Ethical hacking, made easy.

Kitsec is a powerful toolkit CLI designed to help you simplify & centralize your security workflow. Whether you're a seasoned professional or just getting started, Kitsec provides a comprehensive set of tools to help you stay on top of your game.

✨ Features

- **VPS Logger**: Login to your VPS with a single command.
- **Convert**: Applies a specified decoding or hashing function to input data. (ie. URL, HTML, Base64, ASCII, Hex, Octal, Binary & GZIP).
- **Enumerator**: Enumerates subdomains for a given domain using subfinder, amass, assetfinder and findomain and active enumeration.
- **Capture**: Send a GET request to a specified URL, capture the request headers, extract the hostname, path, and cookies and missing headers.
- **Portscan**: Scan a host for common or all possible open ports.
- **Certificate**: Check the SSL/TLS certificate information for a given URL.
- **Storm**: Sends HTTP requests to a given URL with a specified number of attacks and requests.
- **Disturb**: Send multiple HTTP requests to the specified URL with the same payload.
- **Fuzz**: Test your web applications against path fuzzing and file fuzzing.
- **CIDR**: Looks up the CIDR range for a company's domain name from its RDAP record.
- **CVE**: Retrieves CVE data for a specific product name (company name) from NIST's National Vulnerability Database (NVD).

🛣️ Roadmap

- **Add raid types**: Add flood, hybrid and single shot
- **XSS Scan**: Add XSS scanner.

📦 Installation

Assuming that you have Python 3.6+ installed, you can install Kitsec using pip:

```
pip install kitsec
```

Assuming that you have go installed, you can install the additional dependencies:

```
kitsec deps
```

Usage

🧢 VPS Logger

Connects to a remote VPS server and tails the auth.log file.

```
kitsec vps-logger -h <IP ADDRESS> -u <USERNAME> -p <PASSWORD>
```

```
Usage: kitsec vps-logger [OPTIONS]

Connects to a remote VPS server and tails the auth.log file.

Options:
  -h, --host TEXT      The IP address of the VPS server to connect to.
  -u, --username TEXT  The limited user account to use for connecting to
the VPS server.
  -p, --password TEXT  The password for the user account.
  --help               Show this message and exit.

Returns:
- Prints a continuous stream of output from the auth.log file to the
console.

The program attempts to connect to the specified VPS server using SSH,
with the provided
username and password. Once connected, it invokes a shell and sends the
command to tail
the auth.log file using sudo. It then continuously checks for new output
from the file and
prints it to the console as it is received.
```

📸 Capture

Intercept requests to example.com. This will capture the request headers and extract the hostname and path + cookies! :

```
Usage: kitsec capture [OPTIONS] URL

  Captures the request headers for a given URL.

Options:
  --help  Show this message and exit.

Example:
  kitsec capture https://example.com
```

▶ Output

```
GET /mynetwork/ HTTP/1.1
Host: www.website.com
```

```
Accept-Encoding: gzip, deflate, br
Accept: */*
Connection: keep-alive
Cookie: bcookie="v=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
lang=v=2&lang=en-us;
li_gc=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxx;
lidc="b=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx";
JSESSIONID=ajax:xxxxxxxxxxxxxxxxx;
bscookie="v=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx"

Response headers:
    Cache-Control: no-cache, no-store
    Pragma: no-cache
    Content-Length: 7486
    Content-Type: text/html; charset=utf-8
    Content-Encoding: gzip
    Expires: Thu, 01 Jan 1970 00:00:00 GMT
    Vary: Accept-Encoding
    Content-Security-Policy: default-src *; connect-src 'self' *.domain
etc etc etc *
    X-Frame-Options: sameorigin
    X-Content-Type-Options: nosniff
    Strict-Transport-Security: max-age=31536000
    Expect-CT: max-age=86400, report-
uri="https://www.website.com/platform-telemetry/ct"
    X-Li-Fabric: prod-lzx7
    X-Li-Pop: azd-prod-lzx7-x
    X-Li-Proto: http/1.1
    X-LI-UUID: AAX2TIh6unm3s+DezlC6rw==
    X-Cache: CONFIG_NOCACHE
    X-MSEdge-Ref: Ref A: BB20069DED8C4CF68A735496B4DAFD79 Ref B:
PAR02EDGE0721 Ref C: 2023-03-07T10:04:11Z
    Date: Tue, 07 Mar 2023 10:04:11 GMT
```

