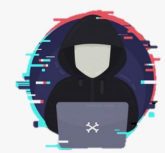
PES UNIVERSITY

PESU ISFCR

PESU Center for Information Security, Forensics and Cyber Resilience

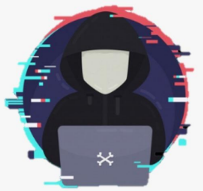


ZERO DAY CYBERSEC

PES UNIVERSITY

PESU ISFCR

PESU Center for Information Security, Forensics and Cyber Resilience



ZERO DAY CYBERSEC

Offline Password Cracking Using HashCat

PES UNIVERSITY

ONLINE

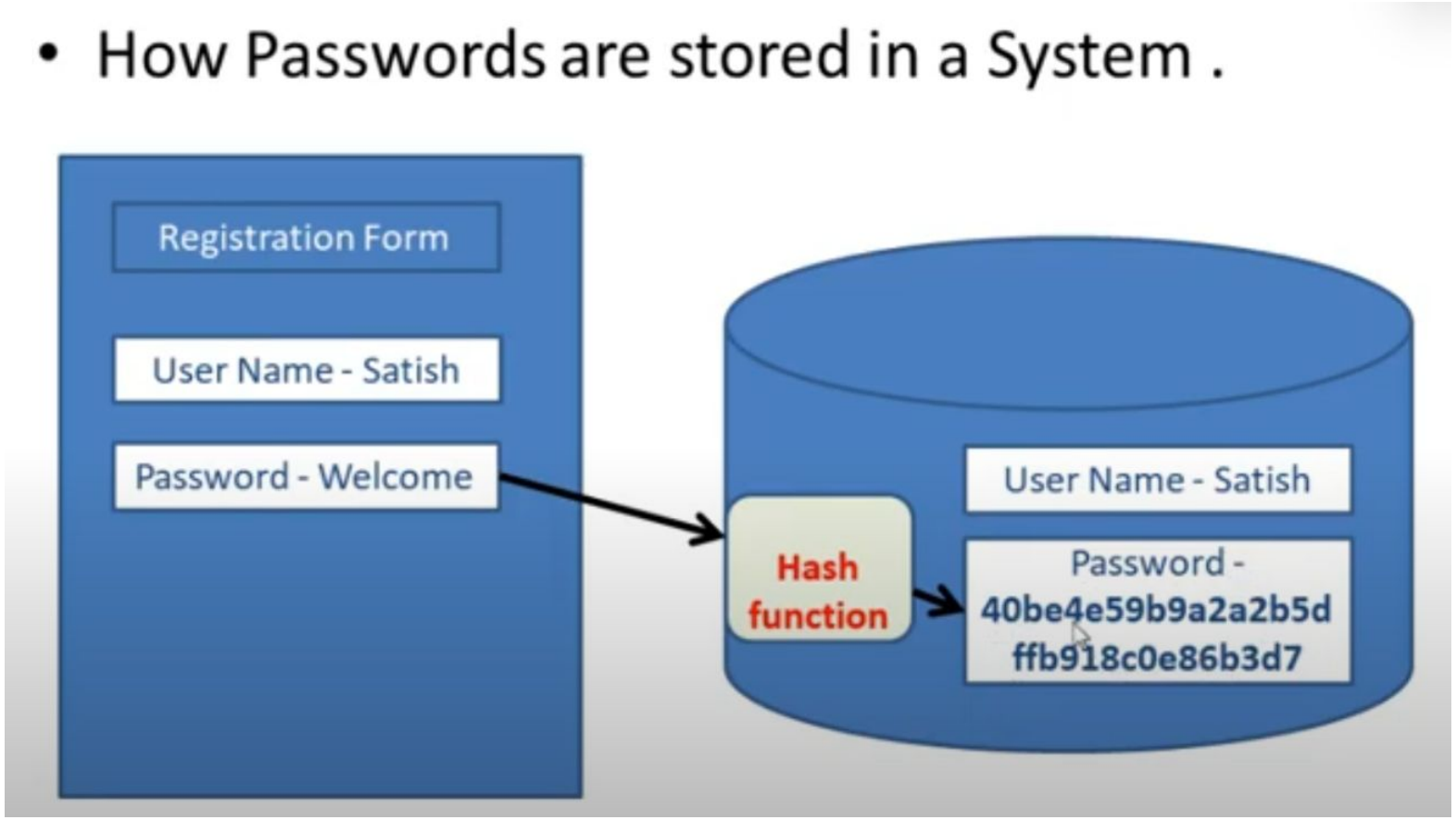
Hydra

OFFLINE

John the Ripper

HashCat

- How Passwords are stored in a System .

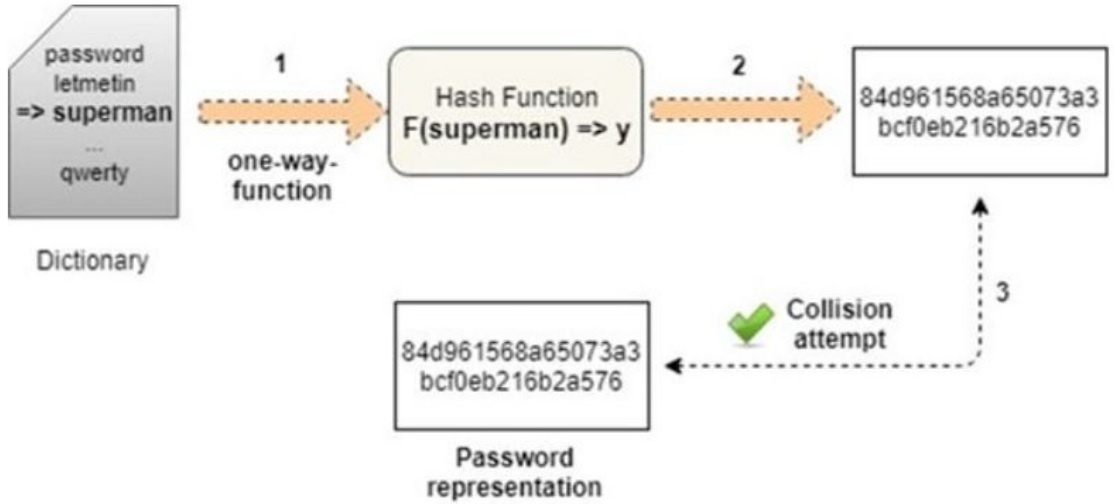


Try the old tradition Dwight Method:

Caption:
Try 000 000
No
Try 000 001

HashCat:

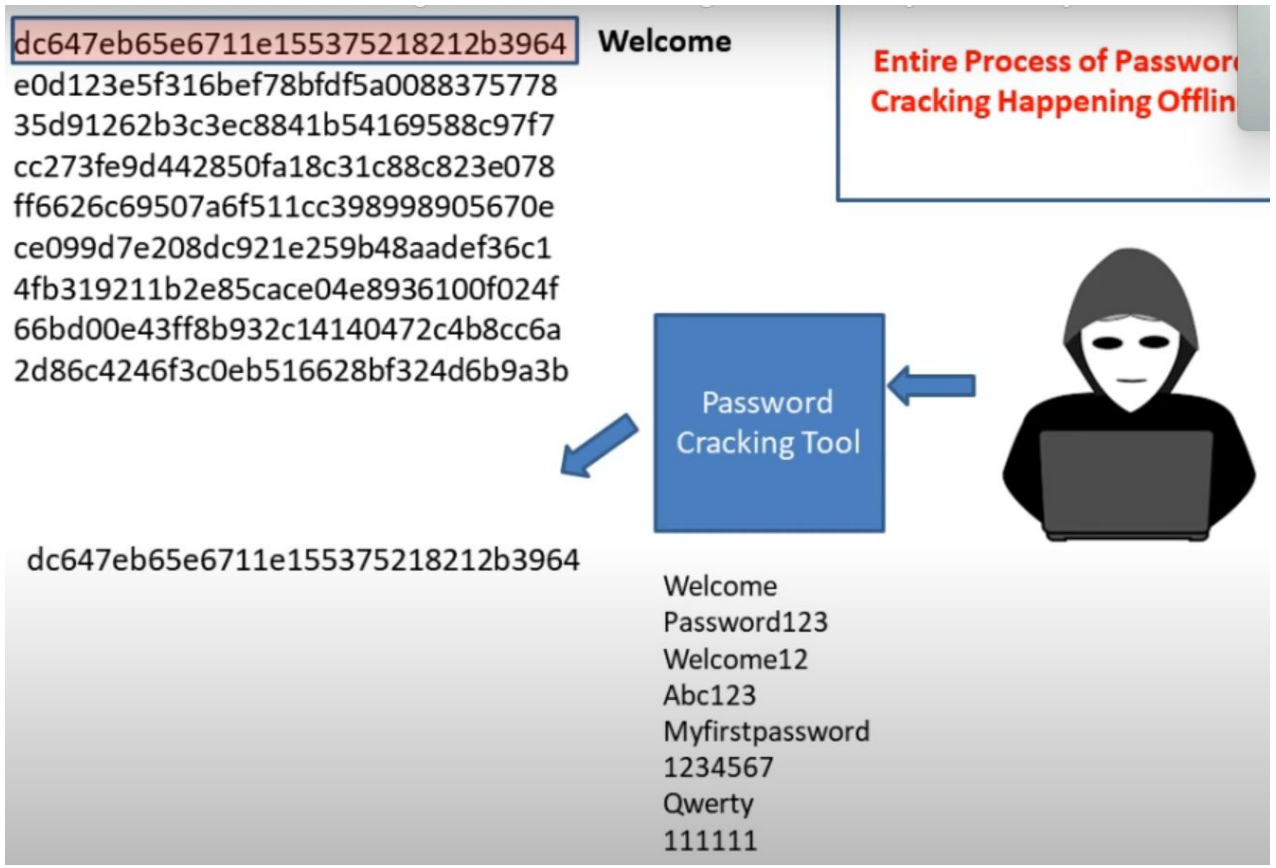
- Hashcat is the self-proclaimed world's fastest CPU-based password recovery tool
- Examples of hashcat supported hashing algorithms are Microsoft LM Hashes, MD4, MD5, SHA-family, Unix Crypt formats, MySQL, Cisco PIX.



PES UNIVERSITY

Algorithms

- MD4
- MD5
- Half MD5 (left, mid, right)
- SHA1
- SHA-256
- SHA-384
- SHA-512
- SHA-3 (Keccak)
- SipHash
- RipeMD160
- Whirlpool
- GOST R 34.11-94
- GOST R 34.11-2012 (Streebog) 256-bit
- GOST R 34.11-2012 (Streebog) 512-bit
- Double MD5
- Double SHA1
- md5($pass.$salt)
- md5($salt.$pass)
- md5(unicode($pass).$salt)
- md5($salt.unicode($pass))
- md5(sha1($pass))
- md5($salt.md5($pass))
- md5($salt.$pass.$salt)
- md5(strtoupper(md5($pass)))
- sha1($pass.$salt)
- sha1($salt.$pass)
- sha1(unicode($pass).$salt)
- sha1($salt.unicode($pass))
- sha1(md5($pass))
- sha1($salt.$pass.$salt)
- sha256($pass.$salt)
- sha256($salt.$pass)
- sha256(unicode($pass).$salt)
- sha256($salt.unicode($pass))
- sha512($pass.$salt)
- sha512($salt.$pass)
- sha512(unicode($pass).$salt)
- sha512($salt.unicode($pass))
- HMAC-MD5 (key = $pass)
- HMAC-MD5 (key = $salt)
- HMAC-SHA1 (key = $pass)
- HMAC-SHA1 (key = $salt)
- HMAC-SHA256 (key = $pass)



Attack-Modes

- Straight *
- Combination
- Brute-force
- Hybrid dict + mask
- Hybrid mask + dict
- Association *

* accept Rules

PESU ISFCR

ZERO DAY CYBERSEC

Credits: Prof. Satish C J - VIT

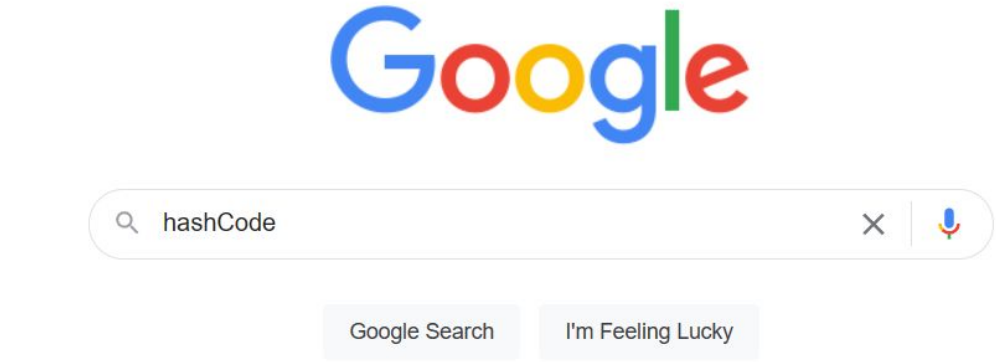
1.Dictionary Attack

1. String samples are essentially taken from a specific wordlist, text file, a dictionary, or past cracked passwords.
2. They are then encrypted identically to the method, key, and algorithm in which the desired password was encrypted originally
3. Dictionary words could also be altered in a randomized manner to check if they work this way
4. Single attack mode of John the Ripper can do such alterations. Accordingly, different hashes' variations are compared when using different alterations.

2.Brute Force Attack

1. All possible plaintexts composed of usernames with encrypted passwords are all exhausted to find the right one
2. They are all hashed and compared to the originally inputted hash.
3. Character frequency tables are used by the program for the sake of including the most probable used characters first.
4. This method is so slow, yet it could identify those passwords having no existence in a dictionary.

Download HashCat for Windows: https://hashcat.net/hashcat/



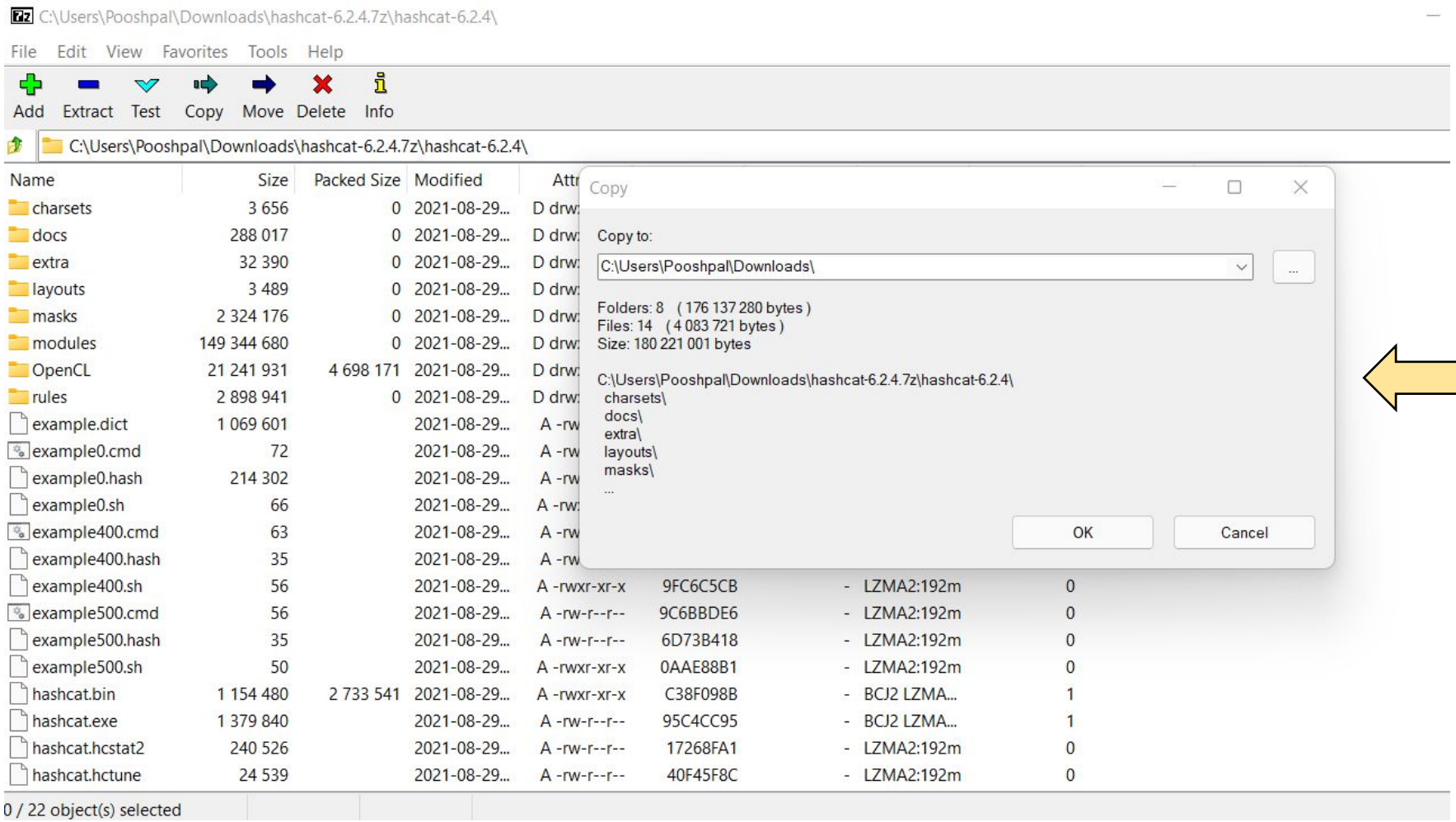
First Link On Top

hashcat
advanced password recovery

Download

Name	Version	Date	Download	Signature
hashcat binaries	v6.2.4	2021.08.29	Download	PGP
hashcat sources	v6.2.4	2021.08.29	Download	PGP

