# Conpot Documentation

*Release 0.6.0*

MushMush Foundation

Jul 31, 2019

# Contents

**1 Installation** **3**
- 1.1 Quick Installation using Docker . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
- 1.2 Installation on host using Virtualenv . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

**2 Conpot concepts** **5**
- 2.1 Databus . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
- 2.2 ConpotFS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
- 2.3 Internal Interface . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
- 2.4 Protocols . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
- 2.5 Proxy Mode . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
- 2.6 Templates . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

**3 Developmental guidelines** **7**
- 3.1 Development Guidelines . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

**4 Usage and Frequently asked questions** **11**
- 4.1 Frequently Asked Questions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

**5 API reference** **13**
- 5.1 API Reference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

**Python Module Index** **73**

**Index** **75**

Conpot is an ICS honeypot with the goal to collect intelligence about the motives and methods of adversaries targeting industrial control systems.

CHAPTER 1

# Installation

Basics instruction on how to install Conpot:

There are two ways of multiple ways of installing conpot. If you are just tinkering around, it is recommended that you use the quick install method. On the other hand, if you are an advanced user, you should do host installation via *pip*. This is described as quick install.

## 1.1 Quick Installation using Docker

## 1.2 Installation on host using Virtualenv

A generic way to keep Python installations separate is using virtualenv. This way you can run conpot on your machine without littering your machine. This guides assumes you have Python 3.6 installed and running on your computer.

Note that this is also the recommended way of installing conpot on a machine. Installation can be done as follows:-

Install dependencies:

```
apt-get install git libsmi2ldbl smistrip libxslt1-dev python3.5-dev libevent-dev␣
↪default-libmysqlclient-dev
```

Create the virtualenv

```
virtualenv --python=python3.5 conpot
```

Activate the environment

```
source conpot/bin/activate
```

Upgrade any basic tools in the environment and deps

```
pip install --upgrade pip
pip install --upgrade setuptools
pip install cffi
```

Install the table version of Conpot from PyPI:

```
pip install conpot
```

# CHAPTER 2

# Conpot concepts

<todo: add some data here>

## 2.1 Databus

## 2.2 ConpotFS

ConpotFS designed to have “safe to use” *os.** wrappers that could be used by protocols. We cannot allow *chmod()* like commands that may allow attackers to make arbitrary system calls.

At the same time - protocols such as FTP need *chmod()* like methods. Same goes for *stat()* etc. For this reason, we needed a file system that can operate on a layer above the actual file system and still provide the flexibility/robustness.

The Conpot’s file system solves this problem by proxying the actual files kept at a controlled location.

```
+---------------+                      +----------------------+
|               |                      |                      |
|               | <----------------+   |  Actual FileSystem   |
|  Conpot VFS   |       Proxy          |         at           |
|               | +---------------->   |  '/tmp/__conpot__*/' |
|               |                      |                      |
+---------------+                      +----------------------+
```

Consequently, we would keep a cache (a dictionary where we would store all file related data - (information regarding access, permissions, owners, stat etc.). Note that no matter what, we won’t change the actual permissions of the file system.

For the sake of demo, consider the following:

This is what a typical *ls -la* for a user *1337honey* looks like:

```
total 8
drwxrwxr-x 2 1337honey 1337honey 4096 Jul  9 01:20 .
```

(continues on next page)

(continued from previous page)

```
drwxrwxr-x 4 1337honey 1337honey 4096 Jul  9 01:17 ..
-rw-rw-r-- 1 1337honey 1337honey    0 Jul  9 01:20 hacked.png
```

Notice the permissions and the user/group.

```
>>> import conpot.core as conpot_core
>>> conpot_core.initialize_vfs('.', data_fs_path='../data_fs')
>>> vfs = conpot_core.get_vfs()
>>> vfs.listdir('.')
['hacked.png']
>>> [print(i) for i in vfs.format_list('', vfs.listdir('.'))]
rwxrwxrwx   1 root      root            0 Jul 08 19:53 hacked.png
```

As you can see, the permissions have changed and so have the user/groups(By default the *uid:gid* is *0:0* and permissions is *777* - this is configurable). This is not all. Check this out!

```
>>> vfs.register_user('attacker', 2000)
>>> vfs.create_group('attacker', 3000)
>>> vfs.chown('/', uid=2000, gid=3000, recursive=True)
>>> vfs.chmod('/', 0o755, recursive=True)
>>> [print(i) for i in vfs.format_list('', vfs.listdir('.'))]
rwxr-xr-x   1 attacker   attacker          0 Jul 08 19:53 hacked.png
```

There is no change with the uid:gid:perms of the actual *'hacked.png'* file though.

Another big advantage of this approach is : VFS is independent of the physical storage media it is located in. We are currently keeping the contents in '/tmp'. But in future if we want to replace this with somewhat better storage media(or location), we can simply detach the VFS - replace it with new storage media URL and it'll fit right in.

## 2.3 Internal Interface

## 2.4 Protocols

## 2.5 Proxy Mode

## 2.6 Templates

CHAPTER 3

# Developmental guidelines

<todo: add some data here>

## 3.1 Development Guidelines

### 3.1.1 Developers Guide

#### Indentation

- We are using 4 tab-spaces
- No one line conditionals

#### Style

- We obey to the PEP8

#### Copyright

- If you are adding a file/code which is produced only by you, feel free to add the license information and a notice who holds the copyrights.

#### Environment

- PyCharm is recommended.

## Recommended git workflow

### For contributors

0, You can do this step when you are on master, or feature_branch, anytime there are new commits in original project.

Just one-time add of remote:

```
git remote add mushorg https://github.com/mushorg/conpot.git
```

And rebase:

```
git fetch mushorg
git rebase mushorg/master feature_branch
```

This way, your feature_branch or master will be up-to-date.

1, For every feature, create new branch:

```
git checkout -b feature_branch
```

2, State what you do in commit message.

When you create pull request and get review, it is recommended to edit your original commits.

3a, If you want to change the last commit:

```
(make some changes in files)
git add file1 file2
git commit --amend
```

3b, If you want to change any of your previous commits:

```
git rebase -i HEAD~3  (can be HEAD~4, depends which commit you want to change, or you␣
↪can type hash of previous commit)
```

change “pick” to “e”:

```
e e88a2f1 commit 1
pick bfd57e4 commit2
```

and save.

```
(make some changes in files)
git add file1 file2
git rebase --continue
```

Warning: Do not use ‘git commit’ in rebase if you don’t know what you are doing.

4, Look at your changes, and git force push to your branch:

```
git push -f feature_branch
```

5, Comment in pull request to let us know about your new code.

### For maintainers

To avoid additional Merge commits, use cherry-pick:

```
git checkout master
git remote add user https://github.com/user/conpot.git
git fetch user
(look at 'git log user/feature_branch')
git cherry-pick commit_hash
git push origin master
git remote rm user
```

Comment on pull request that you added it to master, and close pull request.

This approach is usefull for majority of pull requests (1-3 commits).

If you expect conflicts (a lot of commits in feature branch with a lot of changes) you can use GitHub Merge button.

Revert will be easier too.

Conflicts should not happen, if feature branch is rebased on current master.

# CHAPTER 4

# Usage and Frequently asked questions

<todo: add some data here>

## 4.1 Frequently Asked Questions

### 4.1.1 Sharing Data

**With whom do we share?**

Everyone who is interested and potentially shares data, results or helps improving the tool.

**What's the data volume?**

Conpot has build-in support for HPFeeds, a generic data sharing protocol we are using in the Honeynet Project. This means that potentially we are going to get all the data from every sensor with HPFeeds enabled.

Right now there is only a very small number of deployed sensors. HPFeeds is not enabled by default and probably nobody is using a HMI to attract adversaries yet. So if you are lucky you will see an event every other day. We know that with a HMI the traffic will be significantly higher as your sensor will be found using search engines.

**What is the data format?**

Raw data in JSON formatting.

**How do I get the data?**

There is a Python client which uses the HPFeeds library. About 40 lines of code. From there it's quite easy to write the data to a database. You can find an explanation on how it works here.

**What do I have to do?**

If you want to have access to the Conpot data, you have to create a HPFriends account. As soon as you accept the share, you can create an authkey. You can modify the client with the auth keys credentials. The client should be self explaining. You can extend the client so it fits your needs (e.g. logging to a database).

**How do I test this?**

As soon as you have Conpot set-up it should be easy to create some traffic for testing.

CHAPTER 5

# API reference

## 5.1 API Reference

### 5.1.1 conpot package

#### Subpackages

#### conpot.core package

#### Subpackages

#### conpot.core.loggers package

#### Submodules

#### conpot.core.loggers.helpers module

`conpot.core.loggers.helpers.`**`json_default`**(*obj*)

#### conpot.core.loggers.hpfriends module

**`class`** `conpot.core.loggers.hpfriends.`**`HPFriendsLogger`**(*host*, *port*, *ident*, *secret*, *channels*)

Bases: `object`

**`log`**(*data*)

### conpot.core.loggers.json_log module

**class** conpot.core.loggers.json_log.**JsonLogger**(*filename*, *sensorid*, *public_ip*)

Bases: object

**log**(*event*)

**log_session**(*session*)

### conpot.core.loggers.log_worker module

**class** conpot.core.loggers.log_worker.**LogWorker**(*config*, *dom*, *session_manager*, *public_ip*)

Bases: object

**start**()

**stop**()

### conpot.core.loggers.mysql_log module

### conpot.core.loggers.sqlite_log module

**class** conpot.core.loggers.sqlite_log.**SQLiteLogger**(*db_path='logs/conpot.db'*)

Bases: object

**log**(*event*)

**log_session**(*session*)

**select_data**()

### conpot.core.loggers.stix_transform module

**class** conpot.core.loggers.stix_transform.**StixTransformer**(*config*, *dom*)

Bases: object

**transform**(*event*)

### conpot.core.loggers.syslog module

**class** conpot.core.loggers.syslog.**SysLogger**(*host*, *port*, *facility*, *logdevice*, *logsocket*)

Bases: object

**log**(*data*)

### conpot.core.loggers.taxii_log module

**class** conpot.core.loggers.taxii_log.**TaxiiLogger**(*config*, *dom*)

Bases: object

**log**(*event*)

### Module contents

### Submodules

### conpot.core.attack_session module

**class** conpot.core.attack_session.**AttackSession**(*protocol*, *source_ip*, *source_port*, *destination_ip*, *destination_port*, *databus*, *log_queue*)

Bases: object

**add_event**(*event_data*)

**dump**()

**set_ended**()

### conpot.core.databus module

**class** conpot.core.databus.**Databus**

Bases: object

**get_shapshot**()

**get_value**(*key*)

**initialize**(*config_file*)

**notify_observers**(*key*)

**observe_value**(*key*, *callback*)

**reset**()

**set_value**(*key*, *value*)

### conpot.core.filesystem module

**class** conpot.core.filesystem.**AbstractFS**(*src_path: str*, *create_mode: int = 511*, *temp_dir: Optional[str] = None*, *identifier: Optional[str] = '__conpot__'*, *auto_clean: Optional[bool] = True*, *ignore_clean_errors: Optional[bool] = True*)

Bases: fs.wrapfs.WrapFS

AbstractFS distinguishes between "real" filesystem paths and "virtual" ftp paths emulating a UNIX chroot jail where the user can not escape its home directory (example: real "/home/user" path will be seen as "/" by the client)

This class exposes common fs wrappers around all os.* calls involving operations against the filesystem like creating files or removing directories (such as listdir etc.)