🪄 Convert

Convert your data from one format to another:

```
Usage: kitsec convert [OPTIONS] INPUT [-t TYPE]

Applies a specified decoding or hashing function to input data.

Arguments:
INPUT The input data to be converted.

Options:
-t, --type TYPE The type of conversion to apply (HTML, Base64, ASCII,
Hex, Octal, Binary & GZIP).
```

```
--help Show this message and exit.

Example:
kitsec convert S2l0c2VjIFJvY2tzIQ== -t Base64
```

▶ Output

```
Kitsec Rocks!
```

🧱 Enumerate

Enumerate subdomains for example.com

```
Usage: kitsec enumerate [OPTIONS] DOMAIN

Enumerates subdomains for a given domain using Subfinder and active
enumeration.

Arguments:
DOMAIN The domain to enumerate subdomains for.

Options:
-r, --request Fetch HTTP response for active subdomains.
-t, --technology Analyze technologies used by subdomains.
-a, --active Perform active enumeration.
--help Show this message and exit.

Example:
kitsec enumerate -r -t -a example.com
```

▶ Output

```
Subdomain                      Status  Reason              Technology
-------------------------  --------  ------------------  -------------
-------------------------------------------------
tracking.webapp.domain1.com       503  Service Unavailable  []
legal.domain1.com                 404  Not Found
['Strikingly', 'Lua', 'jQuery', 'Nginx', 'OpenResty']
help.domain1.com                  403  Forbidden
['Cloudflare']
staging-api.domain1.com           401  Unauthorized         []
api.domain1.com                   401  Unauthorized         []
staging-app.domain1.com           200  OK                   ['Nginx',
'Google Font API', 'React', 'Stripe']
staging-website.domain1.com       200  OK                   ['Nginx',
'Google Font API', 'React', 'Stripe']
```

```
sales.domain1.com                  200  OK                    ['Nginx',
'Google Font API', 'React', 'Stripe']
```

📡 Port Scan

Scan for all or most common open ports on example.com:

```
Usage: kitsec portscan [OPTIONS] HOSTNAME

Performs a TCP port scan on a specified hostname and a range of ports.

Arguments:
HOSTNAME The hostname or URL of the target host.

Options:
-c, --common-ports Scan only the most common HTTP ports (80, 8080, and
443).
--help Show this message and exit.

Example:
kitsec portscan -c example.com
```

▶ Output

```
Open Ports:
example.com:80
example.com:443
```

📶 CIDR

Search for CIDR ranges for a given domain name:

```
Usage: kitsec cidr [OPTIONS] COMPANY_NAME

Look up the CIDR range for a company's domain name.

Arguments:
  COMPANY_NAME  The name of the company's domain name to look up.

Options:
  --help           Show this message and exit.

Returns:
  - The CIDR range for the company's domain name as a string.
  - If an exception is raised during the lookup process,
  an error message will be displayed.
```

```
Example:
  kitsec cidr github.com
```

▶ Output

```
The CIDR range for domain.com is 141.82.112.0/20
```

📜 Certificate

Search for ssl / tlsfor the specified host and port:

```
Usage: kitsec certifcate [OPTIONS] HOSTNAME

Check the SSL/TLS certificate for the specified host and port.

Arguments:
  HOSTNAME  The hostname to check the certificate for.

Options:
  -p, --port INTEGER  The port to connect to. Default is 443.
  --help              Show this message and exit.

Returns:
  None. Displays the certificate information to the console.

Example:
  kitsec certificate github.com
```

▶ Output

```
Hostname: github.com
Not Before: 2023-02-14 00:00:00
Not After: 2024-03-14 23:59:59
```