Signing key on PGP keyservers: RSA, 2048-bit. Key ID: 2048R/8A16544F. Fingerprint: A708 3322 9D04 0B41 99CC 0052 3C17 DA8B 8A16 544F


C:\Users\Pooshpal\Downloads\hashcat-6.2.4.7z\hashcat-6.2.4\
File Edit View Favorites Tools Help
Add Extract Test Copy Move Delete Info
C:\Users\Pooshpal\Downloads\hashcat-6.2.4.7z\hashcat-6.2.4\
Name Size Packed Size Modified Attr
charsets 3 656 0 2021-08-29... D drw
docs 288 017 0 2021-08-29... D drw
extra 32 390 0 2021-08-29... D drw
layouts 3 489 0 2021-08-29... D drw
masks 2 324 176 0 2021-08-29... D drw
modules 149 344 680 0 2021-08-29... D drw
OpenCL 21 241 931 4 698 171 2021-08-29... D drw
rules 2 898 941 0 2021-08-29... D drw
example.dict 1 069 601 2021-08-29... A -rw
example0.cmd 72 2021-08-29... A -rw
example0.hash 214 302 2021-08-29... A -rw
example0.sh 66 2021-08-29... A -rw
example400.cmd 63 2021-08-29... A -rw
example400.hash 35 2021-08-29... A -rw
example400.sh 56 2021-08-29... A -rwxr-xr-x 9FC6C5CB - LZMA2:192m 0
example500.cmd 56 2021-08-29... A -rw-r--r-- 9C6BBDE6 - LZMA2:192m 0
example500.hash 35 2021-08-29... A -rw-r--r-- 6D73B418 - LZMA2:192m 0
example500.sh 50 2021-08-29... A -rwxr-xr-x 0AAE88B1 - LZMA2:192m 0
hashcat.bin 1 154 480 2 733 541 2021-08-29... A -rwxr-xr-x C38F098B - BCJ2 LZMA... 1
hashcat.exe 1 379 840 2021-08-29... A -rw-r--r-- 95C4CC95 - BCJ2 LZMA... 1
hashcat.hcstat2 240 526 2021-08-29... A -rw-r--r-- 17268FA1 - LZMA2:192m 0
hashcat.hctune 24 539 2021-08-29... A -rw-r--r-- 40F45F8C - LZMA2:192m 0
0 / 22 object(s) selected
Copy
Copy to:
C:\Users\Pooshpal\Downloads\
...
Folders: 8 (176 137 280 bytes)
Files: 14 (4 083 721 bytes)
Size: 180 221 001 bytes
C:\Users\Pooshpal\Downloads\hashcat-6.2.4.7z\hashcat-6.2.4\
charsets\
docs\
extra\
layouts\
masks\
...
OK
Cancel

Extract it

Open cmd and go to the directory: hashcat.exe --help

```
C:\Users\Pooshpal\Documents\hashcat-6.2.4>hashcat.exe --help
hashcat (v6.2.4) starting in help mode

Usage: hashcat [options]... hash|hashfile|hccapxfile [dictionary|mask|directory]...

- [ Options ] -

 Options Short / Long          | Type | Description                                          | Example
===============================+======+======================================================+=======================
 -m, --hash-type               | Num  | Hash-type, references below (otherwise autodetect)   | -m 1000
 -a, --attack-mode             | Num  | Attack-mode, see references below                    | -a 3
 -V, --version                 |      | Print version                                        |
 -h, --help                    |      | Print help                                           |
     --quiet                   |      | Suppress output                                      |
     --hex-charset             |      | Assume charset is given in hex                       |
     --hex-salt                |      | Assume salt is given in hex                          |
     --hex-wordlist            |      | Assume words in wordlist are given in hex            |
     --force                   |      | Ignore warnings                                      |
     --deprecated-check-disable|      | Enable deprecated plugins                            |
     --status                  |      | Enable automatic update of the status screen         |
     --status-json             |      | Enable JSON format for status output                 |
     --status-timer            | Num  | Sets seconds between status screen updates to X      | --status-timer=1
     --stdin-timeout-abort     | Num  | Abort if there is no input from stdin for X seconds  | --stdin-timeout-abort=300
     --machine-readable        |      | Display the status view in a machine-readable format |
     --keep-guessing           |      | Keep guessing the hash after it has been cracked     |
     --self-test-disable       |      | Disable self-test functionality on startup           |
     --loopback                |      | Add new plains to induct directory                   |
     --markov-hcstat2          | File | Specify hcstat2 file to use                          | --markov-hcstat2=my.hcstat2
```

```
- [ Basic Examples ] -

  Attack-          | Hash- |
  Mode             | Type  | Example command
 ==================+=======+==================================================================
  Wordlist         | $P$   | hashcat -a 0 -m 400 example400.hash example.dict
  Wordlist + Rules | MD5   | hashcat -a 0 -m 0 example0.hash example.dict -r rules/best64.rule
  Brute-Force      | MD5   | hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a?a
  Combinator       | MD5   | hashcat -a 1 -m 0 example0.hash example.dict example.dict
  Association      | $1$   | hashcat -a 9 -m 500 example500.hash 1word.dict -r rules/best64.rule
```

Open Cmd and check if its running.

The Basics
Minimum Arguments

- **Argument 1 of 4:**
  - **-m (or --hash-type)**
    - For example, MD5, SHA1, etc.
    - Full list of hash types can be found by using "--help"
    - Currently supports **237** different hash types
    - In the first example, we will use "-m 1000" for Windows NT Hashes

The Basics
Minimum Arguments

- **Argument 2 of 4:**
  - **-a (or --attack-mode)**
    - Tells Hashcat how to crack the passwords
    - For example, using a dictionary of words, or brute-force
    - In the first example, we will use -a 0 to use a dictionary attack

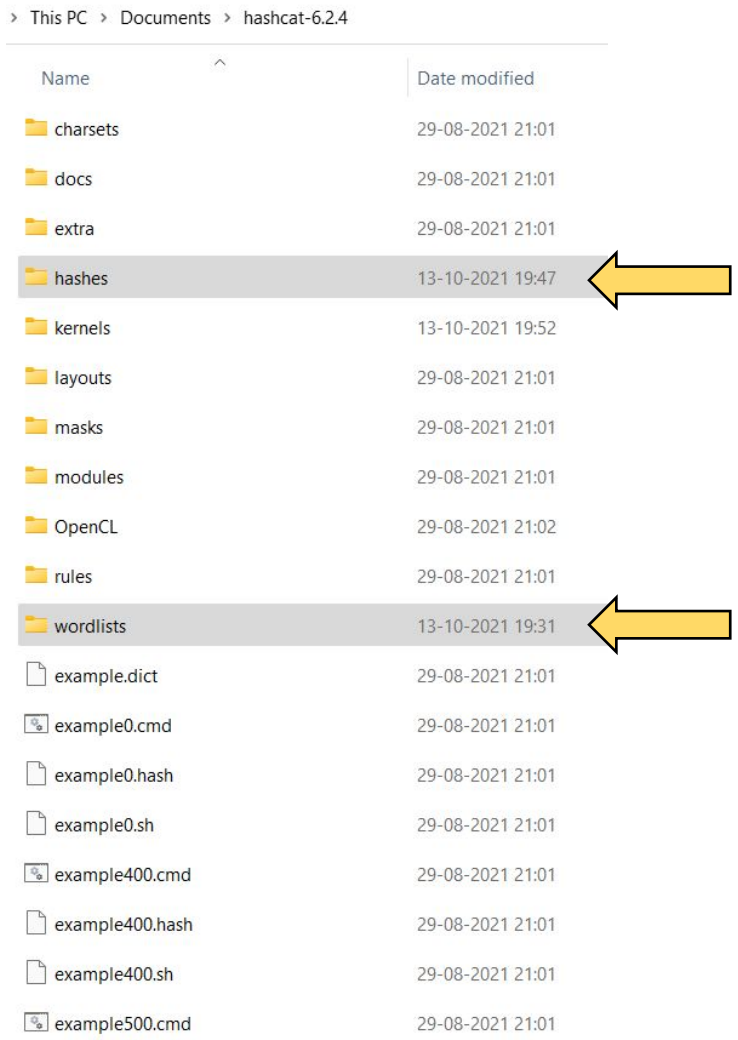
The Basics
Minimum Arguments

- **Argument 3 of 4:**
  - **[*filename* | *hash*]**
    - Specifies a file containing the hash(es) you intend to crack
    - In the first example, we will use ./hashes/ntlm.txt

The Basics
Minimum Arguments

- **Argument 4 of 4:**
  - **[*dictionary* | *mask* | *directory*]**
    - Specifies a dictionary (wordlist), mask, or directory to be used
    - In the first example, we will use ./wordlists/rockyou.txt*
    - *RockYou was a software development company that, in December 2009, experienced one of the largest breaches of cleartext passwords to date. The associated wordlist contains more than 14 million unique credentials, and is often a go-to wordlist to be used during password cracking (recovery) activities.

Credits: 13 Cubed

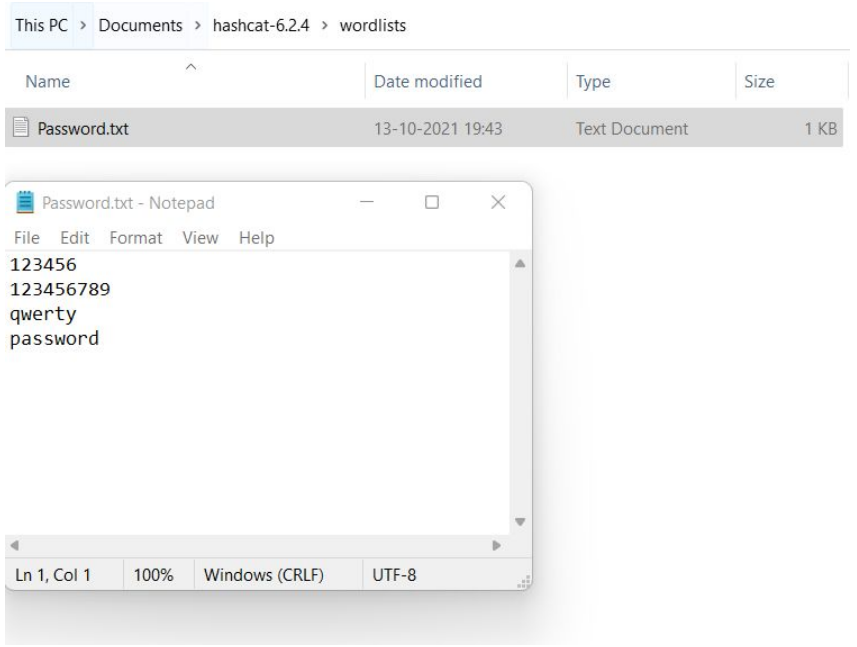


Make 2 new folders:

- Hashes
- WordList

Make a Text File in WordList Folder
And save it as “passwords”

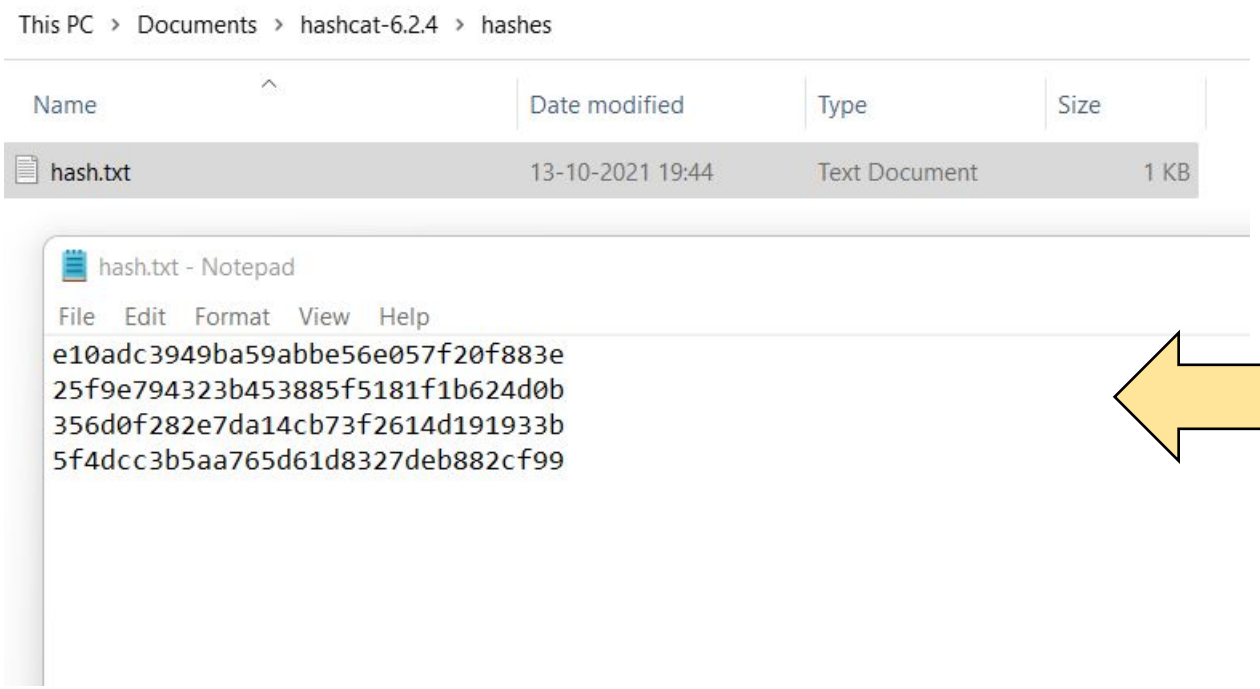
Enter Some Basic Password
Examples In it.



4. Open MD5 Hash Generator: https://www.md5hashgenerator.com/

We Write The Text File Here

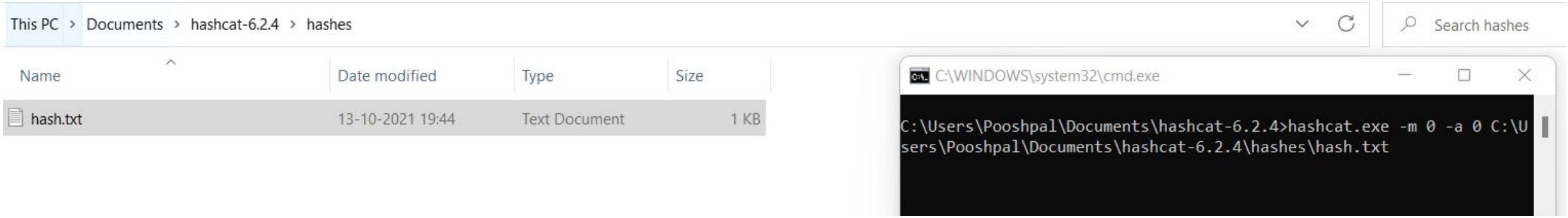
Create a dictionary with MBD5 hashes



Make a Text File in Hashes Folder
And make a textfile named
“hashes.txt” .

Store all hashes here.

Syntax: hashcat.exe –m 0 –a 0 Directory1 Directory2



Explanation:

```
--force = ignore warnings; it is useful if hashcat
          is runnings from a virtual Kali Linux
          machine;

-m 1800 = the -m option indicates the type of
          decryption to be used... in this case
          1800 point to SHA-512 hash  ($6);
          there are many hash types supported
          by hashcat; see hashcat's help for
          full list;
```

Hashcat Attack Modes

Hashcat can perform multiple types of attacks:

- **Dictionary (-a 0)** – Reads from a text file and uses each line as a password candidate.
- **Combination (-a 1)** – Like the Dictionary attack except it uses two dictionaries. Each word of a dictionary is appended to each word in a dictionary.
- **Mask (-a 3)** – Try all combinations in a given keyspace. It is effectively a brute-force on user specified character sets.
- **Hybrid (-a 6 and -a 7)** – A combination of a dictionary attack and a mask attack.

PES UNIVERSITY

Attack Examples

Dictionary attack

I'll first start with a dictionary attack against the list of MD5 hashes.

```
hashcat64.exe -a 0 -m 0 example_md5_hashes.txt
combined_seclists_password_list.txt -O
```

The -O will greatly increase the cracking speed, but will limit the password length that you'll be able to crack. This is usually fine, unless you are cracking passwords greater than 27 characters.

Dictionary attack with rules

Hashcat ships with several rules located in the rules directory. You use the -r <rulefile.rule> option to apply a rule. For example, I'll use the d3ad0ne.rule:

```
hashcat64.exe -a 0 -m 0 example_md5_hashes.txt
combined_seclists_password_list.txt -r rules\d3ad0ne.rule -O
```

Combinator attack

A combinator attack is an attack that combines two dictionaries.

```
hashcat64.exe -a 1 -m 0 example_md5_hashes.txt
combined_seclists_password_list_caps.txt
combined_seclists_password_list_caps.txt -k "$!" -O
```

This attack uses my two dictionaries (I used the same one twice) and also adds a single ! character (-k "$!") to the right of the second dictionary. So if I have the combined word candidate of ThePassword, the -k "$!" transforms it to ThePassword!. Similarly, you can use the -j option to add characters to the left of the second dictionary. For example, if I had also added -j "$&", my word candidate would be The&Password!.