*Implementation Note:* When doing I/O - Always with the check_access and set_access context managers for safe operations.

**access**(*path: str*, *name_or_id: Union[int, str] = None*, *required_perms: str = None*)

Returns bool w.r.t the a user/group has permissions to read/write/execute a file. This is a wrapper around os.access. But it would accept name or id instead of of just ids. Also it can accept required permissions in the form of strings rather than os.F_OK, os.R_OK, os.W_OK etc.

*Implementation Note*: First we would check whether the current user has the required permissions. If not, then we check the group to which this user belongs to. Finally if the user's group also does not meet the perms we check for other permissions.

**add_users_to_group** (*gid: int, uids: List[T]*) → None

Add list of users to an existing group :param gid: Group id of the group. :param uids: List of registers users that belong to this group

**check_access** (*path=None, user=None, perms=None*)

Checks whether the current user has permissions to do a specific operation. Raises FSOperationNotPermitted exception in case permissions are not satisfied. Handy utility to check whether the user with uid provided has permissions specified. Examples:

```
>>> import conpot.core as conpot_core
>>> _vfs, _ = conpot_core.get_vfs('ftp')
>>> with _vfs.check_access(path='/', user=13, perms='rwx'):
>>>     _vfs.listdir('/')
```

```
>>> with _vfs.check_access(path='/', user=45, perms='w'):
>>>     with _vfs.open('/test', mode='wb') as _file:
>>>         _file.write(b'Hello World!')
```

**chmod** (*path: str, mode: oct, recursive: bool = False*) → None

Change file/directory mode. :param path: Path to be modified. :param mode: Operating-system mode bitfield. Must be in octal's form. Eg: chmod with (mode=0o755) = Permissions(user='rwx', group='rx', other='rx') :param recursive: If the path is directory, setting recursive to true would change permissions to sub folders and contained files. :type recursive: bool

**chown** (*fs_path: str, uid: int, gid: int, recursive: Optional[bool] = False*) → None

Change the owner of a specified file. Wrapper for os.chown :param fs_path: path or directory in the VFS where chown would be executed. :param uid: The *uid* of the user. **User must be a registered user on the filesystem or an exception would be thrown. :param gid: The *gid* of the group **Group must be a registered group on the filesystem or an exception would be thrown. :param recursive: If the given path is directory, then setting the recursive option to true would walk down the tree and recursive change permissions in the cache.

** *fs_path* needs to be the absolute path w.r.t to the vfs. If you are in a sub file system, please use *subvfs.getcwd()* to get the current directory. **

**clean** ()

Clean (delete) temporary files created by this filesystem.

**copy** (*src_path, dst_path, overwrite=False*)

Copy file contents from `src_path` to `dst_path`.

**Arguments:** src_path (str): Path of source file. dst_path (str): Path to destination file. overwrite (bool): If *True*, overwrite the destination file

if it exists (defaults to *False*).

**Raises:**

**fs.errors.DestinationExists: If `dst_path` exists,** and `overwrite` is *False*.

**fs.errors.ResourceNotFound: If a parent directory of** `dst_path` does not exist.

**create_group** (*name: str, gid: int*) → None

Store all group related data for the file system. :param name: Name of the group :param gid: gid of the group

**create_jail**(*path*)
Returns chroot jail sub system for a path

**format_list**(*basedir*, *listing*)
Return an iterator object that yields the entries of given directory emulating the "/bin/ls -lA" UNIX command output. This is how output should appear: -rw-rw-rw- 1 owner group 7045120 Sep 02 3:47 music.mp3 drwxrwxrwx 1 owner group 0 Aug 31 18:50 e-books -rw-rw-rw- 1 owner group 380 Sep 02 3:40 module.py

**Parameters**

- **basedir** – (str) must be protocol relative path
- **listing** – (list) list of files to needed for output.

**get_permissions**(*path*)
Get permissions for a particular user on a particular file/directory in 'rwxrx—' format

**getcwd**()

**getfile**(*path*, *file*, *chunk_size=None*, ***options*)
Copies a file from the filesystem to a file-like object.

This may be more efficient that opening and copying files manually if the filesystem supplies an optimized method.

**Arguments:** path (str): Path to a resource. file (file-like): A file-like object open for writing in

binary mode.

**chunk_size (int, optional): Number of bytes to read at a** time, if a simple copy is used, or *None* to use sensible default.

****options: Implementation specific options required to open** the source file.

Note that the file object `file` will *not* be closed by this method. Take care to close it after this method completes (ideally with a context manager).

**Example:**

```
>>> with open('starwars.mov', 'wb') as write_file:
...     my_fs.download('/movies/starwars.mov', write_file)
```

**Note:** Deprecated since version 2.2.0: Please use *~download*

**getinfo**(*path: str*, *get_actual: bool = False*, *namespaces=None*)
Get information about a resource on a filesystem.

**Arguments:** path (str): A path to a resource on the filesystem. namespaces (list, optional): Info namespaces to query

(defaults to *[basic]*).

**Returns:** ~fs.info.Info: resource information object.

For more information regarding resource information, see info.

**getmeta**(*namespace='standard'*)
Get meta information regarding a filesystem.

**Arguments:**

**namespace (str): The meta namespace (defaults** to `"standard"`).

**Returns:** dict: the meta information.

Meta information is associated with a *namespace* which may be specified with the `namespace` parameter. The default namespace, `"standard"`, contains common information regarding the filesystem's capabilities. Some filesystems may provide other namespaces which expose less common or implementation specific information. If a requested namespace is not supported by a filesystem, then an empty dictionary will be returned.

The `"standard"` namespace supports the following keys:

| key | Description |
|---|---|
| case_insensitive | *True* if this filesystem is case insensitive. |
| invalid_path_chars | A string containing the characters that may not be used on this filesystem. |
| max_path_length | Maximum number of characters permitted in a path, or *None* for no limit. |
| max_sys_path_length | Maximum number of characters permitted in a sys path, or *None* for no limit. |
| network | *True* if this filesystem requires a network. |
| read_only | *True* if this filesystem is read only. |
| supports_rename | *True* if this filesystem supports an *os.rename* operation. |

Most builtin filesystems will provide all these keys, and third- party filesystems should do so whenever possible, but a key may not be present if there is no way to know the value.

**Note:** Meta information is constant for the lifetime of the filesystem, and may be cached.

**`getmtime`**(*path*)

Return the last modified time as a number of seconds since the epoch.

**`groups`**

**`listdir`**(*path*)

Get a list of the resource names in a directory.

This method will return a list of the resources in a directory. A *resource* is a file, directory, or one of the other types defined in *~fs.ResourceType*.

**Arguments:** path (str): A path to a directory on the filesystem

**Returns:** list: list of names, relative to `path`.

**Raises:** fs.errors.DirectoryExpected: If `path` is not a directory. fs.errors.ResourceNotFound: If `path` does not exist.

**`makedir`**(*path*, *permissions=None*, *recreate=True*)

Make a directory.

**Arguments:** path (str): Path to directory from root. permissions (~fs.permissions.Permissions, optional): a

*Permissions* instance, or *None* to use default.

**recreate (bool): Set to *True* to avoid raising an error if** the directory already exists (defaults to *False*).

**Returns:** ~fs.subfs.SubFS: a filesystem whose root is the new directory.

**Raises:** fs.errors.DirectoryExists: If the path already exists. fs.errors.ResourceNotFound: If the path is not found.

**`mount_fs`**(*dst_path: str*, *fs_url: str = None*, *owner_uid: Optional[int] = 0*, *group_gid: Optional[int] = 0*, *perms: Union[fs.permissions.Permissions*, *int*, *None] = 493*) → fs.subfs.SubFS

To be called to mount individual filesystems. :param fs_url: Location/URL for the file system that is to be mounted. :param dst_path: Place in the Conpot's file system where the files would be placed. This should be relative to FS root. :param owner_uid: The owner *user* **UID** of the directory and the sub directory. Default is root/ :param group_gid: The group 'group' to which the directory beings. Defaults to root. :param perms: Permission UMASK

**move** (*src_path*, *dst_path*, *overwrite=False*)

Move a file from `src_path` to `dst_path`.

**Arguments:** src_path (str): A path on the filesystem to move. dst_path (str): A path on the filesystem where the source

file will be written to.

**overwrite (bool): If *True*, destination path will be** overwritten if it exists.

**Raises:**

**fs.errors.FileExpected: If `src_path` maps to a** directory instead of a file.

**fs.errors.DestinationExists: If `dst_path` exists,** and `overwrite` is *False*.

**fs.errors.ResourceNotFound: If a parent directory of** `dst_path` does not exist.

**norm_path** (*path*)

**open** (*path*, *mode='r'*, *buffering=-1*, *encoding=None*, *newline="*, *line_buffering=False*, ***options*)

Open a file.

**Arguments:** path (str): A path to a file on the filesystem. mode (str): Mode to open the file object with

(defaults to *r*).

**buffering (int): Buffering policy (-1 to use** default buffering, 0 to disable buffering, 1 to select line buffering, of any positive integer to indicate a buffer size).

**encoding (str): Encoding for text files (defaults to** `utf-8`)

**errors (str, optional): What to do with unicode decode errors** (see *codecs* module for more information).

newline (str): Newline parameter. **options: keyword arguments for any additional information

required by the filesystem (if any).

**Returns:** io.IOBase: a *file-like* object.

**Raises:** fs.errors.FileExpected: If the path is not a file. fs.errors.FileExists: If the file exists, and *exclusive mode*

is specified (`x` in the mode).

fs.errors.ResourceNotFound: If the path does not exist.

**openbin** (*path*, *mode='r'*, *buffering=-1*, ***options*)

Open a file in the ConpotFS in binary mode.

**opendir** (*path*, *factory=<class 'conpot.core.fs_utils.SubAbstractFS'>*)

Get a filesystem object for a sub-directory.

**Arguments:** path (str): Path to a directory on the filesystem. factory (callable, optional): A callable that when invoked

with an FS instance and `path` will return a new FS object representing the sub-directory contents. If no `factory` is supplied then *~fs.subfs.SubFS* will be used.

**Returns:** ~fs.subfs.SubFS: A filesystem representing a sub-directory.

**Raises:**

**fs.errors.DirectoryExpected: If `dst_path` does not** exist or is not a directory.

**`readlink`**(*path*)

Perform a readlink() system call. Return a string representing the path to which a symbolic link points. :param path: (str) must be protocol relative path

**`register_user`**(*name: str*, *uid: int*) → None

Store all user related data for the file system.

