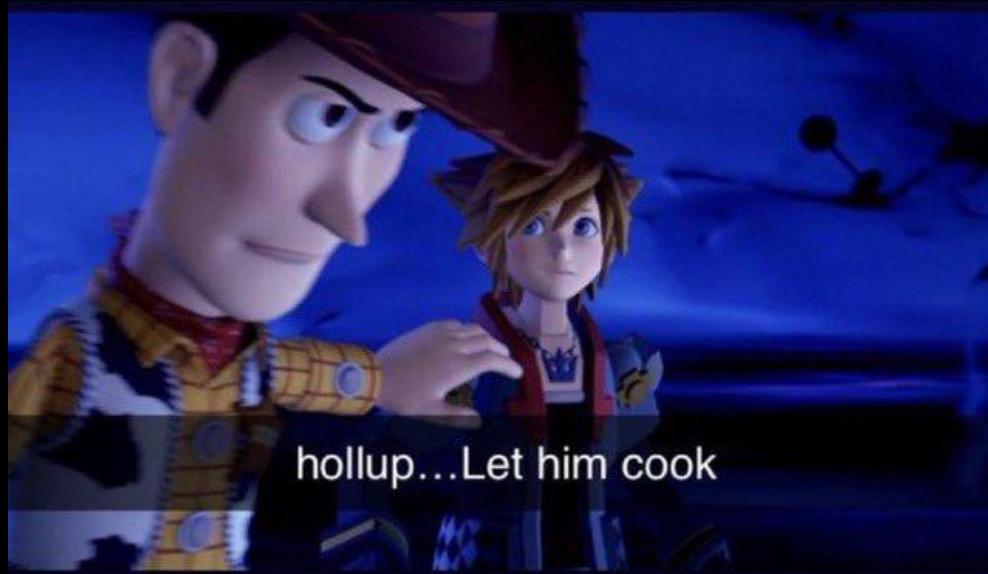


# Cooking for Hashcat, Improving Old Recipes and Exploring New Ones



**First of all...**

---

Hello  
Wisconsin!



**My name is**

Vitaliy Tally

Vitaly Taliy

Vitali Tals

Vitalii Talie Tale

Vitale

Tailiy

Vitality

Kelly

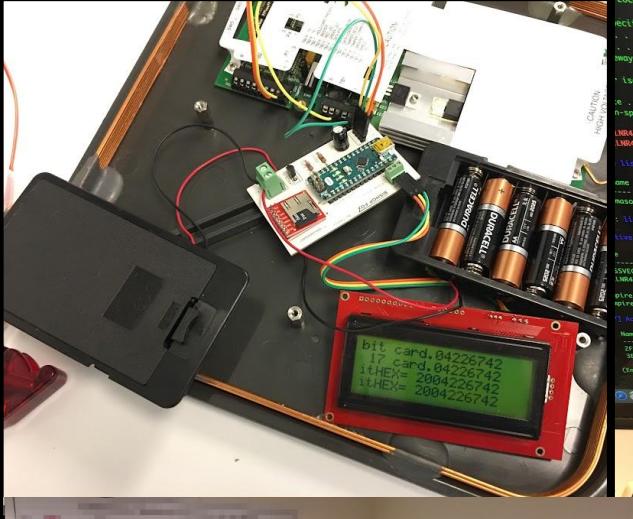


# What do I do professionally

---

- Director of Adversary Ops (I still execute projects) at SEVN-X
- I help figure out ways to crack things open when someone hits a wall
  - e.g., fancy EDR blocking SAM/LSA/LSASS access, ask Taliy
  - e.g., running kali without admin rights, ask Taliy.
- To be clear, I find the dumbest methods that work, just goal oriented mindset required
- 11 years of infosec
  - ~7 years of offsec consulting (started here)
  - ~4 years of defensive shenanigans (took a consulting break) with SIEM, EDRs, etc.
- Use GitHub as a search engine & surprisingly don't hate Excel

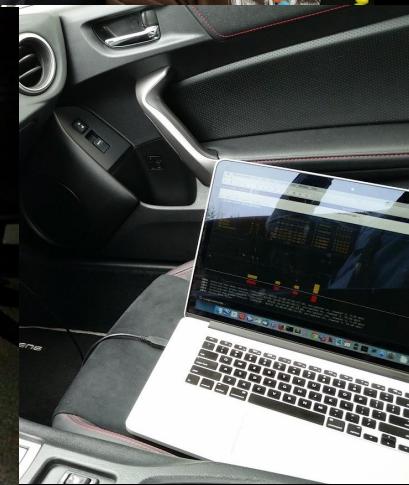
# Visual representation over the years



```
Local Area Connection:  
inetifcic DNS Suffix:   
azupwn@LASALLE-PC1:~/jackit  
[+] Scanning every 5s CTRL-C when ready.  
Key Address Channels Count Seen Type Packet  
1 EC:CF:79:05:07 65 5 0:00:06 ago Logitech HID 00:C2:00:00:FD:7F:FF:00:00:03  
[+] Select target keys: (-+) separated by commas, or 'all': [all]: i  
Ping success on channel 65  
[+] Pending attack to EC:CF:79:05:07 [Logitech HID] on channel 65  
Listeners:  
one masonic https://10.0.0.10:443  
listeners > agents  
Active agents:  
Name Internal IP  
ZFSSEGREGATIONBOX 172.16.158.134  
172.16.158.128  
SELMAAZAMITY12AA  
Powerline 172.16.158.134  
SELMAAZAMITY12AA -> agents  
[Impire: agents > ]  
[+] Active agents:
```

A screenshot of a terminal window showing network monitoring and attack logs. It includes a list of active agents with their internal IP addresses and some configuration details.

```
C:\Users\cas\printadmin>ipconfig  
Windows IP Configuration  
  
Ethernet adapter Local Area Connection:  
Connection-specific DNS Suffix: 129.175.76.62  
IPv4 Address . . . . . 129.175.76.62  
Subnet Mask . . . . . 255.255.255.0  
Default Gateway . . . . . 129.175.76.1  
  
Tunnel adapter istcp...CP3BHD32-00B0-40E8-9741-6197DPE1790:  
Media State . . . . . Media disconnected  
Connection-specific DNS Suffix: .  
Tunnel 1 Adapter Local Area Connection 9:  
Media State . . . . . Media disconnected  
Connection-specific DNS Suffix: .  
Tunnel 0 Adapter G104 Adapter:  
IPv4 Address . . . . . 2002:00af:1e63::100f:1e3e  
Subnet Mask . . . . . ::ffff:ffff  
Default Gateway . . . . . ::  
C:\Users\cas\printadmin>cd desktop  
C:\Users\cas\printadmin>cd desktop>ren save M1.exe ren save  
C:\Users\cas\printadmin>cd desktop>ren save M1.exe ren system.exe  
The operation completed successfully.  
C:\Users\cas\printadmin>cd desktop>ren save M1.exe ren security.exe  
The operation completed successfully.  
C:\Users\cas\printadmin>cd desktop>del security.exe  
C:\Users\cas\printadmin>cd desktop>del security.exe  
C:\Users\cas\printadmin>cd desktop>..  
C:\Users\cas\printadmin>
```

A screenshot of a terminal window showing network configuration and command-line history. It includes several commands related to file renaming and deletion.

# My alphabet soup

---

- A+ (certified for family helpdesk support)
- Sec+ (look ma, I'm doing it)
- Net+ (Please Do Not Throw Sausage Pizza Away)
- SEE EH embarrassed to list this one, government likes it (lapsed)
- OSCP (PCI & HR compliant pen tester)
- OSWP (Wi-Fi shenanigans is how I got roped in like mid 2000s)
- OSCE (remember to attach debugger before starting the process)
- CISSP (cert that has data diddling as a term)
- CCSP (I know about securing buckets)

ComptIA®



OFFENSIVE®  
security

(ISC)<sup>2</sup>®

# A Note on wardriving, warwalking, warflying, etc.

So, what drove me dig deeper into password cracking outside of password getting more complex?