Mask attack

Next, let's try a mask attack (-a 3).

```
hashcat64.exe -a 3 -m 0 example_md5_hashes.txt ?u?l?l?l?d?d?d?d
```

The -a 3 specifies a mask attack, -m 0 specifies MD5 as the hash type, and example_md5_hashes.txt is my file containing the hashes. After that, you have the mask. This particular mask will attempt to bruteforce an 8 character password, where the first character (?u) is an uppercase letter, the next three characters (?l?l?l) are lowercase letters, and the last four characters (?d?d?d?d) are digits. Hashcat has the following charsets built-in:

https://www.browserling.com/tools/ntlm-hash

NTLM Password Hasher

Hash generator

cross-browser testing tools

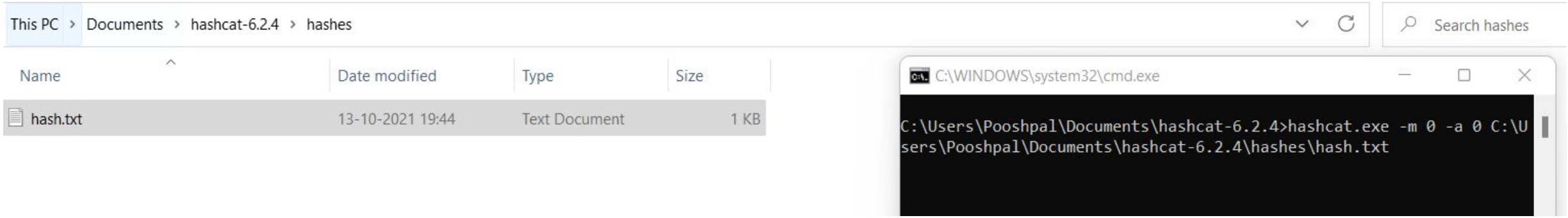
World's simplest online NTLM hash generator for web developers and programmers. Just paste your password in the form below, press the Calculate NTLM Hash button, and you'll get an NTLM hash. Press a button – get a hash. No ads, nonsense, or garbage.

Like 51K

Announcement: We just launched DEVURLS – a neat developer news aggregator. Check it out!

Calculate NTLM Hash | Copy to clipboard

Syntax: hashcat.exe –m 1000 –a 0 Directory1 Directory2



PES UNIVERSITY

MASK ATTACK

You can create or create multiple masks to cycle through each one. The only rule is the mask file you create has the '.hcmask' extension.

```
Hashcat charsets
    ?l = abcdefghijklmnopqrstuvwxyz

    ?u = ABCDEFGHIJKLMNOPQRSTUVWXYZ

    ?d = 0123456789

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

    ?a = ?l?u?d?s

    ?b = 0x00 - 0xff
```

Advantage over Brute-Force

The reason for doing this and not to stick to the traditional Brute-Force is that we want to **reduce** the password candidate **keyspace** to a more efficient one.

Here is a single example. We want to crack the password: *Julia1984*

PESU ISFCR

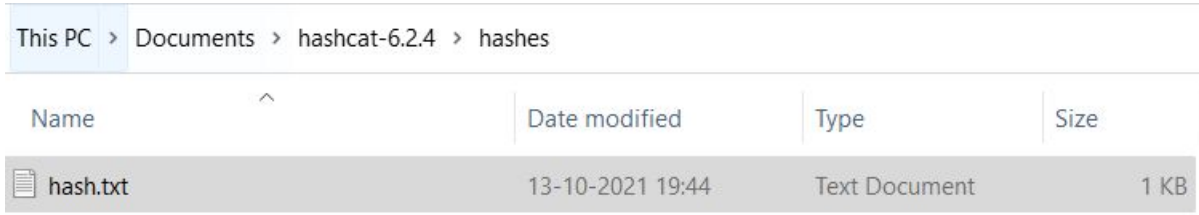
ZERO DAY CYBERSEC

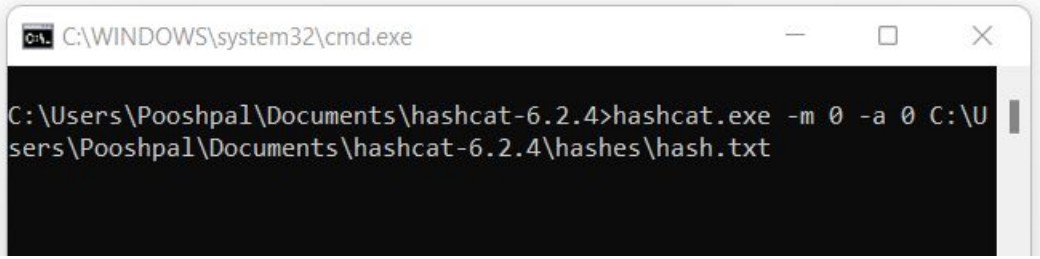
Syntax for Mask Attack:

```
hashcat.exe -a 3 -m 0 maskhashes.txt ?l?l?l?l?l?l?l?l
```

```
? | Charset
===+==========
l | abcdefghijklmnopqrstuvwxyz
u | ABCDEFGHIJKLMNOPQRSTUVWXYZ
d | 0123456789
h | 0123456789abcdef
H | 0123456789ABCDEF
s |  !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
a | ?l?u?d?s
b | 0x00 - 0xff
```

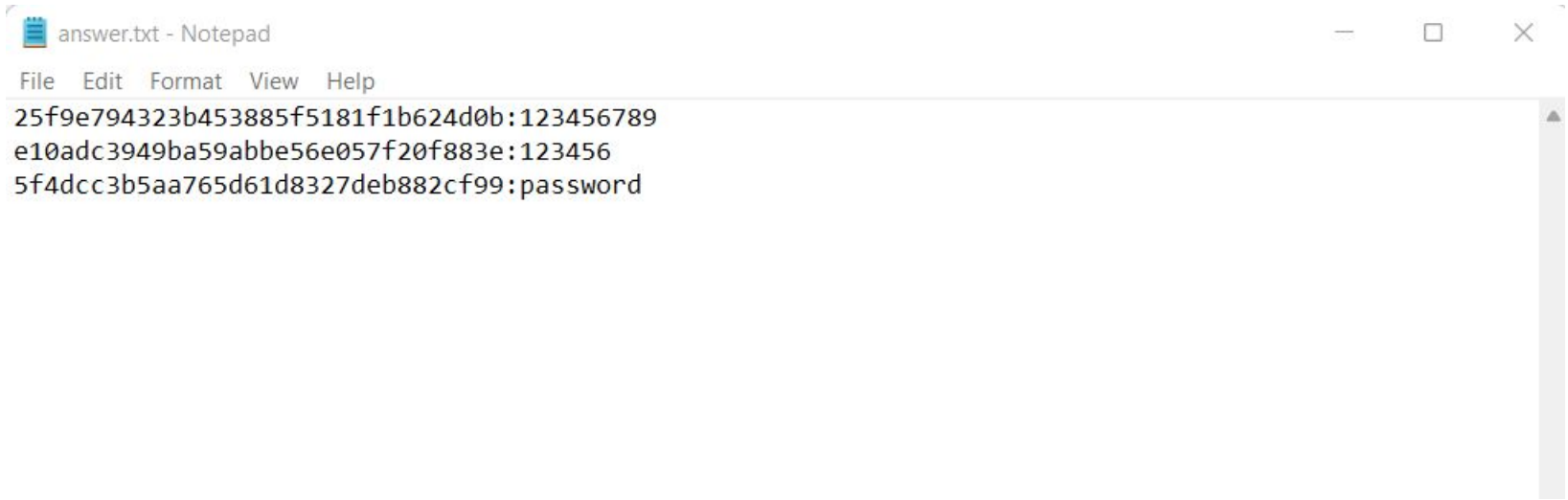
Syntax: hashcat.exe –m 1000 –a 3 Directory1 ?d?d?d?d?d?d –force --show






```
C:\Users\Pooshpal\Documents\hashcat-6.2.4>hashcat.exe -m 0 -a 3 ./hashes/hash.txt ?d?d?d?d?d?d --force --self-test-disable --show
e10adc3949ba59abbe56e057f20f883e:123456
25f9e794323b453885f5181f1b624d0b:123456789
5f4dcc3b5aa765d61d8327deb882cf99:password
```

Syntax: hashcat.exe –m 1000 –a 3 Directory1 ?d?d?d?d?d?d –o ans.txt --force --show



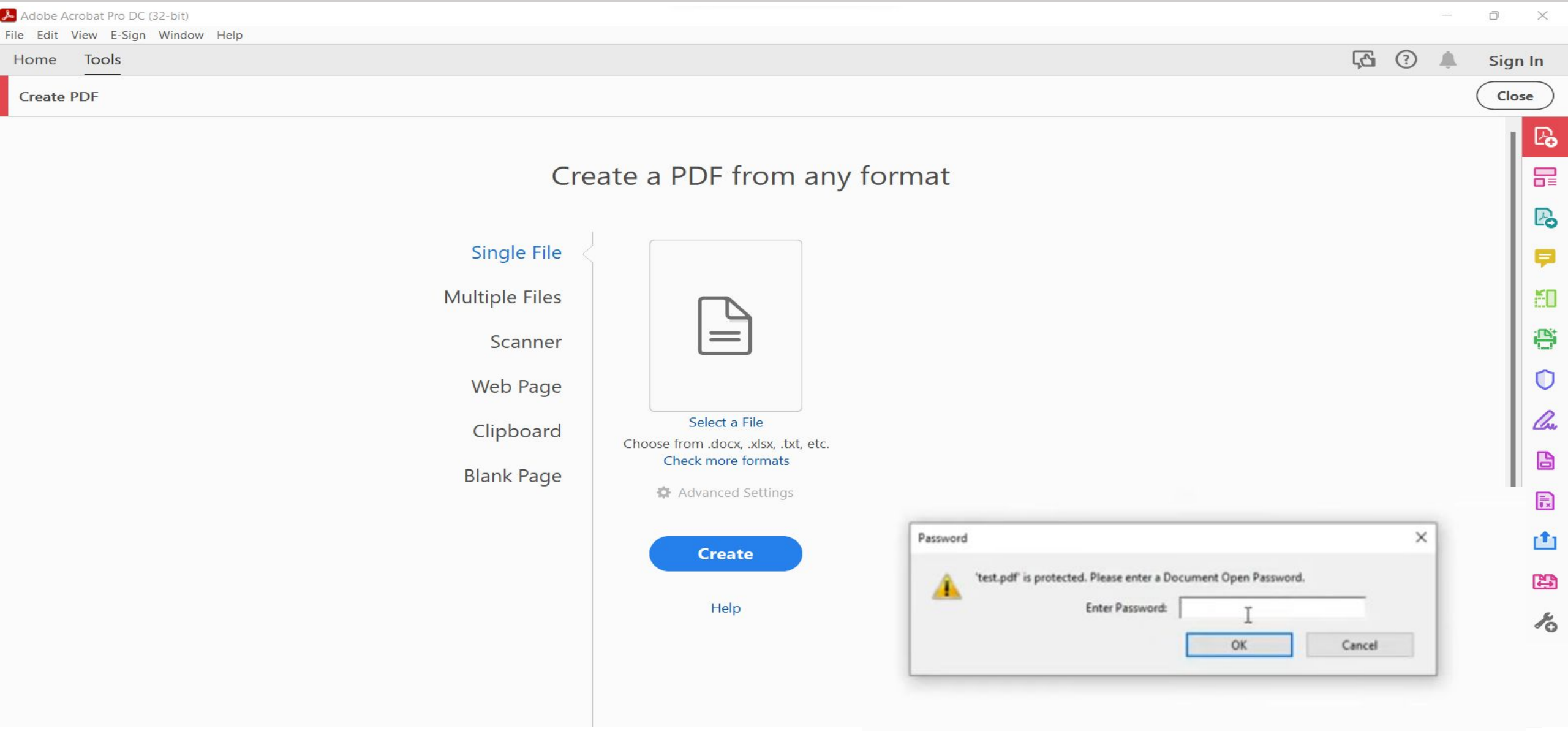
OCTOBER 16 , 2021
Workshop

PES UNIVERSITY

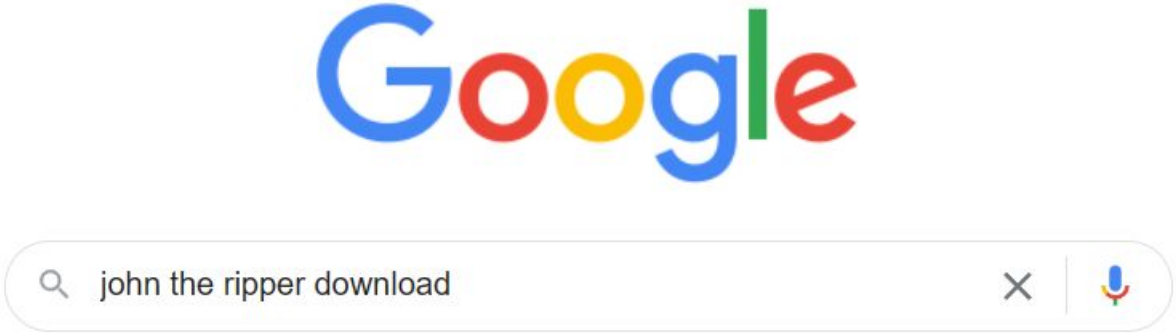
PESU ISFCR

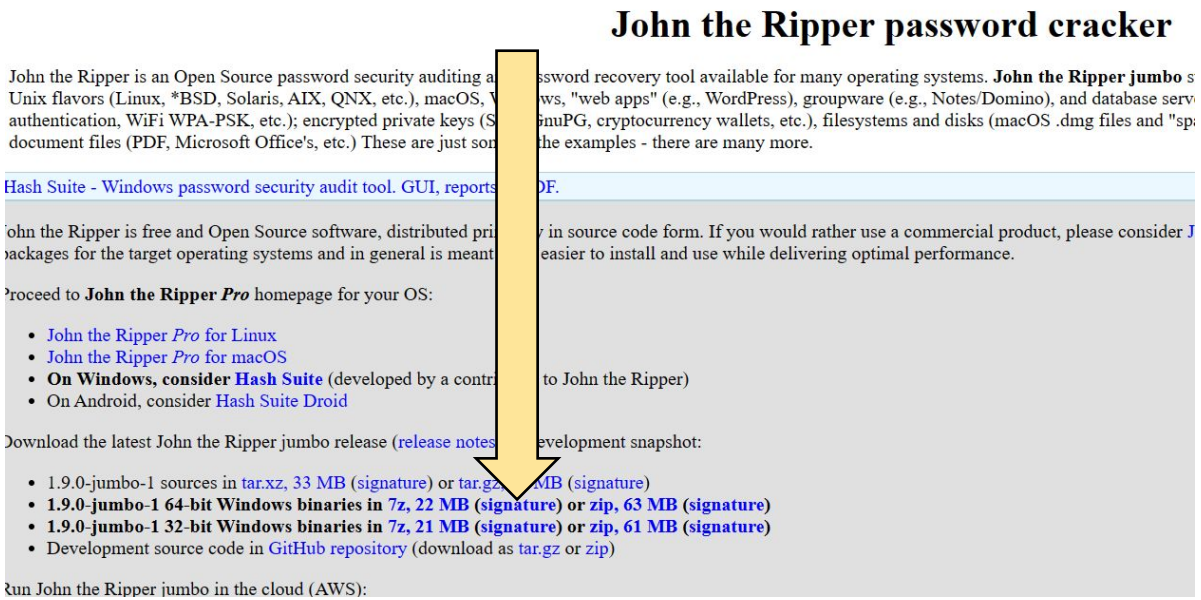
ZERO DAY CYBERSEC

PES UNIVERSITY



https://www.openwall.com/john/





John the Ripper password cracker

John the Ripper is an Open Source password security auditing a ssword recovery tool available for many operating systems. **John the Ripper jumbo** s
Unix flavors (Linux, *BSD, Solaris, AIX, QNX, etc.), macOS, W ws, "web apps" (e.g., WordPress), groupware (e.g., Notes/Domino), and database serv
authentication, WiFi WPA-PSK, etc.); encrypted private keys (S GnuPG, cryptocurrency wallets, etc.), filesystems and disks (macOS .dmg files and "sp
document files (PDF, Microsoft Office's, etc.) These are just son the examples - there are many more.

Hash Suite - Windows password security audit tool. GUI, reports DF.

ohn the Ripper is free and Open Source software, distributed pri y in source code form. If you would rather use a commercial product, please consider J
ackages for the target operating systems and in general is meant easier to install and use while delivering optimal performance.

roceed to **John the Ripper *Pro*** homepage for your OS:

- John the Ripper *Pro* for Linux
- John the Ripper *Pro* for macOS
- **On Windows, consider Hash Suite** (developed by a contr to John the Ripper)
- On Android, consider Hash Suite Droid

ownload the latest John the Ripper jumbo release (release notes evelopment snapshot:

- 1.9.0-jumbo-1 sources in tar.xz, 33 MB (signature) or tar.gz MB (signature)
- **1.9.0-jumbo-1 64-bit Windows binaries in 7z, 22 MB (signature) or zip, 63 MB (signature)**
- **1.9.0-jumbo-1 32-bit Windows binaries in 7z, 21 MB (signature) or zip, 61 MB (signature)**
- Development source code in GitHub repository (download as tar.gz or zip)

un John the Ripper jumbo in the cloud (AWS):

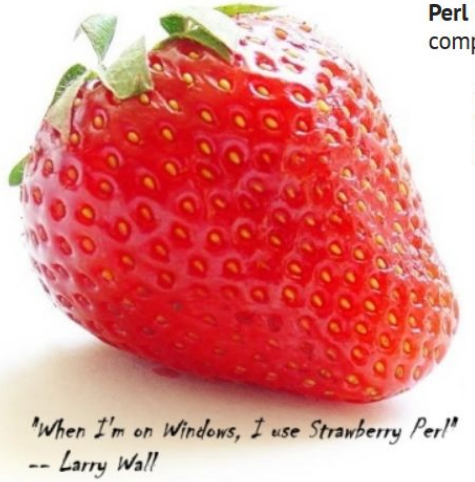
Download From the first Link
Extract and place in Directory.

https://strawberryperl.com/

Google

strawberry perl

The Perl for MS Windows, free of charge!



