assassin

Release 0.0.1

Jul 22, 2020

Contents:


paloalto®
NETWORKS

CHAPTER 1

Introduction

Overview When an attacker looks at an organization as her target, nearly every approach starts with reconnaissance. Gathering information about the organization takes place in many ways, but today's attacker rarely does so in a way that can be detected by most security tools. The purpose of this project is to illustate a common technique that is easily automated and rich in its ability to mine information, giving the attacker a list of potential target devices.

Description Asassin was originally developed by the Palo Alto Networks SecOps CE Team. It uses some scripting to assemble the output of a few well known techniques (Hacker Target, Shodan) to gather information about the publicly registered devices of a target's domain, search for known vulnerabilities associated with those devices and matching up some descriptions of those vulnerabilities with CVE information. This gives the attacker a view of what the target has and how those devices might be attacked. From the attacker's perspective, the key advantage this approach is that from the target's perspective, it is completely silent and invisible. At no time during the process does the attacker ever touch any resource that actually belongs to the target.

CHAPTER 2

Quickstart for Mac

2.1 Install brew

`https://docs.brew.sh/Installation`

```
mkdir homebrew && curl -L https://github.com/Homebrew/brew/tarball/master|tar xz --
↪strip 1 -C homebrew
brew tap caskroom/cask
```

2.2 Install Git

```
brew install git
```

2.3 Install Docker Desktop

Install Docker on MAC: overview.

Download the software package here.

2.4 Configure "apiKeys.py"

- Edit the file assassin/apiKey.py
- Change default values of apiKeys.py
- Save file with update API key values
- Do not commit your API keys to the repo

```
vtKey = 'CHANGEME'
shodanKey = 'CHANGEME'
GoogleMapsKey = 'CHANGEME'
dnsdbKey = 'CHANGEME'
GoogleSafeBrowsingKey = 'CHANGEME'
```

2.5 Disable Global Protect

If the operator runs the tool with Global Protect enabled, the Shodan portion of the tool is blocked.

- The console will show HTTP 503 errors.
- The report output will be incomplete.

2.6 Run the Assassin Tool

```
make docker
cd assassin
python assassin.py
```

2.7 Viewing Reports

Two *.html* files (a detail file and a summary file) will be written to /app/assassin directory with the results of your scans. These files should persist even if the operator exits the Docker container.

To remove these files, execute the *make clean* option from the top level of the repo directory.

CHAPTER 3

Keys

There is a template *apiKeys.py* file that can be populated with various keys to improve the functionality of the tool and the granularity of the report.

3.1 Shodan

The tool will check for the existence of an environment variable containing the value of the Shodan key. For example:

```
export SHODAN_KEY=`abc123keystringValue`
```

If this is not found, the tool will check for a value in the *apiKeys.py* file. Failing this, the tool will log an exception and move on. No Shodan report will be generated for hosts within the domain.

3.2 Google Maps

The tool will check for the existence of an environment variable containing the value of the Google Maps key. For example:

```
export GOOGLE_MAPS_KEY=`abc123keystringValue`
```

If this is not found, the tool will check for a value in the *apiKeys.py* file. Failing this, the tool will log an exception and move on. No Google map will be added to the top of the detail report.

CHAPTER 4

Operation

4.1 Docker

While this tool can be run by installing the dependent Python3 modules to your local machine, it is design so you can install Docker and run the tool inside a container. The goal is to prevent issues with differing Python 2 & 3 versions, as well as possible dependency issues.

You can install Docker to your local machine by clicking this link.

4.2 Make

For ease of operation, this is a Makefile driven project. Once you've installed Docker, type the command *make* to show the options available.

```
user@host: ~   make
clean                Cleanup all the things
docker               build docker container for testing
docs                 Generate documentation
python               setup python3
test                 run tests in container

Use the command `make docker` to set up the container.
```

4.3 Running Scans

To run the detail and summary reports for a domain, change to the /app/assassin directory and run the tool.

```
user@host: ~   make docker
Successfully built f591a2044610
```

(continues on next page)

(continued from previous page)

```
Successfully tagged docker_assassin:latest
root@assassin:/app# cd assassin
root@assassin:/app/assassin# python assassin.py --domain paloaltonetworks.com
```

If you do not pass the *–domain* flag you will be prompted to manually enter a domain.

```
root@assassin:/app/assassin# python assassin.py
Signatures loaded
What domain would you like to search ? paloaltonetworks.com
```

4.4 Logging

The tool is configured to write a log file to */var/log/secops/assassin.log*. See the troubleshooting section for more details.

CHAPTER 5

Assasin Local Dev Env

5.1 Testing

Test cases are located in the /test folder. Test-specific Python3 modules, to be installed in the docker container are listed in */requirements-test.txt*. This includes the tox and coverage modules. The Dockerfile specifies a recent version of Python 3.x.

5.2 Disable Global Protect

If the operator runs the tool with Global Protect enabled, the Shodan portion of the tool is blocked as described in the troubleshooting section.

- The console will show HTTP 503 errors.
- The report output will be incomplete.

5.3 Running Test Suite Manually

Use the Makefile options to start the Docker container and execute the test suite.

```
make docker
make test
```

You should run the command *docker system prune* occasionally to free up space on your filesystem.

5.4 Automated Testing

The file */.github/workflows/python.yml* is the CI test configuration for GitHub Actions. This workflow will install Python dependencies, run tests and lint with a single version of Python (currently 3.8).

Test runs can be viewed (and reviewed for Pull Request errors) by clicking this link.

CHAPTER 6

Reports

Two reports are produced by the tool. A summary report and a detail report.

A document detailing strategy for presenting results to C-Levels from customer is here.:

6.1 Summary Reports

6.2 Detail Reports

CHAPTER 7

Troubleshooting

7.1 Debug Logging

Debug level logging can be enabled for the tool. The results will be written to the file */var/log/secops/assassin.py*.

7.2 Shodan Errors

- Shodan error: HTTP Error 404: Not Found
  - If you see *Processing host:* followed by a 404 error from

  Shodan, this means there is a DNS entry but the host is not up or does not exist. It could also mean the ports on the target host are being protected by a firewall.
- Shodan error: HTTP Error 503: Service Unavailable

You might try disabling Global Protect to get your local machine to properlay access Shodan.

7.3 Reporting Issues/Requesting Help

Please use the issues tab. in the GitHub repo for this tool to report issues. Include a detailed description of the problem as well as the debug log referenced above.