- Cracking PSKs from wardriving
- Verizon APs rabbit hole



Name	Date modified	Type	Size
4_4d.txt	11/19/2022 12:54 PM	Text Document	616,692 KB
4d_4.txt	11/19/2022 12:51 PM	Text Document	616,692 KB
6-3d-.txt	8/7/2023 1:20 AM	Text Document	558,538 KB
6d_3_.txt	8/6/2023 1:07 AM	Text Document	558,538 KB
3_5d.txt	11/19/2022 12:55 PM	Text Document	294,454 KB
3d_5.txt	11/19/2022 12:52 PM	Text Document	294,454 KB
5_3d.txt	11/19/2022 12:55 PM	Text Document	294,454 KB
5d_3.txt	11/19/2022 12:52 PM	Text Document	294,454 KB
3_4d.txt	11/19/2022 12:54 PM	Text Document	145,145 KB
3d_4.txt	11/19/2022 12:49 PM	Text Document	145,145 KB
4_3d.txt	11/19/2022 12:54 PM	Text Document	145,145 KB
4d_3.txt	11/19/2022 12:51 PM	Text Document	145,145 KB
_6_3.txt	8/6/2023 3:48 PM	Text Document	47,875 KB
6_3.txt	8/6/2023 3:46 PM	Text Document	43,886 KB
3_3d.txt	11/19/2022 12:55 PM	Text Document	33,880 KB
3d_3.txt	11/19/2022 12:52 PM	Text Document	33,880 KB
6d.txt	8/6/2023 1:05 AM	Text Document	610 KB
5d.txt	8/6/2023 2:00 AM	Text Document	334 KB
-4d-.txt	8/6/2023 2:03 AM	Text Document	202 KB
-4d.txt	8/6/2023 2:04 AM	Text Document	180 KB
4d.txt	8/6/2023 2:03 AM	Text Document	157 KB
7char.txt	11/19/2022 12:21 AM	Text Document	81 KB
6char-.txt	8/7/2023 1:20 AM	Text Document	61 KB
6char.txt	11/19/2022 12:20 AM	Text Document	55 KB
-3d-.txt	8/6/2023 2:18 AM	Text Document	47 KB
3d-.txt	8/6/2023 2:15 AM	Text Document	41 KB
-3d.txt	8/6/2023 2:02 AM	Text Document	41 KB
3d.txt	8/6/2023 2:01 AM	Text Document	35 KB
5char.txt	11/18/2022 11:48 PM	Text Document	30 KB
4char.txt	8/6/2023 1:03 AM	Text Document	16 KB
-4char-.txt	8/6/2023 1:03 AM	Text Document	16 KB
-3char-.txt	8/6/2023 1:04 AM	Text Document	5 KB
-3char.txt	8/6/2023 1:03 AM	Text Document	4 KB
3char.txt	11/18/2022 11:47 PM	Text Document	3 KB
0-9-.txt	8/6/2023 2:14 AM	Text Document	1 KB
0-9.txt	8/6/2023 12:54 AM	Text Document	1 KB

# What am I here for?

---

- To share learned secrets of improving password cracking success
  - An intro to a little bit of everything in password cracking scene
    - The hardware kitchen setup
    - The cooking utensils
    - Amazing “ingredients” out there
    - Put it all together for feast for our hashcat food critic
    - Feed hashcat the good stuff so you can get those tasty Michelin Star passwords
- Try my best not to screw up
- Few minutes in the end is for password trivia time



# Are you ready?

- Kitchen = hardware
- Cooking utensils = OS/utilities
- Ingredients = rules/wordlists
- Hashcat = food critic
- Michelin Star = cracked password



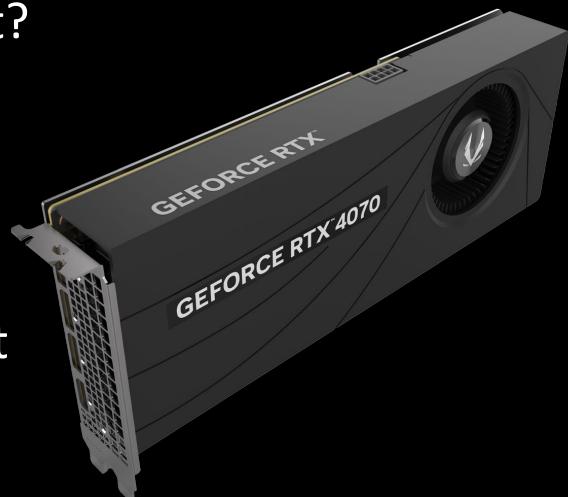
# Earning Michelin Stars

---

- NetNTLMv1 aka -m 5600 (responder, mitm6, ntlmrelayx)
- NetNTLMv1 aka -m 5500 (older systems/lucky with WPA-ENTERPRISE config)
- Kerberoast goodies -m 13100
- Hard to crack kerberoast goodies -m 19700 or -m 19600 (been seeing more of these)
- Domain caches credentials -m 1100 if you're lucky but most likely -m 2100
- IPMI dumped hashes -m 7300 or -m 7350 (neat way to access non-AD resources in our experience)
- Occasional Microsoft DOC or XLS -m 9600 (there's a couple variants depending on the version, usually to get contained creds within)
- NT hashes and LM hashes -m 1000 and -m 3000 (NTDS audits, SAM dumps lsass dumps) only when PTH isn't working like default SID 500
- Sweet Wi-Fi (Pre-Shared Keys) PSKs -m 22000

# Expensive kitchen setup...

- You either have Nvidia stock money or you're running what looks like a crypto mining rig
  - First it was crypto craze, then it was supply chains, now it's ML/AI craze
  - Still rely on consumer rather than pro GPUs (RTX)
  - There's part 1 of a blog I wrote about building out a cracker
  - Blower style consumer cards still exist?
- Water cooling is NOT cheap
- Neither is a server chassis build
- RIP GTX 1080Ti era
- HDDs and SSDs/M.2s
- Summertime curse and wintertime delight



