How to Guess a Password: Your 2024 Guide

May 11, 2024 Richard Dezso

Listen to the article

Is learning how to guess a password a simple process, or does it require different tools and techniques to achieve your desired result?

In this article, we will discuss a part of the Authentication Principle. Authentication fundamentally involves confirming your identity by providing three elements: something you know, like a password; something you possess, such as a security key; and something that represents you, like your username or email.

We will look at something you know (a password). We will discuss common weak passwords, using OSINT to find leaked passwords, rules and tools that can be used to create a wordlist, and what to watch out for when trying to spray passwords.

Common Weak Passwords

The most common way we prove who we are online is by using passwords. But what's a password? It's a mix of letters, numbers, and other symbols to validate to a service we are who we say we are.

Based on an analysis by **CyberNews**, which reviewed over fifteen billion passwords found in data breaches, the following are the top ten most used passwords. These passwords are notably weak and insecure:

1. 123456
2. 123456789
3. qwerty
4. password
5. 12345
6. qwerty123
7. 1q2w3e
8. 12345678
9. 111111
10. 1234567890

Here are other notable statistics on passwords from the **Keeper Security Workplace Password Malpractice Report.**

- Over one-third (37%) of respondents have used their employer’s name in a work-related password.
- Over one-third (34%) have used their significant other’s name or birthday.
- Nearly one-third (31%) have used their child’s name or birthday.

Wordlists are already created for you with these common and weak passwords; Kali, for instance, has a directory with many different wordlists. The most notable being rockyou.txt, nmap.lst, and john.lst. These lists can be found at /usr/share/wordlists.

```
/u/s/wordlists ls -la
total 136660
drwxr-xr-x   2 root root      4096 Jun 21 18:30 ./
drwxr-xr-x 375 root root     12288 Jul 10 14:59 ../
lrwxrwxrwx   1 root root        26 Jun 21 18:30 amass → /usr/share/amass/wordlists/
lrwxrwxrwx   1 root root        25 Jun 21 18:30 dirb → /usr/share/dirb/wordlists/
lrwxrwxrwx   1 root root        30 Jun 21 18:30 dirbuster → /usr/share/dirbuster/wordlists/
lrwxrwxrwx   1 root root        41 Jun 21 18:30 fasttrack.txt → /usr/share/set/src/fasttrack/wordlist.txt
lrwxrwxrwx   1 root root        45 Jun 21 18:30 fern-wifi → /usr/share/fern-wifi-cracker/extras/wordlists/
lrwxrwxrwx   1 root root        28 Jun 21 18:30 john.lst → /usr/share/john/password.lst
lrwxrwxrwx   1 root root        27 Jun 21 18:30 legion → /usr/share/legion/wordlists/
lrwxrwxrwx   1 root root        46 Jun 21 18:30 metasploit → /usr/share/metasploit-framework/data/wordlists/
lrwxrwxrwx   1 root root        41 Jun 21 18:30 nmap.lst → /usr/share/nmap/nselib/data/passwords.lst
-rw-r--r--   1 root root 139921507 May 12 11:14 rockyou.txt
lrwxrwxrwx   1 root root        39 Jun 21 18:30 sqlmap.txt → /usr/share/sqlmap/data/txt/wordlist.txt
lrwxrwxrwx   1 root root        25 Jun 21 18:30 wfuzz → /usr/share/wfuzz/wordlist/
lrwxrwxrwx   1 root root        37 Jun 21 18:30 wifite.txt → /usr/share/dict/wordlist-probable.txt
/u/s/wordlists
```

Another great location for password lists is **SecLists** on GitHub. This resource contains many different password lists, such as “2020-200_most_used_passwords.txt” and “500-worst-passwords.txt”.

SecLists / Passwords /

..	
BiblePass	dos2unix
Common-Credentials	dos2unix
Cracked-Hashes	Quick rename of files
Default-Credentials	update default-passwords.csv
Honeypot-Captures	strip trailing whitespace
Leaked-Databases	Merge pull request #653 from soufianetahiri/master
Malware	Close #291 - Fix encoding issues
Permutations	rename 's/_/-/g'
Software	Close #291 - Fix encoding issues
WiFi-WPA	Add "-" to split up words, moved files since PR accepted
2020-200_most_used_passwords.txt	fix: Dedupe wordlist
500-worst-passwords.txt	fix: Dedupe wordlist
500-worst-passwords.txt.bz2	Add 500 worst passwords
Keyboard-Combinations.txt	Add "-" to split up words, moved files since PR accepted
lost-Popular-Letter-Passes.txt	Add "-" to split up words, moved files since PR accepted

So why do people still use some of these weak passwords in their everyday accounts, including in the workplace? There are several reasons.

