Ethical Hacking and Countermeasures
OSRFramework Cheat Sheet

EC-Council
Building A Culture Of Security

OSRFramework
Source: https://github.com

OSRFramework includes applications related to username checking, DNS lookups, information leaks research, deep web search, regular expression extraction, etc. Tools included in the OSRFramework package to gather information on the target are usufy.py, mailfy.py, searchfy.py, domainfy.py, phonefy.py, and entify.py.

usufy.py – Verifies a user profile against 290 different platforms

Syntax

```
usufy.py
(--info <action> | --license | -b | -f <path_to_fuzzing_list> | -l
<path_to_nick_list> | -n <nick> [<nick> ...] | --show_tags)
[-p <platform> [<platform> ...]] [-t <tag> [<tag> ...]]
[-x <platform> [<platform> ...]] [--avoid_download]
[--avoid_processing] [--fuzz_config <path_to_fuzz_list>]
[--nonvalid <not_valid_characters>]
[-e <sum_ext> [<sum_ext> ...]] [-L <path_to_log_folder] [-m]
[-o <path_to_output_folder>] [-w]
[-F <alternative_header_file>] [-T <num_threads>] [-h]
[-v <verbosity>] [--version]
```

Input options

`--info <action>`	Action to be performed such as list_platforms, list_tags
`--license`	Displays license
`-b, --benchmark`	Benchmarking tasks
`-f <path_to_fuzzing_list>, --fuzz <path_to_fuzzing_list>`	Find URLs similar to usufy
`-l <path_to_nick_list>, --list <path_to_nick_list>`	Specify the path to the file where the list of nicks to verify is stored (one per line)
`-n <nick> [<nick> ...], --nicks <nick> [<nick> ...]`	Specify a list of nicks to process
`--show_tags`	Shows platforms grouped by tags

Platform selection arguments

`-p <platform> [<platform> ...], --platforms <platform> [<platform> ...]`	Specify the platform for searching
`-t <tag> [<tag> ...], --tags <tag> [<tag> ...]`	Specify the list of tags that fit the platforms to perform the search
`-x <platform> [<platform> ...], --exclude <platform> [<platform>`	Specify the platforms you want to exclude from the processing

Processing arguments

`--avoid_download`	Use to prevent storing the downloadable version of the profiles
`--avoid_processing`	Use to prevent performing any processing task without valid profiles
`--fuzz_config <path_to_fuzz_list>`	Specify the path to the fuzzing config details
`--nonvalid <not_valid_characters>`	Specify not valid characters
`-e <sum_ext> [<sum_ext> ...], --extension <sum_ext> [<sum_ext> ...]`	Use to output extension for the summary files. Default: XLS
`-L <path_to_log_folder, --logfolder <path_to_log_folder`	Specify the path to the log folder. If none was provided, ./logs is assumed
`-m, --maltego`	Specify parameter to launch usufy.py by using Maltego Transform

Processing arguments

`-o <path_to_output_folder>, --output_folder <path_to_output_folder>`	Specify Output folder for the generated documents
`-w, --web_browser`	Specify default web browser
`-F <alternative_header_file>, --file_header <alternative_header_file>`	Specify header for the output filenames
`-T <num_threads>, --threads <num_threads>`	Specify the number of threads to be used (default 32)

Arguments

`-h, --help`	Displays help
`-v <verbosity>, --verbose <verbosity>`	Select the verbosity level: 0 - none; 1 – normal (default); 2 - debug
`--version`	Displays the version of the program

usufy.py Commands

Commands	Description
`usufy.py -n <username> -p twitter facebook youtube`	Search for a user on social media sites
`usufy.py -n <username>`	Check username (profile) on various platforms

mailfy.py – Check for the existence of a given mail

Syntax

```
mailfy.py
(--license | -m <emails> [<emails> ...] | -M
<emails_file> | -n <nicks> [<nicks> ...] | -N
<nicks_file> | --create_emails <nicks_file>)
[-e <sum_ext> [<sum_ext> ...]]
[-d <candidate_domains> [<candidate_domains> ...]]
[-o <path_to_output_folder>] [-x <domain> [<domain>
...]]
[-F <alternative_header_file>] [-T <num_threads>]
[--is_leaked] [--quiet] [-h] [--version]
```

Input options

`--license`	Displays license
`-m <emails> [<emails> ...], --emails <emails> [<emails> ...]`	Specify a list of emails for checking
`-M <emails_file>, --emails_file <emails_file>`	Input file containing a list of emails
`-n <nicks> [<nicks> ...], --nicks <nicks> [<nicks> ...]`	Specify a list of nicks for checking against domains selected
`-N <nicks_file>, --nicks_file <nicks_file>`	Input file containing a list of nicks to be checked against domains selected
`--create_emails <nicks_file>`	Input file with the list of nicks to be created in the domains selected