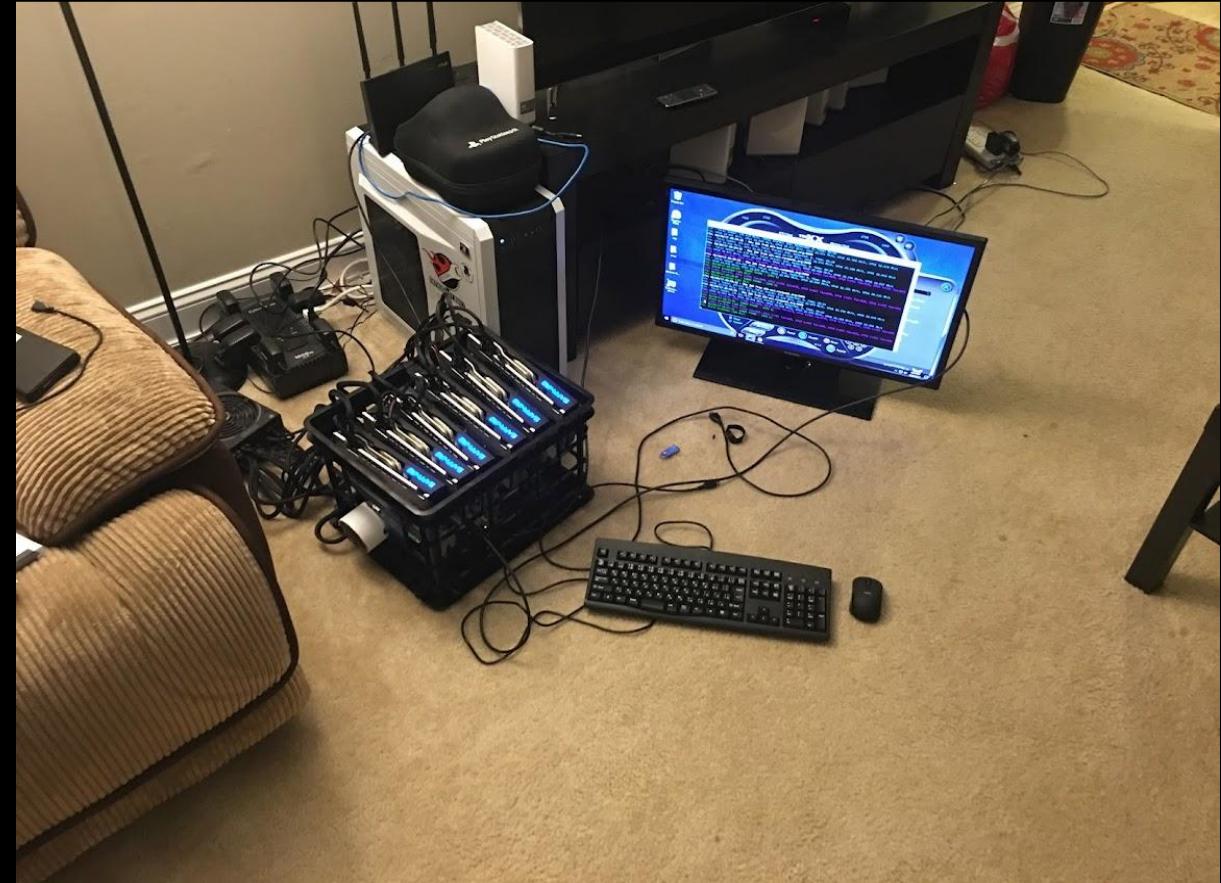
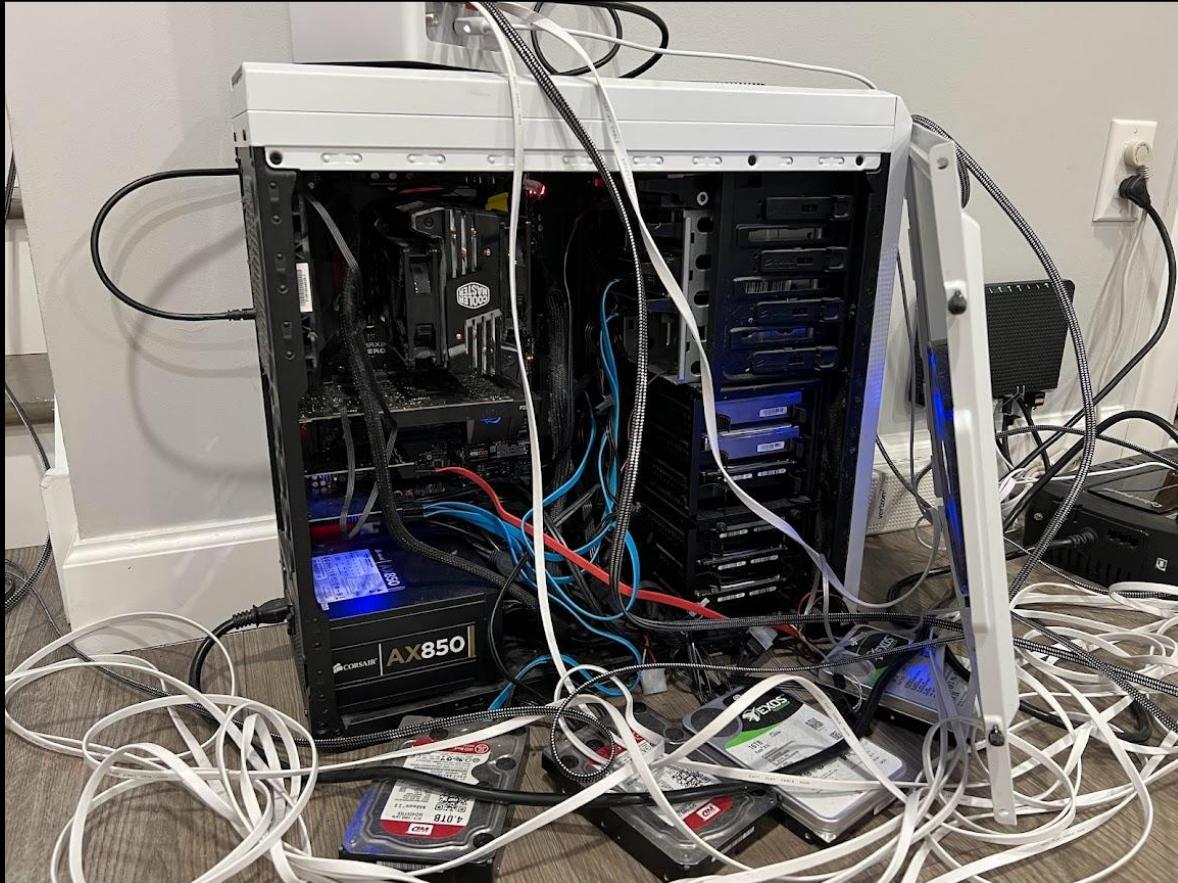
# Kitchen tours

Left: Hashburglar, right: my space heater, both quality craftsmanship

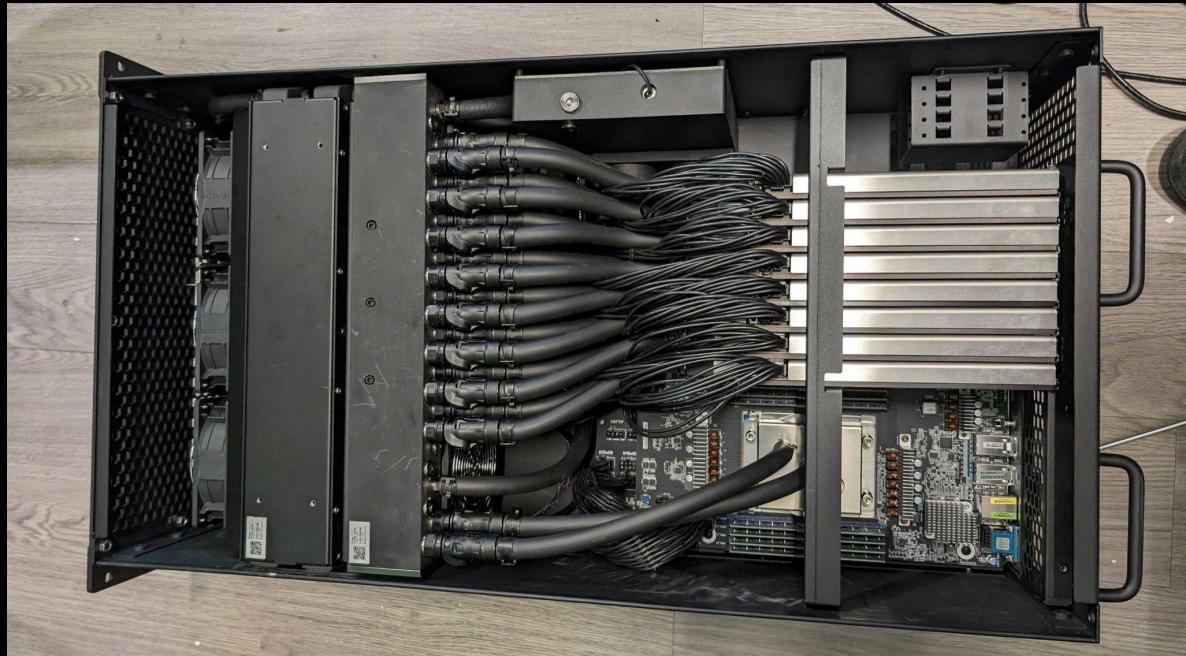


# It's not always about aesthetics, if it works

Left: my Plex server (born 2016), right:~2017 eth mining rig, complete with milk crate/paper towel tube



# Twitter recon



# Cooking utensils

- OS can be Windows, it's FINE, Linux is free, and Windows can be too
  - I'm not a purist
  - Overclocking utilities, hashcat utilities, WSL, etc.

```
C:\Users\dude\Desktop\hashcat>hashcat --version
v6.2.6-850-gfafb277e0

C:\Users\dude\Desktop\hashcat>hashcat.exe -I
hashcat (v6.2.6-850-gfafb277e0) starting in backend information mode

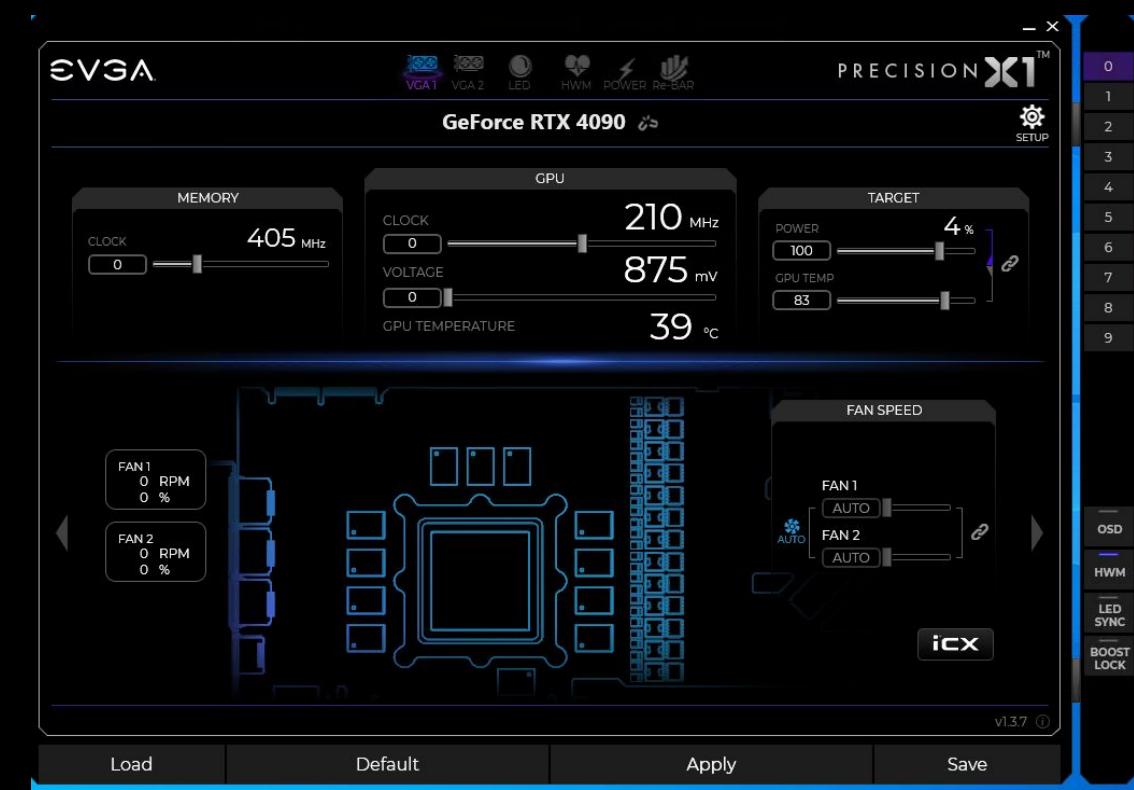
The device #3 specifically listed was skipped because it is an alias of device #1

The device #4 specifically listed was skipped because it is an alias of device #2

CUDA Info:
=====
CUDA.Version..: 12.4

Backend Device ID #1 (Alias: #3)
Name.....: NVIDIA GeForce RTX 4090
Processor(s)...: 128
Clock.....: 2520
Memory.Total...: 24563 MB
Memory.Free....: 22952 MB
Local.Memory...: 99 KB
PCI.Addr.BDFe...: 0000:01:00.0

Backend Device ID #2 (Alias: #4)
Name.....: NVIDIA GeForce RTX 4090
Processor(s)...: 128
Clock.....: 2520
Memory.Total...: 24563 MB
Memory.Free....: 22952 MB
Local.Memory...: 99 KB
PCI.Addr.BDFe...: 0000:08:00.0
```