Perl is a programming language suitable for writing simple scripts as well as complex applications — see https://www.perl.org.

Strawberry Perl is a perl environment for MS Windows containing all you need to run and develop perl applications. It is designed to be as close as possible to perl environment on UNIX systems.

It includes perl binaries, compiler (gcc) + related tools, all the external libraries (crypto, math, graphics, xml...), all the bundled database clients and all you expect from Strawberry Perl.

Recommended version:

strawberry-perl-5.32.1.1-64bit.msi
strawberry-perl-5.32.1.1-32bit.msi

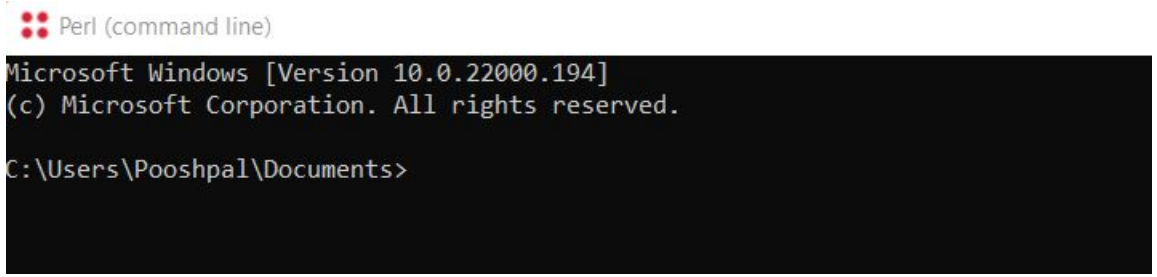
Download and run exe to install.

Perl is a general-purpose programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming, GUI development, and more.

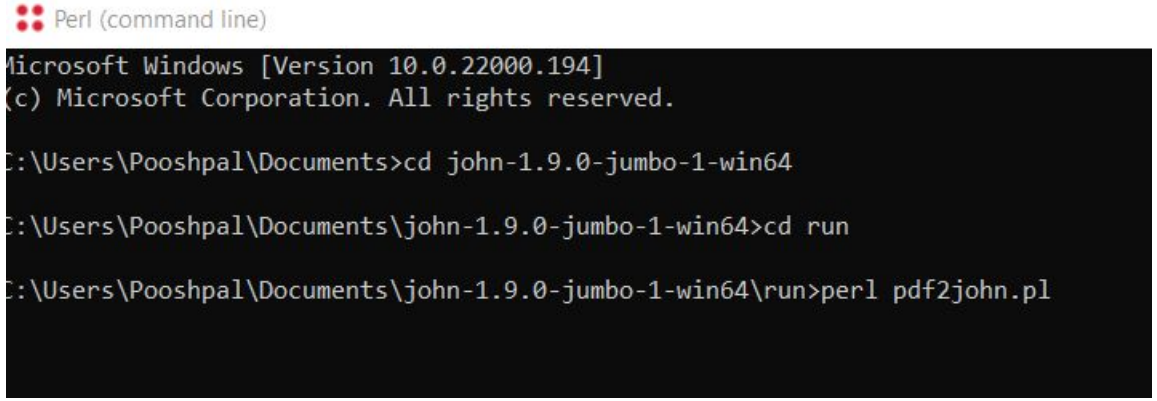
What is Perl?

- Perl is a stable, cross platform programming language.
- Though Perl is not officially an acronym but few people used it as **Practical Extraction and Report Language**.

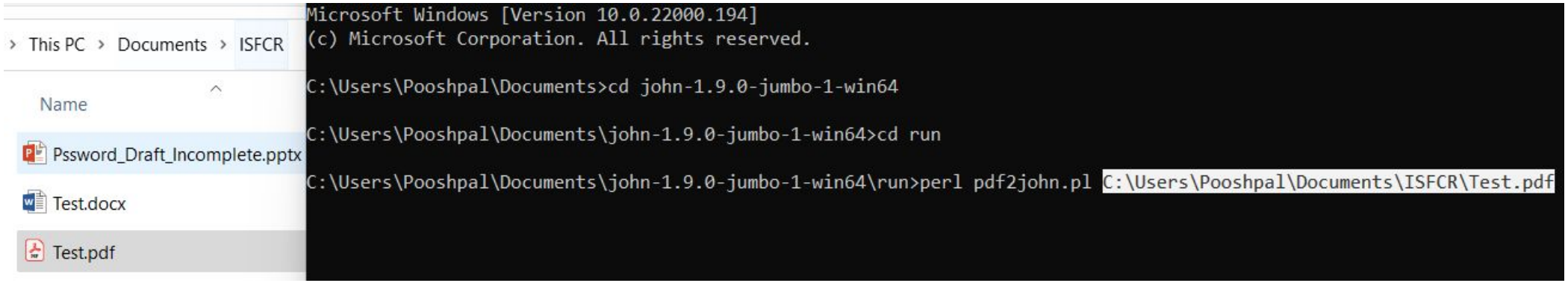
Open Perl(command Line)

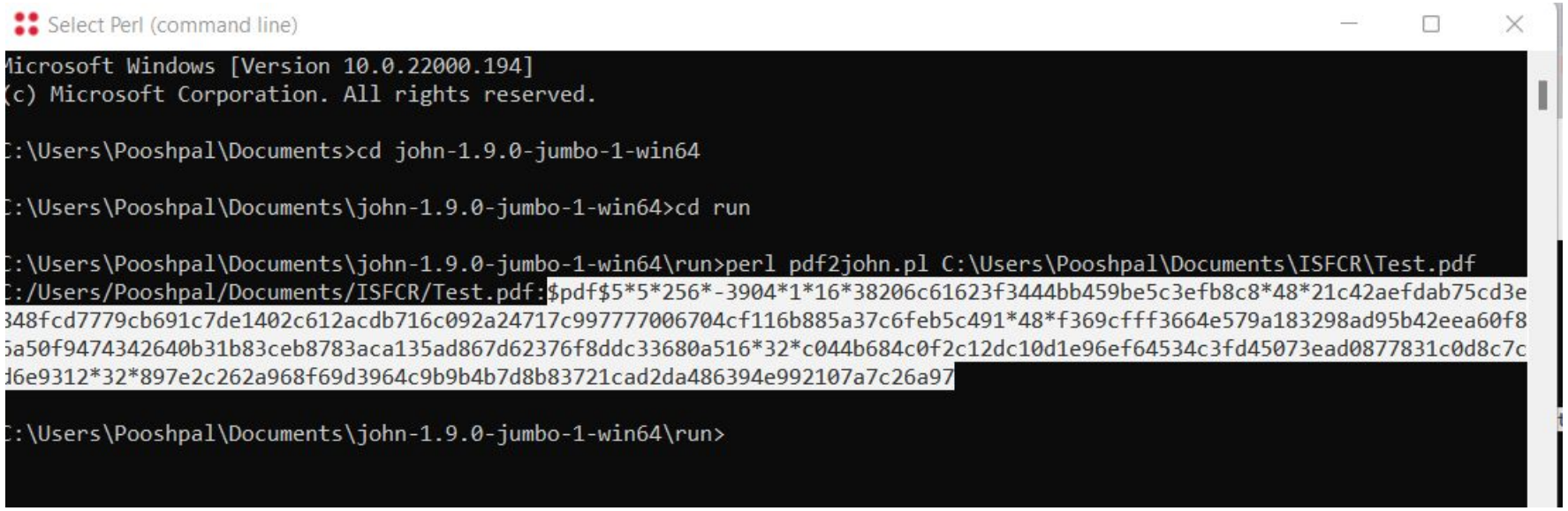


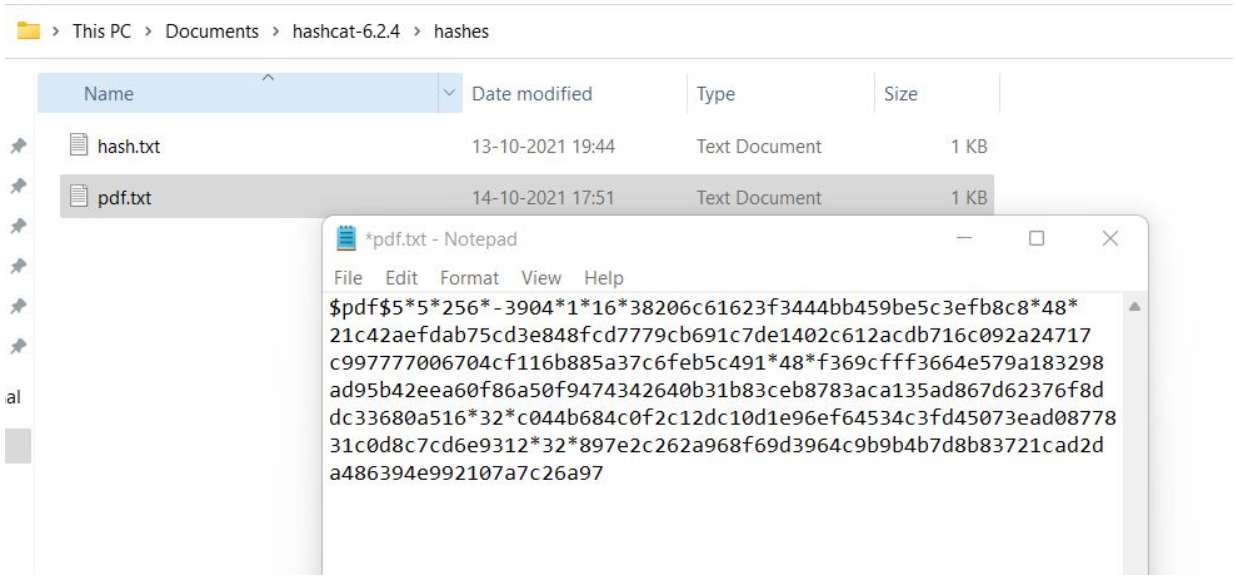
Reach the directory where you installed John The Ripper



Syntax: perl pdf2john.pl Directory1







Hash Generated

What Is John the Ripper?

One of the best security tools which can be used to crack passwords is John the Ripper. It has a high rank among all of its other counterparts in the market, supported by sectools.org which assures such information implying a sort of reliability. In addition, it is a free software which is considered a great characteristic of such program.

Caution:

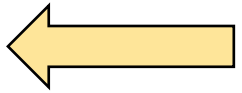
Neither can deal with encrypted files.

```
chird@ubuntu:~/Desktop/JohnTheRipper$ ./run/john --list=formats
descrypt, bsdicrypt, md5crypt, md5crypt-long, bcrypt, scrypt, LM, AFS,
tripcode, AndroidBackup, adxcrypt, agilekeychain, aix-ssha1, aix-ssha256,
aix-ssha512, andOTP, ansible, argon2, as400-des, as400-ssha1, asa-md5,
AxCrypt, AzureAD, BestCrypt, bfegg, Bitcoin, BitLocker, bitshares, Bitwarden,
BKS, Blackberry-ES10, WoWSRP, Blockchain, chap, Clipperz, cloudkeychain,
dynamic_n, cq, CRC32, sha1crypt, sha256crypt, sha512crypt, Citrix_NS10,
dahua, diskcryptor, Django, django-scrypt, dmd5, dmg, dominosec, dominosec8,
DPAPImk, dragonfly3-32, dragonfly3-64, dragonfly4-32, dragonfly4-64, Drupal7,
eCryptfs, eigrp, EncFS, enpass, EPI, EPiServer, ethereum, fde, Fortigate256,
Fortigate, FormSpring, FVDE, geli, gost, gpg, HAVAL-128-4, HAVAL-256-3, hdaa,
hMailServer, hsrp, IKE, ipb2, itunes-backup, iwork, KeePass, keychain,
keyring, keystore, known_hosts, krb4, krb5, krb5asrep, krb5pa-sha1, krb5tgs,
krb5-17, krb5-18, krb5-3, kwallet, lp, lpcli, leet, lotus5, lotus85, LUKS,
MD2, mdc2, MediaWiki, monero, money, MongoDB, scram, Mozilla, mscash,
mscash2, MSCHAPv2, mschapv2-naive, krb5pa-md5, mssql, mssql05, mssql12,
multibit, mysqlna, mysql-sha1, mysql, net-ah, nethalflm, netlm, netlmv2,
net-md5, netntlmv2, netntlm, netntlm-naive, net-sha1, nk, notes, md5ns,
nsec3, NT, o10glogon, o3logon, o5logon, ODF, Office, oldoffice,
OpenBSD-SoftRAID, openssl-enc, oracle, oracle11, Oracle12C, osc, ospf,
Padlock, Palshop, Panama, PBKDF2-HMAC-MD4, PBKDF2-HMAC-MD5, PBKDF2-HMAC-SHA1,
PBKDF2-HMAC-SHA256, PBKDF2-HMAC-SHA512, PDF, PEM, pfx, pgpdisk, pgpsda,
pgpwde, phpass, PHPS, PHPS2, pix-md5, po, postgres, PST, PuTTY, pwsafe, qnx,
RACF, RACF-KDFAES, radius, RAdmin, RAKP, rar, RAR5, Raw-SHA512, Raw-Blake2,
Raw-Keccak, Raw-Keccak-256, Raw-MD4, Raw-MD5, Raw-MD5u, Raw-SHA1,
Raw-SHA1-AxCrypt, Raw-SHA1-Linkedin, Raw-SHA224, Raw-SHA256, Raw-SHA3,
Raw-SHA384, restic, ripemd-128, ripemd-160, rsvp, RVARY, Siemens-S7,
Salted-SHA1, SSHA512, sapb, sapg, saph, sappse, securezip, 7z, Signal, SIP,
skein-256, skein-512, skey, SL3, Snefru-128, Snefru-256, LastPass, SNMP,
solarwinds, SSH, sspr, Stribog-256, Stribog-512, STRIP, SunMD5, SybaseASE,
Sybase-PROP, tacacs-plus, tcp-md5, telegram, tezos, Tiger, tc_aes_xts,
tc_ripemd160, tc_ripemd160boot, tc_sha512, tc_whirlpool, vdi, OpenVMS, vmx,
VNC, vtp, wbb3, whirlpool, whirlpool0, whirlpool1, wpapsk, wpapsk-pmk,
xmpp-scram, xsha, xsha512, zed, ZIP, ZipMonster, plaintext, has-160,
HMAC-MD5, HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512,
dummy, crypt
411 formats (149 dynamic formats shown as just "dynamic_n" here)
chird@ubuntu:~/Desktop/JohnTheRipper$
```

All the formats available and supported in John The Ripper module.

```
root@linuxpitstop:/opt/john/run# ./john
John the Ripper password cracker, ver: 1.7.9-jumbo-7 [linux-x86-64]
Copyright (c) 1996-2012 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--config=FILE             use FILE instead of john.conf or john.ini
--single[=SECTION]        "single crack" mode
--wordlist[=FILE] --stdin wordlist mode, read words from FILE or stdin
                  --pipe  like --stdin, but bulk reads, and allows rules
--loopback[=FILE]         like --wordlist, but fetch words from a .pot file
--dupe-suppression        suppress all dupes in wordlist (and force preload)
--encoding=NAME           input data is non-ascii (eg. UTF-8, ISO-8859-1).
                          For a full list of NAME use --list=encodings
--rules[=SECTION]         enable word mangling rules for wordlist modes
--incremental[=MODE]      "incremental" mode [using section MODE]
--markov[=OPTIONS]        "Markov" mode (see doc/MARKOV)
--external=MODE           external mode or word filter
--stdout[=LENGTH]         just output candidate passwords [cut at LENGTH]
--restore[=NAME]          restore an interrupted session [called NAME]
--session=NAME            give a new session the NAME
--status[=NAME]           print status of a session [called NAME]
--make-charset=FILE       make a charset file. It will be overwritten
--show[=LEFT]             show cracked passwords [if =LEFT, then uncracked]
--test[=TIME]             run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..] [do not] load this (these) user(s) only
--groups=[-]GID[,..]      load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]    load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX]    load salts with[out] COUNT [to MAX] hashes
--pot=NAME                pot file to use
--format=NAME             force hash type NAME: afs bf bfegg bsdi crc32 cryp
                          des django dmd5 dominosec dragonfly3-32 dragonfly3
```



Attack Modes

Brute-Force Using Dict: *Syntax: john --format=raw-md5 –wordlist=Directory1 Directory2*

```
└─$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt hash1.txt
```

Brute-Force Using Single Mode: *Syntax: john –single –format=raw-md5 Directory1*

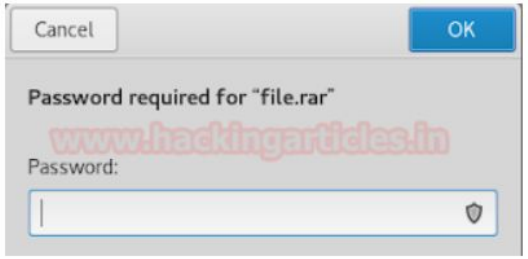
```
john --single --format=raw-md5 hash7.txt
```

```
root@kali:~# john
John the Ripper password cracker, version 1.8.0.6-jumbo-1-
-64]
Copyright (c) 1996-2015 by Solar Designer and others
Homepage: http://www.openwall.com/john/
www.hackingarticles.in

Usage: john [OPTIONS] [PASSWORD-FILES]
--single[=SECTION]        "single crack" mode
--wordlist[=FILE] --stdin wordlist mode, read words from F
                  --pipe  like --stdin, but bulk reads, an
--loopback[=FILE]         like --wordlist, but fetch words
--dupe-suppression        suppress all dupes in wordlist (
--prince[=FILE]           PRINCE mode, read words from FIL
--encoding=NAME           input encoding (eg. UTF-8, ISO-8
                          doc/ENCODING and --list=hidden-o
--rules[=SECTION]         enable word mangling rules for w
--incremental[=MODE]      "incremental" mode [using sectio
--mask=MASK               mask mode using MASK
--markov[=OPTIONS]        "Markov" mode (see doc/MARKOV)
--external=MODE           external mode or word filter
```

- In this manner, a hacker's computer can guess the right password and recover it, especially if the password contains clear-text words for which a "dictionary attack" is where the process is derived.
- On the other hand, a password could be recovered through what is called 'rainbow' table. It is much faster and contains password hashes from which a password is guessed by a computer system.

