using. We're at least mode and incremental mode. Incremental mode is a basic Brute Force attack in which will go

into further in the next section. Well, at least mode as the name implies is a mode where we use a word list or

dictionary to crack the password, also known as dictionary attack, which we'll look at later in a later section. For

your first cracking. You will use a Brute Force attack. So let's get started. Crack your first password, last time, we

cloned a git repository which contains a lot of password files changing to the directory of that repository and then look for the easy

directory. In the easy directory, there will be a file called md5 passwords. You first passwords will be in

that file. So your exercise right now is to try to crack the file with John the Ripper using

the incremental note. If you struggle with John, you can use John minus H, or men, John, or just Google your problem.

And you will find a solution, but to make things a little bit easier for you since we're starting right now. This is

the comment you have to enter John dash dash incremental which will tell John to use the incremental mode. And then

you enter the password file. All on the same line and John will start cracking. The password file.

Just enter this command all on the same line, in your comment prompt using the right file. John dash

dash incremental tells John the Ripper, the use the incremental mode, which is a Brute Force attack, then enter the path

to your password file, which will be the easy, a md5 password file and let it run. John the Ripper will then crack the passwords for

you. If you know what hash function has been used, you can even speed up the process with

an additional option. But which one I will tell you in the next section, but you can already Google for it or try it yourself. So,

just try to crack the file with John incremental mode. Using the EC m, d5s words. Once you've cracked the passwords, you can see what

the passwords are by running. Another command, which is John says Dash show, password file. This will list all

the broken hashes with their corresponding mean, that's it from today. Have fun with your exercise and good

luck. Next time. We will take a look at Brute Force attacks and go more into detail with John the Ripper.

**5. $3.5 Million Cybersecurity Scholarship by EC-Council**

Did you know that you can be part of the lucrative cybersecurity industry? Even top companies like Google, Microsoft, Amazon IBM,

Facebook and Dell all hire cybersecurity professionals. The cybersecurity industry has a 0% unemployment rate. The average salary for an entry level cybersecurity job is

about $100,000 per year in the United States. Furthermore, you don't need to know coding and learn from your home,

and you get a scholarship to kickstart your career. Apply now, EC Council is pledging a $3.5 million CCT scholarship for cybersecurity career starters.

Scan the QR code on the screen to apply for the scholarship. Fill out the form.

**6. Chapter 1 Quiz**

1. How are passwords stored?

* As hashes
* Encrypted
* Plain text
* All of the above

2. Which one of the following is true for hashing functions?

* They encrypt plaintext
* They are reversible
* They cannot be reverted
* They are not complex

**Chapter 2: Brute Force Attacks**

**1. What is a Brute Force Attack and How Does it Work**

Hello world. I'm sending it and welcome back to the Hands-On password cracking and security course. In this section. We will take a

look at Brute Force attacks. We will discuss what they are their pros and cons and how to perform them. This lesson will

cover, what Brute Force attacks are. In short brute force is just trying out every possible combination of characters. Ashton

with the right one way function and compare it with the hashed password. Until we find the right one, the pros of a Brute

Force attack with a Brute Force attack, the password, will eventually be found. Even if the password is just a bunch of random

characters attacks, like dictionary and rainbow table attacks. Could Now find such a pass. It's because they won't be in any

word list. It's pretty easy and the straightforward attack, which is also the reason why we're going to learn it as our first method.

You only need a password cracker and no additional data like a worthless the rainbow table to perform it.

And last but not least with brute force. It is possible to crack every password with every hashing algorithm.

If we have enough time, the cons are that it can take a very long time to perform and needs a lot of Hardware resources. We

trying every possible combination after all. But we can reduce the time. We need to break the password. If we have some information about the

structure of the password. If we have no information about the password, we need to try every possible combination of characters and length. This

means we have a lot more combinations and the higher entropy than when we try to crack it

with some information. We would have to try all combinations, consisting of lowercase letters, then all consisting of uppercase

letters, then all combinations, consisting of numbers. And then we have to combine each combinations with each. Plus,

don't forget about special symbols and characters. This will grow our possible combinations. Immensely. Usually websites and companies have a password policy or

some password requirements in the image. On the right side. You can see such a password requirement. This can

help us reduce the time needed to crack a password because we only need to try combinations within

the requirements. The combinations will consist of lower letters, uppercase letters and numbers with lengths ranging, from 8 to 30.

32 characters, we don't need to try combinations with special characters Encore combinations smaller than 8, characters or bigger than

32 characters since they're not a requirement or not even allowed. Additionally, humans are lazy and often just try to use the minimum requirements. So

we could even try only eight to nine characters combinations, with one, uppercase letter, one number, and 6 to 7 lower.

His letters in the best case. This could even lead to a pattern if we can find a pattern breaking the password will become much

easier for example. Credit card pin patterns are typically four to six characters long and only consist of numbers. Remember,

our goal is to use the information to lower the entropy from our perspective and therefore reduce the amount of possible combinations

for a pass. Word, to give you some numbers. Let's look at an example. The password we trying to crack is p. 86 to,

if we have no information about the password. We have to try a lot of combinations. Our entropy will be around 25 bits and

we will have to try about 60 million combinations just for all possible. One, two, three, and four character passwords.

But if we're given some information, for example, one uppercase letter, and three numbers, we can reduce the number of possible combinations to

under thousand that will give us an entropy of 10 bits. Just to show you some more numbers in the previous example, we

could reduce the entropy to less than half of the original entropy by using information, about the structure of the password roughly

said, this allows us to break the password in half the time, depending on the algorithm to recap. What we learned in this lesson. Brute

Force are simple and easy to use and can crack any password and algorithm. But they can also need a lot of time

and resources. We also saw that we can reduce the time to crack the password. If we have some information about

its structure, and the next lesson. We will take a look at the tools and commands. We will be using to perform our attacks.

**2. Tools &amp; Commands for Brute Force Attacks**

Hello world and Senate and welcome back to the Hands-On, password cracking and security course, in this video. We're

going to discuss the tools and commands or Brute Force attacks. There are multiple tools. We can use to perform a Brute Force attack. We

can even do it manually. Josh Gad is a GPU optimized password cracker. But since we're working in a VM, we will

use John the Ripper. Also, we already know John the Ripper from the first section, and it gives us the possibility to configure it

to our needs. But why? And what should we configure? The white part is easily answered as we saw in the last lesson, we

can reduce the time. It takes to break the Word with some information about its structure, we can configure John to use that information. So

what can we configure? We can set the password length and what type of characters to use the John incremental mode, which

we used in the first section to create. Our first password can be run in different modes by

default. John will offer you four modes all Alpha digits and unknown? They all try password. Things from one to a but with different characters.

We'll try all ASCII characters and is used by default. Also, we try a lower case letters from A to C. Digits,

will try all numbers from 0 to 9, which is perfect for credit card pins and all numb, combines Alpha and digits modes. But

John also offers us the option to create custom modes. This can be useful. If we want to try other characters in length, then

with the four moons that come with. John we can add custom molds to the / editing /, John /. Jung confide.

Let's say we want to try passwords with the lengths from 1 to 9 and lowercase letters.

But the default mode alpha only tries passwords mid lengths from 1 to 8. We can make our custom mode for that the

name in the brackets tells, John that this is a mode for the incremental mode, with the name, lower 19 here

is an example of such a configuration for a custom mode. The mode will be called Lower 19 and is used

in the incremental mode, the name, in the brackets. Tell, John that this is a mold for the incremental mode,

with the name, lower one night. File is the character file that this mode will use. John comes with the bunch of different character files

in this configuration. We're going to use the lower file which contains all lowercase. Letters, Min and max length

are the minimum and maximum length of the passwords and charm count is the amount of characters. That should be used. Here is 26

since the alphabet only contains 26 ladders. Now, that we know how to configure John, we will break a weak password, using a configuration, in

the next lesson.