# Discover new cooking utensils

<https://github.com/NorthwaveSecurity/wikiraider>

```
[kali㉿Windows10] - [~/Desktop/wikiraider]
$ python3 wikiraider.py list
```



Version 1.0.0 - Copyright (c) 2020 Northwave B.V. ([www.northwave-security.com](http://www.northwave-security.com))

```
[INFO] Listing all Wikipedia dumps ...
[SUCCESS] Name: en, URL: https://dumps.wikimedia.org/enwiki/20240501
[SUCCESS] Name: zh, URL: https://dumps.wikimedia.org/zhwiki/20240501
[SUCCESS] Name: ru, URL: https://dumps.wikimedia.org/rwikipedia/20240501
[SUCCESS] Name: vi, URL: https://dumps.wikimedia.org/viwiki/20240501
[SUCCESS] Name: uk, URL: https://dumps.wikimedia.org/ukwiki/20240501
[SUCCESS] Name: sv, URL: https://dumps.wikimedia.org/svwiki/20240501
[SUCCESS] Name: pt, URL: https://dumps.wikimedia.org/ptwiki/20240501
[SUCCESS] Name: pl, URL: https://dumps.wikimedia.org/plwiki/20240501
[SUCCESS] Name: nl, URL: https://dumps.wikimedia.org/nlwiki/20240501
[SUCCESS] Name: ja, URL: https://dumps.wikimedia.org/jawiki/20240501
[SUCCESS] Name: it, URL: https://dumps.wikimedia.org/itwiki/20240501
[SUCCESS] Name: meta, URL: https://dumps.wikimedia.org/metawiki/20240501
[SUCCESS] Name: ko, URL: https://dumps.wikimedia.org/kowiki/20240501
[SUCCESS] Name: he, URL: https://dumps.wikimedia.org/hewiki/20240501
[SUCCESS] Name: hu, URL: https://dumps.wikimedia.org/huwiki/20240501
[SUCCESS] Name: fr, URL: https://dumps.wikimedia.org/frwiki/20240501
[SUCCESS] Name: de, URL: https://dumps.wikimedia.org/dewiki/20240501
[SUCCESS] Name: es, URL: https://dumps.wikimedia.org/eswiki/20240501
[SUCCESS] Name: ceb, URL: https://dumps.wikimedia.org/cebwiki/20240501
[SUCCESS] Name: fa, URL: https://dumps.wikimedia.org/fawiki/20240501
[SUCCESS] Name: sr, URL: https://dumps.wikimedia.org/srwiki/20240501
[SUCCESS] Name: ca, URL: https://dumps.wikimedia.org/cawiki/20240501
[SUCCESS] Name: tr, URL: https://dumps.wikimedia.org/trwiki/20240501
[SUCCESS] Name: fi, URL: https://dumps.wikimedia.org/fiwiki/20240501
[SUCCESS] Name: id, URL: https://dumps.wikimedia.org/idwiki/20240501
[SUCCESS] Name: cs, URL: https://dumps.wikimedia.org/csmediawiki/20240501
[SUCCESS] Name: ar, URL: https://dumps.wikimedia.org/arwiki/20240501
```

```
[SUCCESS] Identified XML file enwiki-20240501-pages-articles23.xml-p49288942p50564553.bz2
[INFO] Investigating file enwiki-20240501-pages-articles24.xml-p50564554p52064553.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles24.xml-p50564554p52064553.bz2
[INFO] Investigating file enwiki-20240501-pages-articles24.xml-p52064554p53564553.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles24.xml-p52064554p53564553.bz2
[INFO] Investigating file enwiki-20240501-pages-articles24.xml-p53564554p55064553.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles24.xml-p53564554p55064553.bz2
[INFO] Investigating file enwiki-20240501-pages-articles24.xml-p55064554p56564553.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles24.xml-p55064554p56564553.bz2
[INFO] Investigating file enwiki-20240501-pages-articles24.xml-p56564554p57025655.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles24.xml-p56564554p57025655.bz2
[INFO] Investigating file enwiki-20240501-pages-articles25.xml-p57025656p58525655.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles25.xml-p57025656p58525655.bz2
[INFO] Investigating file enwiki-20240501-pages-articles25.xml-p58525656p60025655.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles25.xml-p58525656p60025655.bz2
[INFO] Investigating file enwiki-20240501-pages-articles25.xml-p60025656p61525655.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles25.xml-p60025656p61525655.bz2
[INFO] Investigating file enwiki-20240501-pages-articles25.xml-p61525656p62585850.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles25.xml-p61525656p62585850.bz2
[INFO] Investigating file enwiki-20240501-pages-articles26.xml-p62585851p63975909.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles26.xml-p62585851p63975909.bz2
[INFO] Investigating file enwiki-20240501-pages-articles27.xml-p63975910p65475909.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles27.xml-p63975910p65475909.bz2
[INFO] Investigating file enwiki-20240501-pages-articles27.xml-p65475910p66975909.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles27.xml-p65475910p66975909.bz2
[INFO] Investigating file enwiki-20240501-pages-articles27.xml-p66975910p68475909.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles27.xml-p66975910p68475909.bz2
[INFO] Investigating file enwiki-20240501-pages-articles27.xml-p68475910p69975909.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles27.xml-p68475910p69975909.bz2
[INFO] Investigating file enwiki-20240501-pages-articles27.xml-p69975910p71475909.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles27.xml-p69975910p71475909.bz2
[INFO] Investigating file enwiki-20240501-pages-articles27.xml-p71475910p72975909.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles27.xml-p71475910p72975909.bz2
[INFO] Investigating file enwiki-20240501-pages-articles27.xml-p72975910p74475909.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles27.xml-p72975910p74475909.bz2
[INFO] Investigating file enwiki-20240501-pages-articles27.xml-p74475910p75975909.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles27.xml-p74475910p75975909.bz2
[INFO] Investigating file enwiki-20240501-pages-articles27.xml-p75975910p76788691.bz2
[SUCCESS] Identified XML file enwiki-20240501-pages-articles27.xml-p75975910p76788691.bz2
[INFO] Downloading https://dumps.wikimedia.org/enwiki/20240501/enwiki-20240501-pages-articles1.xml-p1p41242.bz2.
| 61505536/278809385 [00:14<00:52, 4169206.55it/s]
```

# Discover new cooking utensils

<https://hashcat.net/wiki/doku.php?id=maskprocessor>

- Generate your own wordlists or password lists with ease

[https://hashcat.net/wiki/doku.php?id=hashcat\\_utils](https://hashcat.net/wiki/doku.php?id=hashcat_utils)