Convenience: Many people prefer to use easy-to-remember passwords. Trying to remember a fourteen-character-long password with letters, numbers, and symbols is difficult, and most people prioritize ease of use over security.

Multiple Accounts: As people sign up for more and more accounts, the need to remember these passwords increases. People often use the same simple password across numerous sites to simplify this.

Lack of Awareness: Some users may not fully understand the importance of having strong passwords or the risk involved with using these weak passwords. They may also underestimate the value of their data and believe that the chance of them being targeted by an attacker is low.

Establish Parameters

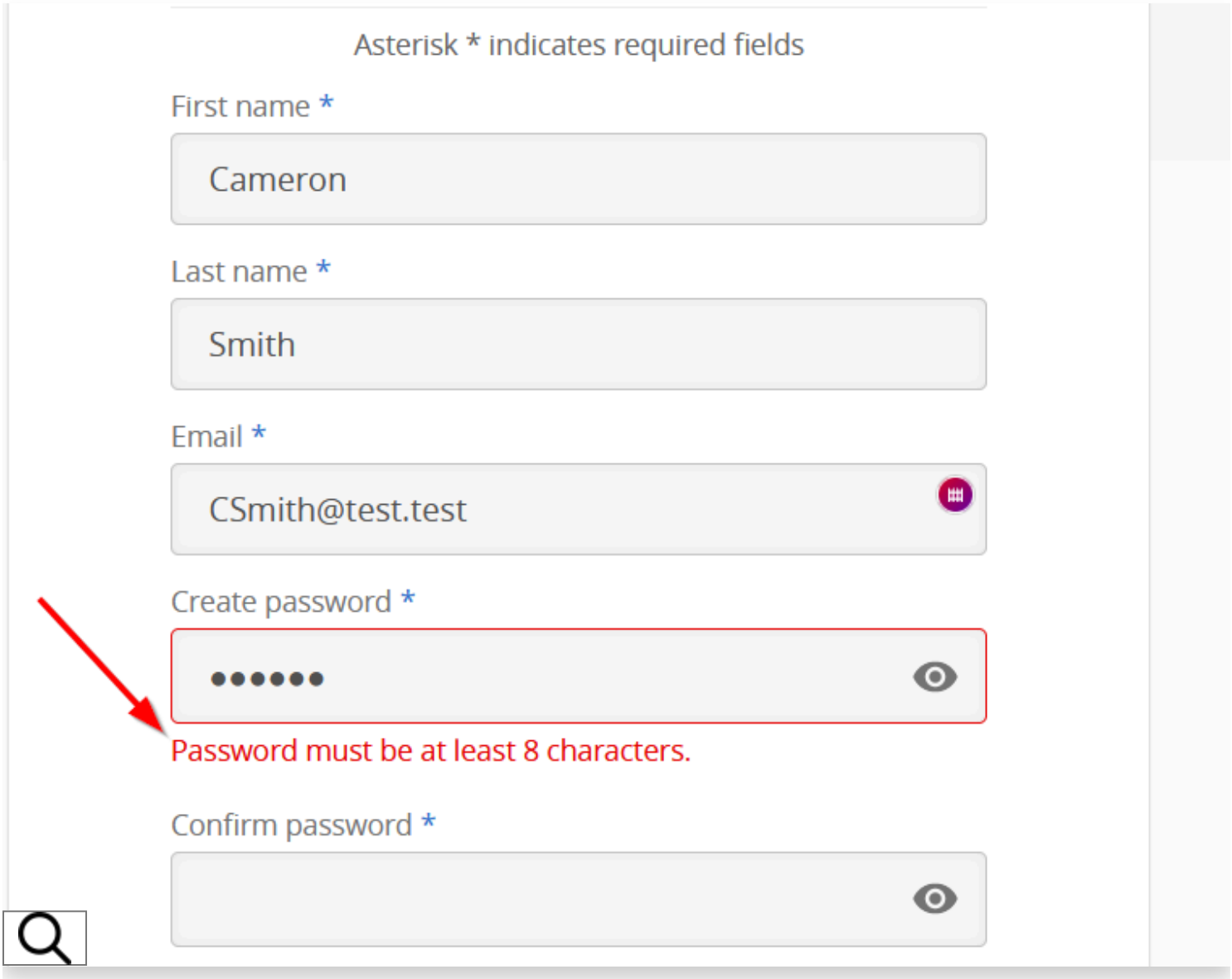
Before we can begin to guess a password, we need to establish a few parameters. The first step in the process is identifying the target. If we are working on a pentest for a company, we need to know the login syntax. Is it Cameron.Smith or C.Smith or maybe CSmith?