**3. Cracking Weak Passwords**

Hello world. I'm Senate and welcome back to the Hands-On password cracking and security course in this video. You will break your first week password with the

custom configuration. Why are we bothering with weak passwords? They're easy and fast to crack. And if the random

there only breakable with Brute Force attack, it's where they Shine for this exercise. The password will be FF F

C6. This password is not to be found in any dictionary word list, rainbow table, which makes it the

perfect example for a Brute Force attack. This password can of course be broken with the default all new mode. But since we

know it's exact length, we can make a custom mode that could break it even faster. You provide the two files in the git repository that

we count. From the first lesson. There are two files. One is called Arnhem FF C 6 md5. And the other is

custom FF C, 6. And the 5, they both contain the same md5 hash of the password, FF C

6. But you have to crack both of them. Use the unknown mode for the unknown file

and the custom mode for the custom file, and compare how long it took for both of them, to break the password.

Have fun with the exercise. In the next video. We will crack a real password that was once used

by a real person.

**4. Brute Forcing Your First Real Password**

Hello world and Senate and welcome back to the Hands-On password cracking and security course. And this lesson, you will crack your first real password using

John and a custom Brute Force Mode. What are some common passwords that we will find in the wild. The more advanced users will

have a password manager and generate a random password. Those are the ones that are impossible to crack with dictionary attacks if they haven't been dumped or

leaked. Common password is usually one or multiple words from a language, like, banana pineapple. Or don't hack me, but

the most common type will be a password consisting of one or more words and the minimum requirements

for the webpage, like one number and one upper case letter, some examples are Jesus want to three password zero and

werewolf written in Elite speak? We will take our passwords for this exercise from the Roku where it lists. The

rock keyword list is a list of the most commonly used passwords taken from Dom's. We will try to crack some of the most commonly used passwords.

Okay. You can find the files for this exercise. In the repository, in the directory, Brute Force, real

life. There are four passwords for you to crack, but eight files. There's always one version of the file where you

have the information about the password and one where you don't have the information. For example, if we look at password free.

This word tree has a version without information and one that tells us it has 428 lowercase letters.

Also, we are trying to crack different. Hashing, algorithms one is md5, which would be pretty easy and the other is shot, too.

Each password file has the hashing function in the name so you don't have to guess it.

To set the hashing algorithm in John, just at the format option or shall to at 4 months. Raw Dash sha-256.

and for md5, And for md5 at the format raw Dash md5. After that, you can add your

After that, you can add the incremental mode and follow it, with the file name. Try making custom modes and comparing how

the performance is between the version with information and the one without f-fun cracking the passwords in the next section. We

will look at a faster method for such passwords dictionary attacks.

**5. Chapter 2 Quiz**

1. What's the name of the tool used for brute force attacks?

* Henry
* John
* Jack
* Bob

2. What can we configure in our tool?

* Language, Speed, Hash function that was used and length
* Length, Characterset, mode
* How many CPU and GPU Cores to use
* All of the above

3. Which one of the following is the best hashing function?

* MD5
* TGF65
* SHA2
* None of the above

**Chapter 3: Dictionary Attacks**

**1. What is a Dictionary Attack and How Does it Work?**

Hello world. I'm Senate and welcome back to the Hands-On password cracking and security course on Code. Red in

this section. We're going to take a look at dictionary attacks. Let's start off with our first topic. What is

a dictionary attack? And how does it work? A dictionary attack is very simple. In theory. You take a

dictionary or a list of words. Words from a language and hash damn. When the hash matches, with the password you're trying

to break, you have found the password in a typical dictionary attack scenario. You will have a list of words. Those

words can be from the English language, like, literally using the Oxford dictionary, and trying every word or

they can be a more nuanced and optimized list or passwords as We will see in some examples later on.

As I already said, in this attack, you'll be hashing every word. So this will take longer depending on the amount of words that

you're using. As we will see in later lessons, you can customize what words to use, add

rules to them. And even modify the words according to a pattern before hashing them. Typically, you will use a dictionary attack.

If you know that you'll be cracking longer words instead of random passwords dictionary. Attacks are based

on a simple. Assumption users don't want or are able to memorize long random sequences of characters.

Yes, they could use password managers. We will see in a later section that they will not solve all your problems. However, users

are able to memorize words, words with modifications and so on. So that's why dictionary attacks are best used

when dealing with passwords that are most likely, single words, or based on words. You can, for example, add numbers or

replace characters with numbers and symbols to make leet-speak words. It also depends on your word list.

If you have a word list, like rock you, txt that stores the most common passwords, that could be even used for for passwords that

are not exactly one word with modifications. Since you most likely will not have random sequences in your word list, unless you explicitly, create

them for the list. You will not be able to break this kind of passwords. It is also difficult to deal with multi-word

passwords. If you do not use the according rules and even if you do takes more time to break

them. Here we have a quick comparison with Brute Force attacks to give you some context. Brute

Force, attacks are best used for short random passwords, where dictionary attacks are the better tool to crack longer passwords that

are based on real words. Or whatever words, Your Dictionary has, however dictionary attacks are a bit harder to set up than Brute Force attacks.

But still not too hard. Now that we have an understanding of dictionary attacks. We will take a look at

what tools and features we can use to perform such an attack in the next video.

**2. Tools &amp; Commands for Dictionary Attacks**

Hello world. I'm Senate and welcome back to the Hands-On password cracking and security course on Code. Red in

this section. We're discussing dictionary attacks. In this video. We're going to take a look at the tools and commands we

can use for dictionary attacks. Our old friend John. And not only be used for Brute Force attacks, but can also perform dictionary

attacks. It's a simple as the command you're seeing on screen. Let's take a closer. Look at the

command from before. They're worthless argument is the list of words that we're going to try and to rules arguments

tells John to apply our rules. We can even apply custom rules. Similar to our Brute Force rules. We can configure dictionary

Attack rules in the John dot count five. Is rules are all in the same section namely

list that rules. And then word list. This is the rule that is going to be used by default. We

can also create our custom section and call it. However, we like, if, for example, I called it list dot rules. This

is the part that has to be always the same. And Then followed, by the name rule name. You

can make your own name and your own. Little sections, so you don't have to try everything every

time. You can use your own section of Rules by passing. The name of the rules. The rules argument namely

here, you see, the example is, the name is real name, and you can pass dash dash rules

equals to rule name. Now, an important thing to consider when doing these rules is there notation? They

have kind of a different notation than the Brute Force attack cruise and we're going to take a look at

how these rules look like in the next slide. You may initially be stunned by the sheer amount of

crazy symbols on the screen. So let me explain them one by one for you. Let's say we want to only check uppercase

alphabetic words with a number or simple punctuation. At the end of the list. We will use the first line, which may seem cryptic

to you and not make any sense. However, the first part smaller than star, and bigger than 3 tells John, to use words

with more than three characters. And no maximum amount. Hence, the smaller than star and bigger than free operator. This

part is called length control commands. The next part is a little crazy. If you have no idea what it means. Now, the

question mark a means to use all letters from A to set, upper, and lower cases. The exclamation point tells John to

ignore this rule, if it contains any other characters. So words like password one will skip this rule because it contains a

number and not only characters. These are called character classes. The next part are called, Simple commands

that you were using converts are worked into an uppercase word. Since we filter out, all the words that only

contain letters, we're bound to use words with lowercase letters at some point, but we don't want to. So we transform them into

uppercase letters. The last part tells John to append a number or special symbol, at the end of the word, it

basically Works similar to rags. You can pass a list of possible characters to append between the two brackets and John

will try all of them. We could, for example, add two numbers with a similar, notation instead of passing each number individually

to the list. We make two lists and pests 029 to each list. So John will try every possible

combination with those two numbers, but be careful as we have seen in the first couple of sections, entropy can

grow fast if you add more possible combinations and it will Take longer to crack a password. Also,

make sure to take into consideration that this Rule and this possible combinations will be applied to all words that

match the criteria. So this could potentially generate a lot of possible words and it will take you longer to crack the password. Thank

you for watching this lesson in the next lesson. We're going to take what we learn from this one.

Set up a word list and start cracking passwords right away using rules and the word list.

See you there.

**3. Crack a Password Using a Dictionary Attack**

Hello world. I'm Senate and welcome back to the Hands-On password cracking and security course on Code. Red in this section. We're

going to take a look at dictionary attacks. In this video. We're going to use the word list to crack

some passwords. So first, how do we use a word list? In this course, we are going to use the

Rock, You txt word list, which contains a bunch of commonly used passwords.