- cutb is one of my favorites but here's plenty of additional goodies

hashcat / maskprocessor Public

Code Issues 15 Pull requests Actions Wiki Security Insights

master 1 Branch 1 Tags Q Go to file Code

**About**

High-Performance word generator with a per-position configurable charset

Readme MIT license Activity

Custom properties 420 stars 39 watching 110 forks

Report repository

**Releases** 1

maskprocessor v0.73 Latest on Feb 9, 2015

**Packages**

No packages published

**Contributors** 7

**Languages**

C 95.1% Makefile 4.9%

hashcat advanced password recovery

hashcat Forums Wiki Tools Events

Recent changes Admin Log In Sitemap

**Table of Contents**

- hashcat-utils
- Description
- Current Version
- Download
- List of Utilities
  - capthcrack
  - clearchrcl
  - combinator
  - combinator3
  - combinatorX
  - combipow
  - cutb
  - ct3\_to\_ntlm
  - cutb
  - daskey\_to\_ntlm
  - expandcr
  - export\_potfile
  - gate
  - generate-rules
  - hcstatgen
  - hcstatgen
  - keyspace
  - lsm
  - ml2
  - morph
  - permute
  - permute\_exist
  - prepare
  - remaining
  - req
  - req-exclude
  - req-include
  - rrl
  - rrl
  - rules\_optimize
  - seprule
  - splitlen
  - strip-bsn
  - strip-cr
  - tmesis
  - tmesis-dynamic
  - topmorph
  - Limitations

**hashcat-utils**

**Description**

Hashcat-utils are a set of small utilities that are useful in advanced password cracking.

They are packed into multiple stand-alone binaries.

All of these utils are designed to execute only one specific function.

Since they all work with STDIN and STDOUT you can group them into **chains**.

hashcat-utils is released as open source software under the MIT license.

**Current Version**

The current version is 1.9.

**Download**

The programs are available for Linux and Windows on both 32-bit and 64-bit architectures, as well as .app binaries for 64-bit OSX/macOS. The project is released as MIT-licensed open source software.

hashcat-utils does not have a dedicated homepage, but this download link always has the latest release:

- hashcat-utils

**List of Utilities**

Each of them is described in detail in the following sections.

# Discover new cooking utensils

<https://github.com/iphelix/pack>

- Generate rules/masks and stats

<https://github.com/hops/pack2>

- Generate rules/masks and stats v2

<https://github.com/digininja/pipal>

- Password analysis

<https://github.com/digininja/CeWL>

- Target site wordlists generator

```
(kali㉿Windows10)-[~]
└─$ cewl --help
CeWL 6.1 (Max Length) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
Usage: cewl [OPTIONS] ... <url>

OPTIONS:
-h, --help: Show help.
-k, --keep: Keep the downloaded file.
-d <x>, --depth <x>: Depth to spider to, default 2.
-m, --min_word_length: Minimum word length, default 3.
-x, --max_word_length: Maximum word length, default unset.
-o, --offsite: Let the spider visit other sites.
--exclude: A file containing a list of paths to exclude.
--allowed: A regex pattern that path must match to be followed.
-w, --write: Write the output to the file.
-u, --ua <agent>: User agent to send.
-n, --no-words: Don't output the wordlist.
-g <x>, --groups <x>: Return groups of words as well.
--lowercase: Lowercase all parsed words.
--with-numbers: Accept words with numbers in as well as just letters.
--convert-umlauts: Convert common ISO-8859-1 (Latin-1) umlauts (ä-ae, ö-oe, ü-ue, ß-ss).
-a, --meta: Include meta data.
--meta_file <file>: Output file for meta data.
-e, --email: Include email addresses.
--email_file <file>: Output file for email addresses.
--meta-temp-dir <dir>: The temporary directory used by exiftool when parsing files, default /tmp.
-c, --count: Show the count for each word found.
-v, --verbose: Verbose.
--debug: Extra debug information.

Authentication
--auth_type: Digest or basic.
--auth_user: Authentication username.
--auth_pass: Authentication password.

Proxy Support
--proxy_host: Proxy host.
--proxy_port: Proxy port, default 8080.
--proxy_username: Username for proxy, if required.
--proxy_password: Password for proxy, if required.
```

```
(kali㉿Windows10)-[~]
└─$ pipal --help
pipal 3.4.0 Robin Wood (robin@digi.ninja) (http://digi.ninja)

Usage: pipal [OPTION] ... FILENAME
      --help, -h, -?: show help
      --top, -t X: show the top X results (default 10)
      --output, -o <filename>: output to file
      --gkey <Google Maps API key>: to allow zip code lookups (optional)
      --list-checkers: Show the available checkers and which are enabled
      --verbose, -v: Verbose

      FILENAME: The file to count
```

```
(kali㉿Windows10)-[~]
└─$
```

```
(kali㉿Windows10)-[~]
└─$ pack2
pack2 0.1.0

USAGE:
      pack2 <SUBCOMMAND>

FLAGS:
      -h, --help      Prints help information
      -V, --version   Prints version information

SUBCOMMANDS:
      cgrams      Splits each line on the charset boundary
      filtermask   Filters a wordlist by a given mask
      help         Prints this message or the help of the given subcommand(s)
      statsgen    Generates statistics from a [input] and writes masks to <output> stats are written to stderr
      unhex       Decodes $HEX[] encoded lines

(kali㉿Windows10)-[~]
└─$
```

```
(kali㉿Windows10)-[~]
└─$
```

# Basic ingredients

- Bundled rules
- Well known wordlists

Name	Date modified	Type
crackstation	1/8/2024 11:14 AM	File folder
hashes.org	1/8/2024 11:14 AM	File folder
hashkiller	1/8/2024 11:14 AM	File folder
rockyou	1/8/2024 11:14 AM	File folder

Name	Date modified	Type	Size
rockyou.txt	12/1/2021 8:21 PM	Text Document	136,643 KB
realhuman_phill.txt	9/5/2010 6:58 PM	Text Document	699,650 KB
realuniq.lst	2/17/2013 5:05 PM	LST File	15,328,241 KB
hashes.org-2019.txt	12/29/2019 9:02 AM	Text Document	5,383,738 KB
hashes.org-2020.txt	12/27/2020 12:04 AM	Text Document	1,948,094 KB
hashes.org-2021.txt	11/19/2022 9:28 PM	Text Document	44,773 KB
hashkiller-dict.txt	11/19/2022 6:06 PM	Text Document	2,892,549 KB

▀ dive.rule	4/17/2024 7:47 PM	RULE File	790 KB
▀ generated2.rule	4/17/2024 7:47 PM	RULE File	410 KB
▀ Incisive-leetspeak.rule	4/17/2024 7:47 PM	RULE File	383 KB
▀ rockyou-30000.rule	4/17/2024 7:47 PM	RULE File	295 KB
▀ d3adOne.rule	4/17/2024 7:47 PM	RULE File	235 KB
▀ TOXICv2.rule	4/17/2024 7:47 PM	RULE File	225 KB
▀ TOXIC_insert_HTML_entities_0_Z.rule	4/17/2024 7:47 PM	RULE File	165 KB
▀ generated.rule	4/17/2024 7:47 PM	RULE File	86 KB
▀ TOXIC-insert_00-99_1950-2050_toprules_0..	4/17/2024 7:47 PM	RULE File	76 KB
▀ unix-ninja-leetspeak.rule	4/17/2024 7:47 PM	RULE File	55 KB
▀ toggles5.rule	4/17/2024 7:47 PM	RULE File	48 KB
▀ InsidePro-HashManager.rule	4/17/2024 7:47 PM	RULE File	35 KB
▀ TOXIC.rule	4/17/2024 7:47 PM	RULE File	34 KB
▀ TOXIC-insert_top_100_passwords_1_G.rule	4/17/2024 7:47 PM	RULE File	34 KB
▀ InsidePro-PasswordsPro.rule	4/17/2024 7:47 PM	RULE File	21 KB
▀ toggles4.rule	4/17/2024 7:47 PM	RULE File	16 KB
▀ toggles3.rule	4/17/2024 7:47 PM	RULE File	4 KB
▀ TOXIC-insert_space_and_special_0_F.rule	4/17/2024 7:47 PM	RULE File	3 KB
▀ specific.rule	4/17/2024 7:47 PM	RULE File	2 KB
▀ TOXIC_3_rule.rule	4/17/2024 7:47 PM	RULE File	2 KB
▀ oscommerce.rule	4/17/2024 7:47 PM	RULE File	2 KB
▀ best66.rule	4/17/2024 7:47 PM	RULE File	1 KB
▀ combinator.rule	4/17/2024 7:47 PM	RULE File	1 KB
▀ toggles2.rule	4/17/2024 7:47 PM	RULE File	1 KB
▀ stacking58.rule	4/17/2024 7:47 PM	RULE File	1 KB
▀ leetspeak.rule	4/17/2024 7:47 PM	RULE File	1 KB
▀ top10_2023.rule	4/17/2024 7:47 PM	RULE File	1 KB
▀ toggles1.rule	4/17/2024 7:47 PM	RULE File	1 KB
▀ hybrid	4/17/2024 7:47 PM	File folder	