**`remove`**(*path*)

Remove a file from the file system.

**`removedir`**(*path*, *rf=True*)

Remove a directory from the file system. :param path: directory path :param rf: remove directory recursively and forcefully. This removes directory even if there is any data in it. If set to False, an exception would be raised

**`root`**

The root directory - where the filesystem is stored

**`setbinfile`**(*path*, *file*)

Set a file to the contents of a binary file object.

This method copies bytes from an open binary file to a file on the filesystem. If the destination exists, it will first be truncated.

**Arguments:** path (str): A path on the filesystem. file (io.IOBase): a file object open for reading in

binary mode.

**chunk_size (int, optional): Number of bytes to read at a** time, if a simple copy is used, or *None* to use sensible default.

****options: Implementation specific options required to open** the source file.

Note that the file object `file` will *not* be closed by this method. Take care to close it after this method completes (ideally with a context manager).

**Example:**

```
>>> with open('~/movies/starwars.mov', 'rb') as read_file:
...     my_fs.upload('starwars.mov', read_file)
```

**Note:** Deprecated since version 2.2.0: Please use ~*upload*

**`setinfo`**(*path*, *info*)

Higher level function to directly change values in the file system. Dictionary specified here changes cache values. :param path: path of the file that is to be changed :param info: Raw Info object. Please check pyfilesystem2's docs for more info.

**`settimes`**(*path*, *accessed=None*, *modified=None*)

Set the accessed and modified time on a resource.

**Arguments:** path: A path to a resource on the filesystem. accessed (datetime, optional): The accessed time, or

*None* (the default) to use the current time.

**modified (datetime, optional): The modified time, or** *None* (the default) to use the same time as the `accessed` parameter.

**stat** (*path*)
Perform a stat() system call on the given path. :param path: (str) must be protocol relative path

**take_snapshot** ()
Take snapshot of entire filesystem. :rtype: dict

**user_groups**
gid: {set of uid of users.}

**users**

### conpot.core.fs_utils module

Utils related to ConpotVFS

**exception** `conpot.core.fs_utils.`**FSOperationNotPermitted** (*msg=None*)
Bases: `fs.errors.FSError`

Custom class for filesystem-related exceptions.

**exception** `conpot.core.fs_utils.`**FilesystemError** (*msg=None*)
Bases: `fs.errors.FSError`

Custom class for filesystem-related exceptions.

**class** `conpot.core.fs_utils.`**SubAbstractFS** (*parent_fs*, *path*)
Bases: `fs.subfs.SubFS`, `typing.Generic`

Creates a chroot jail sub file system. Each protocol can have an instance of this class. Use AbstractFS's create_jail method to access this. You won't be able to cd into an *up* directory.

**access** (*path: str*, *name_or_id: Union[int*, *str] = None*, *required_perms: str = None*)

**check_access** (*path=None*, *user=None*, *perms=None*)

**chmod** (*path: str*, *mode: oct*, *recursive: bool = False*) → None

**chown** (*fs_path: str*, *uid: int*, *gid: int*, *recursive: Optional[bool] = False*)

**default_gid**

**default_group**

**default_perms**

**default_uid**

**default_user**

**format_list** (*basedir*, *listing*)

**get_permissions** (*path*)

**getcwd** ()

**getinfo** (*path: str*, *get_actual: bool = False*, *namespaces=None*)
Get information about a resource on a filesystem.

**Arguments:** path (str): A path to a resource on the filesystem. namespaces (list, optional): Info namespaces to query

(defaults to *[basic]*).

**Returns:** ~fs.info.Info: resource information object.

For more information regarding resource information, see info.

**getmtime** (*path*)

**move** (*src_path*, *dst_path*, *overwrite=True*)

Move a file from `src_path` to `dst_path`.

**Arguments:** src_path (str): A path on the filesystem to move. dst_path (str): A path on the filesystem where the source

file will be written to.

**overwrite (bool): If *True*, destination path will be** overwritten if it exists.

**Raises:**

**fs.errors.FileExpected: If `src_path` maps to a** directory instead of a file.

**fs.errors.DestinationExists: If `dst_path` exists,** and `overwrite` is *False*.

**fs.errors.ResourceNotFound: If a parent directory of** `dst_path` does not exist.

**readlink** (*path*)

**remove** (*path*)

Remove a file from the filesystem.

**Arguments:** path (str): Path of the file to remove.

**Raises:** fs.errors.FileExpected: If the path is a directory. fs.errors.ResourceNotFound: If the path does not exist.

**removedir** (*path*, *rf=False*)

Remove a directory from the filesystem.

**Arguments:** path (str): Path of the directory to remove.

**Raises:**

**fs.errors.DirectoryNotEmpty: If the directory is not empty (** see *~fs.base.FS.removetree* for a way to remove the directory contents.).

**fs.errors.DirectoryExpected: If the path does not refer to** a directory.

**fs.errors.ResourceNotFound: If no resource exists at the** given path.

**fs.errors.RemoveRootError: If an attempt is made to remove** the root directory (i.e. `'/'`)

**root**

**stat** (*path*)

`conpot.core.fs_utils.`**copy_files** (*source*, *dest*, *buffer_size=1048576*)

Copy a file from source to dest. source and dest must be file-like objects.

## conpot.core.internal_interface module

**class** `conpot.core.internal_interface.`**DotDict**

Bases: `dict`

**class** `conpot.core.internal_interface.`**Interface**

Bases: `object`

Conpot's internal interface

**enabled**

**class** conpot.core.internal_interface.**Network**
Bases: object

### conpot.core.protocol_wrapper module

conpot.core.protocol_wrapper.**conpot_protocol**(*cls*)

### conpot.core.session_manager module

**class** conpot.core.session_manager.**SessionManager**
Bases: object

**get_session**(*protocol*, *source_ip*, *source_port*, *destination_ip=None*, *destination_port=None*)

**get_session_count**(*protocol=None*)

**initialize_databus**(*config_file*)

**purge_sessions**()

### conpot.core.virtual_fs module

**class** conpot.core.virtual_fs.**VirtualFS**(*data_fs_path=None*)
Bases: object

Conpot's virtual file system. Based on Pyfilesystem2, it would allow us to have arbitrary file uploads while sand boxing them for later analysis. This is how it should look like:

**[_conpot_vfs]**

|– data_fs (persistent) | |– ftp/uploads | '– misc. | '– protocol_fs (temporary, refreshed at startup)

|– common |– telnet |– http |– snmp '– ftp etc.

**Parameters data_fs_path** – Path for storing data_fs. A dictionary with attribute name _protocol_vfs stores all the

fs folders made by all the individual protocols. :type data_fs_path: fs.open_fs

**add_protocol**(*protocol_name: str*, *data_fs_subdir: str*, *vfs_dst_path: str*, *src_path=None*, *owner_uid=0*, *group_gid=0*, *perms=493) -> (<class 'conpot.core.fs_utils.SubAbstractFS'>*, *<class 'fs.subfs.SubFS'>*)
Method that would be used by protocols to initialize vfs. May be called by each protocol individually. This creates a chroot jail sub file system env which makes easier handling. It also creates a data_fs sub file system for managing protocol specific uploads. :param protocol_name: name of the protocol for which VFS is being created. :param data_fs_subdir: sub-folder name within data_fs that would be storing the uploads for later analysis :param vfs_dst_path: protocol specific sub-folder path in the fs. :param src_path: Source from where the files are to copied. :param owner_uid: UID of a registered user. This is the default owner in the sub file system :param group_gid: GID of a existing group. :param perms: Default permissions of the sub file system. :return: fs object

**Note:** The owner_uid and group_gid must be already registered with the fs. Otherwise an exception would be raised.

**close** (*force=False*)
Close the filesystem properly. Better and more graceful than __del__ :param force: Force close. This would close the AbstractFS instance - without close closing data_fs File Systems

**initialize_vfs** (*fs_path=None*, *data_fs_path=None*, *temp_dir=None*)

### Module contents

conpot.core.**add_protocol** (*protocol_name: str*, *data_fs_subdir: str*, *vfs_dst_path: str*, *src_path=None*, *owner_uid: Optional[int] = 0*, *group_gid: Optional[int] = 0*, *perms: Optional[oct] = 493*) → Tuple

conpot.core.**close_fs** ()
Close the file system. Remove all the temp files.

conpot.core.**get_databus** ()

conpot.core.**get_interface** ()

conpot.core.**get_session** (**args*, ***kwargs*)

conpot.core.**get_sessionManager** ()

conpot.core.**get_vfs** (*protocol_name: Optional[str] = None*) → Union[conpot.core.filesystem.AbstractFS, Tuple]
Get the File System. :param protocol_name: Name of the protocol to be fetched

conpot.core.**initialize_vfs** (*fs_path=None*, *data_fs_path=None*, *temp_dir=None*)

## conpot.emulators package

### Subpackages

### conpot.emulators.misc package

### Submodules

### conpot.emulators.misc.random module

**class** conpot.emulators.misc.random.**Random16bitRegister**
Bases: object

**get_value** ()

**class** conpot.emulators.misc.random.**Random8BitRegisters**
Bases: object

**get_value** ()

### conpot.emulators.misc.uptime module

**class** conpot.emulators.misc.uptime.**Uptime** (*started=-1*)
Bases: object

**get_value** ()

### Module contents

### conpot.emulators.sensors package

### Module contents

### Submodules

### conpot.emulators.proxy module

**class** conpot.emulators.proxy.**Proxy**(*name*, *proxy_host*, *proxy_port*, *decoder=None*, *keyfile=None*, *certfile=None*)

Bases: object

**get_server**(*host*, *port*)

**handle**(*sock*, *address*)

**handle_in_data**(*data*, *sock*, *session*)

**handle_out_data**(*data*, *sock*, *session*)

**stop**()

**class** conpot.emulators.proxy.**ProxyDecoder**

Bases: abc.ABC

**decode_in**(*data*)

Decode data that goes into the proxied device

**decode_out**(*data*)

Decode data that goes out from the proxied device to the connected client(attacker).