🌐 CVE

Search for CVEs for the specified product.

```
Usage: kitsec cve [OPTIONS] PRODUCT_NAME

Retrieves CVE data for a specific product and displays it.

Arguments:
  PRODUCT_NAME  The product name (company name) to search for.

Options:
```

```
  --limit INTEGER  Number of results to display (default=10).
  --help           Show this message and exit.

Example:
  kitsec cve python -l 2
```

▶ Output

```
CVE ID    CVE-2023-26477
CWE       CWE-94: Improper Control of Generation of Code ('Code
Injection') (4.10)
Severity  Severity information not available
Summary   XWiki Platform is a generic wiki platform. Starting in versions
6.3-rc-1 and 6.2.4, it's possible to inject arbitrary wiki syntax
including Groovy, Python and Velocity script macros via the `newThemeName`
request parameter (URL parameter), in combination with additional
parameters. This has been patched in the supported versions 13.10.10,
14.9-rc-1, and 14.4.6. As a workaround, it is possible to edit
`FlamingoThemesCode.WebHomeSheet` and manually perform the changes from
the patch fixing the issue.

CVE ID    CVE-2018-1000802
CWE       CWE-77: Improper Neutralization of Special Elements used in a
Command ('Command Injection') (4.10)
Severity  Severity information not available
Summary   Python Software Foundation Python (CPython) version 2.7 contains
a CWE-77: Improper Neutralization of Special Elements used in a Command
('Command Injection') vulnerability in shutil module (make_archive
function) that can result in Denial of service, Information gain via
injection of arbitrary files on the system or entire drive. This attack
appear to be exploitable via Passage of unfiltered user input to the
function. This vulnerability appears to have been fixed in after commit
add531a1e55b0a739b0f42582f1c9747e5649ace.
```

storm

Send HTTP requests to a given URL with a specified number of Attacks and requests.

```
Usage: kitsec storm [OPTIONS] URL

Sends HTTP requests to a given URL with a specified number of threats and
requests.

Arguments:
URL The URL to send HTTP requests to.

Options:
-a, --num-attacks INT Number of parallel attacks to send requests from.
Default: 6.
```

```
-r, --num-requests INT Number of requests to send from each threat.
Default: 200.
-y, --num-retries INT Number of times to retry failed requests. Default:
4.
-p, --pause-before-retry INT Number of milliseconds to wait before
retrying a failed
request. Default: 3000.
--help Show this message and exit.

Example:
kitsec storm https://example.com/
```

fuzz

`kitsec fuzz example.com`

```
Usage: kitsec fuzz [OPTIONS] BASE_URL

Sends HTTP GET requests to a specified base URL with a given list of
paths.

Arguments:
BASE_URL The base URL to send requests to. The URL must include the
protocol (http or https).

Options:
-p, --path PATH The path to a file or directory containing a list of paths
to send requests
to. Default: ../lists/fuzz/path_fuzz
-f, --file-fuzz Use file format fuzzing
--help Show this message and exit.

Example:
kitsec fuzz example.com
```

Guidelines

Here are some guidelines for using open source tools for ethical hacking:

1. Bug bounties are not a license to hack indiscriminately. Stay withing your scope and safe harbour.

- Ensure you have a strong understanding of the open source tools being used and their impact.
- Always obtain written permission from the owner of the target system before testing.
- Never go beyond the scope of the agreement.
- Be professional in your approach.

Here are some examples of websites that offer bug bounty programs for ethical hackers to test their skills:

- Google Vulnerability Reward Program

- Microsoft Bounty Program
- HackerOne
- Bugcrowd
- Synack

Reporting Bugs and Contributing

If you encounter any bugs or would like to suggest new features https://github.com/kitsec-labs/kitsec/issues/new

Disclaimer

This project is made for educational and ethical testing purposes only. Usage of this tool for attacking targets without prior mutual consent is illegal. Developers assume no liability and are not responsible for any misuse or damage caused by this tool.

Acknowledgements

Thank you to @projectdiscovery, @milo2012, @duyet, @ayoubfathi, @Bo0oM and @Practical-Formal-Methods for opening their tools to the world.

License