We may know this beforehand or need to find out by doing some **investigation online using OSINT**.

Once we have the correct syntax for users, the next step is to figure out information on the service we are attacking, specifically the login information.

Does the login offer hints if the user forgets their login, such as username reminder, custom password hints, or security questions? And what is the password policy of the service? This will give you much more information you can use to formulate a plan of attack.

Let's take a look at the website signup below.



Looking at the form, we know the password must be at least eight characters, and no other requirements must be met. Now we could use this information to create a list of passwords.

OSINT

When trying to guess a password, you will often need to perform some sort of OSINT, which can be incredibly helpful in finding out more about the individual. You can locate the individual on social media and find hints like hobbies, significant dates, pet names, or work details that could be part of their password.

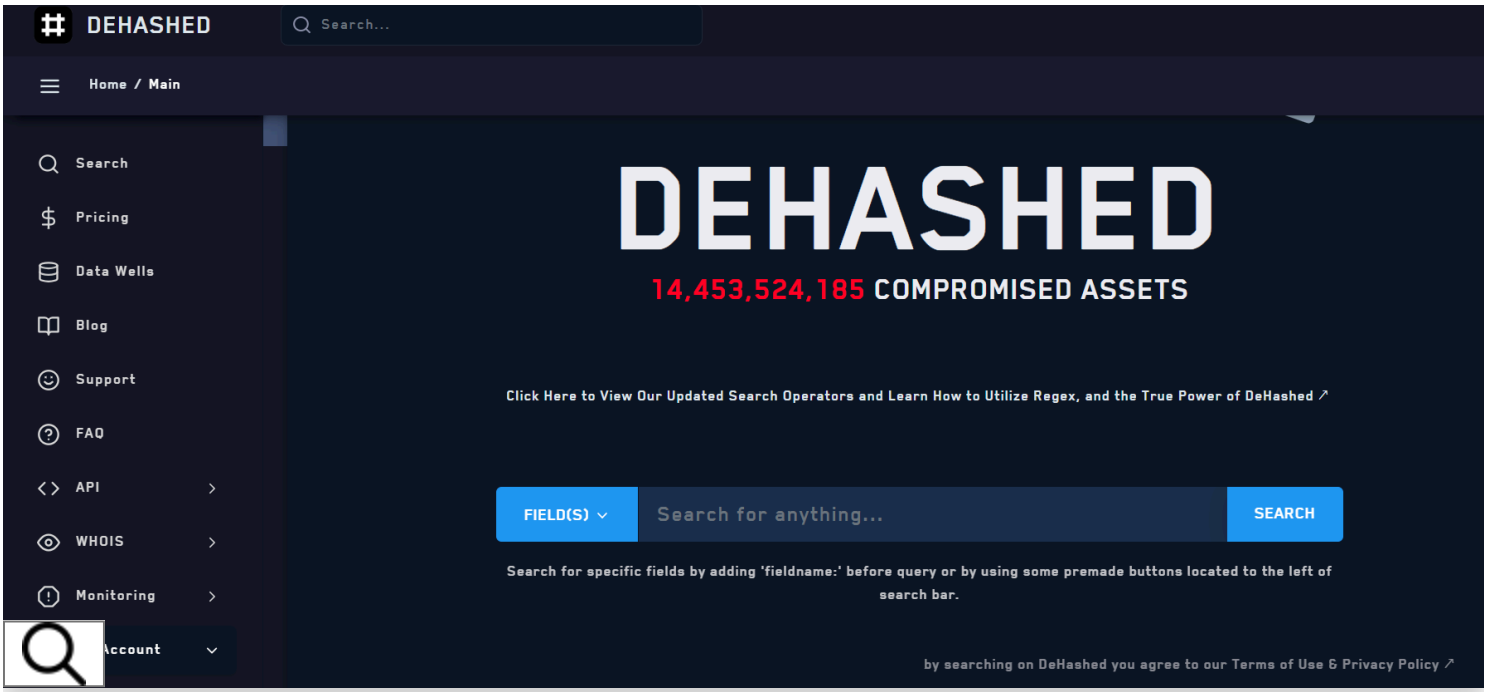
If you are working for a client, find a list of employees and their emails on LinkedIn or via other open-source means. This will give you a list you can manipulate. This list might reveal common themes or patterns, such as the company's username syntax or other potentially useful information.

Many individuals or employees follow similar patterns when generating passwords, often linking them to the specific service used. It is common for employees to include their company's name in their workplace passwords.