### Module contents

### conpot.protocols package

### Subpackages

### conpot.protocols.IEC104 package

### Submodules

### conpot.protocols.IEC104.DeviceDataController module

**class** conpot.protocols.IEC104.DeviceDataController.**DeviceDataController**(*template*)

Bases: object

**check_registers**()

**get_object_from_reg**(*obj_addr*)

**get_registers**()

**set_object_val**(*obj_addr*, *val*)

conpot.protocols.IEC104.DeviceDataController.**addr_in_hex**(*address*)

conpot.protocols.IEC104.DeviceDataController.**hex_in_addr**(*hex_addr*)

conpot.protocols.IEC104.DeviceDataController.**inro_response**(*sorted_reg*, *asdu_type*)

### conpot.protocols.IEC104.IEC104 module

**class** conpot.protocols.IEC104.IEC104.**IEC104**(*device_data_controller*, *sock*, *address*, *session_id*)

Bases: object

**disconnect**()

**static get_infoobj_list**(*frame*)

**handle_double_command46**(*container*)

**handle_i_frame**(*frame*)

**handle_inro_command100**(*container*)

**handle_s_frame**(*frame*)

**handle_setpointfloatpoint_command50**(*container*)

**handle_setpointscaled_command49**(*container*)

**handle_single_command45**(*container*)

**handle_u_frame**(*frame*)

**increment_sendseq**()

**recvseq_increment**()

**restart_t1**()

**send_104frame**(*frame*)

**send_frame_imm**(*frame*)

**show_send_list**()

**class** conpot.protocols.IEC104.IEC104.**frame_object_with_timer**(*frame*)

Bases: object

**build**()

**cancel_t1**()

**getfieldval**(*fieldval*)

**restart_t1**()

### conpot.protocols.IEC104.IEC104_server module

### conpot.protocols.IEC104.errors module

**exception** conpot.protocols.IEC104.errors.**FrameError**(**args*)

Bases: Exception

This error is raised if the IEC104 frame is wrong or ain’t a IEC104 packet at all

**exception** conpot.protocols.IEC104.errors.**InvalidFieldValueException**(**args*)

Bases: ValueError

This error is raised if a field value is not allowed

**exception** conpot.protocols.IEC104.errors.**Timeout_t1**

Bases: BaseException

Base class for exceptions in this module.

**exception** conpot.protocols.IEC104.errors.**Timeout_t1_2nd**

Bases: BaseException

Base class for exceptions in this module.

**exception** conpot.protocols.IEC104.errors.**Timeout_t3**

Bases: BaseException

Base class for exceptions in this module.

### conpot.protocols.IEC104.frames module

**class** conpot.protocols.IEC104.frames.**BCR**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.BCR'>, <class 'scapy.packet.Packe**

**fields_desc = [<Field (BCR).Value>, <Field (BCR).IV>, <Field (BCR).CA>, <Field (BCR).C**

**payload_guess = [({}, <class 'scapy.packet.Padding'>)]**

**class** conpot.protocols.IEC104.frames.**BSI**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.BSI'>, <class 'scapy.packet.Packe**

**fields_desc = [<Field (BSI,asdu_infobj_7,asdu_infobj_8,asdu_infobj_51,asdu_infobj_64).**

**class** conpot.protocols.IEC104.frames.**CP16Time**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.CP16Time'>, <class 'scapy.packet.**

**extract_padding**(*p*)

DEV: to be overloaded to extract current layer’s padding.

**Parameters** **s** (*str*) – the current layer

**Returns** a couple of strings (actual layer, padding)

**fields_desc = [<Field (CP16Time,asdu_infobj_17,asdu_infobj_18,asdu_infobj_19,asdu_info**

**class** conpot.protocols.IEC104.frames.**CP24Time**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.CP24Time'>, <class 'scapy.packet.**

**extract_padding**(*p*)

DEV: to be overloaded to extract current layer’s padding.

**Parameters** **s** (*str*) – the current layer

**Returns** a couple of strings (actual layer, padding)

**fields_desc = [<Field (CP24Time).Ms>, <Field (CP24Time).Min>]**

**class** conpot.protocols.IEC104.frames.**CP56Time**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.CP56Time'>, <class 'scapy.packet.**

**fields_desc = [<Field (CP56Time).Ms>, <Field (CP56Time).Min>, <Field (CP56Time).Hour>,**

**class** conpot.protocols.IEC104.frames.**DIQ**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.DIQ'>, <class 'scapy.packet.Packe**

**fields_desc = [<Field (DIQ).IV>, <Field (DIQ).NT>, <Field (DIQ).SB>, <Field (DIQ).BL>,**

**payload_guess = [({}, <class 'scapy.packet.Padding'>)]**

**class** conpot.protocols.IEC104.frames.**FloatField**(*name*, *default*)

Bases: scapy.fields.Field

**class** conpot.protocols.IEC104.frames.**IOA**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.IOA'>, <class 'scapy.packet.Packe**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**LESignedShortField**(*name*, *default*)

Bases: scapy.fields.Field

**class** conpot.protocols.IEC104.frames.**NVA**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.NVA'>, <class 'scapy.packet.Packe**

**fields_desc = [<Field (NVA,asdu_infobj_9,asdu_infobj_10,asdu_infobj_21,asdu_infobj_34,**

**class** conpot.protocols.IEC104.frames.**NormValueField**(*name*, *default*)

Bases: *conpot.protocols.IEC104.frames.LESignedShortField*

**i2repr**(*pkt*, *x*)

Convert internal value to a nice representation

**class** conpot.protocols.IEC104.frames.**OCI**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.OCI'>, <class 'scapy.packet.Packe**

**fields_desc = [<Field (OCI).Padding>, <Field (OCI).CL3>, <Field (OCI).CL2>, <Field (OC**

**class** conpot.protocols.IEC104.frames.**QDP**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.QDP'>, <class 'scapy.packet.Packe**

**extract_padding**(*p*)

DEV: to be overloaded to extract current layer's padding.

**Parameters** **s** (*str*) – the current layer

**Returns** a couple of strings (actual layer, padding)

**fields_desc = [<Field (QDP).IV>, <Field (QDP).NT>, <Field (QDP).SB>, <Field (QDP).BL>,**

**payload_guess = [({}, <class 'scapy.packet.Padding'>)]**

**class** conpot.protocols.IEC104.frames.**QDS**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.QDS'>, <class 'scapy.packet.Packe**

**extract_padding**(*p*)

DEV: to be overloaded to extract current layer's padding.

**Parameters** **s** (*str*) – the current layer

**Returns** a couple of strings (actual layer, padding)

**fields_desc = [<Field (QDS).IV>, <Field (QDS).NT>, <Field (QDS).SB>, <Field (QDS).BL>,**

**payload_guess = [({}, <class 'scapy.packet.Padding'>)]**

**class** conpot.protocols.IEC104.frames.**QOS**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.QOS'>, <class 'scapy.packet.Packe**

**fields_desc = [<Field (QOS).S/E>, <Field (QOS).QL>]**

**payload_guess = [({}, <class 'scapy.packet.Padding'>)]**

**class** conpot.protocols.IEC104.frames.**SCD**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.SCD'>, <class 'scapy.packet.Packe**

**fields_desc = [<Field (SCD).Status>, <Field (SCD).StatChaDet>]**

**class** conpot.protocols.IEC104.frames.**SEP**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.SEP'>, <class 'scapy.packet.Packe**

**extract_padding**(*p*)

DEV: to be overloaded to extract current layer's padding.

**Parameters** **s** (*str*) – the current layer

**Returns** a couple of strings (actual layer, padding)

**fields_desc = [<Field (SEP).IV>, <Field (SEP).NT>, <Field (SEP).SB>, <Field (SEP).BL>,**

**payload_guess = [({}, <class 'scapy.packet.Padding'>)]**

**class** conpot.protocols.IEC104.frames.**SIQ**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.SIQ'>, <class 'scapy.packet.Packe**

**fields_desc = [<Field (SIQ).IV>, <Field (SIQ).NT>, <Field (SIQ).SB>, <Field (SIQ).BL>,**

**payload_guess = [({}, <class 'scapy.packet.Padding'>)]**

**class** conpot.protocols.IEC104.frames.**SPE**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: `scapy.packet.Packet`

**aliastypes = [<class 'conpot.protocols.IEC104.frames.SPE'>, <class 'scapy.packet.Packe**

**extract_padding**(*p*)

DEV: to be overloaded to extract current layer’s padding.

**Parameters** **s** (*str*) – the current layer

**Returns** a couple of strings (actual layer, padding)

**fields_desc = [<Field (SPE).Padding>, <Field (SPE).SRD>, <Field (SPE).SIE>, <Field (SP**

**payload_guess = [({}, <class 'scapy.packet.Padding'>)]**

**class** conpot.protocols.IEC104.frames.**SVA**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: `scapy.packet.Packet`

**aliastypes = [<class 'conpot.protocols.IEC104.frames.SVA'>, <class 'scapy.packet.Packe**

**fields_desc = [<Field (SVA,asdu_infobj_11,asdu_infobj_12,asdu_infobj_35,asdu_infobj_49**

**class** conpot.protocols.IEC104.frames.**VTI**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: `scapy.packet.Packet`

**aliastypes = [<class 'conpot.protocols.IEC104.frames.VTI'>, <class 'scapy.packet.Packe**

**fields_desc = [<Field (VTI).T>, <Field (VTI).Value>]**

**payload_guess = [({}, <class 'scapy.packet.Padding'>)]**

**class** conpot.protocols.IEC104.frames.**asdu_head**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: `scapy.packet.Packet`

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_head'>, <class 'scapy.packet**

**fields_desc = [<Field (asdu_head).TypeID>, <Field (asdu_head).SQ>, <Field (asdu_head).**

**guess_payload_class**(*payload*)

DEV: Guesses the next payload class from layer bonds. Can be overloaded to use a different mechanism.

**Parameters** **payload** (*str*) – the layer’s payload

**Returns** the payload class

**payload_guess = [({'TypeID': 1}, <class 'conpot.protocols.IEC104.frames.asdu_infobj_1'**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_1**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: `scapy.packet.Packet`

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_1'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_10**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: `scapy.packet.Packet`

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_10'>, <class 'scapy.p**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_100**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_100'>, <class 'scapy.p**

**fields_desc = [<Field (asdu_infobj_100).IOA>, <Field (asdu_infobj_100).QOI>]**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_101**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_101'>, <class 'scapy.p**

**fields_desc = [<Field (asdu_infobj_101).IOA>, <Field (asdu_infobj_101).QCC>]**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_102**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_102'>, <class 'scapy.p**

**fields_desc = [<Field (asdu_infobj_102).IOA>]**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_103**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_103'>, <class 'scapy.p**

**fields_desc = [<Field (asdu_infobj_103).IOA>, <Field (asdu_infobj_103).CP56Time>]**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_11**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_11'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_12**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_12'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_13**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_13'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_14**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_14'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_15**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_15'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_16**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_16'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_17**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_17'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_18**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_18'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_19**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_19'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_2**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_2'>, <class 'scapy.pac**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_20**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_20'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_21**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_21'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_3**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_3'>, <class 'scapy.pac**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_30**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_30'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_31**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_31'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_32**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_32'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_33**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_33'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_34**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_34'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_35**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_35'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_36**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_36'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_37**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_37'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_38**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_38'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_39**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_39'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_4**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_4'>, <class 'scapy.pac**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_40**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_40'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_45**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_45'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_46**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_46'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_47**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_47'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_48**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_48'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_49**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_49'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_5**(*_pkt=b”*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_5'>, <class 'scapy.pac**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asdu**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_50**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_50'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_51**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_51'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_58**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_58'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_59**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_59'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_6**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_6'>, <class 'scapy.pac**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_60**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_60'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_61**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_61'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_62**(_pkt=b”, post_transform=None, _internal=0, _underlayer=None, **fields)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_62'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_63**(_pkt=b”, post_transform=None, _internal=0, _underlayer=None, **fields)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_63'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_64**(_pkt=b”, post_transform=None, _internal=0, _underlayer=None, **fields)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_64'>, <class 'scapy.pa**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_7**(_pkt=b”, post_transform=None, _internal=0, _underlayer=None, **fields)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_7'>, <class 'scapy.pac**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_8**(_pkt=b”, post_transform=None, _internal=0, _underlayer=None, **fields)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_8'>, <class 'scapy.pac**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

**class** conpot.protocols.IEC104.frames.**asdu_infobj_9**(_pkt=b”, post_transform=None, _internal=0, _underlayer=None, **fields)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_9'>, <class 'scapy.pac**

**fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd**

conpot.protocols.IEC104.frames.**calctime**()

**class** conpot.protocols.IEC104.frames.**i_frame**(_pkt=b”, post_transform=None, _internal=0, _underlayer=None, **fields)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.i_frame'>, <class 'scapy.packet.P**

**fields_desc = [<Field (i_frame).Start>, <Field (i_frame).LenAPDU>, <Field (i_frame).Se**

**payload_guess = [({}, <class 'conpot.protocols.IEC104.frames.asdu_head'>)]**

**post_build**(*p*, *pay*)

DEV: called right after the current layer is build.

