CAPA

installation

Download standalone binary from github.com/mandiant/capa/releases

Install Python library

- Download capa rules from github.com/mandiant/capa-rules/releases
- Download FLIRT signatures from github.com/mandiant/capa/tree/master/sigs
- `> pip install flare-capa`

Install source code

- `> git clone git@github.com:mandiant/capa.git /local/path/to/src`
- `> cd /local/path/to/src ; git submodule update --init rules`
- `> pip install -e /local/path/to/src`

show-features

Display extracted features

```
> python capa/scripts/show-features.py /path/to/sample -s /path/to/sigs
```

Display extracted features for specific function

```
> python capa/scripts/show-features.py /path/to/sample -s /path/to/sigs
--function <address>
```

show-capabilities-by-function

Display capabilities grouped by function

```
> python capa/scripts/show-capabilities-by-function.py /path/to/sample -s
/path/to/sigs -r /path/to/rules
```

usage

Run with default output mode

```
> capa /path/to/sample
```

Run with `verbose` output mode

```
> capa /path/to/sample -v
```

Run with `vverbose` output mode

```
> capa /path/to/sample -vv
```

Run using only rules matching a given metadata value

```
> capa /path/to/sample -t <meta_value>
```

Run using a custom rules directory

```
> capa /path/to/sample -r /path/to/rules
```

Print additional options

```
> capa -h
```

capafmt

Format rule as required by the linter

```
> python capa/scripts/capafmt.py /path/to/rule
```

Format rule in-place

```
> python capa/scripts/capafmt.py /path/to/rule -i
```

FLARE

updated 10/03/2023

MANDIANT
NOW PART OF Google Cloud

CAPA

FLARE

rule format

YAML files with top-level element **rule** and two required children elements, **meta** and **features**

- **meta**: contains metadata that identifies rule, groups it via a namespace, and provides references to additional documentation
- **features**: declares logical statements about the features that must exist for rule to match

For additional information, see https://github.com/mandiant/capa-rules/blob/master/doc/format.md

structural expressions

and: match all children expressions

or: match at least one child expression

not: match when child expression does not

***n* or more**: match at least *n or more* children expressions

- **optional**: match *0 or more* children expressions

match: match on other rule matches or namespaces

scopes

file: high level conclusions, like encryptor, backdoor, or statically linked with specific library

function: collection of API calls, constants

basic block: closely related instructions

instruction: specific combination of mnemonics, operands, constants

(global): features available at every scope, like architecture or operating system

features and characteristics

Features are extracted from multiple scopes, starting with most specific (instruction), and working towards most general (file / global)

Characteristics are one-off features that may represent unique or interesting functionality

file

(sub)string	function-name	section
export	namespace	class
import	forwarded export	embedded pe
mixed mode		

function

loop	calls from	calls to
recursive call		

basic block

tight loop	stack string

instruction

namespace	(sub)string	operand
class	bytes	number
api	offset	mnemonic
property	cross section flow	fs access
nzxor	indirect call	gs access
peb access	call $+5	unmanaged call

(global)

os	format	arch

updated 10/03/2023

MANDIANT
NOW PART OF Google Cloud