Cracking the RAR Password Hash



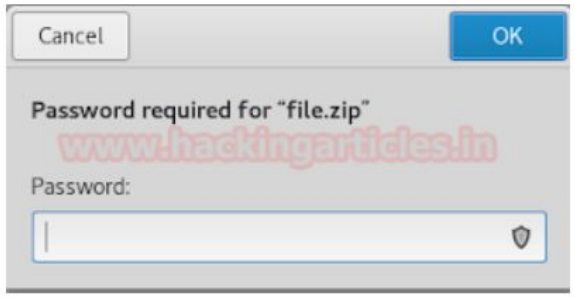
Now John cannot directly crack this key, first, we will have to change it format, which can be done using a john utility called "rar2john".

Syntax: rar2john [location of key]

```
rar2john file.rar > crack.txt
```

Ways to crack RAR file which is password protected.

Cracking the ZIP Password Hash

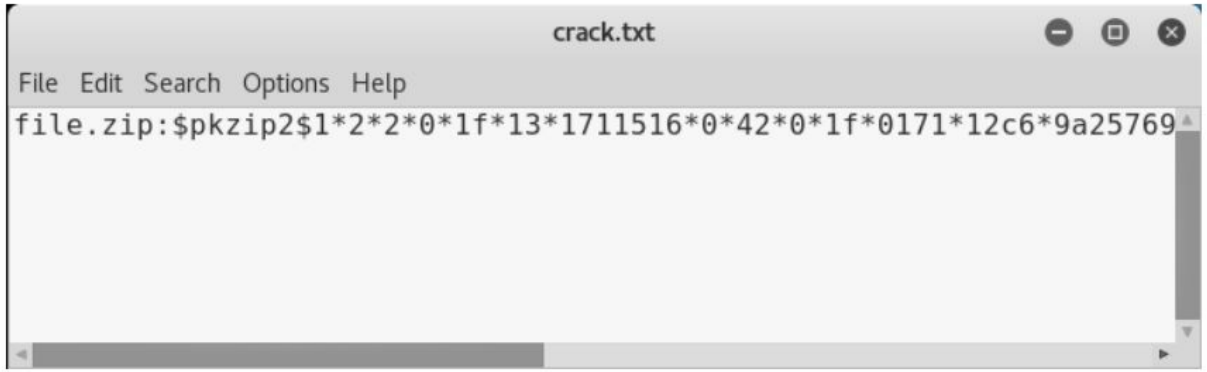


Ways to crack ZIP file which is password protected.

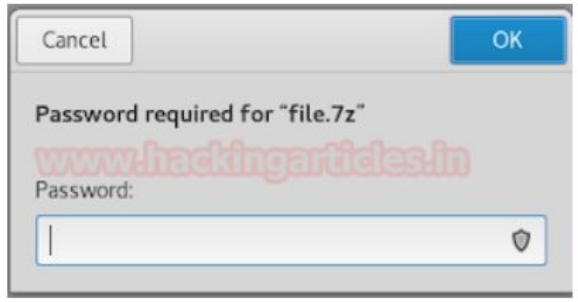
Now John cannot directly crack this key, first, we will have to change its format, which can be done using a john utility called "zip2john".

Syntax: zip2john [location of key]

```
zip2john file.zip > crack.txt
```



Cracking the 7-Zip Password Hash



Ways to crack 7-Zip file which is password protected.

Now John cannot directly crack this key, first, we will change its format, which can be done using a john utility called "7z2john". This is not inbuilt utility, It can be downloaded from here.

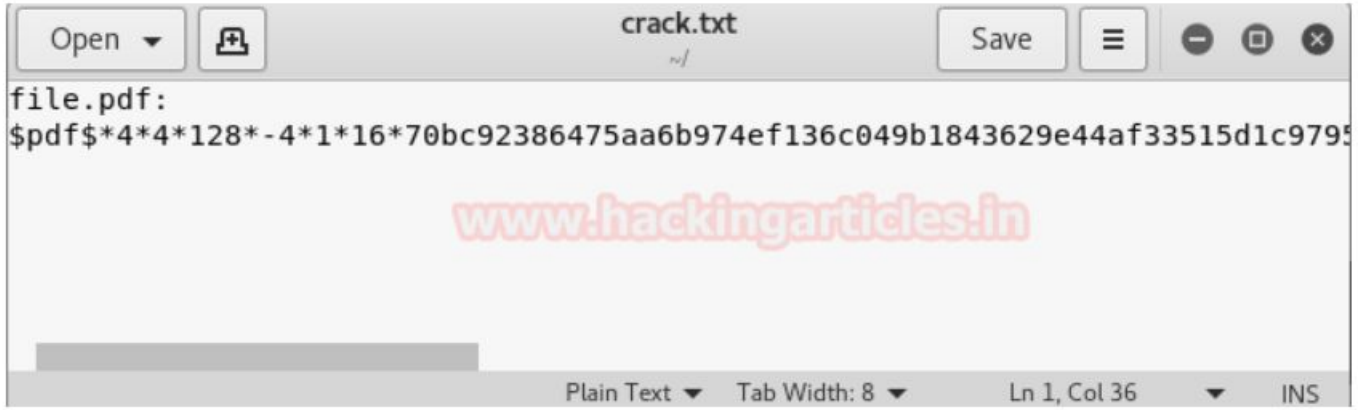
Syntax: zip2john [location of key]

```
python 7z2john.py file.7z > crack.txt
```

Cracking the PDF Password Hash

Syntax: pdf2john [location of key]

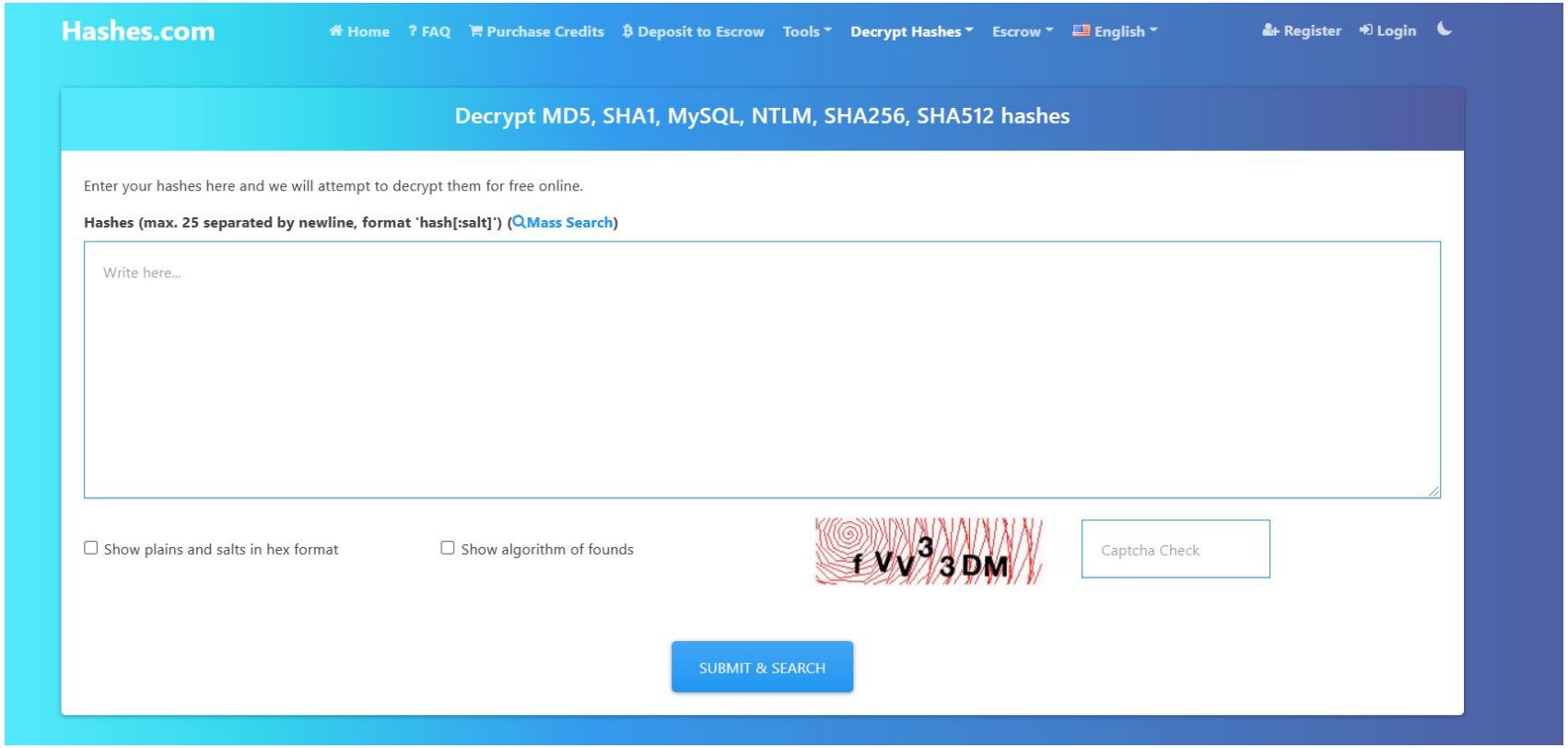
```
python pdf2john.py file.pdf > crack.txt
```



Ways to crack PDF file which is password protected.

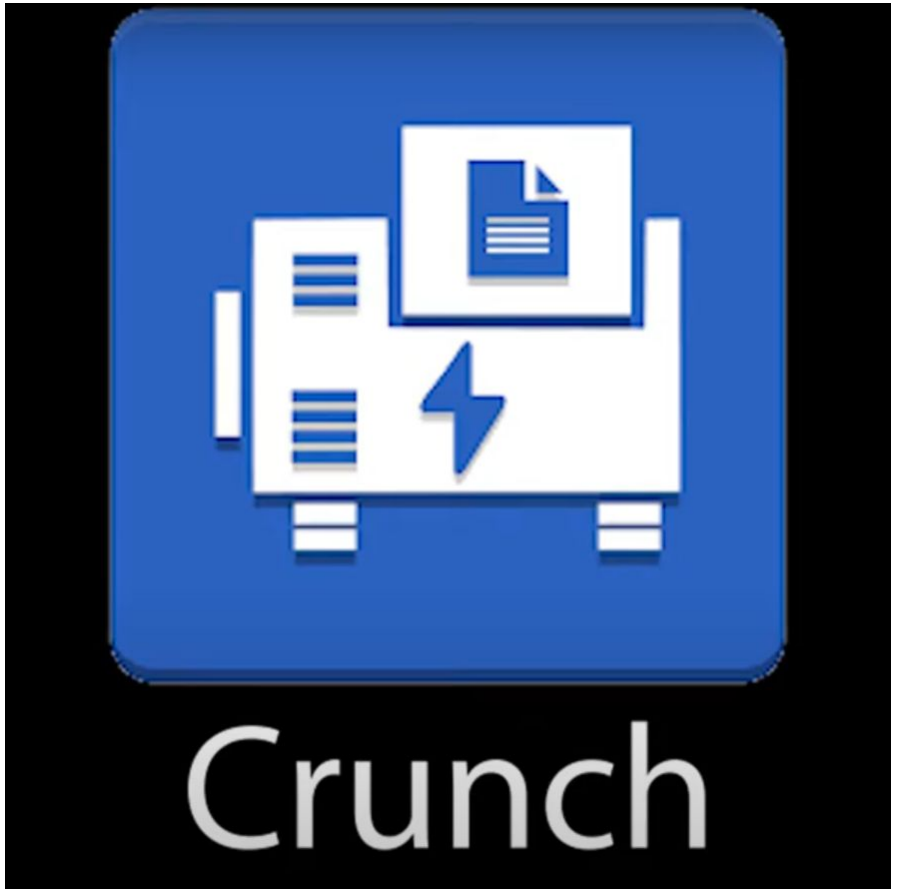
Now let's use John the Ripper to crack this hash.

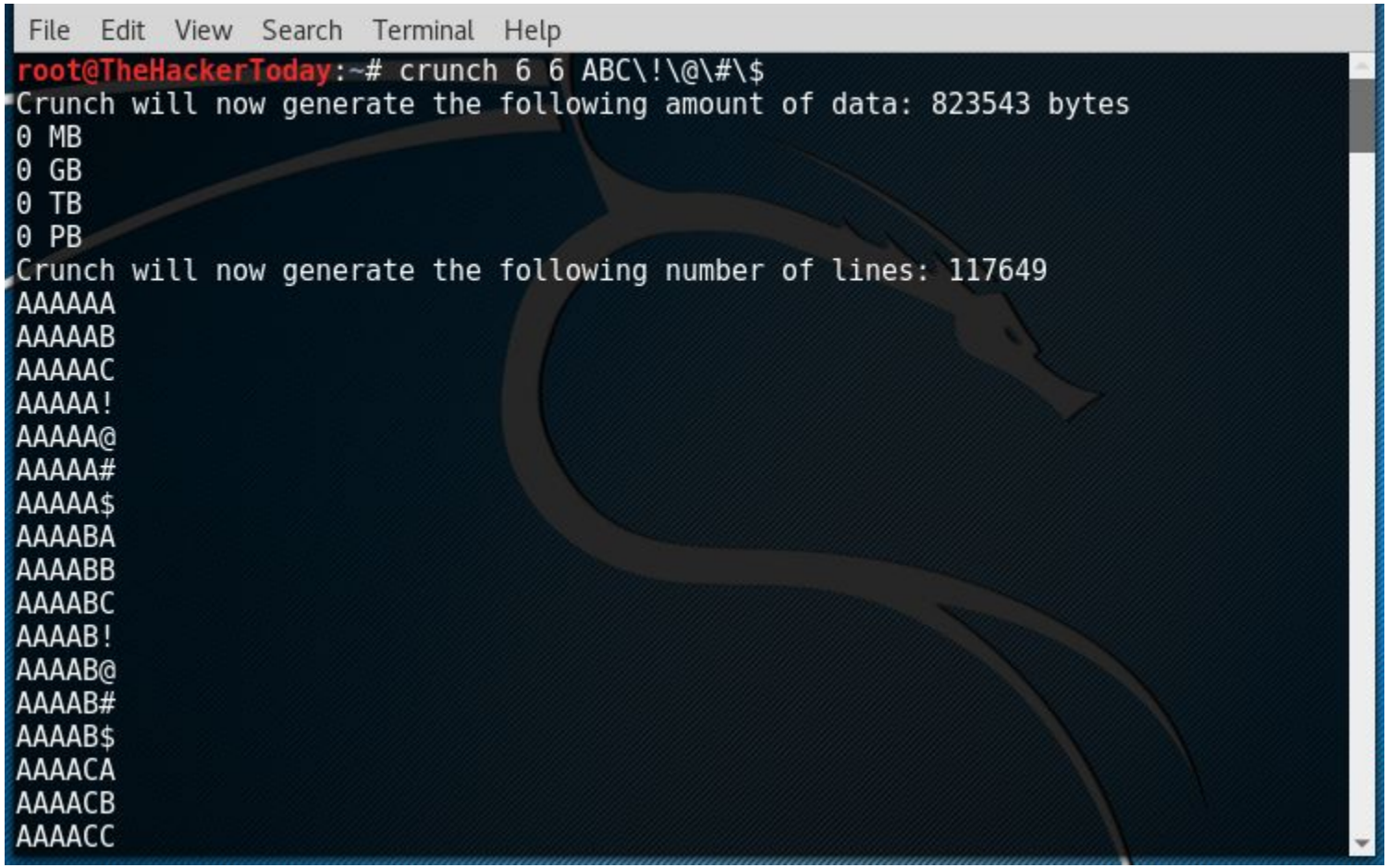
Hash Identifier Website:
https://hashes.com/en/decrypt/hash



PES UNIVERSITY

Creating wordlists with Crunch




File Edit View Search Terminal Help
root@TheHackerToday:~# crunch 6 6 ABC\!\@\#\$
Crunch will now generate the following amount of data: 823543 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 117649
AAAAAA
AAAAAB
AAAAAC
AAAAA!
AAAAA@
AAAAA#
AAAAA$
AAAABA
AAAABB
AAAABC
AAAAB!
AAAAB@
AAAAB#
AAAAB$
AAAACA
AAAACB
AAAACC

Wordlist Generator Website: https://www.securesafepro.com/pasgen.html

SecureSafe Pro
Password Manager

Password Manager Download Buy Free Password Generator

Free Password Generator

Generate any quantity of random, strong and secure passwords with one mouse click just in seconds with Free Password Generator software.

Free Password Generator application will create strong and secure passwords instead of you.

With **Free Password Generator** you will use only strong random passwords, that can protect your identity from potential harm.