In order to do this, start up your Kelly machine and open a terminal. Then change the directory into user share word

lists. Here you will have a bunch of directories and word list, but we are interested in

the Rock, You txt. As you can see by the file ending, the rocket EXT is not in a text format. As

you can see by the file ending. It's not a txt file. It's a gzip file. So we first want to unpack this file.

You can unzip this file by running sudo G, unzip. Rocky txt .gc.

After you unzip this file, you'll find the text file in this directory. Now, if we take a look at this file, we can see that it

contains a bunch of different passwords that are commonly used like abc123 Nicole. Danielle baby, girl, lovely. I

love you query, tea, and other words. We will be using this list of words to crack some passwords.

You may remember the command from the second lesson in the section. Which is used to use the word list with Drew's and the password file.

You just have to replace the password list file with where it lists equals user. Share, where

it lists Rock. You Dot txt. And this is how we'll use. A word list. Now, this exercise wouldn't be

complete if it didn't have some passwords. So let's take a look at where to find them and

how to crack them.

The passwords to crack can be found in the Repository. In the dictionary folder. If you go into the dictionary folder, you will find

poor password files. Now this are the exact same passwords that we already cracked with the Brute Force attacks and

some Roots. But now, since all those passwords were part of the rocket exe file, we can use

the rocket exe file as a word list to crack this for passwords. Try to perform a dictionary attack on all those for passwords

and maybe also try to remember how long it took for The Brute Force attacks and how long it took for

the dictionary attack. So you have a little comparison on what performs better in, what attack. I hope you will have some fun cracking those passwords and

I will see you in the next lesson where we're going to take a look how to further use

rules to crack passwords using dictionary attacks.

**4. Crack a Password Using a Dictionary Attack and Rules**

Hello world. I'm Senate and welcome back to the Hands-On password cracking and security course on Code. Red in this section. We're going to take a

look at dictionary attacks. In this video. We're going to take a look how to further use routes to crack passwords using dictionary

attacks. We have learned from previous lessons that different rules can be used to apply Transformations and filter to

our word list. So, how can we do this?

You just need to open the /. It is a /john /. John dot confide in your favorite

editor.

Then you can search for the according section namely list rules word list. And you can see that there are already a couple of

rules in this file. Now, you can still use this rules or you can create your own section.

Like this and add your rules beneath it.

One important thing is, you might want to open this file as a root user. So just type

in sudo. The name of your favorite editor. and, The puff to the John dot con file.

Sorry.

You might need root user privileges to edit this file. So just enter sudo the name of your favorite editor.

The path to the file hit enter and you have opened the file as root user. One little hint. If you're unfamiliar with him,

you can also use Nano or a console editor. Or you can use. Mouse pad. Or a standard editor with a gooey.

So, what passwords will you encounter in this exercise? Just to prepare you for the rules? You will be writing. These

are some examples. These are not the actual passwords that you're going to break. But these are the classification of the password. So one is

like football, a lowercase letters, only password. Then the second type will be like, Terraces, which is just secret reversed. So

you will have to reverse the words. And we have something like Jordan 23, which is a lowercase word with two

numbers. We also have something like, exclamation point slick not 777 which is basically a special symbol in front of reversed. Where

it in this case, it's angels and then three numbers at the end. Then we have something like flowers where you'll have

to replace a hose with zeros. And also something like Amanda, which is to capitalize the first letter and add a number suffix.

If you go into the repository, you will find the directory called. Dictionary rules. And this directory, you'll have a couple of files which

contains md5 hashes of passwords. Now, every file is basically an explanation of the rule. You'll have to write

to crack this file. Efficiently. For example, you can see lowercase, add two numbers will have a word, that's lowercase,

and contains two numbers at the end. And capitalized at number will be a word that starts with an

uppercase letter. The rest will be lower case and at the end of the word will be a

number like Amanda nine. In our example.

So your exercise for this video is to create rules for the passwords shown before but be cautious. The exercise passwords

are just similar types. Not the exact same passwords. Let's see how efficient rules. You can write and how fast you

can crack. The passwords. Thank you for watching the section about dictionary attacks. In the next section We're Gonna Save Computing Time by

using rainbow tables.

**5. Chapter 3 Quiz**

1. When should you use a dictionary attack?

* Single Word Passwords
* Random Sequences
* Multi word passwords
* Hashed Passwords

2. Why should you use rules for dictionary attacks?

* So we don't break the law
* We can use password requirements from the page to reduce the time to crack passwords
* So that our tool knows which language we're using
* All of the above

**Chapter 4: Rainbow Table Attacks**

**1. What is a Rainbow Table Attack and How Does it Work?**

Hello world. I'm Senate and welcome to the Hands-On password cracking, and security course on Code. Red. In

this section. We will discuss rainbow table attacks. Let's start as usual and discuss what rainbow table. Attacks are and how they

work. Rainbow table attacks work similar to word list, but they need less computing power. They were intentionally created

to consume less computing power at the cost of using more space. So why should we use space instead of

computing? Power? It's simple. If we compare the pricing for Space versus computing power space has always been the cheaper option.

Even nowadays in Cloud providers like, Google Cloud, you pay around 20 cents for one gigabyte of storage per

month, depending on location for the cheapest standard E2, machine with 8 gigabytes of RAM. You pay around 70 cents per

hour and it doesn't even have a Jeep, you to accelerate a cracking. So usually storage is cheaper

than Computing. Which also mean, there is usually more storage space than computing power available. Additionally, it can be faster especially

with the methods that the rainbow tables use, which we are going to take a look at in this

section. And last, but not least, you can use the spare Computing, resources, to do other stuff. Maybe even cracking another

password in the meantime, depending of your actual Computing resources. No, sir, rainbow table attacks. Rick similar to dictionary attacks,

but use a rainbow table, instead of a word list. Let's think of a rainbow table as a table, with two columns, in the first one, you have

hashes in the second one. You have to plain text for the hashes. All you need to do now is search for the right hash in

column 1 and return. The plain text from column to. This is basically what a rainbow table attack does.

Now that we cleared some Basics, let's dig a little deeper into what rainbow table attacks are. Rainbow table attacks. Use rainbow tables. These

tables are precomputed listings will create one later in this section. Well now, just think of them as tables from one column is the hash. And

the second column is the password. This is an oversimplification and not entirely correct, but more in the next lesson.

So, now we can work in Reverse instead of trying every possible word or combination and hash it. We just check if

we find the hash of the password in the first row and return, the second row, this offers

a couple of advantages. The tanker has a reusable table mapping. It can even abuse hash collisions and

all he has to do during the attack is a simple lookup, which is very fast. But they come at a price, all

the Computing has to be done up front. The hashes have to be calculated at some point and

store it. But once you've got that out of the way, if got yourself a rainbow table with

that table, you can start cracking multiple passwords. For example, if you want to crack 10 passwords, you don't need to

calculate the hashes for every combination or word in a list every time for each password. You already did that once and can just

look it up in a Brute Force attack. You will always have to start at the beginning of all possible

combinations and try them over hash time. Compare them with the hash of the password and do this for every password. So in this example, you'd

have to do it 10 times. Whereas in the rainbow table attack. You just have to do it once

and then look and times where you can find the hash. With the rainbow table, you call clean, it all hashes ones and

now you just need to find the right one. I hope you got the gist of what rainbow table attacks

are in the next lesson. I'm going to show you how this tables really look like, since they're not just simple to column tables.

And I'm also going to show you how you can build your own tables.

