Classified Documentation

Release 1.3.0

Wijnand Modderman-Lenstra

October 13, 2014

Contents

1 Requirements **3**

2 Requirements (optional) **5**

3 Table Of Contents **7**
- 3.1 Getting started . . . 7
- 3.2 Reports . . . 8
- 3.3 Configuration . . . 9

4 Indices and tables **17**

Classified is a fast forensic tool that aids in scanning for sensitive data, such as unencrypted PAN (Primary Account Number) data, passwords, network traffic dumps, and so on. You can use this utility to assist in getting and maintaining PCI DSS compliance.

CHAPTER 1

Requirements

Classified is suitable for Python 2.6 - Python 2.7. With little effort it could be ported to Python 3.x as well.

Required:

- Python 2.6 - 2.7
- python-magic, for mime type detection

The current reporting code will not work on Python version 2.4 or 2.5, because we rely on PEP 3101 compatible string formatting.

CHAPTER 2

Requirements (optional)

Optionally, install:

- backports.lzma, to inspect LZMA compressed files and archives
- rarfile, to inspect RAR archives

CHAPTER 3

Table Of Contents

3.1 Getting started

3.1.1 Using pip

The easiest way to install classified is to use pip:

```
~ $ sudo pip install classified
Downloading/unpacking classified
  Downloading classified-1.3.0.tar.gz
  Running setup.py egg_info for package classified

Installing collected packages: classified
  Running setup.py install for classified
    building 'classified._platform' extension

    changing mode of /usr/bin/classified to 755
Successfully installed classified
Cleaning up...
```

3.1.2 On Linux, using Debian (wheezy) or Ubuntu

Firstly, install the required dependancies:

```
~ $ sudo apt-get install python-magic python-lzma python-jinja2
...
```

Grab a copy of the `rarfile` module from PyPi and install it:

> https://pypi.python.org/pypi/rarfile

Now you can install classified:

```
~ $ wget https://pypi.python.org/packages/source/c/classified/classified-1.3.0.tar.gz
~ $ tar -xzf classified-1.3.0.tar.gz
~ $ cd classified-1.3.0
classified-1.3.0 $ sudo python setup.py install
```

3.2 Reports

There are several reporting options available. The report format is chosen as a command line option, see the report modules documentation below for more information.

3.2.1 Variables

The templates use Jinja2 formatting, the report engine has globally available variables. The probes may also export probe-specific variables.

fqdn Full qualified domain name of the system.

filename Filenames discovered in all the probes.

hostname Hostname of the system.

user The name of the effective user identifier (euid).

username Usernames discovered in all the probes.

probe Iterable results from the probes.

3.2.2 Available reports

Documentation on probes:

Report: HTML

The report collects all the results in a single HTML page. The page uses a Jinja2 template, which can be overridden.

Configuration

template Path to the template file.

Report: Mail

The report collects all the results in a single e-mail. The page uses a Jinja2 template, which can be overridden.

Configuration

sender Envelope sender.

server Address or hostname of the SMTP server.

subject Subject of the message.

template Path to the template file.

Report: Syslog

The report collects all the results to syslog as they come in. You can specify a report format per probe. See the example configuration for examples.

Configuration

format_* Per-probe format strings.

syslog_facility Syslog facility, see syslog(3) for more information.

3.3 Configuration

The configuration uses INI-style syntax. The configuration sections and options are case sensitive.

3.3.1 Configuration option types

string

String options can be bare words, single or double quoted strings.

numeric

Numeric options ca be long intergers or floating point numbers.

boolean

Boolean options can be specified as follows.

Valid `true` values are:

- `true`
- `yes`
- `on`
- `1`

Valid `false` values are:

- `false`
- `no`
- `off`
- `0`

3.3.2 Default section

The global configuration is defined under the `[DEFAULT]` section.

DEFAULT.db_path

Path where various database files can be stored. The value can be used in other sections if referenced by `%(db_path)s`.

3.3.3 Other sections

Other configuration sections have their own documentation:

Scanner configuration

The scanner takes care of running the actual probes.

Scanner options

These options are configurable under the `[scanner]` configuration section.

scanner.deflate

If enabled, the scanner will use all available decompression techniques to descend into (tar, rar, zip) archives. It will transparently decompress files.

Note: This functionality highly depends on the availability of optionally installed decompression libraries for Python.

---

scanner.deflate_limit

Size limit for archived files (in bytes).

scanner.include_probes

List of enable probe types.

scanner.exclude_link

If enabled, symlinks will be ignored globally.

scanner.exclude_dirs

List of excluded directory names. The directory name can be either a full path or a glob.

Example:

```
[scanner]
exclude_dirs = /tmp
               /home/*/tmp
```

scanner.exclude_fs

List of excluded file system types. The file system type can be a glob.

Example:

```
[scanner]
exclude_fs = tmpfs
             ext?fs
```

scanner.exclude_type

List of excluded mime types. This mime type can be a glob.