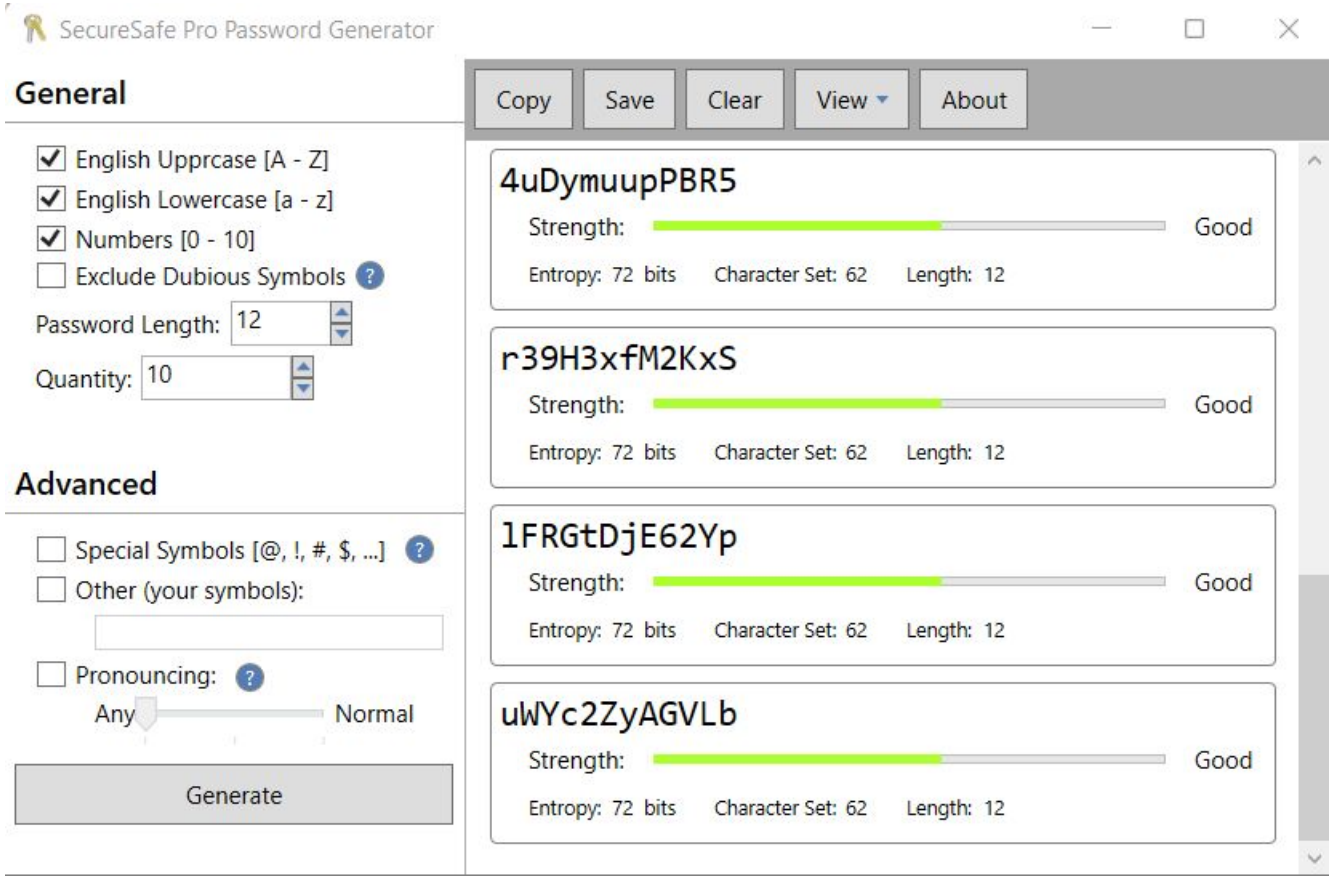
Download Free Password Generator

Latest version: 5.43 (January 13, 2017) | Size: 0,7 Mb
For **Windows 10 / Windows 8.1 / Windows 8 / Windows 7** / Windows XP / Windows Vista

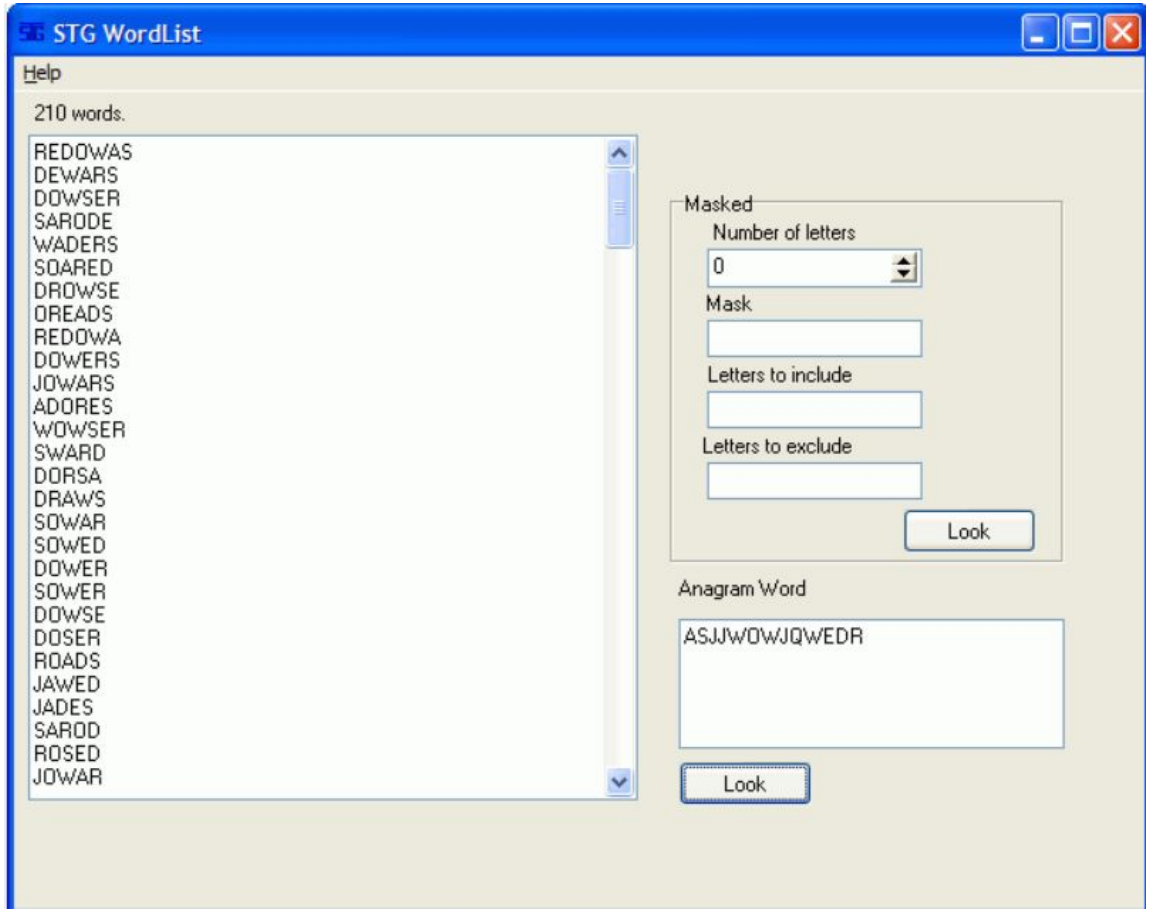
Download Free Portable Password Generator

For **Windows 10 / Windows 8.1 / Windows 8 / Windows 7** / Windows XP / Windows Vista

Enter the Parameters here.



Creating wordlists **STG Wordlist**
https://www.stgsys.com/wordlist.asp



ONLINE

Hydra

Hydra (**better known as "thc-hydra"**) is an online password attack tool. It brute forces various combinations on live services like telnet, ssh, http, https, smb, snmp, smtp etc. Hydra supports 30+ protocols including their SSL enabled ones. It brute forces on services we specify by using **user-lists & wordlists**.

PES UNIVERSITY

Hydra works in 4 modes :

- One username & one password
- User-list & One password
- One username & Password list
- User-list & Password list

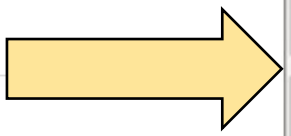
Target - Settings of various target oprions.

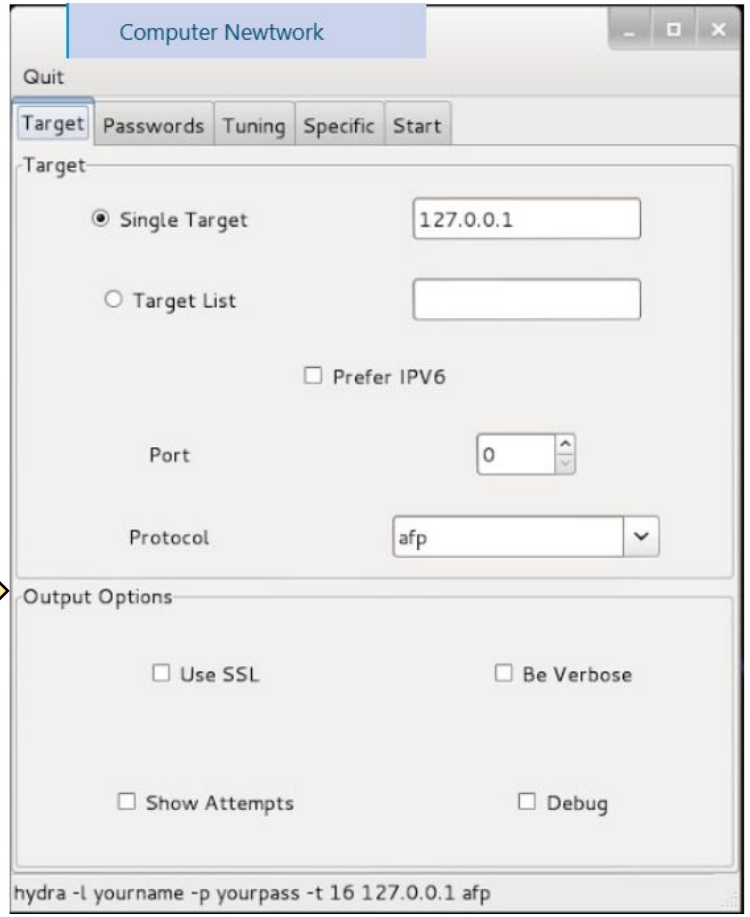
Passwords – Specify password options & wordlists.

Tuning – Secify how fast should hydra work. Other timing options are also available.

Specific – For testing on specific targets like a domain, https proxy etc.

Start – Start/Stop & shows the output.





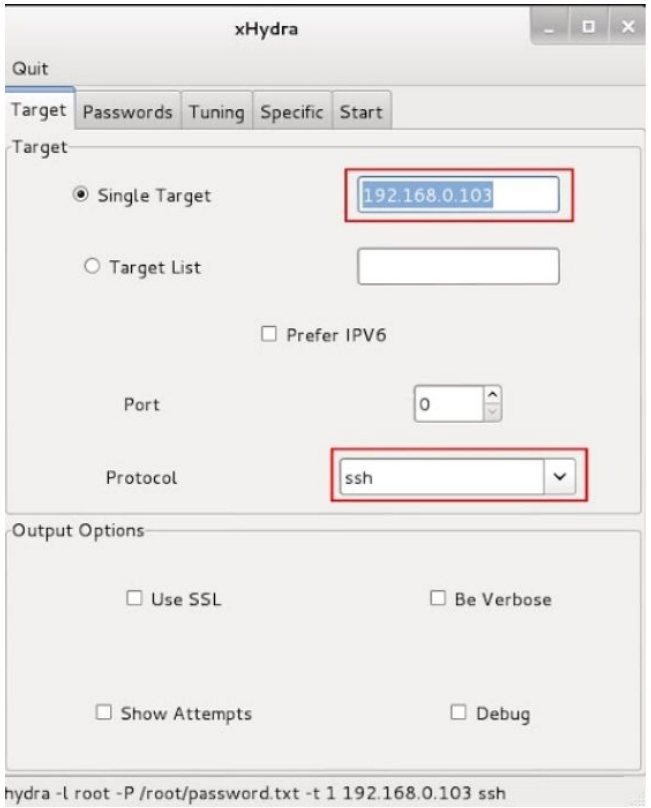
PESU ISFCR

ZERO DAY CYBERSEC

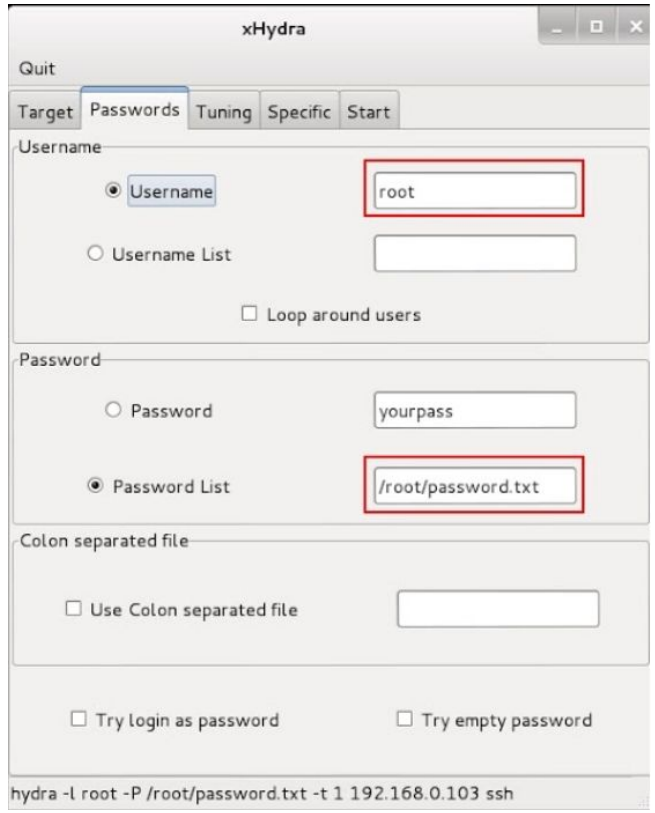
PES UNIVERSITY

Breaking an ssh with wordlist attack – Hydra

Set Target & protocol in the target tab.



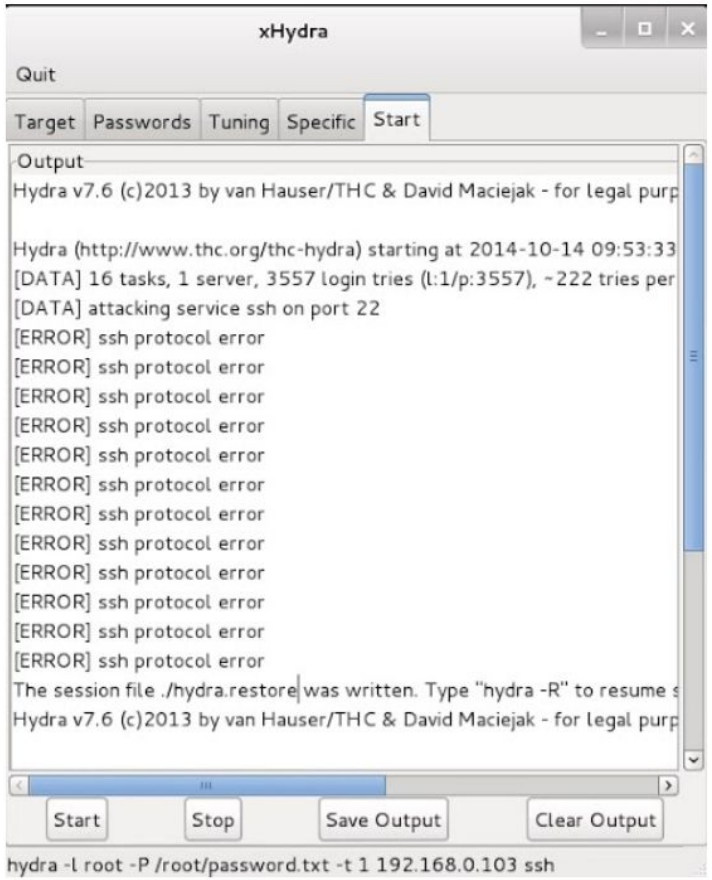
Set the username as root & specify the location for a wordlist in passwords tab.