**2. What are Rainbow Tables?**

Hello world. I'm Senate and welcome to the Hands-On password cracking, and security course on Code. Red in this section. We're discussing

rainbow table attacks. In today's lesson. We're going to take a closer. Look at rainbow tables in

the last video. I told you to imagine the rainbow tables are a table with two columns, one

with hashes and the second with the corresponding plaintext, but that was just an over simplification there, more than

that. Storing, all hashes will take too much storage. Rainbow tables are, in fact, the compromise between

pre-computation and low storage usage. First it uses what is called a reduction function reduction functions map a plain

text value to a certain hash, but they do not reverse. The hashing operation as we saw. This is

not possible reduction functions are a one-way function like hash functions. They can be as simple as taking the first not five numbers

from the hash. So we first hash to plain text with the given algorithm, for example, md5 and then take the

first five numbers from the hash. This generates another plain text from the hash, which is not the original plain text, but the completely different

one. This reduction functions are used to build chains. The rainbow tables consists of chains for each, plaintext a

chain starts with the random plaintext, ashes it. Using the hashing function. Then reduces the hash to another plain text ashes,

the new plain text. And so, on the table Only Stores, the starting plain text and the final hash you choose to end with. And

so a chain. Millions of hashes can be represented with only a single starting plaintext and one hash.

The final table is a lookup table with one, plain text and one hash, but they are not the corresponding hashes for the plain

text. So you may ask yourself. How can we use this table? It's useless. But it's not.

We can use an algorithm, you first need to look for the hash in the table. If you can't find it you reduce the hash into another

plain text and hash the new one. Now, you try to look it up in the list of hashes.

When the hash matches, with one in the table, you get the starting plain text from the table, Ash

and reduce it until you find the corresponding hash with its plain text. This way. You can check every hash stored in

the chain backwards from the last column in the chain. You check whether the hash exist in the last column in

any of the chains by reducing and hashing. The given hash once, and then check it against the chain ends.

Or the second last column we reduce and has twice and check for the chain ends or the third, you do it three

times and so on. This sounds pretty complicated. But luckily for us. We have a couple of tools at our disposal to

create and use the rainbow tables. Let's start by creating one. Please take a minute to study the arguments to

the tool.

To create a rainbow table, we can use our tejan from the rainbow Craig suit, the example commands, generates a rainbow table with

thousand chains. Consisting of lowercase letter passwords with the length from one to seven. Don't worry, if your to over hand by it. We're

going to generate a table in a later lesson in this section. This one argument that I left out here namely Dash

bench, which will determine how long it will take to generate a rainbow table. I hope you have some idea how rainbow tables work

and how we can build and use them. Don't worry. There will be an exercise at the end of the section where I'm going to show you step-by-step how

to generate and use a rainbow table. But first, we need to get familiar with the tools and commands to perform such an attack in

the next lesson.

**3. Tools &amp; Commands for Rainbow Attacks**

Hello world. I'm tenant and welcome to the Hands-On password cracking, and security course on Code. Red. In this section. We will discuss rainbow

table attacks. Today's lesson will consist of the tools and commands used for the rainbow table attacks.

So the tools that we are going to use are all tools from the Rainbow Crack suit, it

comes with a lot of tools, 12 is called are crack, which is the tool. We will use to crack passwords using rainbow

tables. Another tool is called our tejan which we took a look at in the last lesson.

It allows us to create our own tables. Additionally, the Rainbow Crack suit also offers more tools like tools to

sort and convert rainbow tables. A big bonus from this tool is that it's pretty easy to install on Kali Linux.

Before we can even start, installing it, make sure you have all the repository lists in your Source list for opt

to do this. Run the command grab Dash V, single quote hashtag single, quote slash Etc, slash

upped /, sources.list pipe, sort s, you If not edit this file, as a root user and add this line at the

end of the file. Once you have done it, you can run. Sudo apt update. This will update the repository and the package that

will be available to you. Next, you can run the command sudo apt-get, install, rainbow crack. Since I have already done it, there is nothing for my

Cali insulation to add, but this will add a couple of tools. Namely the ones we mentioned before to your Kali installation.

And that's all for it. You can test out if you have the aforementioned tool by just running them.

If this works on your machine, you have successfully installed the Rainbow Crack suit. Now, we are ready to go, which is exactly what

we will do. In the next lesson. We will take the tools. We just installed, create a rainbow table

and use it to crack passwords.

**4. Crack a Password Using a Rainbow Table**

Hello world. I'm Senate and welcome to the Hands-On password cracking, and security course on Code. Red. In

this section. We will discuss rainbow table attacks. In this lesson, we're going to generate our first rainbow table and use it to

crack passwords. A little disclaimer upfront. Since I don't want to waste your time with creating huge

rainbow tables or downloading them. We're going to crack easy passwords. You can then further use the knowledge and tools to create

a more sophisticated table and use it for stronger passwords. So what do I mean by simple passwords? We're

going to create a rainbow table for passwords consisting of exactly four numbers. I have provided you with a couple of passwords to crack with

this table. Additionally, each password is present twice once to crack it with a Brute Force attack and

wants to crack it with a rainbow table. This way. You can compare the two. Of course. This is not a real scenario and the results

May differ when using strong passwords. Now, let's go on to create a table. To create a rainbow table. Use the tool

are tejan since the passwords. I provided, our md5 choose the md5 hash functions set the charset in

America, since all passwords are numbers and for the Charmin and Char max amount select for since, they're all exactly four

characters. The rest, you can just copy and use it to your discretion and tweak a little bit with it.

If you don't get the results that you want. But basically, we are creating a ton of chains with

all the hashes in it. Now, I've created the rainbow table. So with this rainbow table, you're going to

crack a couple of passwords. In the repository, you will find a directory called rainbow. If you go into that directory.

You will find three files. If you go into the rainbow directory, you will find 3 files. One is called example. One is called

passwords brute. And one is called passwords rainbow. You will use a Brute Force attacks for the passwords

brute file. And the rainbow table attack for the passwords rainbow file. Each of these file contains the same md5

hashes and each of them contains 10 of them. So let me show you an example. The

password I'm going to crack is one, two, three, four.

That rainbow tables, we created can be found. At user share rainbow cracked. Here you can see I've created a couple more, but

I think all you need for this exercise. Is this one which we created together?

Now, I'm going to show you how to crack a list of hashes. You enter pseudo aircrack, then select the path with all

your rainbow tables. In this case. It's user, share Rainbow Crack the shell, and Then followed by the file with

the list of the hashes.

But before we can do that, we have to sort the rainbow tables. We can do this with to do our t-shirt and the path to

the directory. Containing the tables.

as you can see, I was not lucky enough to find it, but maybe

I went so far as to create 20 rainbow table files. You can do this. By changing this number and increment. It one

more for example from 19 to 20. So I execute this command 20 time. One time with zero

and with one, then with 2 and so forth. Then I sorted the tables with Artie sword and tap have

to the tables. So, let's see if this is enough. To crack our example, password. Apparently, this was not enough.

So apparently the current version of are crack that we're using in. This caliber machine has some issues with md5 hash

list. So what we can do at are negatively is pass it. The hash directly in the comment line in this example. I

use the same command as before, but I changed that Dash L to a dash H and followed

it by the Md5, hash of the password 123. and as you can see, It found it very quickly.

Now, what can you do now with this information? You can open the hashes for the rainbow attack

in a text document. And execute this command with the hashes from the document. By copying.

And pasting them through the command line argument. And the first one is one, two, three, four. So

now go ahead, copy and paste. This hashes. And try to crack them all. So now you can go ahead and try

to crack all of the passwords. So now you can go ahead and crack all of the passwords.

And that's all for the section, on Rainbow table attacks in the next section. We're going to take a look at the

downside of passwords. I will teach you about other vulnerabilities and negative aspects of passwords that you might want to consider. I

hope to see you there.

**5. Chapter 4 Quiz**

1. Why should you use a rainbow table attack?

* The search is faster
* It prints out colorful results
* You should never use them
* This attack protects you from getting caught

2. What can't you configure when creating rainbow tables?

* Which language to use
* Password length
* How many files it should create
* None of the above

**Chapter 5: The Downside of Passwords**

**1. What Will be Covered in This Section?**

Hello world. I'm Senate and welcome to the Hands-On password cracking, and security course on Code Red.

In this section, we will cover the downsides of passwords and miscellaneous winner, abilities of passwords and their Miss usage will

also take a brief look at the alternatives to passwords in this video. I'll be laying out the topics that will

be covered in this section. In this section, we're going to cover different attacks on password-based authentication and how they can be prevented. We

will further discuss why passwords are not the best choice for authentication at what alternatives we have. The text.