**Parameters**

- **pkt** (*str*) – the current packet (build by self_buil function)
- **pay** (*str*) – the packet payload (build by do_build_payload function)

**Returns** a string of the packet with the payload

**class** conpot.protocols.IEC104.frames.**s_frame**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.s_frame'>, <class 'scapy.packet.Pa**

**fields_desc = [<Field (s_frame).Start>, <Field (s_frame).LenAPDU>, <Field (s_frame).Ty**

**class** conpot.protocols.IEC104.frames.**u_frame**(*_pkt=b"*, *post_transform=None*, *_internal=0*, *_underlayer=None*, ***fields*)

Bases: scapy.packet.Packet

**aliastypes = [<class 'conpot.protocols.IEC104.frames.u_frame'>, <class 'scapy.packet.Pa**

**fields_desc = [<Field (u_frame).Start>, <Field (u_frame).LenAPDU>, <Field (u_frame).Ty**

### conpot.protocols.IEC104.i_frames_check module

conpot.protocols.IEC104.i_frames_check.**check_asdu_1**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_100**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_11**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_12**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_13**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_14**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_2**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_3**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_30**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_31**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_35**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_36**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_4**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_45**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_46**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_47**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_48**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_49**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_50**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_asdu_51**(*frame*, *direction*) conpot.protocols.IEC104.i_frames_check.**check_command**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_information_with_time**(*frame*, *direction*)

conpot.protocols.IEC104.i_frames_check.**check_information_without_time**(*frame*, *direction*)

### conpot.protocols.IEC104.register module

**class** conpot.protocols.IEC104.register.**IEC104Register**(*category_id*, *addr*, *val*, *relation*)

Bases: `object`

**set_val**(*val*)

### Module contents

### conpot.protocols.bacnet package

### Submodules

### conpot.protocols.bacnet.bacnet_app module

**class** conpot.protocols.bacnet.bacnet_app.**BACnetApp**(*device*, *datagram_server*)

Bases: `bacpypes.app.BIPSimpleApplication`

BACnet device emulation class. BACnet properties are populated from the template file. Services are defined. Conpot implements a smart sensor and hence - DM-RP-B (execute ReadProperty) - DM-DDB-B (execute Who-Is, initiate I-Am) - DM-DOB-B (execute Who-Has, initiate I-Have) services are supported.

**add_object**(*obj*)
Add an object to the local collection.

**add_property**(*prop_name*, *prop_value*)

**get_objects_and_properties**(*dom*)
parse the bacnet template for objects and their properties

**iAm**(**args*)

**iHave**(**args*)

**indication**(*apdu*, *address*, *device*)
logging the received PDU type and Service request

**readProperty**(*request*, *address*, *invoke_key*, *device*)

**response**(*response_apdu*, *address*)

**whoHas**(*request*, *address*, *invoke_key*, *device*)

**whoIs**(*request*, *address*, *invoke_key*, *device*)

### conpot.protocols.bacnet.bacnet_server module

### Module contents

## conpot.protocols.enip package

### Submodules

### conpot.protocols.enip.enip_server module

**class** conpot.protocols.enip.enip_server.**EnipConfig**(*template*)

Bases: object

Configurations parsed from template

**class Tag**(*name*, *type*, *size*, *value*, *addr=None*)

Bases: object

Represents device tag setting parsed from template

**parse_template**()

### Module contents

## conpot.protocols.ftp package

### Submodules

### conpot.protocols.ftp.ftp_base_handler module

**class** conpot.protocols.ftp.ftp_base_handler.**FTPHandlerBase**(*request*, *client_address*, *server*)

Bases: socketserver.BaseRequestHandler

Base class for a full duplex connection

**authentication_ok**(*user_pass*)

Verifies authentication and sets the username of the currently connected client. Returns True or False Checks user names and passwords pairs. Sets the current user and uid.

**config = None**

**class false_request**

Bases: object

**finish**()

End this client session

**ftp_path**(*path*)

Clean and sanitize ftp paths relative fs instance it is hosted in.

**handle**()

Actual FTP service to which the user has connected.

**handle_cmd_channel**()

Read data from the socket and add it to the _command_channel_input_q for processing

**handle_data_channel**()

**host = None**

**port = None**

**process_ftp_command**()

**push_data**(*data*)

Handy utility to push some data using the data channel

**recv_file**(*_file*, *_file_pos=0*, *cmd='STOR'*)

Receive a file - to be used with STOR, REST and APPE. A copy would be made on the _data_fs. :param _file: File Name to the file that would be written to fs. :param _file_pos: Seek file to position before receiving. :param cmd: Command used for receiving file.

**respond**(*response*)

Send processed command/data as reply to the client

**send_file**(*file_name*)

Handy utility to send a file using the data channel

**setup**()

Connect incoming connection to a FTP session

**start_data_channel**(*send_recv='send'*)

Starts the data channel. To be called from the command process greenlet. :param send_recv: Whether the event is a send event or recv event. When set to 'send' data channel's socket writes data in the output queues else when set to 'read' data channel's socket reads data into the input queue. :type send_recv: str

**stop_data_channel**(*abort=False*, *purge=False*, *reason=None*)

**classmethod stream_server_handle**(*sock*, *address*)

Translate this class for use in a StreamServer

**class** conpot.protocols.ftp.ftp_base_handler.**FTPMetrics**

Bases: object

Simple class to track total bytes transferred, login attempts etc.

**get_elapsed_time**()

**get_metrics**(*user_name*, *uid*, *failed_login_attempts*, *max_login_attempts*, *client_address*)

**timeout**

### conpot.protocols.ftp.ftp_handler module

**class** conpot.protocols.ftp.ftp_handler.**FTPCommandChannel**(*request*, *client_address*, *server*)

Bases: *conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase*

FTP Command Responder. Implementation of RFC 959.

**do_ABOR**(*arg*)

Aborts a file transfer currently in progress.

**do_ALLO**(*arg*)

Allocate bytes for storage (noop).

**do_APPE**(*file*)

Append data to an existing file on the server. On success return the file path, else None.

**do_BYE**(*arg*)

**do_CDUP** (*arg*)
Change into the parent directory. On success return the new directory, else None.

**do_CWD** (*path*)
Change the current working directory.

**do_DELE** (*path*)
Delete the specified file.

**do_HELP** (*arg*)
Return help text to the client.

**do_LIST** (*path*)

**do_MDTM** (*path*)
Return last modification time of file to the client as an ISO 3307 style timestamp (YYYYMMDDHHMMSS) as defined in RFC-3659. On success return the file path, else None.

**do_MKD** (*path*)
Create the specified directory. On success return the directory path, else None.

**do_MODE** (*line*)
Set data transfer mode ("S" is the only one supported (noop)).

**do_NLST** (*path*)
Return a list of files in the specified directory in a compact form to the client.

**do_NOOP** (*arg*)
Do nothing. No params required. No auth required and no permissions required.

**do_PASS** (*arg*)

**do_PASV** (*arg*)
Starts a Passive Data Channel using IPv4. We don't actually need to start the full duplex connection here. Just need to figure the host ip and the port. The DTP connection would start in each command.

**do_PORT** (*arg*)
Starts an active data channel by using IPv4. We don't actually need to start the full duplex connection here. Just need to figure the host ip and the port. The DTP connection would start in each command.

**do_PWD** (*arg*)
Return the name of the current working directory to the client.

**do_QUIT** (*arg*)

**do_REIN** (*arg*)
Reinitialize user's current session.

**do_REST** (*line*)
Restart a file transfer from a previous mark.

**do_RETR** (*arg*)
Fetch and send a file. :param arg: Filename that is to be retrieved

**do_RMD** (*path*)
Remove the specified directory. On success return the directory path, else None.

**do_RNFR** (*path*)
Rename the specified (only the source name is specified here, see RNTO command)

**do_RNTO** (*dst_path*)
Rename file (destination name only, source is specified with RNFR).

**do_SITE_CHMOD** (*path*, *mode*)
Change file mode. On success return a (file_path, mode) tuple.

**do_SITE_HELP** (*line*)
Return help text to the client for a given SITE command.

**do_SIZE** (*path*)
Return size of file in a format suitable for using with RESTart as defined in RFC-3659.

**do_STAT** (*path*)
If invoked without parameters, returns general status information about the FTP server process. If a parameter is given, acts like the LIST command, except that data is sent over the command channel (no PORT or PASV command is required).

**do_STOR** (*file*, *mode='w'*)
Store a file (transfer from the client to the server).

**do_STOU** (*line*)
Store a file on the server with a unique name.

**do_STRU** (*line*)
Set file structure ("F" is the only one supported (noop)).

**do_SYST** (*arg*)
Return system type (always returns UNIX type: L8).

**do_TYPE** (*line*)
Set current type data type to binary/ascii

**do_USER** (*arg*)
USER FTP command. If the user is already logged in, return 530 else 331 for the PASS command :param arg: username specified by the client/attacker

**do_XCUP** (*arg*)
Change into the parent directory. On success return the new directory, else None.

**do_XCWD** (*path*)
Change the current working directory.

**do_XMKD** (*path*)
Create the specified directory. On success return the directory path, else None.

**do_XPWD** (*arg*)
Return the name of the current working directory to the client.

**do_XRMD** (*path*)
Remove the specified directory. On success return the directory path, else None.

**process_ftp_command** ()
Handle an incoming handle request - pick and item from the input_q, reads the contents of the message and dispatch contents to the appropriate do_* method. :param: (bytes) line - incoming request :return: (bytes) response - reply in respect to the request

### conpot.protocols.ftp.ftp_server module

**class** conpot.protocols.ftp.ftp_server.**FTPConfig** (*template*)
Bases: object

**get_gid** (*uid*)
Get group id of a user from it's uid

**get_uid**(*user_name*)
Get uid from a username

### conpot.protocols.ftp.ftp_utils module

**exception** conpot.protocols.ftp.ftp_utils.**FTPException**
Bases: Exception

General FTP related exceptions.