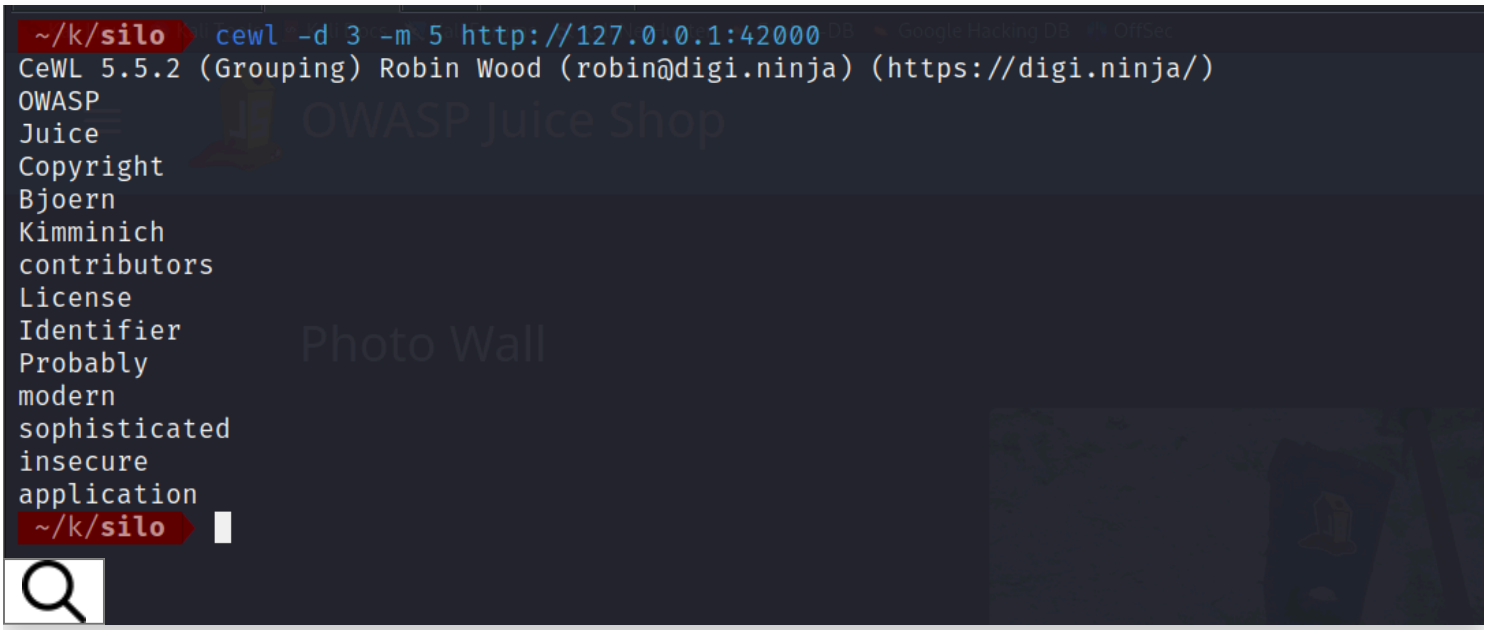
Moreover, people's personal interests and preferences, such as their pets' names, friends' or spouse/partner names, activities, preferred sports, and numerous other aspects of their lives, frequently influence their password selection.

You can use online OSINT sites such as **Have I Been Pwned** or **Dehashed** to see if the company has been involved in any breaches (and may still have employees using compromised passwords); you can also search specific individuals and discover if **they** have been involved in any breaches, where you may find passwords you can reuse.

> In 2011, Aaron Barr, the CEO of the cyber security consulting firm HBGary Federal, was hacked by Anonymous after they discovered he used the **same password** for his business email, Twitter, Facebook, Yahoo, and World of Warcraft accounts.



Another way you can find potential passwords is by using a tool such as CeWL, a Ruby application designed to create custom word lists for password-cracking tools. It spiders a website, collecting words for a password list.



You can also use **social engineering** techniques to create fake login portals or even watering hole websites, these tactics can be effective in obtaining user credentials by duping the users into thinking they're logging into a legitimate service. Tools such as **BeEF**, **Social Engineer Toolkit**, or **ChatGPT** can help.

Rules

When creating a wordlist, we can also use rules to help take a password and modify it. We may want to append a password, such as adding numbers or symbols to the end of them (Password123@) or we may want to substitute characters (such as P@$$W0RD), or we could even reverse the password (like "drowssaP"). In addition, we could incorporate leet speak substitutions, such as replacing 'i' with '1', 'e' with '3', 'a' with '4', etc.