# Finding new ingredients to spruce things up

<https://hashmob.net/resources/hashmob>

The screenshot shows the HashMob website interface. The left sidebar contains navigation links such as 'Search Hashes', 'Submit Founds', 'Hash Lists' (with sub-links for 'User lists', 'Official lists', 'Premium lists', and 'Combined Official Left Lists'), 'Wordlists, Rules & More' (selected), 'HashMob Wordlists' (highlighted in blue), 'Other Wordlists', 'Rules', 'Writeups', 'Tools' (with sub-links for 'Verifier', 'Password Strength', and 'Programs'), and 'API Information'. The main content area has a header 'Information' and a detailed description of the wordlists. Below this is a section titled 'HashMob Official Resources' with a table listing various wordlists. The table columns are: Wordlist Name, Release Date, Size, Unzipped Size, Type, and Download link. The listed wordlists are:

Wordlist Name	Release Date	Size	Unzipped Size	Type	Download
HashMob Combined Full	2024-06-02	4.95 GB	15.82 GB	Wordlist	<a href="#">Download</a>
HashMob Official Full	2024-06-02	4.84 GB	15.56 GB	Wordlist	<a href="#">Download</a>
HashMob Huge Combined	2024-06-02	1.73 GB	5.31 GB	Wordlist	<a href="#">Download</a>
HashMob Larger Combined	2024-06-02	412.72 MB	1.51 GB	Wordlist	<a href="#">Download</a>
HashMob User Full	2024-06-02	219.82 MB	717.97 MB	Wordlist	<a href="#">Download</a>
HashMob Large Combined	2024-06-02	134.69 MB	476.75 MB	Wordlist	<a href="#">Download</a>
HashMob Medium Combined	2024-06-02	35.06 MB	114.82 MB	Wordlist	<a href="#">Download</a>
HashMob Small Combined	2024-06-02	6.09 MB	17.44 MB	Wordlist	<a href="#">Download</a>

# Finding new ingredients to spruce things up

<https://weakpass.com/wordlist>

The screenshot shows the WEAKPASS website interface. At the top, there is a navigation bar with links for Home, Download, Lists (with a dropdown menu), All-in-One, Hash cracker, Generators (with a dropdown menu), and social media icons for GitHub, LinkedIn, Twitter, and a coffee cup icon labeled "Buy me a coffee".

The main content area displays a grid of wordlist entries:

- weakpass\_3a** (ID 100): 104.8 GB, 16 GB, 8799064934. MD5: 13 min, NTLM: 9 min, NetNTLMv2: 10 min. md5crypt: 7 H, sha512crypt: 79 H, WPA2: 8 H. New version of weakpass\_2a.
- weakpass\_3** (ID 99): 28.29 GB, 6.03 GB, 2579783050. MD5: 4 min, NTLM: 3 min, NetNTLMv2: 3 min. md5crypt: 2 H, sha512crypt: 23 H, WPA2: 2 H. Updated Weakpass wordlist for general purpose.
- cyclone.hashesorg.hashkiller.combined.txt** (ID 97): 15.02 GB, 6.53 GB, 1469156499. MD5: 2 min, NTLM: 93 sec, NetNTLMv2: 100 sec. md5crypt: 1 H, sha512crypt: 13 H, WPA2: 1 H.
- dicassassin** (ID 96): 262.17 GB, 32.91 GB, 23109038633. MD5: 34 min, NTLM: 25 min, NetNTLMv2: 26 min. md5crypt: 18 H, sha512crypt: 208 H, WPA2: 22 H.
- Hashes.org** (ID 95): 13.99 GB, 3.44 GB, 1397237946. MD5: 2 min, NTLM: 89 sec, NetNTLMv2: 95 sec. md5crypt: 1 H, sha512crypt: 13 H, WPA2: 1 H. Hashes.org full archive.
- hashesorg2019** (ID 95): 12.79 GB, 4.16 GB, 1279729139. MD5: 114 sec, NTLM: 81 sec, NetNTLMv2: 87 sec. md5crypt: 59 min, sha512crypt: 12 H, WPA2: 1 H.
- weakpass\_2a** (ID 94): 104.8 GB, 16 GB, 8799064934. MD5: 13 min, NTLM: 9 min, NetNTLMv2: 10 min. md5crypt: 7 H, sha512crypt: 79 H, WPA2: 8 H.
- xsukax-Wordlist-All.txt** (ID 94): 104.8 GB, 16 GB, 8799064934. MD5: 13 min, NTLM: 9 min, NetNTLMv2: 10 min. md5crypt: 7 H, sha512crypt: 79 H, WPA2: 8 H.
- rockyou2021.txt** (ID 93): 104.8 GB, 16 GB, 8799064934. MD5: 13 min, NTLM: 9 min, NetNTLMv2: 10 min. md5crypt: 7 H, sha512crypt: 79 H, WPA2: 8 H.

Each entry includes a "Download" and "Torrent" button at the bottom.

# Finding new ingredients to spruce things up

<https://github.com/n0kovo/hashcat-rules-collection>

# Finding new ingredients to spruce things up

<https://github.com/initstring/passphrase-wordlist>

The screenshot shows the GitHub repository page for 'passphrase-wordlist'. The repository has 96 commits, 2 branches, and 1 tag. The README.md file contains information about the project, including a list of files like hashcat-rules, utilities, LICENSE, and README.md. The README section includes an 'Overview' with a note about passphrases, a 'To use this project, you need:' section with two bullet points, and a 'WORDLIST LAST UPDATED: November 2022' note. The 'Usage' section provides instructions for using hashcat with the wordlist.

source file name	source type	description
wiktionary-2022-11-19.txt	dynamic	Article titles scraped from Wiktionary's index dump <a href="#">here</a> .
wikipedia-2022-11-19.txt	dynamic	Article titles scraped from the Wikipedia <a href="#">pages-articles-multistream-index</a> dump generated 29-Sept-2021 <a href="#">here</a> .
urban-dictionary-2022-11-19.txt	dynamic	Urban Dictionary dataset pulled using <a href="#">this script</a> .
know-your-meme-2022-11-19.txt	dynamic	Meme titles from KnownYourMeme scraped using my tool <a href="#">here</a> .
imdb-titles-2022-11-19.txt	dynamic	IMDB dataset using the "primaryTitle" column from <a href="#">title.basics.tsv.gz</a> file available <a href="#">here</a>
global-poi-2022-11-19.txt	dynamic	<a href="#">Global POI dataset</a> using the 'allCountries' file from 29-Sept-2021.
billboard-titles-2022-11-19.txt	dynamic	Album and track names using <a href="#">Ultimate Music Database</a> , scraped with <a href="#">a fork of mwkling's tool</a> , modified to grab Billboard Singles (1940-2021) and Billboard Albums (1970-2021) charts.
billboard-artists-2022-11-19.txt	dynamic	Artist names using <a href="#">Ultimate Music Database</a> , scraped with <a href="#">a fork of mwkling's tool</a> , modified to grab Billboard Singles (1940-2021) and Billboard Albums (1970-2021) charts.
book.txt	static	Kaggle dataset with titles from over 300,000 books.
rstone-top-100.txt	static (could be dynamic in future)	Song lyrics for Rolling Stone's "top 100" artists using my <a href="#">lyric scraping tool</a> .
cornell-movie-titles-raw.txt	static	Movie titles from this <a href="#">Cornell project</a> .
cornell-movie-lines.txt	static	Movie lines from this <a href="#">Cornell project</a> .
author-quotes-raw.txt	static	<a href="#">Quotables</a> dataset on Kaggle.
1800-phrases-raw.txt	static	<a href="#">1,800 English Phrases</a> .
15k-phrases-raw.txt	static	<a href="#">15,000 Useful Phrases</a> .