**exception** conpot.protocols.ftp.ftp_utils.**FTPMaxLoginAttemptsExceeded**
Bases: *conpot.protocols.ftp.ftp_utils.FTPException*

**exception** conpot.protocols.ftp.ftp_utils.**FTPPrivilegeException**
Bases: *conpot.protocols.ftp.ftp_utils.FTPException*

conpot.protocols.ftp.ftp_utils.**get_data_from_iter**(*iterator*)
This utility function generates data from iterators and returns them as string

### Module contents

### conpot.protocols.guardian_ast package

### Submodules

### conpot.protocols.guardian_ast.guardian_ast_server module

Service support based on gaspot.py [https://github.com/sjhilt/GasPot] Original authors: Kyle Wilhoit and Stephen Hilt

### Module contents

### conpot.protocols.http package

### Submodules

### conpot.protocols.http.command_responder module

**class** conpot.protocols.http.command_responder.**CommandResponder**(*host*, *port*, *template*, *docpath*)
Bases: object

**serve_forever**()

**stop**()

**class** conpot.protocols.http.command_responder.**HTTPServer**(*request*, *client_address*, *server*)
Bases: http.server.BaseHTTPRequestHandler

**do_GET**()
Handle GET requests

**do_HEAD**()
Handle HEAD requests.

**do_OPTIONS**()
Handle OPTIONS requests.

**do_POST**()
Handle POST requests

**do_TRACE**()
Handle TRACE requests.

**get_entity_headers**(*rqfilename*, *headers*, *configuration*)

**get_entity_trailers**(*rqfilename*, *configuration*)

**get_status_headers**(*status*, *headers*, *configuration*)

**get_status_trailers**(*status*, *configuration*)

**get_trigger_appendix**(*rqfilename*, *rqparams*, *configuration*)

**load_entity**(*requeststring*, *headers*, *configuration*, *docpath*)
Retrieves status, headers and payload for a given entity, that can be stored either local or on a remote system

**load_status**(*status*, *requeststring*, *requestheaders*, *headers*, *configuration*, *docpath*, *method='GET'*, *body=None*)
Retrieves headers and payload for a given status code. Certain status codes can be configured to forward the request to a remote system. If not available, generate a minimal response

**log**(*version*, *request_type*, *addr*, *request*, *response=None*)

**send_chunked**(*chunks*, *payload*, *trailers*)
Send payload via chunked transfer encoding to the client, followed by eventual trailers.

**send_error**(*code*, *message=None*)
Send and log an error reply. This method is overloaded to make use of load_status() to allow handling of "Unsupported Method" errors.

**send_response**(*code*, *message=None*)
Send the response header and log the response code. This function is overloaded to change the behaviour when loggers and sending default headers.

**substitute_template_fields**(*payload*)

**class** conpot.protocols.http.command_responder.**SubHTTPServer**(*server_address*, *RequestHandlerClass*, *template*, *docpath*)
Bases: conpot.protocols.http.command_responder.ThreadedHTTPServer

this class is necessary to allow passing custom request handler into the RequestHandlerClass

**config_sanitize_tarpit**(*value*)

**daemon_threads = True**

**do_tarpit**(*delay*)

**class** conpot.protocols.http.command_responder.**TemplateParser**(*data*)
Bases: html.parser.HTMLParser

**handle_startendtag**(*tag*, *attrs*)
handles template tags provided in XHTML notation.

Expected format: <condata source="(engine)" key="(descriptor)" /> Example: <condata source="databus" key="SystemDescription" />

at the moment, the parser is space- and case-sensitive(!), this could be improved by using REGEX for replacing the template tags with actual values.

**class** `conpot.protocols.http.command_responder.`**`ThreadedHTTPServer`**(*server_address*, *RequestHandlerClass*, *bind_and_activate=True*)

Bases: `socketserver.ThreadingMixIn`, `http.server.HTTPServer`

Handle requests in a separate thread.

### conpot.protocols.http.web_server module

### Module contents

## conpot.protocols.ipmi package

### Submodules

### conpot.protocols.ipmi.fakebmc module

**class** `conpot.protocols.ipmi.fakebmc.`**`FakeBmc`**(*authdata*, *port*)

Bases: `pyghmi.ipmi.bmc.Bmc`

**`cold_reset`**()

**`get_boot_device`**()

**`get_power_state`**()

**`power_cycle`**()

**`power_off`**()

**`power_on`**()

**`power_reset`**()

**`power_shutdown`**()

**`set_boot_device`**(*bootdevice*)

### conpot.protocols.ipmi.fakesession module

**class** `conpot.protocols.ipmi.fakesession.`**`FakeSession`**(*bmc*, *userid*, *password*, *port*)

Bases: `pyghmi.ipmi.private.session.Session`

**`send_data`**(*packet*, *address*)

**`send_ipmi_response`**(*data=None*, *code=0*)

**`send_payload`**(*payload=()*, *payload_type=None*, *retry=True*, *delay_xmit=None*, *needskeepalive=False*)

Send payload over the IPMI Session

**Parameters**

- **needskeepalive** – If the payload is expected not to count as 'active' by the BMC, set this to True to avoid Session considering the job done because of this payload. Notably, 0-length SOL packets are prone to confusion.
- **timeout** – Specify a custom timeout for long-running request

### conpot.protocols.ipmi.ipmi_server module

**class** conpot.protocols.ipmi.ipmi_server.**IpmiServer**(*template*, *template_directory*, *args*)

Bases: object

**close_server_session**()

**handle**(*data*, *address*)

**handle_client_request**(*request*)

**initiate_session**(*data*, *address*, *session*)

**send_auth_cap**(*myaddr*, *mylun*, *clientaddr*, *clientlun*, *sockaddr*)

**start**(*host*, *port*)

**stop**()

### Module contents

### conpot.protocols.kamstrup package

### Subpackages

### conpot.protocols.kamstrup.management_protocol package

### Submodules

### conpot.protocols.kamstrup.management_protocol.command_responder module

**class** conpot.protocols.kamstrup.management_protocol.command_responder.**CommandResponder**

Bases: object

**COMMAND_NOT_FOUND = "\r\n? Command not found.\r\nSend 'H' for help.\r\n"**

**respond**(*request*)

### conpot.protocols.kamstrup.management_protocol.commands module

**class** conpot.protocols.kamstrup.management_protocol.commands.**AccessControlCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**CMD_OUTPUT = '\r\n{access_control_status} \r\n [1] {access_control_1}\r\n [2] {access_c**

**HELP_MESSAGE = "!AC: Access control.\r\n Used for simple IP address firewall filtering**

**run**(*params=None*)

**set_access_ip** (*number*, *ip_string*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**AlarmServerCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**CMD_OUTPUT = '\r\nAlarm server:  {alarm_server_output} '**

**HELP_MESSAGE = '!AS: Alarm Server.\r\n Used to set IP and port of server to handle ala**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**BaseCommand**

Bases: object

**CMD_OUTPUT = ''**

**HELP_MESSAGE = ''**

**INVALID_PARAMETER = "\r\n? Invalid parameter.\r\nTry 'H cmd' for specific help.\r\n Ie**

**help** ()

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**GetConfigCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**CMD_OUTPUT = 'Device Name :  {device_name}\r\nUse DHCP : {use_dhcp}\r\nIP addr. :  {ip_**

**HELP_MESSAGE = '!GC: Get Config.\r\n Returns the module configuration.\r\n'**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**HelpCommand** (*commands*)

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**CMD_OUTPUT = '==============================================================================**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**RequestConnectCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**HELP_MESSAGE = '!RC: Request connect\r\n Makes the module crate a ChA or ChB socket to**

**run** (*params*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**RequestRestartCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**HELP_MESSAGE = '!RR: Request restart (*1).\r\n'**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**SetConfigCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**CMD_OUTPUT = '\r\nService server hostname.:  {}\r\n'**

**HELP_MESSAGE = '!SC: Set Config (*1).\r\n Configures the module.\r\n Format:  !SC DHCP**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**SetDeviceNameCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**HELP_MESSAGE = '!SD: Set device name (*1).\r\n Option for individual naming of the mod**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**SetIPCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**CMD_OUTPUT = '\r\nUse DHCP : {use_dhcp}\r\n\r\nIP addr. :  {ip_addr}\r\n'**

**HELP_MESSAGE = "!SI: Set IP (enter either valid IP or 0 to force DHCP)(*1).\r\n Used f**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**SetKap1Command**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**CMD_OUTPUT = '\r\nService server addr.:  {kap_a_output}\r\n'**

**HELP_MESSAGE = '!SA: Set KAP Server IP and port (*1).\r\n Used for setting the IP of t**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**SetKap2Command**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**CMD_OUTPUT_DOUBLE = '\r\n{}\r\nService server addr.:  {}:{} (from DNS)\r\nand fallback**

**CMD_OUTPUT_SINGLE = '\r\n{}\r\nService server addr.:  {}:{} (from DNS)\r\nNo redundanc**

**HELP_MESSAGE = '!SB: Set 2nd KAP Server IP and port.\r\n Used for redundancy with two**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**SetLookupCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**HELP_MESSAGE = '!SH: Set KAP Server lookup (DNS or DHCP)\r\n Used for setting the DNS**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**SetNameserverCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**HELP_MESSAGE = '!SN: Set IP for DNS Name servers to use.\r\n Format:  !SN DNS1 DNS2 DN**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**SetPortsCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**CMD_OUTPUT = '\r\n{}\r\nKAP on server:  {}\r\nChA on module:  {}\r\nChB on module:  {}**

**HELP_MESSAGE = '!SP: Set IP Ports\r\n Format:  !SP [KAP CHA CHB CFG]\r\n Example:  !SP**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**SetSerialCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**HELP_MESSAGE = "!SS: Set Serial Settings.\r\n Used for setting the serial interface fo**

**run** (*params=None*)

**class** conpot.protocols.kamstrup.management_protocol.commands.**SetWatchdogCommand**

Bases: *conpot.protocols.kamstrup.management_protocol.commands.BaseCommand*

**CMD_OUTPUT = 'Software watchdog:  {0}\r\nKAP Missing warning:  {1}\r\nKeep alive timer**

**HELP_MESSAGE = '!SK: Set KAP watchdog timeout(WDT).\r\n Used for setting KeepAlive wat**

**run** (*params=None*)

**class** `conpot.protocols.kamstrup.management_protocol.commands.`**`SoftwareVersionCommand`**
Bases: *`conpot.protocols.kamstrup.management_protocol.commands.BaseCommand`*