We're going to look into our credential stuffing password, spraying and keylogger attacks. We will learn how these attacks

will work in theory, but not actually perform them. A because it will be too hard to

perform them alone without an actual Target, or you wouldn't have the same benefit and be the point is

only to show you more attacks that are not necessarily hash-based, but circumvent the whole hashing part 5 using other methods.

We will also see how to prevent those attacks except for the keylogger. Attacks remedy for keylogger. Tax will be

covered in the next section together with the rest of the remedies. At the end of this section, we will have a little discussion

about passwords. Why am I not such a huge fan of passwords and what we could use

instead we'll also see why we're currently stuck with password authentication and not use other methods or why

we only partially use other methods. That's all for this lesson in the next one. We will discuss what credentials stuffing is

will also see how it has influenced companies in the real world and how you can prevent this from happening to you

or your company.

**2. Credential Stuffing**

Hello world. I'm sanded and welcome to the Hands-On password cracking and security course on Code. Red. In this section. We will cover

the downsides of passwords & misc, Elena's vulnerabilities and the Miss usage of passwords. We will also take a brief look at alternatives

to passwords. In this lesson, we will discuss what credentials stuffing is. We will also see how this

has influenced companies in the real world and how you can prevent this from happening to you, or

your company. So what even is credential stuffing? Presidential stuff in is a subcategory of Brute Force attacks. It doesn't require hashing algorithms. But

instead tries a list of commonly used username and password pairs. This is possible because most users use

the same password and username for every application. They use. So, once a company gets hacked and the username password, pairs dumped in, plain text

somewhere, an attacker can use those pairs to hack into other accounts. For example, if you were to have the

email Senate at example.com with the password 1234 56 as your Facebook credential, but also, as your

credentials for, let's say, Twitter, you could be a target for this attack. If Twitter gets hacked, and all passwords. Get dumped an

attacker can now enter into your Twitter account, but not only that because you're using the same password.

And username on Facebook and attacker. Cannot try the pair from Twitter on Facebook and login as you.

One big problem here is that sysadmins cannot force users to have unique passwords for every platform. So in the worst case, an

employee might be even using the same password and or username that was already leaked or dumped or

example in 2016 or 3 billion credentials have been dumped and the number is constantly growing. The attack itself is also pretty straightforward.

You first need to find a dump and then use a tool like selenium to automate to logins with the list. You

just got after a while. You will have cracked some login. So already know hashing needed. So, how do we solve this issue? Well,

one is for sure. We can't force the users to create a unique password for every platform, but we can do other

things. We can force them to use a random number for every log. Instead, what something like, multi-factor Authentication.

The best option to remedy this issue is to use multi-factor authentication. In fact, you can use something like a tense or Google

Authenticator and similar apps. Tell me banking platform even have a unique second factor. For example, I have to enter

my username and password and then use a box where I have to put my heart in, enter

my pin, and then I get a code to enter for the login. If for some reason MFA or multi-factor authentication

is not possible, we can also try to stop the attacker by using captures. This may not be as effective as

MFA but still better than nothing. This will reduce the amount of attacks because they have to find a way to

automate the select the traffic like thing from Google. We can also retroactively, do something with IP blacklisting and checking for leaks.

If we monitor suspicious activity from one IP, we can block that one. We can also use the same method as the attackers to

check. If any company employee or user has leaked passwords. We can run a tool that automates the process

and then automatically resets the leaked passwords. And that's all about credential stuffing in the next lesson. We're going to take a look at the similar Tech

bastard. Spraying. It's similar to credential stuffing but works differently.

**3. Password Spraying**

Hello world. I'm Senate and welcome to the Hands-On password cracking and security course on Code, Red dissection,

recovering the downsides of passwords. We will also take a brief look at the alternatives to passwords at the

end of the section. In this lesson. We will take a look at password. Spraying passwords praying is another sub attack of Brute Force

attacks. So how does passwords brain work? The attacker will first find a common password, for example, exclamation-point password

123. He now targets a company Network and tries to log in with this password. He tries every possible user name, email

address or account name from the company network with this, common password, and will eventually succeed. The more users the company has the

higher the probability of actually finding a valid credential pair. This attack is very dangerous because the

attacker doesn't get locked out except afterwards when blacklisting his IP, but because he's using a different account every time,

the typical locking mechanisms, like blocking account after five wrong password attempts, don't work. The best option to defend against such attacks is

to use. MFA, multi-factor Authentication. This will keep the attacker out. Even, he finds a valid credentials pair because he

doesn't have the victims phone eyes or finger. Another additional measure can be a strong password policy

combined with regular password resets. A strong policy alone won't necessarily prevent your users from choosing common

password. For example, the password exclamation point password 123 with an uppercase p is also technically a

strong password. It has more than 8 characters, upper and lower case letters special symbols and numbers, but it's

a fairly common combination and usually policies. Not check for that. We can of course increase the

password policy requirements and say that you need at least 16 characters. But the problem is that

after a while there will be 16 characters, passwords. That will be common. So if we can regular reset the

password, the user is forced to change his passwords. So at least he will not use the same common password every

time. Another good way is to educate the user base about this, topic with security trainings and wellness

workshops. We can now start to see that passwords are problematic for one main reason, they rely heavily

on the user or whatever the users pick as pastured and humans, make mistakes and are lazy. So the passwords are usually bad or

just the minimum requirement from the password policy. In the next lesson, we will take a look at keylogger

text and see how they can be used to steal credentials.

**4. Keylogger Attacks**

Hello world. I'm sending and welcome to the Hands-On password cracking, and security course on Code. Red in

this section. We're discussing the downsides of passwords. We will also take a brief look at the Alternatives at the end of this section. In

this video. We're going to take a look at keylogger tags. One of the oldest form of cyber threats,

which is still used today. A keylogger is basically malware on the victim computer that will read all the

key presses and lock them to the attacker. Additionally to just logging the keystrokes to a locked file. An attacker can use a command and control

infrastructure, which allows him to see the key strokes in almost real time. The lockers can also be optimized to recognize patterns,

like, when the user is, using an at symbol and only record or highlight these parts, it can

make finding passwords very easy as you don't have to scroll through the entire log or the log only contains password pairs. Key

loggers also come in different form. The most commonly known one small where keylogger can come a small where

it is typically installed on the victims Machine by means of fishing. The attacker sends a forged email from a company telling

the victim to download install or run some executable to solve a specific problem and take see, cutable will

turn out to be a key logger. An example could be an app on your smartphone. Which promises you to optimize the

speed of your phone, but instead just locks everything you do. It can also come from downloading software from

than internet without checking it on your PC. For example. Some of them even disguise themselves as

anti viruses. So you download it in the hopes that it will make you more secure. But instead you're actually downloading a keylogger. You

can check the files with services like virustotal before opening or running them in a virtual machine to make sure they're not harmful.

USB sticks. Another way to spread the keylogger, or any malware for that matter. Is to uninstall it on a USB stick, like the rubber

ducky and plant them on strategic places. For example, you could add a label with the text salary report 2020 and drop it

somewhere in the garage near car. The user that will go to that car. Well, find the steak in front of his car and

we'll probably be tempted to plug it into the computer, and the stick will do its magic. You can even hide the key logging

program in an Excel file. So he will actually open the keylogger himself and that's all done. Without any social interaction or

fishing by the hacker. And if the attacker wants to increase his chances, Chances of getting into computer,

he just has to drop more USB sticks in front of multiple cars. Ox or another way we lock your keystrokes or

more specifically software and Hardware box. For example, last year, a Logitech bug allowed attackers to sniff all the signals of a wireless mouse. The

attacker could do a man-in-the-middle attack on the signal sent from the wireless mouse to the USB receiver. Not only could be read and

log everything but he could also control it entirely. Meaning he It sent Mouse and even keystrokes. We

are they hijacked signals. If a wireless keyboard from a manufacturer happens, to have a similar book attacker. Could also

lock the key strokes from there. So far we have seen that passwords can have a lot of vulnerabilities and Cave it and

most importantly they are controlled by the users. But what alternatives do we have? This is what will be answered in the

next lesson.