Here are some tools that can help you manipulate passwords in the ways we described above.

John The Ripper

John the Ripper's rule syntax is extensive, but we'll provide a simple example. Rules in John are specified in the configuration file or on the command line using the -rules: option.

Here's an example of a rule that appends the numbers 0-9 to each word in the wordlist:

```
[List.Rules:MyRule]

$[0-9]
```

If you saved this in your john.conf file under [List.Rules:MyRule], you could then use this rule with:

```
john --wordlist=wordlist.txt --rules:MyRule hashes.txt
```

Hashcat

Hashcat, like John, can also manipulate a password list by using what's known as a "rule-based attack." You can specify a file containing rules to modify the words in the wordlist. This allows Hashcat to attempt variations on the words in the list, such as lowercase all letters or appending the character X to the end.

Here's an example of a command you might run with the best64 rule The "best64" rule is a collection of commonly used rules:

```
hashcat -m 1400 -a 0 -r rules/best64.rule hash.txt wordlist.txt
```

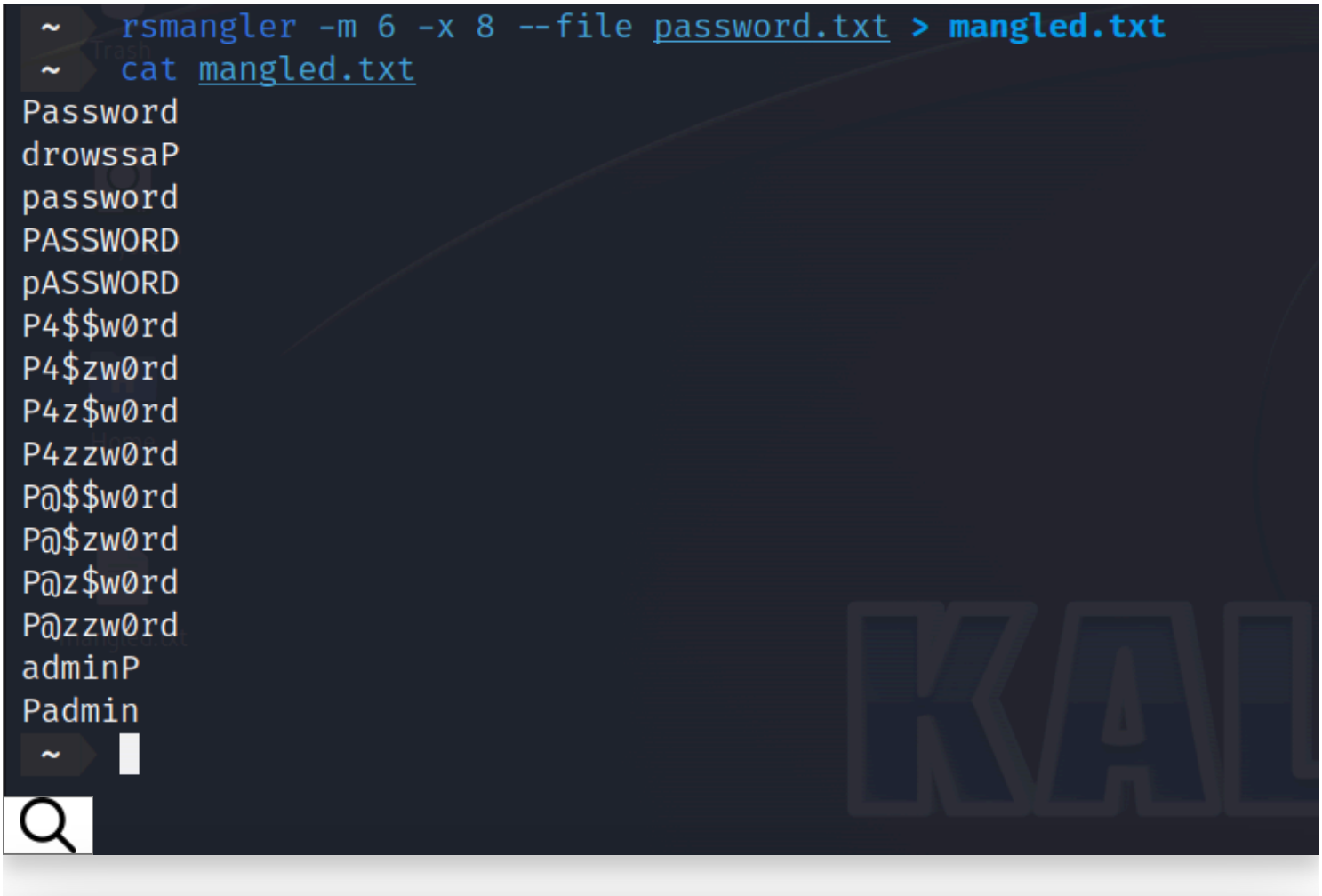
RSMangler

RSMangler is a wordlist manipulation tool. It takes an input file, such as a wordlist, and applies various transformations to the words in the list to generate a larger set of possible passwords. It performs a variety of transformations, including adding years to the end of the word, or adding the following words to the start and end: admin, sys, pw, pwd, numbers 01 - 09, etc.