# Growing your own ingredients

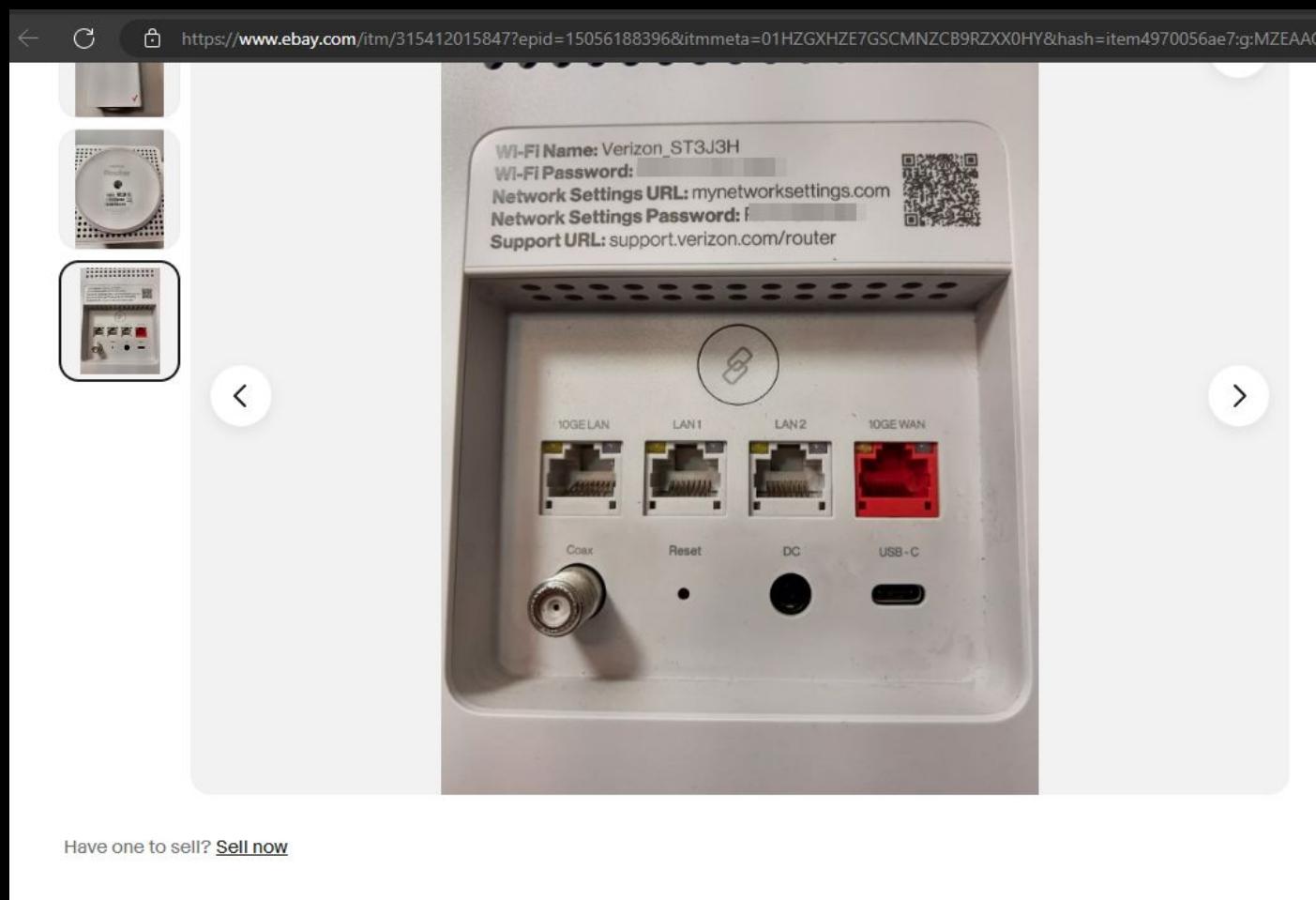
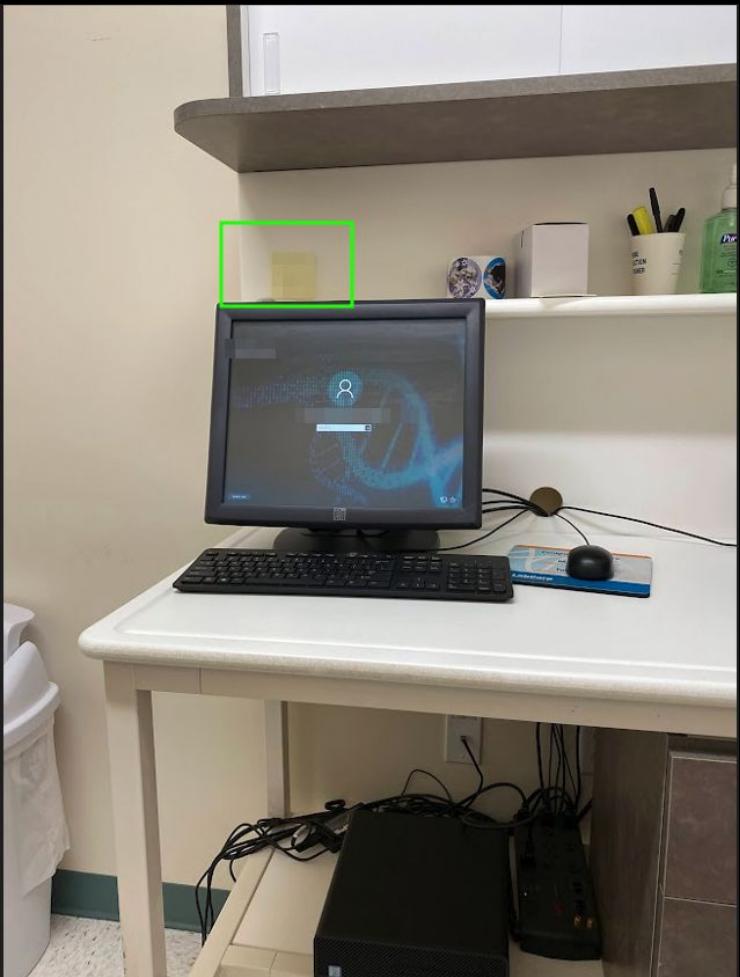
---

Repeat after me, custom rules and masks are not that hard!

- Manual small rule sets to combine in attacks
- Automated analysis tool outputs to feed into lists, etc.
- Automated mask generators



# Gathering your own ingredients



# Forbidden salad example

The forbidden salad:

Making rockyou into rocksalad with cutb.exe and WSL sort -u (original name rocksalad)

```
cutb.exe 0 0 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 1 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 2 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 3 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 4 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 5 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 6 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 7 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 8 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 9 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 10 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 11 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 12 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 13 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 14 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 15 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 16 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 17 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 18 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 19 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 20 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 21 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 22 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 23 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 24 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 25 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 26 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 27 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 28 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 29 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 30 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 31 < cracked.txt >> cracked_chop_051623.txt
cutb.exe 0 32 < cracked.txt >> cracked_chop_051623.txt
```

Example wordlist file:

```
$ cat wordlist
apple1234
theman
fastcars
```

Example positive offset and fixed length (first 4 characters):

```
$ ./cutb.bin 0 4 < wordlist
apl
them
fast
```

Example positive offset, no length (returns remaining characters in string):

```
$ ./cutb.bin 4 < wordlist
e1234
an
cars
```

Example negative offset (last 4 characters in string):

```
$ ./cutb.bin -4 < wordlist
1234
eman
cars
```

Example negative offset, fixed length:

```
$ ./cutb.bin -5 3 < wordlist
e12
hem
tca
```

Name	Date modified	Type	Size
rockyou_salad.txt	11/7/2022 3:43 AM	Text Document	1,114,469 KB
unique_salad.txt	6/4/2024 2:17 AM	Text Document	1,114,469 KB

```
ubuntu@chaoticneko: /mnt/c/Windows/system32$ sort -u /mnt/f/Hashcat/salads/rockyou_salad.txt >> /mnt/f/Hashcat/salads/unique_salad.txt
ubuntu@chaoticneko: /mnt/c/Windows/system32$
```

# Let's get creative

---

combining 2+ rules

combining salads

FOR looping recipes

become ungovernable



stdout ingredient maker

Sublime recipes

-a 1 sandwiches

-a 3 but not raw

-a 6 or -a 7

# Chef's not so secret secrets

Simple rule combinations using common patterns of digits and special characters

- 1900-2100 + toggles OR + insert character OR + replace Nth character, etc.

Client addresses

- Appending St/Rd/Blvd/ and digits

Using DNS records/hostnames and user lists as wordlists

- User as pass worked as of ~3 weeks ago, no joke

Pay attention to patterns if you have a larger volume of hashes (not always picked as pattern)

- e.g., color+ animal, last name + several digit ID,

Longer passwords include multiple words, so adjust for that

- Passphrases, sandwiches on top of sandwiches

PTH when you can like NT or relay NetNTLMs

- Don't always have to spin cycles



# Sous chef joins the chat

- ChatGPT can help with analysis and much more (a lot of AI stuff is dumb though)

The code appears to be correctly structured and should function as expected. It analyzes the input strings, identifies patterns, and outputs various statistics and analysis results to both the terminal and an output CSV file. If you encounter any errors or need further assistance, please feel free to ask!