Example:

```
[scanner]
exclude_type = text/html
               application/*
```

scanner.mindepth

Minimal file system recursion depth, set to `-1` to disable.

scanner.maxdepth

Maximal file system recursion depth, set to `-1` to disable.

scanner.incremental

If enabled, only scan files that have changed. See below for the incremental configuration.

Incremental

These options are configurable under the `[incremental]` configuration section.

The scanner allows you to run in incremental mode, skipping files that have been scanned previously:

incremental.database

Path to the dbm cache files.

Example:

```
[incremental]
database = %(db_path)s/incremental.db
```

incremental.algorithm

Selected checksum algorithm, available options are:

Algorithm	Description
`mtime`	Do not compare file contents, use the file modification time.
`adler32`	Adler-32 checksum algorithm, 16 bit.
`crc32`	Cyclic Redundancy Check, 32 bit.
`md5`	MD5 Message Digest, 128 bit.
`sha1`	SHA-1 Cryptographic Hash, 160 bit.
`sha224`	SHA-2 Cryptographic Hash, 224 bit.
`sha256`	SHA-2 Cryptographic Hash, 256 bit.
`sha384`	SHA-2 Cryptographic Hash, 384 bit.
`sha512`	SHA-2 Cryptographic Hash, 512 bit.

Clean false positives

These options are configurable under the `[clean]` configuration section.

You can specify a clean section per probe, to skip false positives. You can do this by either specifying checksums for files to skip, or you can skip file name patterns using globs.

clean.algorithm

Default checksum algorithm used by the clean operations. Used if the probe-specific section has no algorithm configured.S see `incremental.algorithm` for an overview of available algorithms.

clean.context

Default context to use for specifying clean operations, valid options are:

Option	Description
`file`	Checksum the whole file.
`line`	Checksum the matching line.
`format`	Checksum the formatted result, requires `clean.format` to be set.

clean.*.ignore_hash

Ignores content from the configured `clean.context` that matches the checksum configured in `clean.algorithm`.

clean.*.ignore_name

Ignores filenames that match the list of path globs.

clean.*.ignore_repo

Ignores files that are stored in a version control repository. This is a list of key-value pairs, stored as `repository_type:path glob`. Supported repository types are:

Type	Description
arch	GNU Arch repository.
bzr	Bazaar repository.
cvs	CVS or CVSINFO repository.
darcs	DARCS repository.
git	Git repository or bare repository.
hg	Mercurial repository.
monontone	Monotone repository.
rcs	RCS repository.
svn	Subversion repository or subversion checkout.

Example

An example configuration for per-probe clean operations may be as follows:

```
[clean:pan]
algorithm   = sha1
context     = line
ignore_hash = # The following SHA1 checksums appear in the (Debian)
              # openssh-blacklist package and are false positives
              25aafa4ee3132e56cc546bea0978408adcf93e4b  # blacklist.RSA-4096
              385fbbe7ed554bc62fc26880d657584f679595fc  # blacklist.DSA-1024
              513f8822b16bbb5e0761d241d9f8dd5be25dd686  # blacklist.RSA-4096
              5f7de0813134057412ad8e3210a447310c49d0cd  # blacklist.RSA-2048
              5fa84fb55b7c3670b7117763858f21e89aabfb3a  # blacklist.DSA-1024
              6291e6fd865ed2518138c1bef4fdee5d354f735e  # blacklist.DSA-1024
              7cb6ac88eb2d3022e4ad4d6c29b5649e86c3c927  # blacklist.RSA-4096
              8abea0ce82f30ec53c4b71fe6b623790e58b9714  # blacklist.RSA-2048
              8ebc560b38f3f49d34fac44c23a6840b4c9ad45a  # blacklist.RSA-1024
              989288e4e077043545f7c5a6e3bc1c9fd29cdd42  # blacklist.DSA-1024
              9d30bee3aa225289187e56e92f2b830b891680ca  # blacklist.RSA-1024
              a4913bdef39174229f749b835e29d9ccff0003af  # blacklist.RSA-2048
              a5e3cc59ac5759aba8b29e1ffca9c49979d505cf  # blacklist.RSA-2048
              a908941f167a2ec96a56784d9dc6eb71d3705aaa  # blacklist.RSA-4096
              e2cbb90c60d7d2b61c34b9e43f9fb7ba9ea603d4  # blacklist.DSA-1024
              e9e17d0c00992e7418c9491dd5669f364c55ebb9  # blacklist.RSA-1024
              edf70456d1f98bb30e62713f3669afbb21421ffb  # blacklist.RSA-4096
              f3a17cd5676efcdf5755519a1253b469a4f2132b  # blacklist.RSA-2048
              f71117a3513a7b59b1024675f808bf6bd0416cf7  # blacklist.RSA-1024
              824248e0f8c50bf57ebe587f66c4347f6220de28  # blacklist.RSA-1024

[clean:pcap]
context     = file

[clean:ssl]
algorithm   = sha1
context     = file
ignore_name = /etc/ssl/private/*                      # Debian
              /etc/ssl/certs/*                        # Red Hat
ignore_hash = 0000000000000000000000000000000000000000  # Test hash
```

```
              c7f8cfcd962fc09c653555723639feacdc9c0ced  # Found in testdata/key-dsa
              ffffffffffffffffffffffffffffffffffffffff  # Test hash

[clean:password]
ignore_name = /etc/*
              /usr/local/etc/*
ignore_hash = 0000000000000000000000000000000000000000  # Test hash
              23a7753c047eebdc57c2927856ae497c7655d240  # Found in testdata/.pgpass
              ffffffffffffffffffffffffffffffffffffffff  # Test hash
ignore_repo = git:/usr/local/git/*
```