**`CMD_OUTPUT = '\r\nSoftware Version:  {software_version}\r\n'`**

**`HELP_MESSAGE = '!GV: Software version.\r\n Returns the software revision of the module`**

**`run`**(*params=None*)

**class** `conpot.protocols.kamstrup.management_protocol.commands.`**`WinkModuleCommand`**
Bases: *`conpot.protocols.kamstrup.management_protocol.commands.BaseCommand`*

**`CMD_OUTPUT = '\r\n\r\nOK\r\n'`**

**`HELP_MESSAGE = '!WM: Wink module.\r\n Causes the WINK LED on the module to blink for pl`**

`conpot.protocols.kamstrup.management_protocol.commands.`**`parse_ip`**(*ip_string*)

`conpot.protocols.kamstrup.management_protocol.commands.`**`parse_port`**(*port_string*)

`conpot.protocols.kamstrup.management_protocol.commands.`**`try_parse_uint`**(*uint_string, min_value=0, max_value=254*)

#### conpot.protocols.kamstrup.management_protocol.kamstrup_management_server module

#### Module contents

### conpot.protocols.kamstrup.meter_protocol package

#### Submodules

#### conpot.protocols.kamstrup.meter_protocol.command_responder module

**class** `conpot.protocols.kamstrup.meter_protocol.command_responder.`**`CommandResponder`**(*template*)
Bases: `object`

**`respond`**(*request*)

#### conpot.protocols.kamstrup.meter_protocol.decoder_382 module

**class** `conpot.protocols.kamstrup.meter_protocol.decoder_382.`**`Decoder382`**
Bases: `object`

**`REGISTERS = {1:  'Energy in', 2:  'Energy out', 13:  'Energy in hi-res', 14:  'Energy`**

**`decode_in`**(*data*)

**`decode_out`**(*data*)

**`classmethod valid_crc`**(*message*)

#### conpot.protocols.kamstrup.meter_protocol.kamstrup_constants module

**class** `conpot.protocols.kamstrup.meter_protocol.kamstrup_constants.`**`MeterTypes`**
Bases: `enum.Enum`

An enumeration.

**K162M = (2,)**

**K351C = (3,)**

**K382M = (1,)**

**OMNIA = (4,)**

**Unknown = (0,)**

### conpot.protocols.kamstrup.meter_protocol.kamstrup_server module

### conpot.protocols.kamstrup.meter_protocol.messages module

**class** conpot.protocols.kamstrup.meter_protocol.messages.**KamstrupProtocolBase**(*communication_ada*

Bases: object

**class** conpot.protocols.kamstrup.meter_protocol.messages.**KamstrupRequestBase**(*communication_addr* *command*, *message_bytes*)

Bases: *conpot.protocols.kamstrup.meter_protocol.messages.KamstrupProtocolBase*

**class** conpot.protocols.kamstrup.meter_protocol.messages.**KamstrupRequestGetRegisters**(*communic* *command_byt* *message_byte*

Bases: *conpot.protocols.kamstrup.meter_protocol.messages.KamstrupRequestBase*

**command_byte = 16**

**class** conpot.protocols.kamstrup.meter_protocol.messages.**KamstrupRequestUnknown**(*communication_* *command_byte*, *message_bytes*)

Bases: *conpot.protocols.kamstrup.meter_protocol.messages.KamstrupRequestBase*

**class** conpot.protocols.kamstrup.meter_protocol.messages.**KamstrupResponseBase**(*communication_ada*

Bases: *conpot.protocols.kamstrup.meter_protocol.messages.KamstrupProtocolBase*

**classmethod escape**(*message*)

**serialize**(*message*)

**class** conpot.protocols.kamstrup.meter_protocol.messages.**KamstrupResponseRegister**(*communicatio*

Bases: *conpot.protocols.kamstrup.meter_protocol.messages.KamstrupResponseBase*

**add_register**(*register*)

**serialize**(*message=None*)

### conpot.protocols.kamstrup.meter_protocol.register module

**class** conpot.protocols.kamstrup.meter_protocol.register.**KamstrupRegister**(*name*, *units*, *length*, *unknown*, *databus_key*)

Bases: object

### conpot.protocols.kamstrup.meter_protocol.request_parser module

**class** conpot.protocols.kamstrup.meter_protocol.request_parser.**KamstrupRequestParser**

Bases: object

**add_byte**(*byte*)

**get_request**()

**classmethod valid_crc**(*message*)

### Module contents

### Submodules

### conpot.protocols.kamstrup.usage_simulator module

**class** conpot.protocols.kamstrup.usage_simulator.**UsageSimulator**(**args*)

Bases: object

**initialize**()

**stop**()

**usage_counter**()

### Module contents

### conpot.protocols.misc package

### Submodules

### conpot.protocols.misc.ascii_decoder module

**class** conpot.protocols.misc.ascii_decoder.**AsciiDecoder**

Bases: *conpot.emulators.proxy.ProxyDecoder*

**decode_in**(*data*)

Decode data that goes into the proxied device

**decode_out**(*data*)

Decode data that goes out from the proxied device to the connected client(attacker).

### Module contents

### conpot.protocols.modbus package

### Submodules

### conpot.protocols.modbus.modbus_block_databus_mediator module

**class** conpot.protocols.modbus.modbus_block_databus_mediator.**ModbusBlockDatabusMediator**(*datab*
*starting_a*

Bases: object

This class represents the values for a range of addresses

**is_in**(*starting_address*, *size*)
Returns true if a block with the given address and size would overlap this block

### conpot.protocols.modbus.modbus_server module

### conpot.protocols.modbus.slave module

**class** conpot.protocols.modbus.slave.**MBSlave**(*slave_id*, *dom*)
Bases: modbus_tk.modbus.Slave

Customized Modbus slave representation extending modbus_tk.modbus.Slave

**add_block**(*block_name*, *block_type*, *starting_address*, *size*)
Add a new block identified by its name

**handle_request**(*request_pdu*, *broadcast=False*)
parse the request pdu, makes the corresponding action and returns the response pdu

### conpot.protocols.modbus.slave_db module

**class** conpot.protocols.modbus.slave_db.**SlaveBase**(*template*)
Bases: modbus_tk.modbus.Databank

Database keeping track of the slaves.

**add_slave**(*slave_id*, *unsigned=True*, *memory=None*)
Add a new slave with the given id

**handle_request**(*query*, *request*, *mode*)
Handles a request. Return value is a tuple where element 0 is the response object and element 1 is a dictionary of items to log.

### Module contents

### conpot.protocols.s7comm package

### Submodules

### conpot.protocols.s7comm.cotp module

**class** conpot.protocols.s7comm.cotp.**COTP**(*tpdu_type=0*, *opt_field=0*, *payload=''*, *trailer=''*)

Bases: object

**pack**()

**parse**(*packet*)

**class** conpot.protocols.s7comm.cotp.**COTPConnectionPacket**(*dst_ref=0*, *src_ref=0*, *opt_field=0*, *src_tsap=0*, *dst_tsap=0*, *tpdu_size=0*)

Bases: object

**dissect**(*packet*)

**class** conpot.protocols.s7comm.cotp.**COTP_ConnectionConfirm**(*dst_ref=0*, *src_ref=0*, *opt_field=0*, *src_tsap=0*, *dst_tsap=0*, *tpdu_size=0*)

Bases: *conpot.protocols.s7comm.cotp.COTPConnectionPacket*

**assemble**()

**class** conpot.protocols.s7comm.cotp.**COTP_ConnectionRequest**(*dst_ref=0*, *src_ref=0*, *opt_field=0*, *src_tsap=0*, *dst_tsap=0*, *tpdu_size=0*)

Bases: *conpot.protocols.s7comm.cotp.COTPConnectionPacket*

**assemble**()

### conpot.protocols.s7comm.exceptions module

**exception** conpot.protocols.s7comm.exceptions.**AssembleException**(*protocol*, *reason*, *payload=''*)

Bases: Exception

**exception** conpot.protocols.s7comm.exceptions.**ParseException**(*protocol*, *reason*, *payload=''*)

Bases: Exception

### conpot.protocols.s7comm.s7 module

**class** conpot.protocols.s7comm.s7.**S7**(*pdu_type=0*, *reserved=0*, *request_id=0*, *result_info=0*, *parameters=''*, *data=''*)

Bases: object

**handle**(*current_client=None*)

**pack**()

**parse**(*packet*)

**plc_stop_signal**(*current_client*)

**request_diagnostics**()

**request_not_implemented**()

**request_ssl_17**(*data_ssl_index*)

**request_ssl_28**(*data_ssl_index*)

**ssl_lists = {}**

### conpot.protocols.s7comm.s7_server module

conpot.protocols.s7comm.s7_server.**cleanse_byte_string**(*packet*)

### conpot.protocols.s7comm.tpkt module

**class** conpot.protocols.s7comm.tpkt.**TPKT**(*version=3*, *payload=''*)

Bases: object

**pack**()

**parse**(*packet*)

### Module contents

### conpot.protocols.snmp package

### Submodules

### conpot.protocols.snmp.build_pysnmp_mib_wrapper module

conpot.protocols.snmp.build_pysnmp_mib_wrapper.**compile_mib**(*mib_name*, *output_dir*)

Compiles the given mib_name if it is found in the internal MIB file map. If the MIB depends on other MIBs, these will get compiled automatically. :param mib_name: Name of mib to compile (string). :param output_dir: Output directory (string).

conpot.protocols.snmp.build_pysnmp_mib_wrapper.**find_mibs**(*raw_mibs_dirs*, *recursive=True*)

Scans for MIB files and populates an internal MIB->path mapping. :param raw_mibs_dirs: Directories to search for MIB files (list of strings). :param recursive: If True raw_mibs_dirs will be scanned recursively. :return: A list of found MIB names (list of strings).

conpot.protocols.snmp.build_pysnmp_mib_wrapper.**generate_dependencies**(*data*, *mib_name*)

Parses a MIB for dependencies and populates an internal dependency map. :param data: A string representing an entire MIB file (string). :param mib_name: Name of the MIB (string).

conpot.protocols.snmp.build_pysnmp_mib_wrapper.**mib2pysnmp**(*mib_file*, *output_dir*)

The 'build-pysnmp-mib' script we previously used is no longer available Latest pysmi has the ability to generate a .py file from .mib automatically

**Parameters**

- **mib_file** – path to the .mib file we want to compile
- **output_dir** – path to the output directory

**Returns** True if we successfully compile the .mib to a .py

### conpot.protocols.snmp.command_responder module

**class** conpot.protocols.snmp.command_responder.**CommandResponder**(*host*, *port*, *mibpaths*)

Bases: object

**addSocketTransport**(*snmpEngine*, *transportDomain*, *transport*)
Add transport object to socket dispatcher of snmpEngine