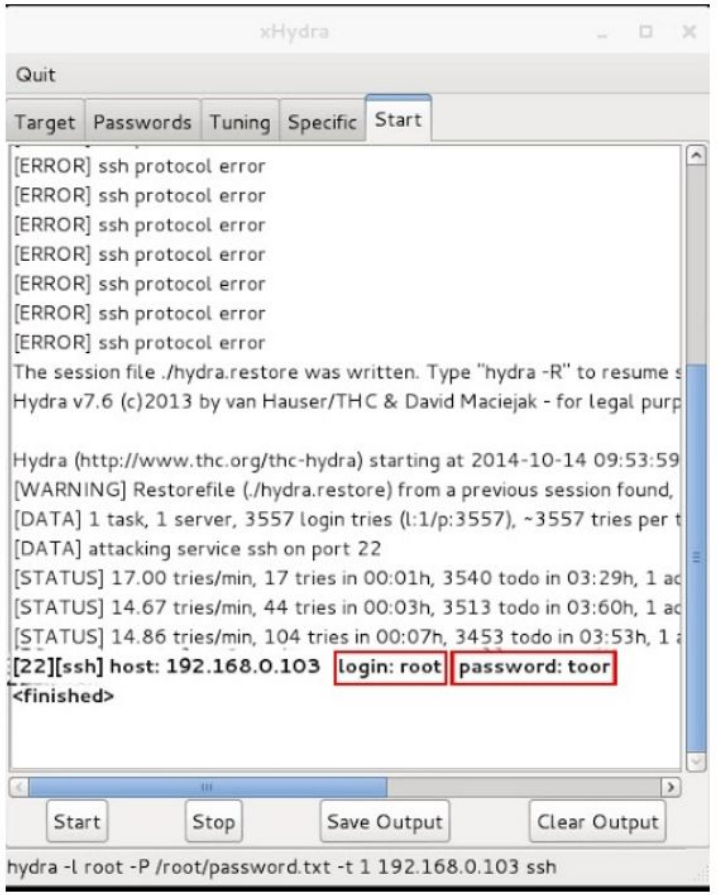
PESU ISFCR

ZERO DAY CYBERSEC

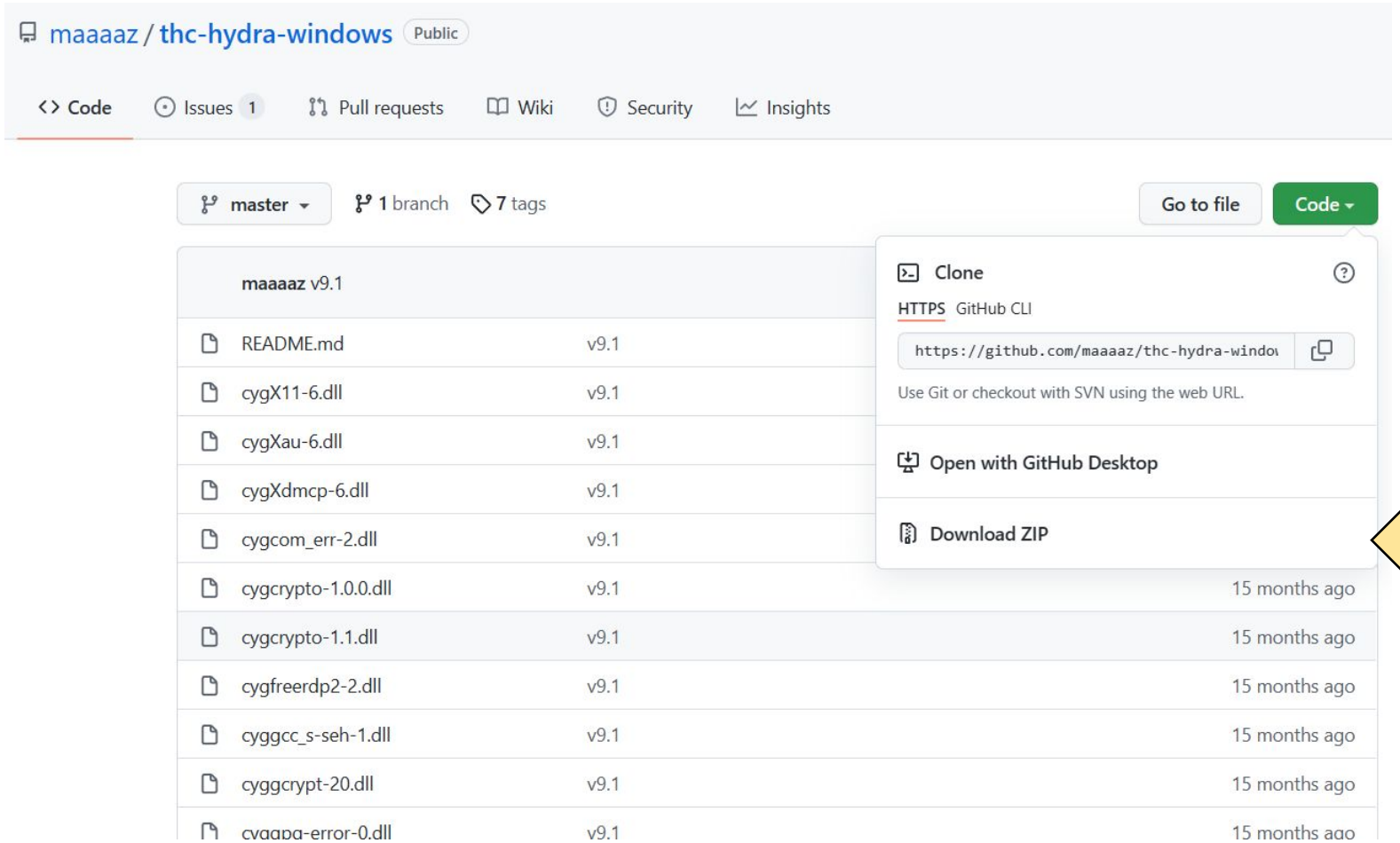
Start the thc-hydra from Start tab.



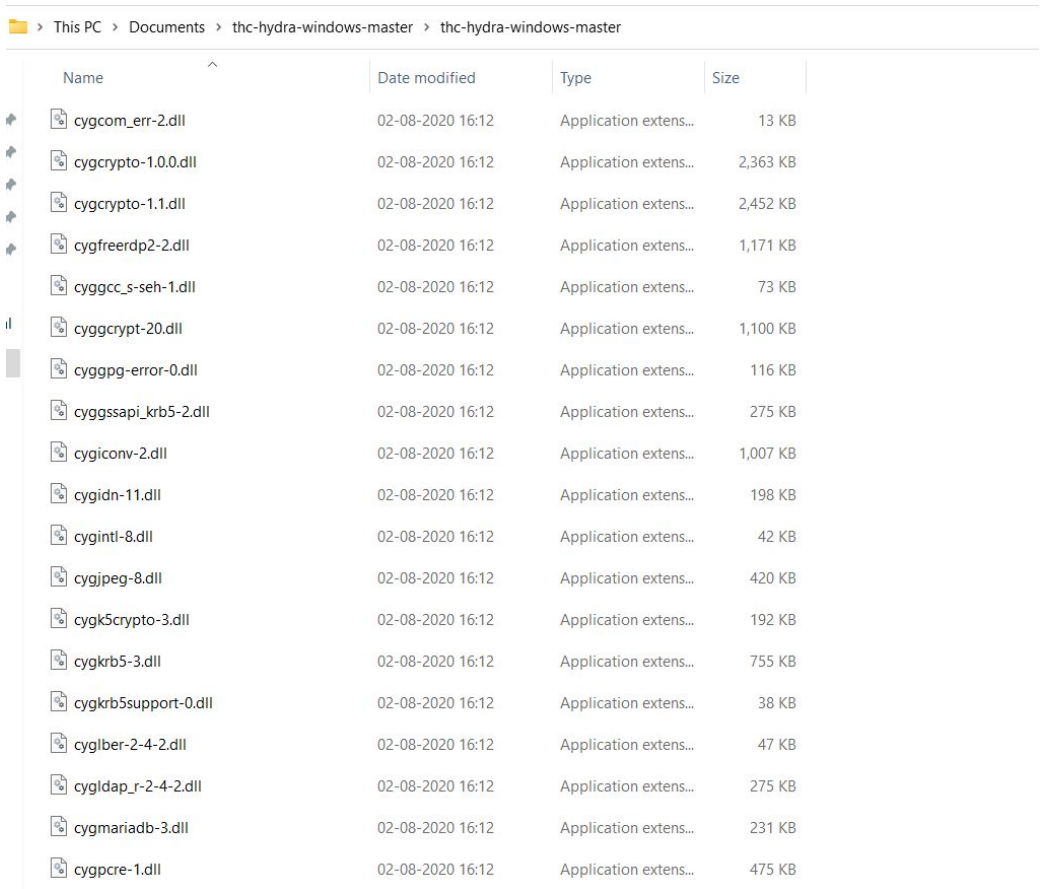
Scroll Down & Wait until the password gets cracked



Download THC Hydra :https://github.com/maaaaz/thc-hydra-windows

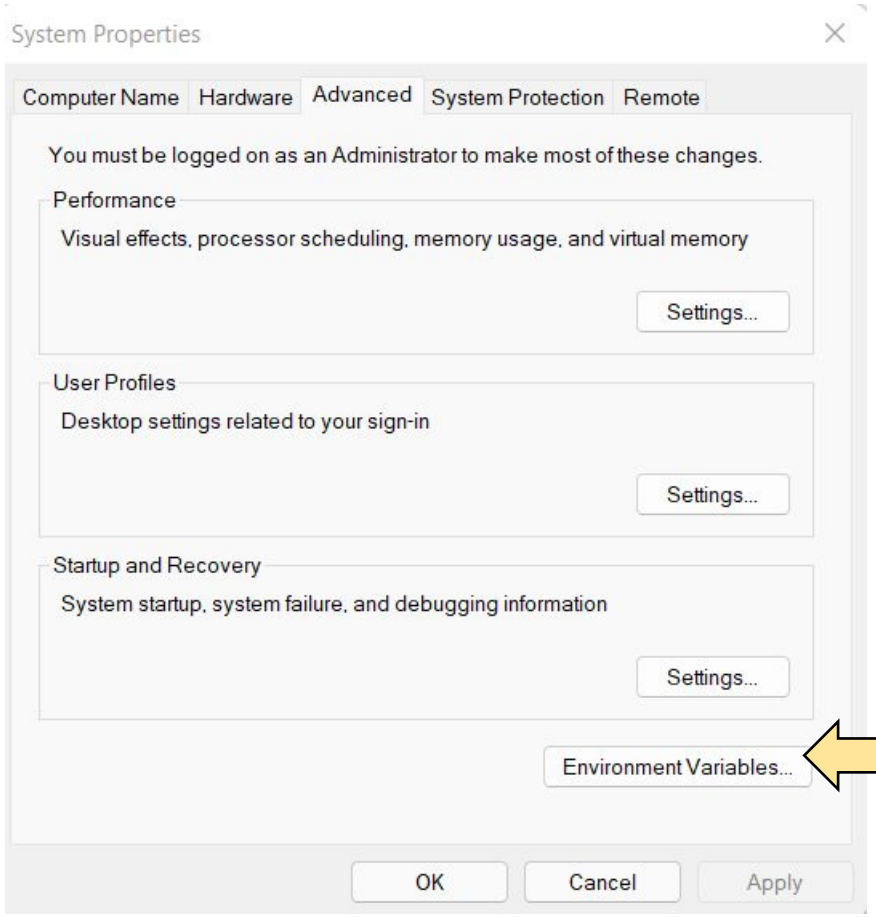


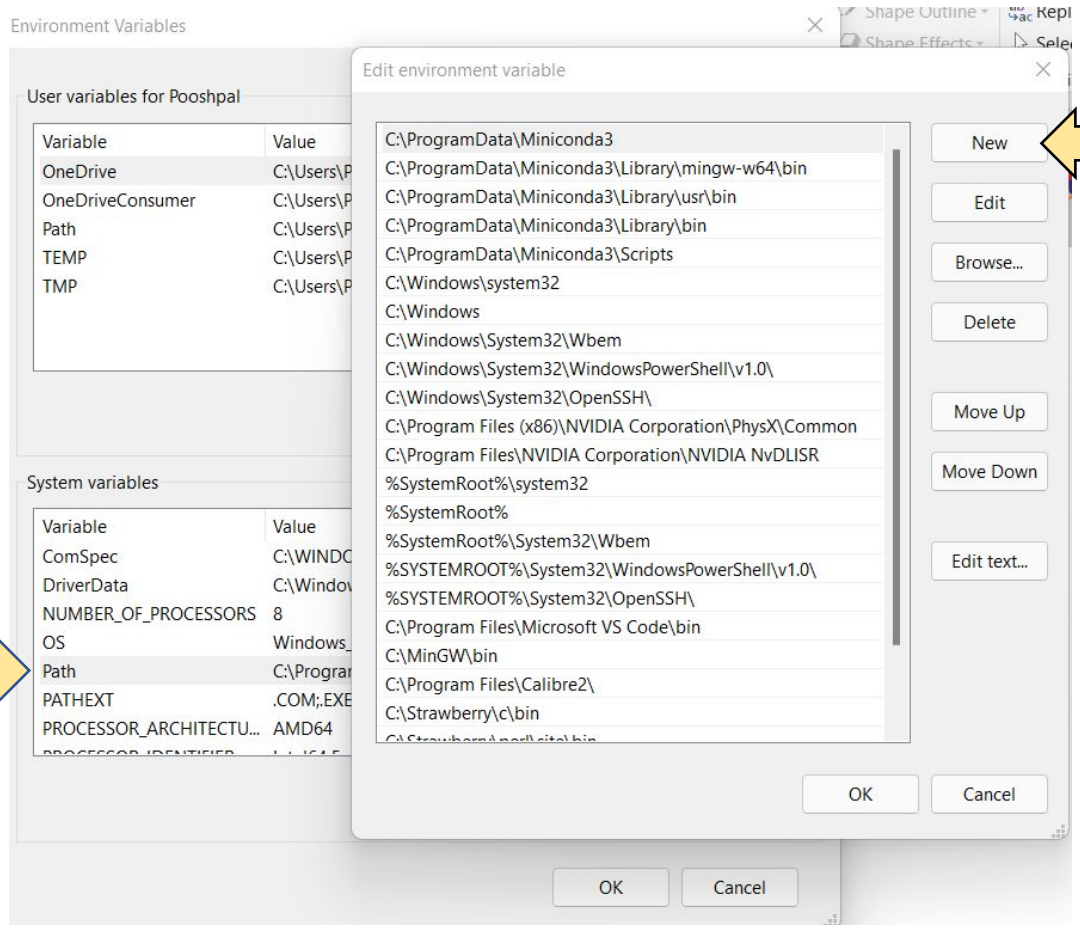
Download

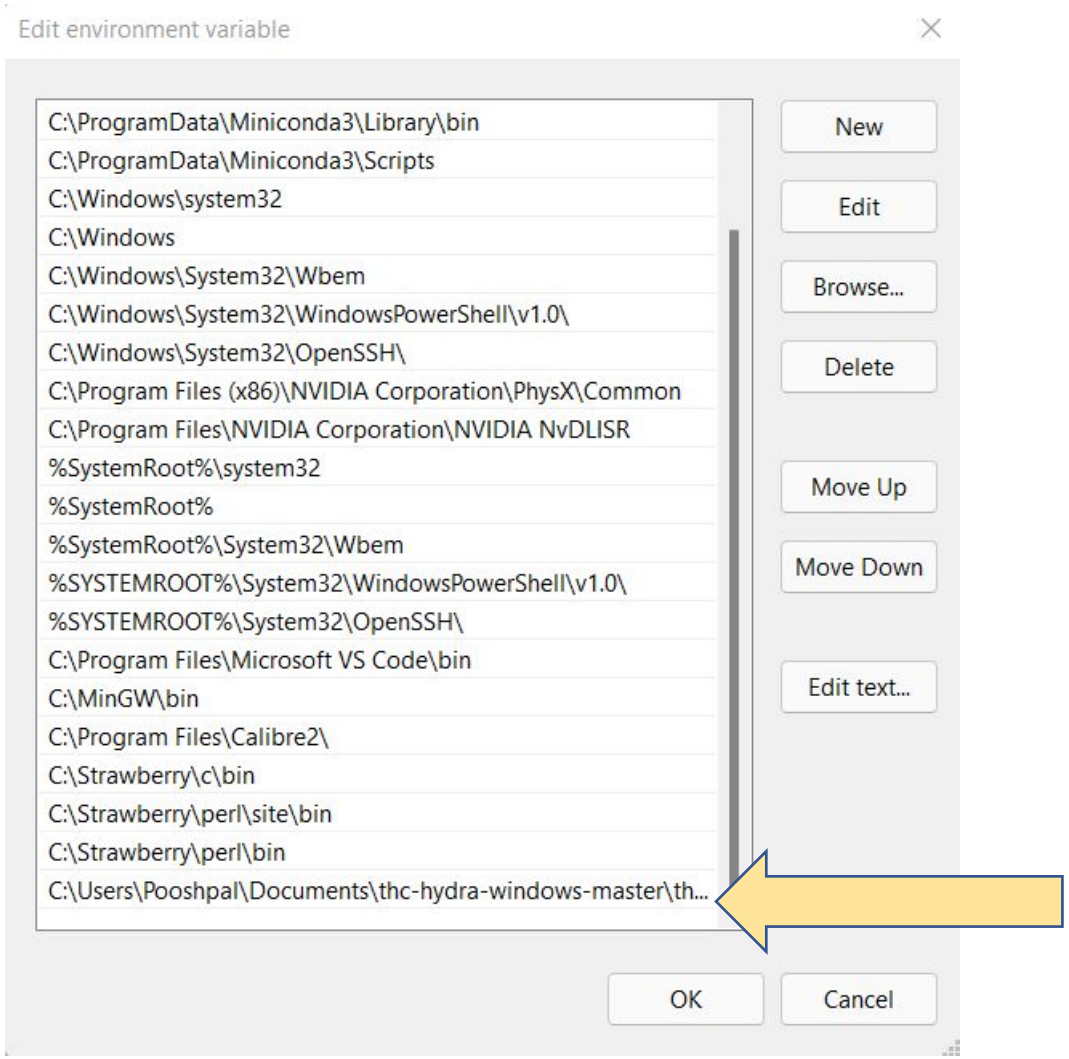


Extract in desired Location.

Then copy address/directory.


