Viper - Using MISP from your terminal

MISP - Malware Information Sharing Platform & Threat Sharing

CIRCL
Computer Incident
Response Center
Luxembourg

Raphaël Vinot - *TLP:WHITE*

MISP Training - 20160905

Viper - Main ideas

Viper is a **binary analysis and management framework**. *Its fundamental objective is to provide a solution to* **easily organize** *your collection of* **malware** *and* **exploit samples** *as well as your collection of* **scripts** *you created or found over the time to facilitate your daily research. Think of it as a* **Metasploit for malware researchers**: *it provides a terminal interface that you can use to* **store**, **search** *and* **analyze** *arbitraty files with and a framework to* **easily create plugins** *of any sort.*

Viper

- **Solid CLI**
- Plenty of modules (PE files, *office, ELF, APK, ...)
- Connection to **3rd party services** (MISP, VirusTotal, cuckoo)
- Connectors to **3rd party tools** (IDA, radare)
- **Locale storage** of your own zoo
- Django interface is available (I've been told)

Viper

Command	Description
apk	Parse Android Applications
clamav	Scan file from local ClamAV daemon
cuckoo	Submit the file to Cuckoo Sandbox
debup	Parse McAfee BUP Files
editdistance	Edit distance on the filenames
elf	Extract information from ELF headers
email	Parse eml and msg email files
exif	Extract Exif MetaData
fuzzy	Search for similar files through fuzzy hashing
html	Parse html files and extract content
ida	Start IDA Pro
idx	Parse Java IDX files
image	Perform analysis on images
jar	Parse Java JAR archives
koodous	Interact with Koodous
lastline	Submit files and retrieve reports from LastLine (default will print short summary)
macho	Get Macho OSX Headers
misp	Upload and query IOCs to/from a MISP instance
office	Office Document Parser
pdf	Parse and analyze PDF documents
pdns	Query a Passive DNS server
pe	Extract information from PE32 headers
pssl	Query a Passive SSL server
pst	Process PST Files for Attachment
r2	Start Radare2
rat	Extract information from known RAT families
reports	Online Sandboxes Reports
shellcode	Search for known shellcode patterns
size	Size command to show/scan/cluster files
strings	Extract strings from file
swf	Parse, analyze and decompress Flash objects
triage	Perform some initial triaging and tagging of the file

PyMISP & Viper

- Full featured **CLI for MISP**
- **Remote storage** of your zoo
- Search / **Cross check with VirusTotal**
- Create / Update / Show / Publish Event
- Download / Upload Samples
- Mass export / Upload / Download
- Get Yara rules

MISP Module

```
viper > misp -h
usage: misp [-h] [--url URL] [-k KEY] [-v]
            {upload,download,search,check_hashes,yara,pull,create_event,add,show,open,
publish,version,store}
            ...

Upload and query IOCs to/from a MISP instance

positional arguments:
  {upload,download,search,check_hashes,yara,pull,create_event,add,show,open,publish,ve
rsion,store}
    upload              Send malware sample to MISP.
    download            Download malware samples from MISP.
    search              Search in all the attributes.
    check_hashes        Crosscheck hashes on VT.
    yara                Get YARA rules of an event.
    pull                Initialize the session with an existing MISP event.
    create_event        Create a new event on MISP and initialize the session
                        with it.
    add                 Add attributes to an existing MISP event.
    show                Show attributes to an existing MISP event.
    open                Open a sample from the temp directory.
    publish             Publish an existing MISP event.
    version             Returns the version of the MISP instance.
    store               Store the current MISP event in the current project.

optional arguments:
  -h, --help            show this help message and exit
  --url URL             URL of the MISP instance
  -k KEY, --key KEY     Your key on the MISP instance
  -v, --verify          Disable certificate verification (for self-signed)
```

Viper & VT

- Searches for hashes/ips/domains/URLs from the current MISP event, or download the samples
- Download samples from current MISP event
- Download all samples from all the MISP events of the current session

VirusTotal Module

```
Lookup the file on VirusTotal

optional arguments:
  -h, --help            show this help message and exit
  --search SEARCH       Search a hash.
  -c COMMENT [COMMENT ...], --comment COMMENT [COMMENT ...]
                        Comment to add to the file
  -d, --download        Hash of the file to download
  -dl, --download_list  List the downloaded files
  -do DOWNLOAD_OPEN, --download_open DOWNLOAD_OPEN
                        Open a file from the list of the DL files (ID)
  -don DOWNLOAD_OPEN_NAME, --download_open_name DOWNLOAD_OPEN_NAME
                        Open a file by name from the list of the DL files
                        (NAMe)
  -dd DOWNLOAD_DELETE, --download_delete DOWNLOAD_DELETE
                        Delete a file from the list of the DL files can be an
                        ID or all.
  -s, --submit          Submit file or a URL to VirusTotal (by default it only
                        looks up the hash/url)
  -i IP, --ip IP        IP address to lookup in the passive DNS
  -dm DOMAIN, --domain DOMAIN
                        Domain to lookup in the passive DNS
  -u URL, --url URL     URL to lookup on VT
  -v, --verbose         Turn on verbose mode.
  -m {hashes,ips,domains,urls,download,download_all}, --misp {hashes,ips,domains,urls,
download,download_all}
                        Searches for the hashes, ips, domains or URLs from the
                        current MISP event, or download the samples if
                        possible. Be carefull with download_all: it will
                        download *all* the samples of all the MISP events in
                        the current project.
```

Extra features

- Link to a MISP event
- Local storage of the MISP event
- On the fly cross-check of MISP atributes with 3rd party services
- Never leaving your CLI!

Other modules

- Fully featured CLI for **Passive SSL**
- Fully featured CLI for **Passive DNS**
- Can launch Radare2 or IDA

Passive SSL

```
viper > pssl -h
usage: pssl [-h] [--url URL] [-u USER] [-p PASSWORD] [-i IP] [-c CERT]
            [-f FETCH] [-v] [-m {ips}]

Query a Passive SSL server

optional arguments:
  -h, --help            show this help message and exit
  --url URL             URL of the Passive SSL server (No path)
  -u USER, --user USER  Username on the PSSL instance
  -p PASSWORD, --password PASSWORD
                        Password on the PSSL instance
  -i IP, --ip IP        IP to query (can be a block, max /23).
  -c CERT, --cert CERT  SHA1 of the certificate to search.
  -f FETCH, --fetch FETCH
                        SHA1 of the certificate to fetch.
  -v, --verbose         Turn on verbose mode.
  -m {ips}, --misp {ips}
                        Searches for the ips from the current MISP event
```

Passive DNS

```
viper > pdns -h
usage: pdns [-h] [--url URL] [-u USER] [-p PASSWORD] [-v] [-m {ips,domains}]
            [query]

Query a Passive DNS server

positional arguments:
  query                 Domain or IP address to query

optional arguments:
  -h, --help            show this help message and exit
  --url URL             URL of the Passive DNS server
  -u USER, --user USER  Username on the PDNS instance
  -p PASSWORD, --password PASSWORD
                        Password on the PDNS instance
  -v, --verbose         Turn on verbose mode.
  -m {ips,domains}, --misp {ips,domains}
                        Searches for the ips or domains from the current MISP
                        event
```

Q&A

- `https://github.com/MISP/PyMISP`
- `https://github.com/MISP/`
- `https://github.com/viper-framework/viper`
- We welcome new functionalities and pull requests.