Processing arguments

`-e <sum_ext> [<sum_ext> ...], --extension <sum_ext> [<sum_ext> ...]`	Specify the output extension for the summary files. Default: XLS
`-d <candidate_domains> [<candidate_domains> ...], --domains <candidate_domains> [<candidate_domains> ...]`	-d <candidate_domains> [<candidate_domains> ...], --domains <candidate_domains> [<candidate_domains> ...]
`-o <path_to_output_folder>, --output_folder <path_to_output_folder>`	Specify the output folder for the generated documents
`-x <domain> [<domain> ...], --exclude <domain> [<domain> ...]`	Specify the domains to be excluded from the search
`-F <alternative_header_file> , --file_header <alternative_header_file>`	Specify the Header for the output filenames to be generated
`-T <num_threads>, --threads <num_threads>`	Specify the number of threads to be used (default 16
`--is_leaked`	Specify whether mailfy.py should search for leaked emails
`--quiet`	Inform the program not to show anything

Arguments

`-h, --help`	Shows help
`--version`	Displays version of the program and exists

mailfy.py Commands

Commands	Description
`mailfy.py -n <target email>`	Validates the existences of a given email address
`mailfy.py -m <target email>`	Check whether the given email address was used on the Internet
`mailfy.py -m <target email> <website>`	Check whether the given email address was used on the specified web site

searchfy.py – Performs a query on the platforms in OSRFramework

Syntax

```
searchfy.py
(--license | -q <searches> [<searches> ...])
[-e <sum_ext> [<sum_ext> ...]]
[-F <alternative_header_file>]
[-m] [-o <path_to_output_folder>]
[-p <platform> [<platform> ...]] [--process] [-w]
[-x <platform> [<platform> ...]] [-h] [--version]
```

Input options	
--license	Displays license
-q <searches> [<searches> ...], --queries <searches> [<searches> ...]	Specify a list of queries to be performed

Processing arguments	
-e <sum_ext> [<sum_ext> ...], --extension <sum_ext> [<sum_ext> ...]	Specify the output extension for the summary files. Default: xls
-F <alternative_header_file>, --file_header <alternative_header_file>	Specify header for the output filenames to be generated
-o <path_to_output_folder>, --output_folder <path_to_output_folder>	Specify the output folder for the generated documents
-p <platform> [<platform> ...], --platforms <platform> [<platform> ...]	Specify the platforms to perform the search
--process	Specify whether to process the info in the profiles recovered
-w, --web_browser	Open URIs returned in the default web browser
-x <platform> [<platform> ...], --exclude <platform> [<platform> ...]	Specify the platforms you want to exclude from the processing

Arguments	
-h, --help	Shows help
--version	Displays version of the program

domainfy.py – Checks for the existence of domains

Syntax

```
domainfy.py (--license | -n <nicks> [<nicks> ...] | -N
                        <nicks_file>)
             [-e <sum_ext> [<sum_ext> ...]] [-o
                <path_to_output_folder>]
              [-t <tld_type> [<tld_type> ...]]
 [-u <new_tld> [<new_tld> ...]] [-x <domain> [<domain>
                          ...]]
  [-F <alternative_header_file>] [-T <num_threads>]
                      [--quiet]
                  [-h] [--version]
```

Input options	
--license	Display license
-n <nicks> [<nicks> ...], --nicks <nicks> [<nicks> ...]	Specify a list of nicks to be checked in the selected domains
-N <nicks_file>, --nicks_file <nicks_file>	Specify the file having the list of nicks to be checked in the selected domains

Processing arguments	
-e <sum_ext> [<sum_ext> ...], --extension <sum_ext> [<sum_ext> ...]	Specify parameter to launch usufy.py by using Maltego Transform
-o <path_to_output_folder>, --output_folder <path_to_output_folder>	Specify the output folder for the generated documents
-x <domain> [<domain> ...], --exclude <domain> [<domain> ...]	Specify the domains to be excluded from the search
-F <alternative_header_file>, --file_header <alternative_header_file>	Specify header for the output filenames to be generated
-T <num_threads>, --threads <num_threads>	Specify the number of threads to be used (default 16)
-t <tld_type> [<tld_type> ...], --tlds <tld_type> [<tld_type> ...]	Specify the list of TLD types where the nick will be searched for
--quiet	Informs the program not to show anything