Here is a sample command that will mangle the given wordlist.

```
rsmangler -m 6 -x 8 --file password.txt > mangled.txt
```

It reads password.txt as an input file, applies a variety of transformations to each word, and saves the results in mangled.txt. -m 6 specifies a minimum word length of 6 characters, while -x 8 specifies a maximum of 8. Any words generated outside of the specified range will be discarded. This is helpful if you know the length of the password.



Become a Professional ☠ Hacker

Tools

Once you're ready to try the passwords or hashes, let's discuss some tools you can use.

Attacking Login Portals

Hydra

Hydra is a popular login brute force tool that performs dictionary attacks against many services such as SSH, FTP, or web servers. It attempts to log in to the service using the username provided and all the passwords in your list.

> See **“How to Use Hydra to Crack Passwords: The Complete Guide”** for more information.

BurpSuite

BurpSuite is a collection of testing tools for web applications designed for penetration testing. It has a feature called “Intruder” that allows you to replace the username and password fields with values from a wordlist.

> See **“How to Use Burp Suite: Discover & Master Powerful Features”** for more

information.

Cracking Password Hashes

Hashcat

Hashcat is a powerful password-cracking tool that uses the power of your GPU(Graphics Processing Unit) to crack various hashes with different types of attack modes, including brute force, dictionary, combination, and rule-based attacks.

> See **“How to Use Hashcat for Password Cracking: A Hacking Guide”** for more information.

John The Ripper

John is another great password-cracking tool that employs various methods to attempt to crack a password. The most common technique is a dictionary attack, which attempts a list of possible passwords. It is also capable of brute-force attacks, attempting every possible combination of characters. In addition, it is capable of rule-based attacks, in which it modifies the words in a wordlist based on predefined or custom rules.

> See **“How to Use John the Ripper: A Quick and Easy Guide”** for more information.

Create Custom Password Lists

CUPP

CUPP, which stands for Common User Passwords Profiler, is a tool used to generate targeted wordlists based on personal information. It uses details about a target, such as their name, pet's name, birthday, etc., to create a custom wordlist that can be used in a dictionary attack.

What to Watch Out For

You will undoubtedly encounter some of the following when attempting to log in using brute force or password-spraying methods.

Account Lockout Policies

Before spraying passwords, try to understand the account lockout policy of the targeted service. Let's take Active Directory as an example. In Active Directory, most often, the lockout policy will be in place with, let's say, ten invalid login attempts. If users enter an incorrect password ten times consecutively, their account will be locked. The lockout duration is set to 30 minutes by default, which means the user will be unable to log in for that time period.

Your best bet in situations like this is to try to spread out your attempts by trying a single password across multiple different account names. Try to use some of the most common passwords used with Active Directory, such as:

- P@ssw0rd01, Password123, Password1, Hello123, Welcome1/Welcome01

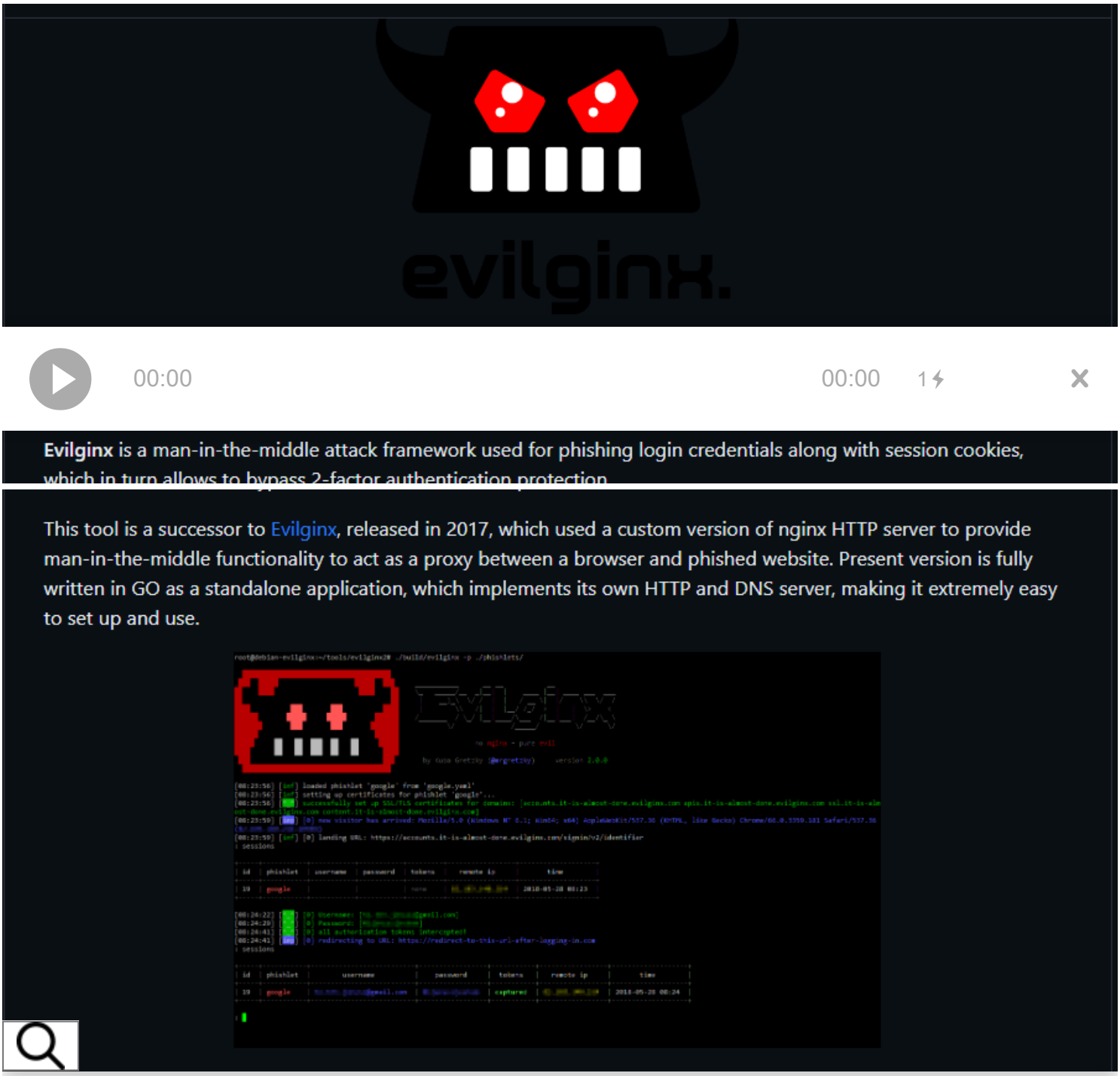
- $Companyname1
- Winter2023*, Spring2023!, Summer2023?, Summer2023, July2023! (Depending on the time of year your testing is taking place)

Multi-Factor Authentication

You may also run into a situation where you successfully log in to a service, but then you realize the user has set up MFA (Multi-Factor Authentication). Where do you go from here? There are a few ways you can accomplish this.

token. This method, however, requires timing and precision because MFA tokens usually expire quickly.

Man-in-the-Middle (MitM) Attacks: In this method, you place yourself between the user's communication and the service. When the user enters their MFA token, you intercept and use it to authenticate their session. **Evilginx2** and **CredSniper** are a couple of tools that can help with this.



CSRF Tokens

In certain situations, you may be up against anti-CSRF tokens when attempting a brute-force attack. A new **CSRF token** must be fetched from the server for every login attempt, as each token is typically unique per session or request. This effectively means that for each login attempt, we would need to fetch a new login page, parse it to extract the CSRF token, and then use it in the login request.

To overcome this, you could use advanced tools like **Burp Suite** to update CSRF tokens while using Intruder automatically or automate the process by writing scripts in Python.

IP Blacklists

While attempting brute force attacks, you may also be subject to IP blacklisting. IP blacklisting is a security measure that blocks traffic originating from particular IP addresses. Systems may implement this to prevent repeated failed login attempts, indicating a brute force attack.

There are a few ways we can circumvent this. To bypass the blacklist, you can use a proxy server or VPN to change your IP address, utilize cloud-based services, or use virtual machines to test from different IP addresses.

Or use a script like **TREVORspray** from GitHub. TREVORspray is a password sprayer that can take advantage of SSH proxying. It logs in to multiple different systems (such as AWS virtual machines, each with a different IP address) and takes turns attacking a password portal from each to avoid blacklisting the IPs because of too many failed attempts in a short period from a single IP.