System Properties
Computer Name
Hardware
Advanced
System Protection
Remote
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...
User Profiles
Desktop settings related to your sign-in
Settings...
Startup and Recovery
System startup, system failure, and debugging information
Settings...
Environment Variables...
OK
Cancel
Apply


Environment Variables
User variables for Pooshpal
Variable
Value
OneDrive
OneDriveConsumer
Path
TEMP
TMP
System variables
Variable
Value
ComSpec
DriverData
NUMBER_OF_PROCESSORS
OS
Path
PATHEXT
PROCESSOR_ARCHITECTU...
AMD64
OK
Cancel
Edit environment variable
C:\ProgramData\Miniconda3
C:\ProgramData\Miniconda3\Library\mingw-w64\bin
C:\ProgramData\Miniconda3\Library\usr\bin
C:\ProgramData\Miniconda3\Library\bin
C:\ProgramData\Miniconda3\Scripts
C:\Windows\system32
C:\Windows
C:\Windows\System32\Wbem
C:\Windows\System32\WindowsPowerShell\v1.0\
C:\Windows\System32\OpenSSH\
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Microsoft VS Code\bin
C:\MinGW\bin
C:\Program Files\Calibre2\
C:\Strawberry\c\bin
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
OK
Cancel



Add the Directory in Paths in Environment Variables of the system.

Create new and add.

```
C:\Users\Pooshpal>hydra
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS]
] [service://server[:PORT][/OPT]]

Options:
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to attack, one entry per line, ':' to specify port
  -t TASKS  run TASKS number of connects in parallel per target (default: 16)
  -U        service module usage details
  -m OPT    options specific for a module, see -U output for information
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cvs ftp[s] http[s]-{head|get|post} http
nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin

Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)

Example:  hydra -l user -P passlist.txt ftp://192.168.0.1

C:\Users\Pooshpal>
```

Now open cmd and type hydra

```
Examples:
 hydra -l user -P passlist.txt ftp://192.168.0.1
 hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN
 hydra -C defaults.txt -6 pop3s://[2001:db8::1]:143/TLS:DIGEST-MD5
 hydra -l admin -p password ftp://[192.168.0.0/24]/
 hydra -L logins.txt -P pws.txt -M targets.txt ssh
```

http://127.0.0.1/dvwa/login.php

Username

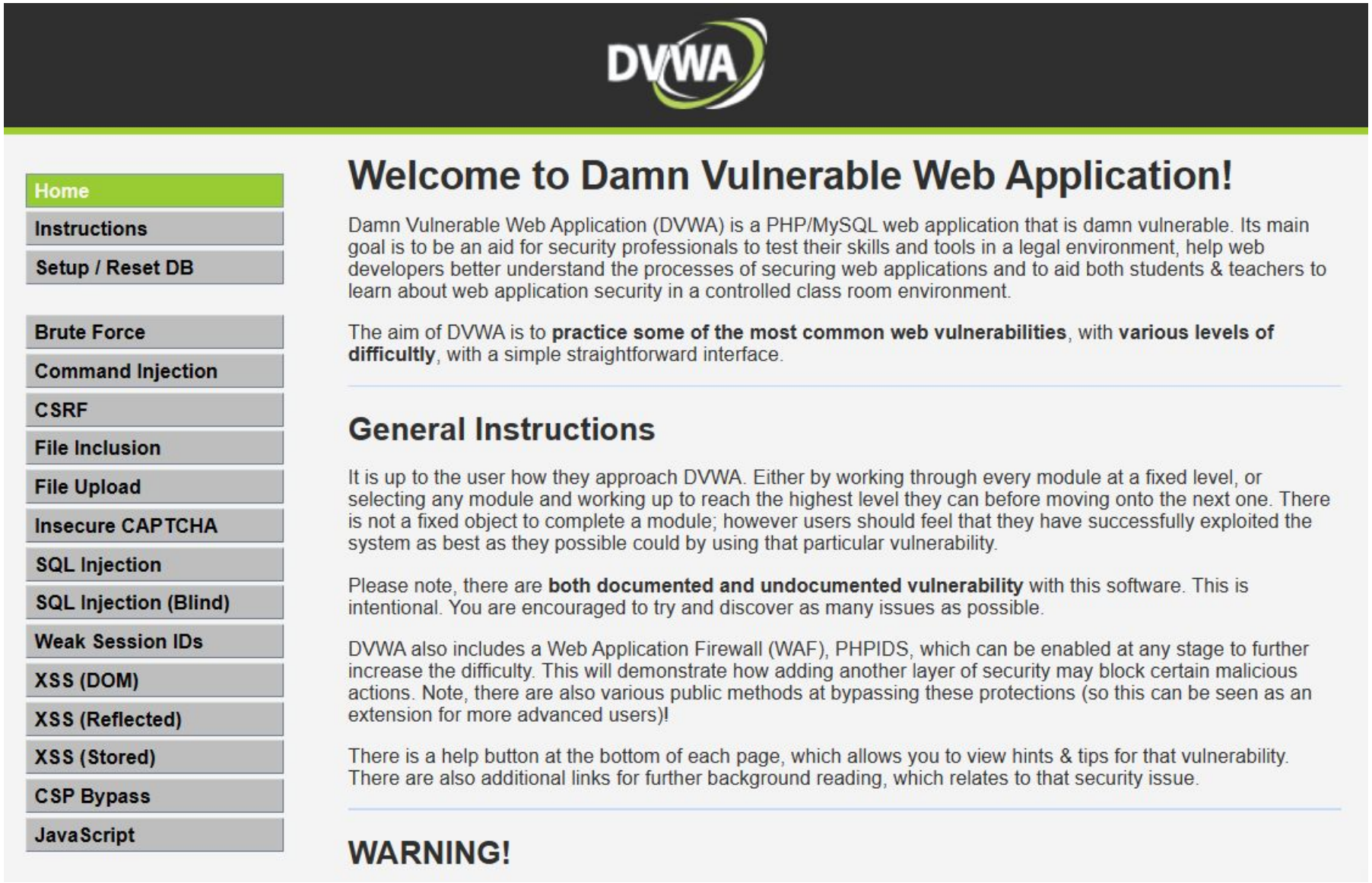
Password

Login

How To Install:

https://www.linkedin.com/pulse/how-setup-dvwa-windows-10-using-xampp-shubham-yadav#:~:text=%20How%20To%20Setup%20DVWA%20In%20Windows%2010,7%20Click%20on%20%E2%80%98Create%2FReset%20Database%E2%80%99.%20%20More%20

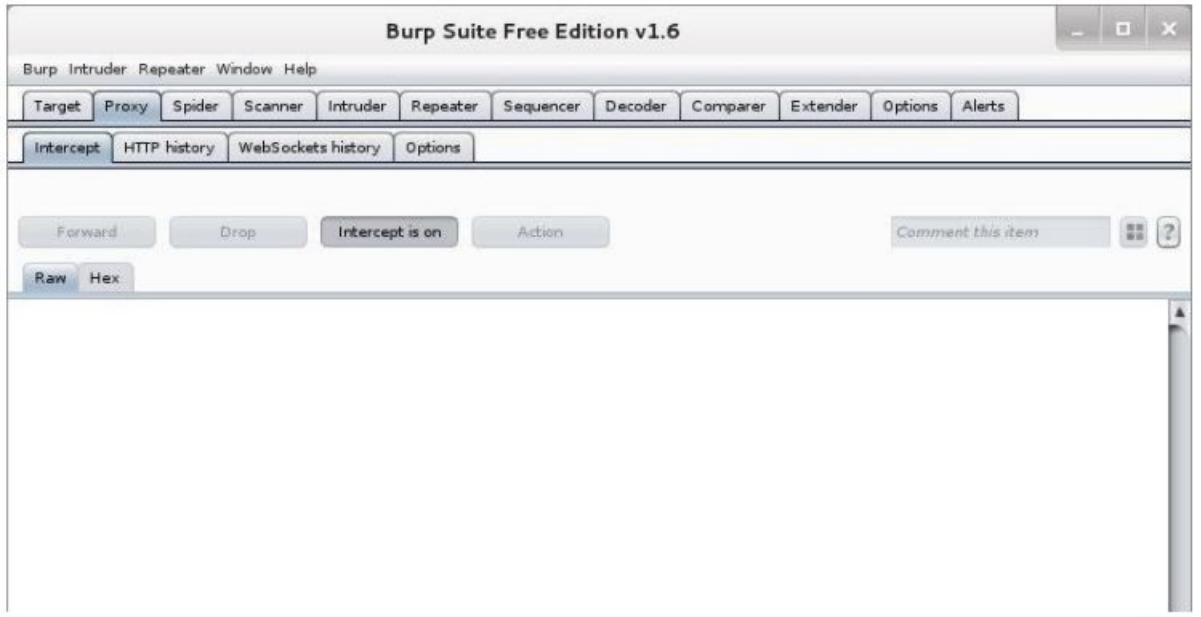
PES UNIVERSITY



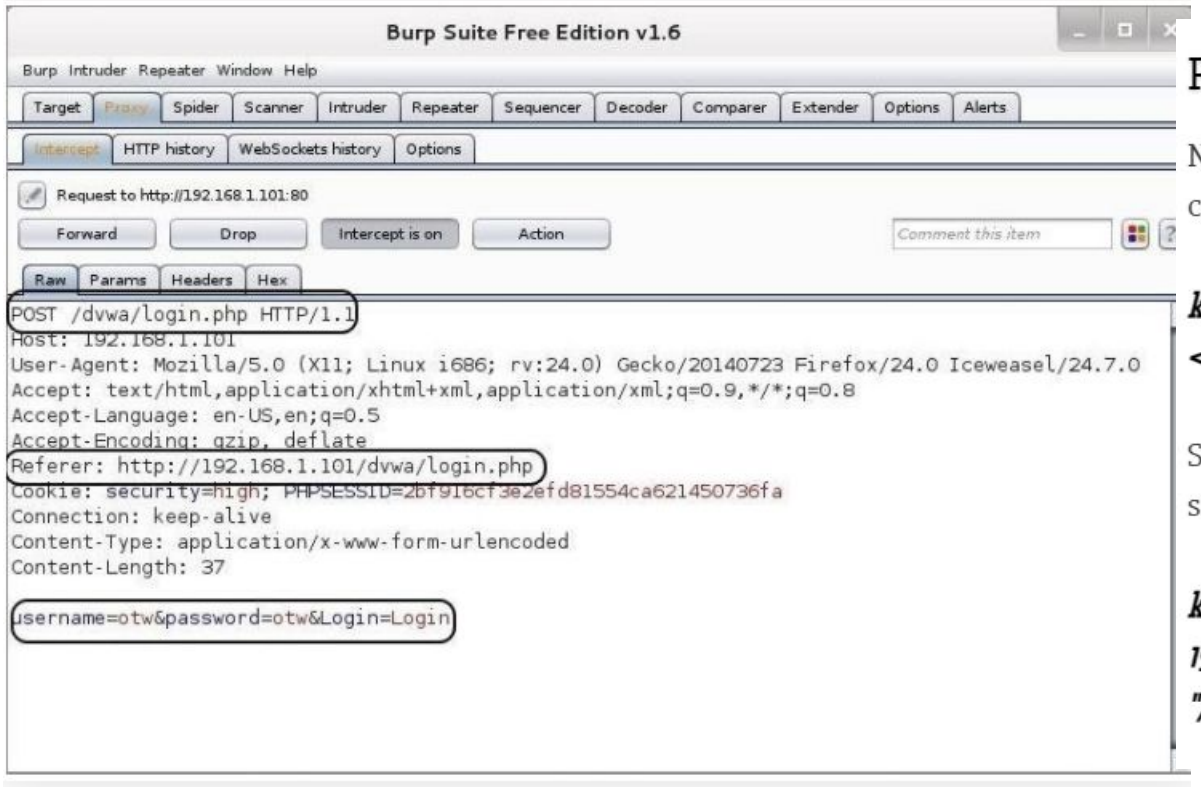
PESU ISFCR

ZERO DAY CYBERSEC

Next step - Intercepting HTTP traffic with Burp Proxy



https://download.freedownloadmanager.org/Windows-PC/Burp-Suite-Community-Edition/FREE-2021.3.3.html
Needs JDK 17



Place the Parameters into Your THC Hydra Command

Now, that we have the parameters, we can place them into the THC-Hydra command. The syntax looks like this:

kali > hydra -L <username list> -p <password list> <IP Address> <form parameters> <failed login message>

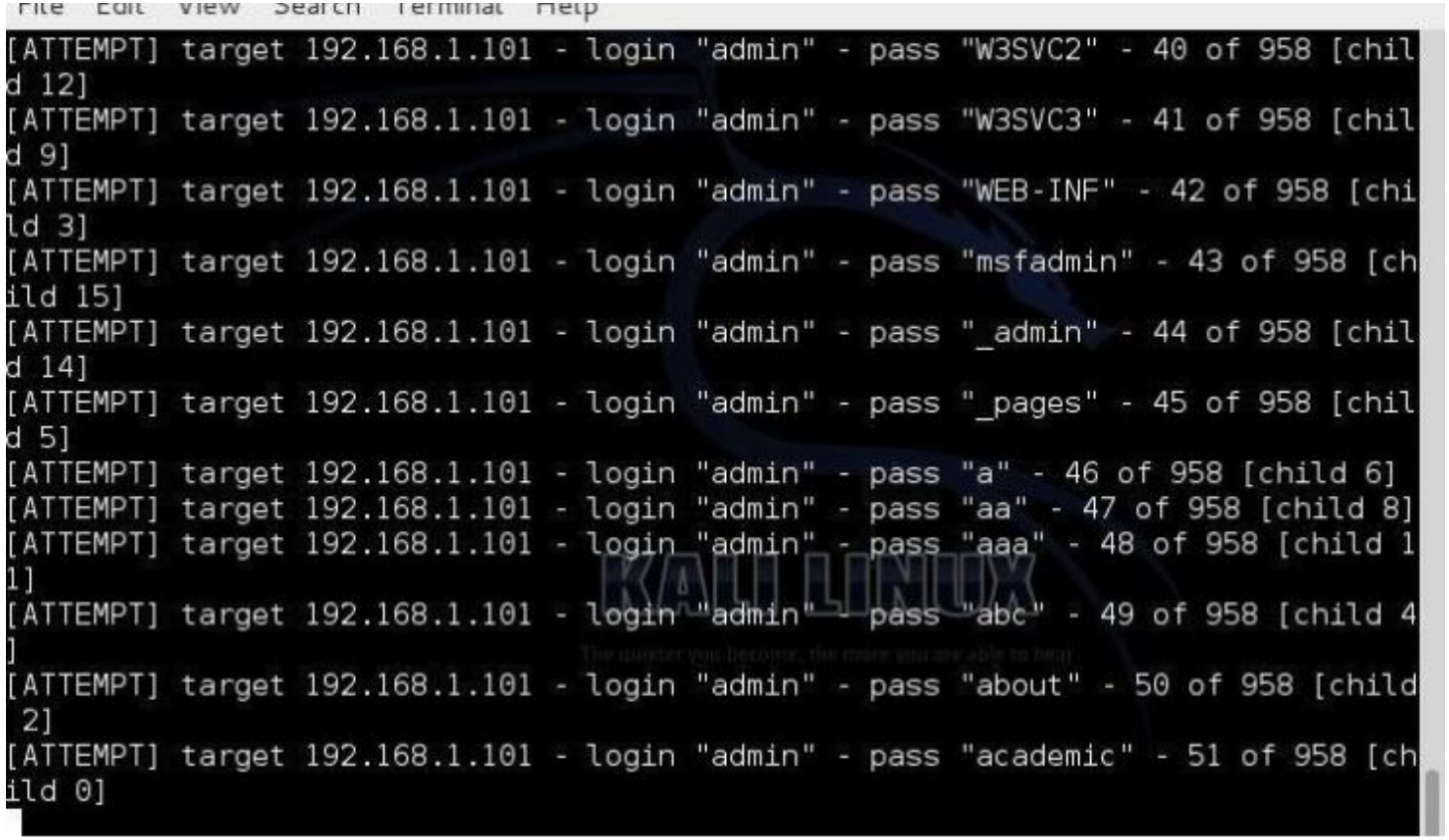
So, based on the information we have gathered from Burp Suite, our command should look something like this:

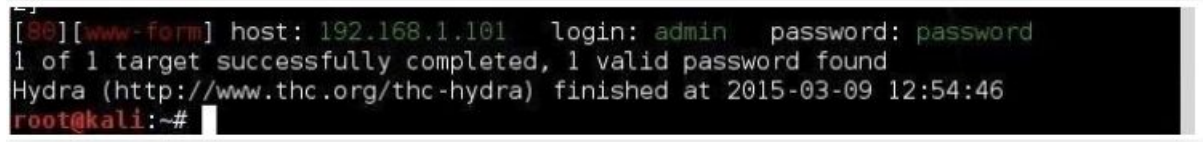
kali >hydra -L <wordlist> -P<password list>
192.168.1.101 http-post-form
"/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:Login failed"

Syntax:
> hydra -l admin -P /usr/share/dirb/wordlists/small.txt 192.168.1.101 http-post-form "/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:Login failed" -V

- **-l** indicates a single username (use **-L** for a username list)
- **-P** indicates use the following password list
- **http-post-form** indicates the type of form
- **/dvwa/login-php** is the login page URL
- **username** is the form field where the username is entered
- **^USER^** tells Hydra to use the username or list in the field
- **password** is the form field where the password is entered (it may be passwd, pass, etc.)
- **^PASS^** tells Hydra to use the password list supplied
- **Login** indicates to Hydra the login failed message
- **Login failed** is the login failure message that the form returned
- **-V** is for verbose output showing every attempt

hydra -l admin -P /usr/share/john/password.lst 10.100.100.11 http-post-form "/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:incorrect:H=Cookies: PHPSSID:alskdjffjdklsl234432; security=high;"





PES UNIVERSITY

PESU ISFCR

PESU Center for Information Security, Forensics and Cyber Resilience

ZERO DAY CYBERSEC

Thank You