Probes

Configuration

The `[probe]` section is a mapping between mime type mappings (globs) and probes. The probes themselves have a per-probe configuration section, identified as `[probe:<name>]`. See the probe documentation for possible configration options.

Available probes

Documentation on probes:

Probe: Primary Account Numbers (PAN)

About The Primary Account Number (PAN) or Band Card Number are found on payment cards, such as credit cards and debit cards.They have a certain amount of internal structure and share a common numbering scheme. Bank card numbers are allocated in accordance with ISO/IEC 7812.

Configuration

`probe.pan.ignore`
List of hexadecimal characters that are ignored in between sequences of potential PAN characters. You may chose to ignore characters such as NULL, space or other whitespace characters.

`probe.pan.format`

Default reporting format. Available format options:

Option	Description
`card_number`	The full credit card number.
`card_number_masked`	The masked credit card number, suitable for printing in reports.
`company`	Company that issued the credit card number.
`filename`	Full path to the file.
`filename_relative`	Path to the file relative to the current working directory.
`line`	Line number of find.

`probe.pan.limit`

The maximum number of findings reported per file. Set to 0 to disable the limit.

Reference documents

- PCI-DSS v2.0 published October, 2010
- *ISO/IEC 7812-1:2006*, Identification cards, Identification of issuers, Part 1: Numbering system
- *ISO/IEC 7812-2:2007*, Identification cards, Identification of issuers, Part 2: Application and registration procedures
- US patent 2950048, Computer for verifying numbers
- List of issuer identification numbers

Merchant Reference documents

- Maestro Global Rules, published November 9, 2012
- VISA PAN truncation best practices, published July 14, 2010
- VISA Best Practices for Tokenization Version, published July 14, 2010

Probe: Password

About The password probe scans for stored (plain text) passwords.

Configuration
`probe.password.format`
Default reporting format, available options:

Option	Description
`filename`	Full path to the file.
`filename_relative`	Path to the file relative to the current working directory.
`line`	Line number of finding.
`password`	Password as discovered.
`password_masked`	Masked password as discovered, suitable for reporting.
`text`	Full line of the finding.
`text_masked`	Full line of the finding with the password masked, suitable for reporting.
`pattern`	Regular expression that scans for passwords. The regular expression is a Python-compatible regular expression and must include at least a `password` capture group.

Probe: Packet Capture files (PCAP)

About This probe may identify pcap dump files.

Configuration
`probe.pcap.format`
Default reporting format. Available options:

Option	Description
`filename`	Full path to the file.
`filename_relative`	Path to the file relative to the current working directory.
`linktype`	Link type of the packet capture file.
`line`	Line number of find.
`version`	Packet capture file version.

Probe: Secure Sockets Layer (SSL)

About The Secure Sockets Layer (SSL) probe scans for cryptographic private keys, that are either not properly secured or have no passphrase set.

Configuration

`probe.ssl.format`

Default reporting format. Available options:

Option	Description
`filename`	Full path to the file.
`filename_relative`	Path to the file relative to the current working directory.
`gid`	Numeric group identifier.
`key_info`	Information about the discovered key.
`key_type`	Type of the discovered key.
`line`	Line number of find.
`username`	Name of the user that owns the file.
`uid`	Numeric user identifier.

CHAPTER 4

Indices and tables

- *genindex*
- *modindex*
- *search*

B

boolean
    command line option, 9

C

clean.algorithm, 11
clean.context, 11
clean.format, 11
command line option
    boolean, 9
    numeric, 9
    string, 9

E

environment variable
    clean.*.ignore_hash, 11
    clean.*.ignore_name, 11
    clean.*.ignore_repo, 12
    clean.algorithm, 11
    clean.context, 11
    clean.format, 11
    DEFAULT.db_path, 9
    incremental.algorithm, 11
    incremental.database, 11
    probe.pan.format, 13
    probe.pan.ignore, 13
    probe.pan.limit, 13
    probe.password.format, 14
    probe.pcap.format, 14
    probe.ssl.format, 15
    scanner.deflate, 10
    scanner.deflate_limit, 10
    scanner.exclude_dirs, 10
    scanner.exclude_fs, 10
    scanner.exclude_link, 10
    scanner.exclude_type, 10
    scanner.include_probes, 10
    scanner.incremental, 10
    scanner.maxdepth, 10
    scanner.mindepth, 10

I

incremental.algorithm, 11

N

numeric
    command line option, 9

S

string
    command line option, 9