Arguments	
-h, --help	Shows help
--version	Displays version of the program

domainfy.py Commands

Commands	Description
domainfy.py -n <nicks> -t all	Verify all top-level domains (-t all) for a registered name
domainfy.py –whois -n <Target IP>	Display results in table format for whois search

phonefy.py – Checks for the existence of a given series of phones

Syntax

```
   phonefy.py (--license | -n <phones> [<phones> ...]) [-e
  <sum_ext> [<sum_ext> ...]] [-o <path_to_output_folder>]
          [-p <platform> [<platform> ...]] [-F
     <alternative_header_file>] [--quiet] [-w] [-x
      <platform> [<platform> ...]] [-h] [--version]
```

Input options	
--license	Displays License
-n <phones> [<phones> ...], --numbers <phones> [<phones> ...]	Specify a list of phones to process

Processing arguments	
-e <sum_ext> [<sum_ext> ...], --extension <sum_ext> [<sum_ext> ...]	Specify the output extension for the summary files. Default: XLS
-o <path_to_output_folder>, --output_folder <path_to_output_folder>	Specify the output folder for the generated documents
-F <alternative_header_file>, --file_header <alternative_header_file>	Specify header for the output filenames to be generated
-p <platform> [<platform> ...], --platforms <platform> [<platform> ...]	Specify the platforms where you want to perform the search
-w, --web_browser	Open the URIs returned in the default web browser
-x <platform> [<platform> ...], --exclude <platform> [<platform> ...]	Specify the platforms that you want to exclude from the processing

Arguments	
-h, --help	Displays help
--version	Displays version of the program

domainfy.py Commands

Commands	Description
phonefy.py -n <number>	Finds URL location and platform for a given telephone number

entify.py – Use regular expressions to extract entities

Syntax

```
        entify.py (-r <name> [<name> ...] | -R
                <regular_expression>)
          (-i <path_to_input_folder> | -w <url>)
            [-e <sum_ext> [<sum_ext> ...]] [-o
              <path_to_output_folder>]
   [-v <verbosity>] [-F <alternative_header_file>]
                         [-q]
     [-L <path_to_log_folder] [--recursive] [-h]
                     [--version]
```

Input options	
-r <name> [<name> ...], --regexp <name> [<name> ...]	Specify the regular expressions to be searched for
-R <regular_expression>, --new_regexp <regular_expression>	specify a new regular expression
-i <path_to_input_folder>, --input_folder <path_to_input_folder>	Specify the path to the folder to analyze
-w <url>, --web <url>	Specify URI to be recovered and analyzed

Processing arguments	
-e <sum_ext> [<sum_ext> ...], --extension <sum_ext> [<sum_ext> ...]	Specify the output extension for the summary files. Default: XLS
-o <path_to_output_folder>, --output_folder <path_to_output_folder>	Specify the output folder for the generated documents
-F <alternative_header_file>, --file_header <alternative_header_file>	Sepcify the verbosity level: 0 - none; 1 – normal (default); 2 - debug
-v <verbosity>, --verbose <verbosity>	Sepcify the verbosity level: 0 - none; 1 – normal (default); 2 - debug
-L <path_to_log_folder, --logfolder <path_to_log_folder	Specify the path to the log folder. If none was provided, ./logs is assumed
--recursive	Input variable to inform the system to perform a recursive search on the folder tree

Arguments	
-h, --help	Shows help
--version	Displays version of the program

osrfconsole.py – Open Sources Research Framework console interface

Syntax
`osrfconsole.py`

Modules Available	
`- usufy`	Checks if a username exists in 291 platforms
`- mailfy`	Verifies if a username has been registered in up to 22 email providers
`- searchfy`	Finds profiles using full names and other information in 7 platforms
`- domainfy`	Verifies the existence of a given domain in up to 1567 different TLD
`- phonefy`	Specify the platforms to perform the search
`--process`	Verifies if a phone number has been linked to spam practices in 4 platforms
`- entify`	Util to look for regular expressions using 13 patterns

osrframework_server.py – Open Sources Research Framework web interface

Syntax
`./osrframework-server.py [--host <IP>] [--port <PORT>] [--debug] [-h] [--version]`

Configuration arguments	
`--host <IP>`	Select the host in which the server will be accessible Default: localhost
`--port <PORT>`	Select the port in which the server will be accessible. Use this carefully
`--debug`	Select whether error messages will be deployed

Arguments	
`-h, --help`	Shows help
`--version`	Version of the program and exists

osrframework_server.py Commands

Commands	Description
`osrframework_server.py -h`	Start a local server