It supports attacking various services like Office 365, Active Directory Federated Services, Outlook Web App, Okta SSO, and Cisco VPN. It also supports Office 365 MFA bypass.

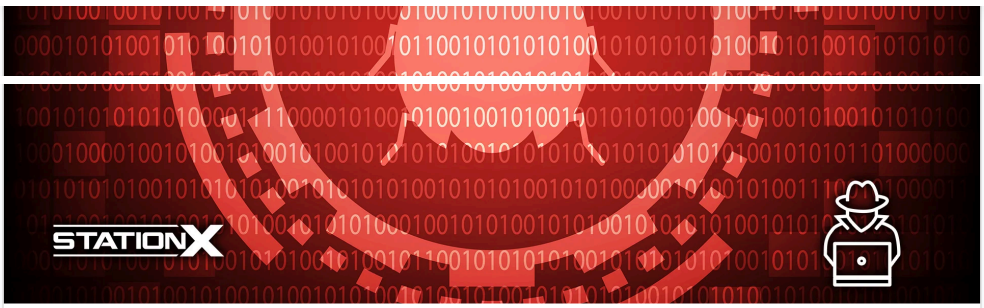
Conclusion

You should now better understand how to guess a password.

We've walked you through common weak passwords, what parameters must be established before beginning, how to use OSINT to find passwords, different tools used while cracking or brute forcing, and what you need to watch out for when performing attacks.

You can continue your journey by utilizing our courses to learn new techniques and skills.

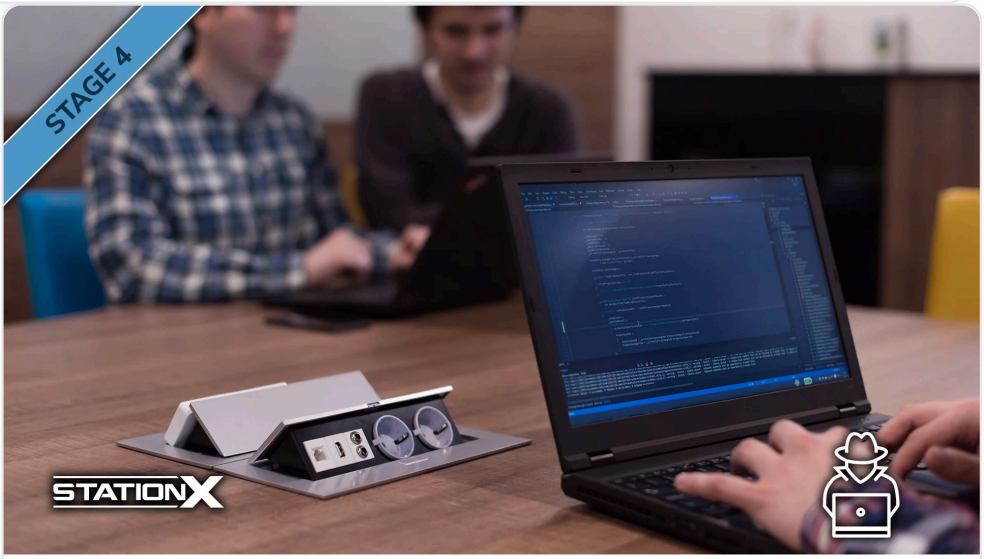
00:00 00:00 1



The Complete Web Penetration Testing & Bug Bounty Course

4.8 ★★★★★

STATIONX



The Complete Ethical Hacking Bootcamp

4.9 ★★★★★

STATIONX

00:00 00:00 1

Ethical Hacking - Hands-On Training - Part I

4.8 ★★★★★

STATIONX

Frequently Asked Questions

What Are the Top 10 Passwords?

The top 10 most used passwords are:
123456
123456789
qwerty
password
12345
qwerty123
1q2w3e
12345678
111111
1234567890

How Long Does It Take To Crack a Password?

Can AI Be Used to Guess Passwords?

What Are Good Rules for Strong Passwords?

Richard Dezso

Richard is a cyber security enthusiast, eJPT, and ICCA who loves discovering new topics and never stops learning. In his home lab, he's always working on sharpening his offensive cyber security skills. He shares helpful advice through easy-to-understand blog posts that offer practical support for everyone. Additionally, Richard is dedicated to raising awareness for mental health. You can find Richard on **LinkedIn**, or to see his other projects, visit his **Linktree**.

Related Articles

Metasploit Cheat Sheet: Master the Modules

Read More »

Top 20 Network Penetration Testing Tools for 2024

Read More »

How to Scan Vulnerabilities With Nmap: A Comprehensive Guide

Read More »

Nathan House