**5. Alternatives to Password Authentication**

Hello world. I'm Senate and welcome to the Hands-On password cracking and security course on Code. Red. In this section. We're covering the downsides of

passwords and at the end which is this lesson. Now, we will take a brief look at alternatives to password.

So as mentioned before, we will take a look at alternatives to password in this lesson, most of the

Alternatives that users will accept are not really Alternatives but rather additions like MFA multi-factor authentication or intense, but

some real alternative exists, like using Biometrics like Windows, hello or face unlock tubes. Or another alternative is to use something different,

like, certificates and encryption keys, or pki keys. MF A&M tents are not really an alternative but

it's easy to setup since all you need. Most of the time is just a phone. Now the use of Biometrics instead of password

has quickly risen in popularity, almost any new phone. Now supports some sort of virus fingerprint or face

can to login. But on phones, there is usually still an option to use a password. In case, the log in

Via Biometrics is not successful. Like when your hand is wet or when your hand isn't actually the one of the owner of

the phone, which makes the use of Biometrics questionable in terms of security. In terms of usability, it's great. If

the technology advances far enough, this could really be the future and we could completely ditch passwords for the most part.

Certificate and keys are by far my favorite alternative to passwords. They are based on math and cryptography and cannot be cracked

or broken like password for the most part. I prefer to use SSH keys to connect two machines over the SSH protocol. Instead

of the typical password login. The problem is that you kind of have to handle the keys, usually yourself and a

bit finicky if you're not accustomed to it and it's not exactly what you would refer to as user friendly.

And certificates are used in websites that use the HTTP protocol. And the problem there is that you have to build a chain of trust with

Ash to start somewhere. Usually your browser comes with the set of prints tailed or you th that you

automatically trust but do you really trust them? And I think that that's an old another topic for maybe even another course, so

I'll leave the question to you. Now, in conclusion, from what we've seen, we're bound to stick with

passwords for a while. Everybody knows how to use them. Although most of them use them wrong because they use weak

and common passwords, but the user is accustomed to this, login screen with username and password and changing that can be a kind of

a hassle. So the most feasible option of all seem to be something like MFA. Where we still use a password, but the user has

to provide an additional Factor like Biometrics or 10 like codes or has to authenticate when the additional app, like with

Google Authenticator app. That's all for the downsides of the passwords. In the next section. We will cover how to prevent their

text learned. In this course. We will also take a look at password managers.

**6. Chapter 5 Quiz**

1. What can you use to stop most attacks covered in Section 5?

* Password policies
* Password resets
* Hashing
* MFA/2FA

2. How does password spraying work?

* Take list of common pairs, use a tool to automate login with those credentials, evaluate the results
* Use a common password and try it on all users
* Install malware on a victims computer
* None of the above

**Chapter 6: Remedies and Mitigations**

**1. Brute Force and Dictionary Attacks**

Hello world. I'm Senate and welcome to the Hands-On password cracking and security course on Code. Red. In this section. We will cover the

remedies for the attacks demonstrated in the course. We will also take a look at different password managers

and how they can or cannot solve our problems with passwords. This lesson focuses on remedies for

brute force and dictionary attacks. They both can be prevented with similar prevention techniques. Let's start with some basic remedies, that

should be easy to implement and can have an huge impact on online cracking, their best practices, and should be implemented, either way. Number

one on the list is MFA, multi-factor Authentication. Since we have discussed, this method, already in the

previous section, I won't get into too much detail on it. As we have seen in the last section, MFA can be used to prevent

an attacker to login with stolen credentials. As long as they are missing, the second Factor like Biometrics or devices, while

this won't prevent the attack itself, because an attacker can find a dump of hashes and crack them offline.

It will make any crack at hash useless. Knocking out the count goes in the same direction. With this method you basically lock an account. After

a certain number of failed attempts. This will stop an attacker or at least slow him down when doing an online, brute, force or dictionary attack,

but it's completely useless against an offline attack since the attacker only needs to try it one time after they have cracked the password. They

will not get locked out. Using capture is also method to prevent online attacks. It slows down, or completely stops,

automated the tags, but if they attack regained the, plain text of the hash offline, this will not help in any way. Since the attacker

is most probably human. You can prevent the worst from happening by disabling administrator and Drew two accounts for your machines, or

at least disable the access to Tim from outside, the corporate Network. Even if an attack defies, the password cracks it

and tries to login. This will stop him from accessing than as administrator accounts. Of course, this is completely useless when the attacker is

inside the network. So, how do we prevent offline cracking? The best solution and probably the only one

out there is to use a strong. Random password is password, should not use words from a language made be an imaginary one or

a real one. There are word lists for the Klingon language, ready to use to crack passwords. The password should also not be a common one.

A password might look safe at first, but if it uses a word or is a common password,

then the security of it will be very low. So, how can we do that? It's not possible for

the most part. We cannot force our users to use random and uncommon passwords that are not based on a word, but

we can enforce some things on them. We can enforce a strong password policy, like 10 characters. At least one number one,

lower case. One upper case letter and the special symbol, but I say users can end will still find.

Common passwords. That can be even based on words like password, which will meet the requirements, but it's still pretty unsafe to

counted. This fact, we can refresh passwords on a regular basis. For example, every three months. If we keep doing that. We

will have a couple of advantage. Once the attack a cracks, a hash, it might be already too late to use it, depending

on the refresh interval. The password might already be changed. If we keep the history of the old passwords users are inclined to

come up with new passwords. We can even extend this to check. If the new password contains the

old, password, username or other information so that they do not just add another number at the end of the password, and call it a day.

This will make the passwords less common as the more users need to change. The less common passwords are

available from all possible passwords. If and only if the user followed, the best practices of not using words and

common passwords, the hashes will be harder to crack but worthless and take much longer with Brute Force

attacks in the next lesson. I will teach you how to prevent rainbow table attacks on your passwords or passwords of

your application.

**2. Rainbow Table Attack**

Hello world. I'm Senate and welcome to the Hands-On cracking and security course on Code Red. In this section. We will cover the

remedies for the attacks demonstrated in the course. We will also take a look at different password managers and

how they can or cannot solve our problems with passwords. This lesson focuses on prevention, methods for

rainbow table attacks, that can be used to protect your passwords or the passwords of your application. The

remedy is mentioned, in the last lesson can also be used to prevent rainbow table attacks. So if you

need more methods, make sure to check out the last lesson first. The only real method to stop rainbow table

attacks is to season your password with either cells or Peppers or both. A salt is basically a string of random data that

is added to the plain text before. It's hashed. It is often stored along the password in a database and this unique and random

for each password. This is very powerful unique, and random. Peppers are similar to salts. A paper is a random data that is added to the input before

being hashed. But the difference here is that peppers are never stored together with the passwords are usually stored in the configuration of

the application or another secure location. They need to be at least 112 bits. Otherwise an attacker only

needs to know. One plain text password to crack the pepper, which could be his own. Usually, the pepper is application wide and

not unique for every password. Why can we prevent rainbow tables with the two methods? With salt and pepper, we can messy. If we

increase the security of our application, if the attacker has no access to the pepper. It will never find the right password. The same is valid for

salts plus salts can result in bigger rainbow tables or rainbow tables that will need to compute more. Generally, speaking salt

and pepper can increase the security for passwords, even if common and weak passwords are used. So, it is definitely something to

Implement either way, as you can, also protect you against brute force and dictionary attacks. Sounds great. How can we do that?

Or the software Engineers Among Us, use the library that helps you implement salting and peppering of hashes without having to create a lot of

additional code or don't Implement with authentication at all. And instead use something like Key Club where you can configure the authentication mechanisms

and even select to use salt. If you have no access to source code and are stuck with an application, try to

find the configuration to either use salt and pepper. If possible that identity provider for the application that will support salt and pepper some

systems and software, even already use salts or peppers. Like the Unix system credentials used in unix-like systems.

So no worries there. With our now, spicy passwords, we can have pretty safe passwords, even if they're weak, but

what about the actual typing of the password? The topic for the next lesson will be to prevent keyloggers from reading your keystrokes?