Gets this error: Traceback (most recent call last):

```
  File "C:\Users\dude\PycharmProjects\pythonProject\venv\string_analyzer.py", line 138, in <module>
    special_char_positions = find_special_char_positions(input_strings)

  ^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\dude\PycharmProjects\pythonProject\venv\string_analyzer.py", line 96, in find_special_char_positions
    special_char_positions = {char: Counter() for char in string.punctuation}
                           ^^^^^^
```

UnboundLocalError: cannot access local variable 'string' where it is not associated with a value

It seems like there might be a naming conflict with the variable name "string", which is also the name of the Python built-in module "string". This is causing the UnboundLocalError because the interpreter is trying to access the local variable "string" before it's assigned a value.

To resolve this issue, you can rename the variable "string" in your function to something else to avoid the conflict with the built-in module na<sup>↓</sup>. Here's the updated function:

```
print("\nTop 100 most common 15-character or longer patterns:")
for pattern, count in most_common_long_patterns:
    print(f"{pattern}: found in {count} strings")

print("\nTop 100 most common 5 or more last characters of the string patterns:")
for pattern, count in most_common_last_patterns:
    print(f"{pattern}: found in {count} strings")

print("\nMost common character position for each digit 0 to 9:")
for digit, positions in digit_positions.items():
    if positions:
        most_common_position = positions.most_common(1)[0][0]
        print(f"Digit {digit}: Most common position is {most_common_position}")

print("\nMost common character position for each capital letter A to Z:")
for letter, positions in letter_positions_upper.items():
    if positions:
        most_common_position = positions.most_common(1)[0][0]
        print(f"Uppercase Letter {letter}: Most common position is {most_common_position}")

print("\nMost common character position for each lowercase letter a to z:")
for letter, positions in letter_positions_lower.items():
    if positions:
        most_common_position = positions.most_common(1)[0][0]
        print(f"Lowercase Letter {letter}: Most common position is {most_common_position}")

print("\nTop 5 most common character positions for each special character:")
```

# TLDR/TLDL

Lots of free available resources and tools provided by the awesome individuals & teams free to use

- Tons of passwords, dictionaries and rules out there to use

Even mixing up current resources with novel approaches can yield new results

- Using tools like cutb or multiple smaller rulesets in combination yields new results

Step 1: grab linked resources

Step 2: get cooking

Step 3: tell me your password cracking success story



# Trivia

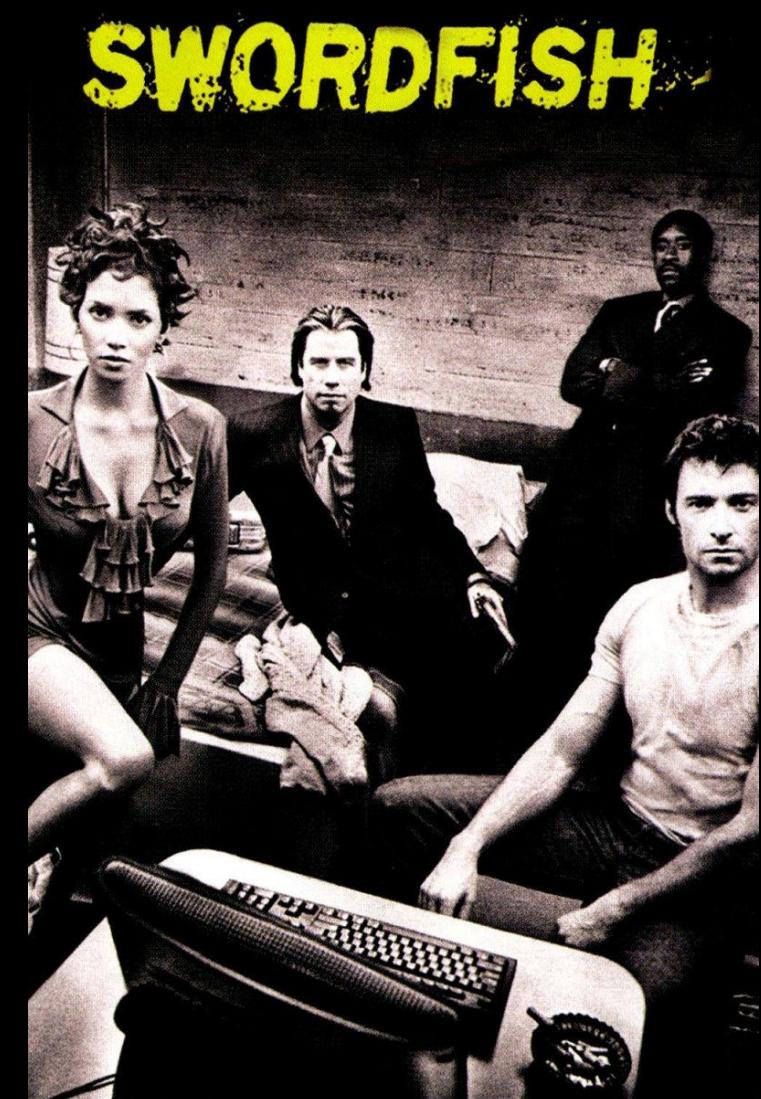
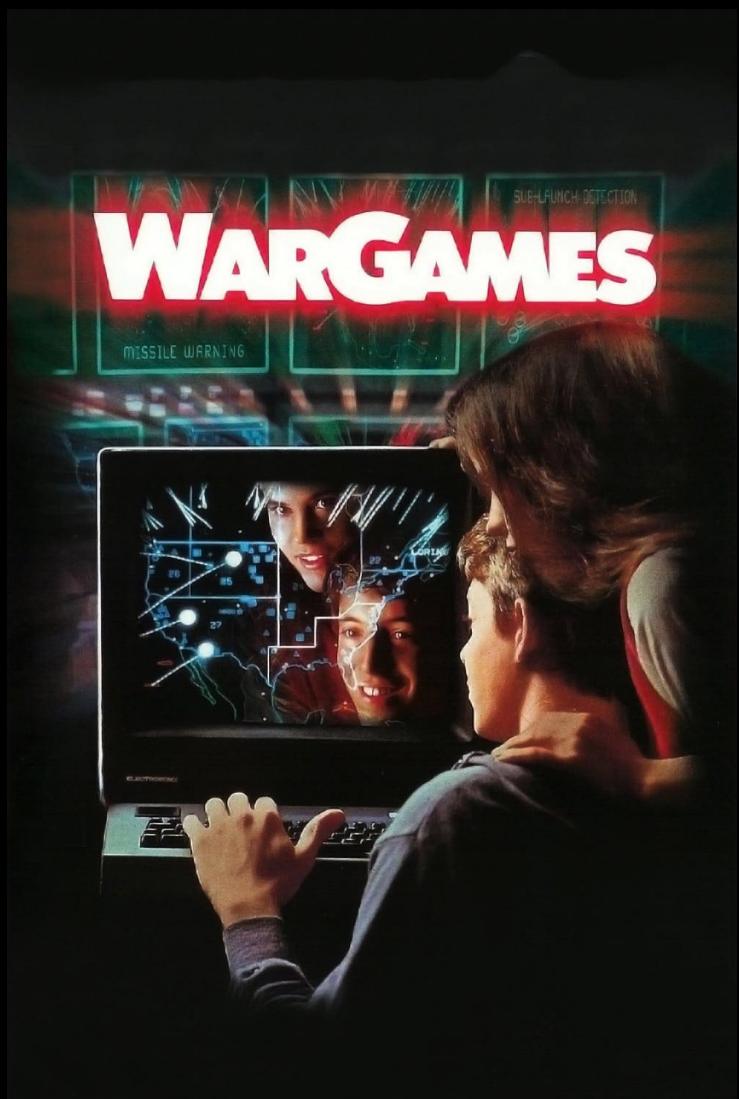
---

Cheese fest trivia time - text ALL answers at ONCE in order:  
Letter case DOES NOT matter, just all lowercase letters are fine. First  
person gets a prize from me 😊

**(920) 659-7883**

1. Password David used to start a Global Thermonuclear War?
2. Password Joey used to access Ellingson Mineral Corporation system or second password Lord Nikon while shoulder surfing with flowers?
3. Password Lord Nikon saw first while shoulder surfing with flowers?
4. Ginger Knowles' bank password?

# Answers



# Q&A

<https://www.linkedin.com/in/taliy/>

[tally@sevnx.com](mailto:tally@sevnx.com) or [taliy@sevx.com](mailto:taliy@sevx.com) or [vitaliy.kovalchuk@sevnx.com](mailto:vitaliy.kovalchuk@sevnx.com)



---

**BIG Thank You  
GrassROOTz Organizers!**