**3. Keylogger**

Other words unsanded and welcome to the Hands-On. Password cracking, and security course on Code. Red in this section. We're covering the remedies for

the attacks demonstrated in the course, and we will also take a look at different password managers and

how they can or cannot solve our problems with passwords. How do you prevent a key logger from recording? All your keystrokes? That's

what I'm going to enter in this lesson. If it wasn't clear for you by now, I'm a big fan

of multi-factor authentication while it is not an all-in-one solution to all password problems. It helps prevent a

lot of common cases. If you use MFA, the locked password will be useless for the attacker, unless he also owns the

second factor for the authentication. So use multi-factor Authentication. However, there are some more remedies for this. Make

sure to update your systems. Regularly. This will allow you to close security vulnerabilities, that allow an attacker to lock, your keystrokes, like the Logitech one.

We discussed in the last section use a tool like a scramble. You can even use a tool like key Scrambler that will

encrypt your keystrokes into unreadable garbage for the attack this way. Even if you're infected, the attacker cannot.

Gain, the password, unless his malware also attacks such tools. Another tip is to never install cracked

software. Not only, is it a messy security risk? Because it's an unsigned application from somewhere in the internet that anybody could

have uploaded. It's also illegal in most countries and use an anti-malware software or antivirus to protect from

scan against and detect unwanted malware like keyloggers. This will help you immensely to prevent your It's getting stolen by a keylogger. It's

also the most used measure in Enterprise as there are great solutions to use and Deploy on multiple machines and manage them.

If for some reason you cannot use an anti-malware software, you can detect keyloggers in task manager on Windows,

but this can be very hard to almost impossible depending on the malware efficient, malware will manipulate the colonel internal

structure. So as not to appear at all in the list of process. Another way is to use firewalls, use

them to your advantage, block all incoming and outgoing traffic, that seems suspicious or better yet block. Everything and only open

needed ports in the first place. Of course. This is only effective as long as the open ports are

not Deport required by the malware. In this case, block the installation or use of specific ports on the devices itself.

For example, block all traffic on Port, 80 and 443, except for browsing the internet with browsers as

restrictive as possible without interfering too much with the user experience and productivity. In the next lesson, we will focus on General tips

to prevent your passwords from being hacked.

**4. Other Considerations**

Hello world. I'm Senate and welcome to the Hands-On password cracking and security course on code. Red and

the section. We are covering the remedies for the attacks demonstrated in the course, and we will later take a look at password managers.

In this lesson, I'm showing you General best practices lose and tones to prevent your password from getting stolen.

First, off some basic rules. This should be pretty straightforward and easy to follow. And I think most of

you are already doing that. So first off, never store your passwords in, plain text. If you are a user, do not store the passwords in

an Excel or a text file in plain text and then even name it passwords, And if you're a developer, make sure

to only save the hash of the password, never the actual password. Of course, this ties nicely with the next one which

is to not store or writer passwords on visible areas. Never, write your passwords down in the first place and especially not

on postage which you will hide under your keyboard or even attach to the bottom of the computer monitor.

Now, to some additional rules. If you watched my last couple of lessons, it should be clear by now that

you should use a strong and unique password for everything that you use. You should never have the same password on multiple Services computer

or the other devices. I know that can be overwhelming but luckily there is a great tool that can help you manage

your passwords for each service, a password manager. We'll take a closer. Look at password managers in the next lesson.

Basically, all you need to remember for a password manager is one master password, which should of course be random and secure

and strong. So how do we create a random and secure password that we actually can remember? Let me

show. To create a random and strong password. All you need to do is come up with a sentence that has a special characters

and numbers need, let me show you an example for it. I'm a software engineer. So I must make six figures.

Now, let's take a look at this. These contain special characters like the coma. The single quote. The periods is

contains a number and you can even sprinkle it with more numbers. If you turn some words to Lead speak. Let's see what

that would look like. I'm a software engineer. So I must make six figures. And this time I wrote engineer with the

free instead of an e. Okay. Now we have an additional number, but this password is too long to type and

this is where the next step comes in since this is not our actual password, take the first letter of each word every number and

punctuation, and highlighted then remove the rest. And you are ready. Have a pretty random and safe

password IAS, 3 SI M, M6 f. Period. I think I forgot a comma there, but this is safe enough

as it is now. And you can see, it contains numbers, special, characters is long and random, but most importantly, you

can remember it easily with the sentence from before. I'm a software engineer. So I must make six figures, just

make sure to make a memorable sentence. Now, I can use the secure password created in that manner for

our master password, in a password manager. But how do they work? And what even are they? Let

me show you in the next lesson.

**5. Password Managers**

Hello, I'm Janet and welcome to the Hands-On password cracking and security course on Code. Red. In this section. We've covered the remedies

for the text demonstrated in the rest of the course. Now, we're going to take a look at

password managers password managers are a great tool to boost your password security. Let's take a look at what we can use them

for and if they will solve all our problems, spoiler. They don't. A password manager is an application which

you can use to generate and store passwords safely as an added bonus. Most passwords managers will allow you to out to feel

credentials on websites and other places which is great because you never have to know or type the password again, so

they can be as complicated as you want and keyloggers can't lock them as most password managers use a chi-square.

Amber, when Auto filling? There are two categories of password managers which are offline, password managers, which

you have to keep local on your computer, but there are also cloud-based ones. So what are the differences?

Offline. Password managers may be more secure than cloud-based ones. They're stored on your hard drive or on a stick that you can bring along with you. But the

problem is that you need to manually synchronize the passwords. If you want to use them on multiple devices or

share them with other users or some smartphones. There isn't even another option than to send it via email. Upload it to a cloud or

something like that. So you lose the security aspect, if you don't Oh nice. Damn, right? Other than that offline. Password managers,

like keepass are a great tool if you need pure security and will synchronize them properly or don't even need to synchronize them. There

are often used in companies for employee specific credentials. That should never leave the machine. Anyways. Online or cloud-based password managers are really helpful if

you're handling multiple devices. It's also great if you have to share a specific passwords in the company with other users.

Enter are some versions that can be deployed on-premise data. However is usually stored in the cloud. Usually,

it is encrypted multiple times using your master password. So it is really important to have a good master password. Like I

showed you in the last lesson. Since the data is in the cloud. It is really easy to use it on multiple devices, including

smart watches. No need to synchronize manually. If you create a new entry while on your laptop, it

will automatically be available from your phone. Most of them offer apps for Android and iOS and have browser plug-ins

for most popular browsers. But the problem is, if the provider of the password managers get hacked, your data could

be damned leaked and misused. Some examples for this kind of password managers are LastPass and one password which

both integrate into multiple browsers and devices. So, our password matches the all-in-one solution for our problems.

Sadly, no password managers. Come with their own set of problems. For example, you cannot use the autofill feature for

logging into your computer. You still have to type it by hand. So, the best would be to use a method, like, I showed you in the last lesson for

your PC, login, another password, as your password manager, password, and the rest can be saved in the password manager.

But what happens if you have two devices, three devices and then you go to work and have another

device, go at home, and you have another device, you would have to create passwords for each of them. Additionally. There

are more problems. What happens if you lose your master password, you cannot recover the data or reset the password as

it is used as the encryption key. So you have no chance of recovery if you lose it.

Some password managers offer a solution to this but I never had to use it. So I cannot vouch for

them. Just don't lose your master password. What happens if you lose the password database, well, you lost all your password with absolutely zero

chance of recovering them ever, unless you're backed it up somewhere. This is only a problem for local, password managers. As

the cloud ones will probably not disappear. As long as the company still exists, the best. Way to prevent this would be to not lose your database or

at least make a backup copy of it. Every time you update it and store it on a different device in your home. Maybe something like

an ass or an external hard drive or a USB stick. Another solution would be to have one email account with the password that

is not stored in the password manager, and use that one to reset. All your password on websites, like Facebook

or Twitter, if you lose the database. But that leaves an opportunity for an attacker. He can abuse your one week password to reset

all your passwords, so don't do that. Now, I hope I didn't scare you off too much

from password managers, as they are still a great tool and combined with the other remedies that we discussed in this section. It will massively

improve your password security. So make sure to at least use multi-factor, authentication, salt and pepper, your passwords, and

store your passwords in a password managers. Generate them randomly, or if you cannot store them in the password manager, use the method showed

in the last lesson in the next section. We will look at some real life cases where passwords were

stolen or cracked and abused by an attacker will also take a look at how it could have been prevented

and what the consequences. Fences for the company were.

**6. Chapter 6 Quiz**

1. Why should you use a password manager?

* It allows you to create strong and unique passwords for every platform you use, without remembering them
* It's faster to fill out login forms
* So my passwords won't get stolen
* All of the above

2. Which problem can't be solved with a password manager?

* Creating strong passwords
* Remembering all passwords
* Logging into your computer (eg. Windows login)
* None of the above

**Chapter 7: Case Studies**

**1. Case Studies**

Hello world. I'm Senate and welcome to the Hands-On password cracking and security course on code. Red in the section. We will look at

some real life cases where passwords were stolen or cracked and abused by an attacker. We will also take a look at how it could

have been prevented and what the consequences for the company were. And as the title implies, this lesson will focus on

data breaches and attacks that led to password dumps. Now, the first attack that we're taking a look at has,

actually nothing to do with password thumbs, but it's the most current one. So let's take a quick Glimpse over it in.

July of 2020. Attackers have gained access to Twitter employee account by means of social engineering. This means that

rather than taking a hash and crack it, or try to break a password. They convinced the users

to give them the passwords by posing as a third party like a bank or company. For example, Microsoft.

They then use those accounts to enter an Administration console and altered the verification emails for multiple users, including Barack Obama and

Elon Musk with the email altered. They could simply use something like a forgot password mechanism to reset

the passwords and enter in their accounts. Then they tweeted a bit point game using this high-profile accounts.

Then they tweeted a Bitcoin scam, using this high-profile account. Alphas attack has to do with zoom and I'm certain that most of you at

least in 2020 have had to use soon at least, once with our current situation and hackers know this as well. So they

take tune and clicked about half a million credentials up for sale. How did they do it with credential stuff? They

collected multiple terms of licked credentials in the dark, net. Or Net crack them and use them with

automated tools on tsums login page. They could then evaluate the response and determine the valid credentials, which they put up for sale

in order to not get their IP blacklisted or blocked. The attackers used something called the botnet. So they always use

a new IP as we saw in the last couple of sections. This is only possible because so many people use the same credentials for

multiple websites. The only thing to do at this point for zoom, Was to let them change their passwords, which

they did. They requested that every user should reset their password or change it. In January of twenty twenty conv became

aware that 4 million cracked credentials were leaked. However, the attack took place long before that in May 20, 19, the

attackers were able to get 137 million user accounts with her hashed passwords and salts. The attacker could use

the solids plus a combination of brute force and dick. Nari attacks to break the passwords. And

as you can see after seven months, they were able to crack about 4% of them. It is unclear at this point. If can

we use the pepper to slow the attackers down, but what we know is that they use big trip to H2 passwords.

This shows that something does not necessarily make your password safer. If the attacker can get, hold of the hash and the salt. So make

sure to use the pepper as well. Apparently, the attackers were even able to gain. Oh Alf, Logan tokens from users using social

sign-in via Google, but that's just a rumor at this point and it hasn't been confirmed. The only thing the canva could do at this

point was to a reset all the passwords or B notify the users, we crack credentials so they could change them on other sides. If

they were using the same ones which as we saw before quickly to credential stuffing attacks on other services. And so that's

what can validate and we hope that it worked out. In 2014, the entire user base of eBay was leaked, including

hashed passwords that used week hashes. Some sources suggest that the attackers used a credentials to access them and that they went

unnoticed for 229 days. How the attackers were able to get hold of the employee passwords is uncertain, but I

think it could have been either through means of social engineering or some attack like credential stuffing or password, spraying, all they could do

and did was to reset all the passwords for the users. In 2012, nearly six point, five million

user credentials were stolen by Russian cyber criminals from LinkedIn. Later in 2016 LinkedIn. Discovered, 100 million email addresses, and

hashed passwords were leaked. The whole list of partially correct credentials. Went up for sale for the price of five Bitcoin, which

at the time was around two to three thousand US dollars. What makes this attack so interesting for us

is that LinkedIn failed to use salts for the passwords and toss. Were easier to crack some sources. Even claim that

linked induced a sha-1 hashing algorithm, which only made it easier to crack. Most of the credentials, were probably

cracked with the rule-based dictionary attack as we ourselves did, in the section about dictionary attacks. While the rest would have been cracked with Brute

Force. Text there is however, still a chance that some passwords were not cracked with the passwords were random and strong enough as

soon as LinkedIn discomforted. So many passwords were leaked. They forced users to change their passwords. If they didn't change them.

Since 2012 already following the breach, some uses filed lawsuits against LinkedIn, some of which could have cost up to

five million dollars for LinkedIn and this This is special here because this is one of the first in our cases which

actually had to pay money. Instead of just resetting, the passwords for everybody in July 2018 and preacher. Kurt at

8 feet. 8 feet is a health and fitness app and 15 to 20 million user credentials were leaked and

what's even worse. It went unnoticed until February of Nineteen eight feet, then went on to encourage the users to change their passwords

as well as passwords for other services. They started investigating the bridge as soon as possible and according to

their own site. They started finding and fixing the vulnerability, the cost of bridge. In 2013, Adobe has also

suffered from a data Bridge different sources. Cite different numbers of stolen credentials, ranging from 35 million, 250

million. And not only that, the attackers were also able to grab parts of the source code for cold, fusion Photoshop and

the Acrobat Reader. Adobe as all other companies. In these examples, reset the passwords of all their users and claimed that they did not see

any indication of unauthorized activity on the bridged accounts only known the bridge took six weeks to be noticed since the

attacker upload the source code on a website that then later was found by security bloggers and researchers.

An adobe is one of the faster companies that we saw in our examples with only six weeks. And six weeks is a

long time for attackers to crack passwords or dump them somewhere in the dark, net, and sell them even if they're hashed.

So not only did the attackers in this example create credentials. They also stole parts of the source code. This would allow them to

either investigate ticket to find flaws and vulnerabilities that they could then later exploit or they could sell

it online which seems to be what they did. Luckily, they did it in such a prominent way

that some researchers found it. And the breach was found in only. Lee, six weeks. So what can we learn from these examples and case studies?

Everyone can be a Target big companies, small companies social media sites, free tools, like conv and even fitness and

health apps. Attackers are usually focused on getting as many credentials as possible and then sell them in bulk on the

darknet or Internet. Other attackers can buy them and use them for passwords praying and credential stuffing. It takes

So make sure to never reuse your passwords on other platforms. Attack your game, Password as a means to sell them

or abused them. In other ways. They usually find the vulnerability of fish, credentials of Administrators, and then dump the passwords

often the attack. Go unnoticed for weeks months or even years, which gives the attacker plenty of

time to crack a chunk of those credentials and abuse them. In order to protect us and our users from

this Menace, which should implement the things discussed in former sections, like salt pepper, multi-factor authentication, strong, password

policies, and good hashing functions. It is important to use random strong and unique passwords. Every time we need the new one and password managers

can help us with that. Or we can use the method learned in the last section. Another thing we can learn is that some companies may

get fined for having a data breach, depending on if they used current technology or not. So if you're using an old hashing algorithm,

you can be fined for that. If an attacker leaks, you don't dumps, if an attacker dumps your passwords and

they were cracked because it was your mistake that you didn't use a good enough. Hashing function. So also make sure to

use good hashing functions whenever you can. And with that, thank you very much for taking part in

this course, on Code Red. I sincerely hope you enjoyed it and learned some things along the way that

you can either use to make your application secure or handle your passwords better. I hope to see you in a future course or on my channel

death Knoll over on YouTube.

**2. Chapter 7 Quiz**

1. What is the best way to handle a data breach?

* Ignore it and hide it from the public
* Notify all users to change their passwords
* Delete all affected users
* Change the setup

2. What did Canva do when they found out about their data breach?

* Ignored it and hidden it from the public
* Notified all users to change their passwords
* Deleted all affected users
* Changed the setup