**has_mib**(*mibname*)

**register**(*mibname*, *symbolname*, *instance*, *value*, *profile_map_name*)
Register OID

**serve_forever**()

**stop**()

**class** conpot.protocols.snmp.command_responder.**SNMPDispatcher**

Bases: gevent.server.DatagramServer

**getTimerResolution**()

**handle**(*msg*, *address*)

**registerRecvCbFun**(*recvCbFun*, *recvId=None*)

**registerTimerCbFun**(*timerCbFun*, *tickInterval=None*)

**registerTransport**(*tDomain*, *transport*)

**sendMessage**(*outgoingMessage*, *transportDomain*, *transportAddress*)

### conpot.protocols.snmp.conpot_cmdrsp module

**class** conpot.protocols.snmp.conpot_cmdrsp.**c_BulkCommandResponder**(*snmpEngine*, *snmpContext*, *databus_mediator*, *host*, *port*)

Bases: pysnmp.entity.rfc3413.cmdrsp.BulkCommandResponder, *conpot.protocols.snmp.conpot_cmdrsp.conpot_extension*

**handleMgmtOperation**(*snmpEngine*, *stateReference*, *contextName*, *PDU*, *acInfo*)

**class** conpot.protocols.snmp.conpot_cmdrsp.**c_GetCommandResponder**(*snmpEngine*, *snmpContext*, *databus_mediator*, *host*, *port*)

Bases: pysnmp.entity.rfc3413.cmdrsp.GetCommandResponder, *conpot.protocols.snmp.conpot_cmdrsp.conpot_extension*

**handleMgmtOperation**(*snmpEngine*, *stateReference*, *contextName*, *PDU*, *acInfo*)

**class** conpot.protocols.snmp.conpot_cmdrsp.**c_NextCommandResponder**(*snmpEngine*, *snmpContext*, *databus_mediator*, *host*, *port*)

Bases: pysnmp.entity.rfc3413.cmdrsp.NextCommandResponder, *conpot.protocols.snmp.conpot_cmdrsp.conpot_extension*

**handleMgmtOperation**(*snmpEngine*, *stateReference*, *contextName*, *PDU*, *acInfo*)

**class** conpot.protocols.snmp.conpot_cmdrsp.**c_SetCommandResponder**(*snmpEngine*, *snmpContext*, *databus_mediator*, *host*, *port*)

Bases: pysnmp.entity.rfc3413.cmdrsp.SetCommandResponder, *conpot.protocols.snmp.conpot_cmdrsp.conpot_extension*

**handleMgmtOperation**(*snmpEngine*, *stateReference*, *contextName*, *PDU*, *acInfo*)

**class** conpot.protocols.snmp.conpot_cmdrsp.**conpot_extension**

Bases: object

**check_evasive**(*state*, *threshold*, *addr*, *cmd*)

**do_tarpit**(*delay*)

**log**(*version*, *msg_type*, *addr*, *req_varBinds*, *res_varBinds=None*, *sock=None*)

### conpot.protocols.snmp.databus_mediator module

**class** conpot.protocols.snmp.databus_mediator.**DatabusMediator**(*oid_mappings*)

Bases: object

**get_response**(*reference_class*, *OID*)

**set_value**(*OID*, *value*)

**update_evasion_table**(*client_ip*)

updates dynamic evasion table

### conpot.protocols.snmp.snmp_server module

### Module contents

### conpot.protocols.tftp package

### Submodules

### conpot.protocols.tftp.tftp_handler module

**class** conpot.protocols.tftp.tftp_handler.**TFTPContextServer**(*host*, *port*, *timeout*, *root*, *dyn_file_func=None*, *upload_open=None*)

Bases: tftpy.TftpContexts.TftpContextServer

Simple TFTP server handler wrapper. Use conpot's filesystem wrappers rather than os.*

**end**()

Finish up the context.

**file_path = None**

**start**(*buffer*)

Start the state cycle. Note that the server context receives an initial packet in its start method. Also note that the server does not loop on cycle(), as it expects the TftpServer object to manage that.

**class** conpot.protocols.tftp.tftp_handler.**TFTPServerState**(*context*)

Bases: *conpot.protocols.tftp.tftp_handler.TFTPState*

The base class for server states.

**data_fs = None**

**full_path = None**

**handle**(*pkt*, *raddress*, *rport*)

An abstract method for handling a packet. It is expected to return a TftpState object, either itself or a new state.

**serverInitial**(*pkt*, *raddress*, *rport*)

**vfs = None**

**class** conpot.protocols.tftp.tftp_handler.**TFTPState**(*context*)

Bases: tftpy.TftpStates.TftpState

**handle**(*pkt*, *raddress*, *rport*)

An abstract method for handling a packet. It is expected to return a TftpState object, either itself or a new state.

**class** conpot.protocols.tftp.tftp_handler.**TFTPStateServerRecvRRQ**(*context*)

Bases: *conpot.protocols.tftp.tftp_handler.TFTPServerState*

**handle**(*pkt*, *raddress*, *rport*)

Handle an initial RRQ packet as a server.

**class** conpot.protocols.tftp.tftp_handler.**TFTPStateServerRecvWRQ**(*context*)

Bases: *conpot.protocols.tftp.tftp_handler.TFTPServerState*

This class represents the state of the TFTP server when it has just received a WRQ packet.

**handle**(*pkt*, *raddress*, *rport*)

Handle an initial WRQ packet as a server.

**make_subdirs**()

The purpose of this method is to, if necessary, create all of the subdirectories leading up to the file to the written.

**class** conpot.protocols.tftp.tftp_handler.**TFTPStateServerStart**(*context*)

Bases: *conpot.protocols.tftp.tftp_handler.TFTPState*

The start state for the server. This is a transitory state since at this point we don't know if we're handling an upload or a download. We will commit to one of them once we interpret the initial packet.

**handle**(*pkt*, *raddress*, *rport*)

Handle a packet we just received.

### conpot.protocols.tftp.tftp_server module

### Module contents

### Module contents

### conpot.tests package

### Subpackages

### conpot.tests.helpers package

### Submodules

### conpot.tests.helpers.s7comm_client module

`conpot.tests.helpers.s7comm_client.`**`AddOptions`**(*parser*)

`conpot.tests.helpers.s7comm_client.`**`BruteTsap`**(*ip, port, src_tsaps=(256, 512), dst_tsaps=(258, 512, 513)*)

**`class`** `conpot.tests.helpers.s7comm_client.`**`COTPConnectionPacket`**(*dst_ref=0, src_ref=0, dst_tsap=0, src_tsap=0, tpdu_size=0*)

Bases: `object`

COTP Connection Request or Connection Confirm packet (ISO on TCP). RFC 1006

**`pack`**()

make Connection Request Packet

**`unpack`**(*packet*)

parse Connection Confirm Packet (header only)

**`class`** `conpot.tests.helpers.s7comm_client.`**`COTPDataPacket`**(*data=''*)

Bases: `object`

COTP Data packet (ISO on TCP). RFC 1006

**`pack`**()

**`unpack`**(*packet*)

`conpot.tests.helpers.s7comm_client.`**`GetIdentity`**(*ip, port, src_tsap, dst_tsap*)

**`exception`** `conpot.tests.helpers.s7comm_client.`**`S7Error`**(*code*)

Bases: `Exception`

**`class`** `conpot.tests.helpers.s7comm_client.`**`S7Packet`**(*_type=1, req_id=0, parameters='', data=''*)

Bases: `object`

S7 packet

**`pack`**()

**`unpack`**(*packet*)

**`exception`** `conpot.tests.helpers.s7comm_client.`**`S7ProtocolError`**(*message, packet=''*)

Bases: `Exception`

`conpot.tests.helpers.s7comm_client.`**`Scan`**(*ip, port*)

`conpot.tests.helpers.s7comm_client.`**`Split`**(*ar, size*)

split sequence into blocks of given size

`conpot.tests.helpers.s7comm_client.`**`StripUnprintable`**(*msg*)

**class** conpot.tests.helpers.s7comm_client.**TPKTPacket**(*data=”*)

Bases: object

TPKT packet. RFC 1006

**pack**()

**unpack**(*packet*)

**class** conpot.tests.helpers.s7comm_client.**s7**(*ip*, *port*, *src_tsap=512*, *dst_tsap=513*, *timeout=8*)

Bases: object

**Connect**()

Establish ISO on TCP connection and negotiate PDU

**Function**(*_type*, *group*, *function*, *data=”*)

**NegotiatePDU**(*pdu=480*)

Send negotiate pdu request and receive response. Reply no matter

**ReadSZL**(*szl_id*)

**Request**(*_type*, *parameters=”*, *data=”*)

Send s7 request and receive response

**plc_stop_function**()

### conpot.tests.helpers.snmp_client module

**class** conpot.tests.helpers.snmp_client.**SNMPClient**(*host*, *port*)

Bases: object

**cbFun**(*sendRequestHandle*, *errorIndication*, *errorStatus*, *errorIndex*, *varBindTable*, *cbCtx*)

**get_command**(*OID=((1, 3, 6, 1, 2, 1, 1, 1, 0), None)*, *callback=None*)

**set_command**(*OID*, *callback=None*)

**walk_command**(*OID*, *callback=None*)

## Module contents

## Submodules

### conpot.tests.test_bacnet_server module

**class** conpot.tests.test_bacnet_server.**TestBACnetServer**(*methodName=’runTest’*)

Bases: unittest.case.TestCase

All tests are executed in a similar way. We initiate a service request to the BACnet server and wait for response. Instead of decoding the response, we create an expected response. We encode the expected response and compare the two encoded data.

**setUp**()

Hook method for setting up the test fixture before exercising it.

**tearDown**()

Hook method for deconstructing the test fixture after testing it.

**test_no_response_requests**()
When the request has apduType not 0x01, no reply should be returned from Conpot

**test_readProperty**()

**test_whoHas**()

**test_whoIs**()

### conpot.tests.test_base module

**class** conpot.tests.test_base.**TestBase**(*methodName='runTest'*)
Bases: unittest.case.TestCase

**setUp**()
Hook method for setting up the test fixture before exercising it.

**tearDown**()
Hook method for deconstructing the test fixture after testing it.

**test_base**()

### conpot.tests.test_docs module

**class** conpot.tests.test_docs.**TestMakeDocs**(*methodName='runTest'*)
Bases: unittest.case.TestCase

**setUp**()
Hook method for setting up the test fixture before exercising it.

**tearDown**()
Hook method for deconstructing the test fixture after testing it.

**test_make_docs**()

### conpot.tests.test_enip_server module

**class** conpot.tests.test_enip_server.**TestENIPServer**(*methodName='runTest'*)
Bases: unittest.case.TestCase

**attribute_operations**(*paths*, *int_type=None*, ***kwds*)

**setUp**()
Hook method for setting up the test fixture before exercising it.

**tearDown**()
Hook method for deconstructing the test fixture after testing it.

**test_list_identity_tcp**()

**test_list_identity_udp**()

**test_list_interfaces_tcp**()

**test_list_interfaces_udp**()

**test_list_services_tcp**()

**test_list_services_udp**()

**test_malformend_request_tcp**()

**test_malformend_request_udp**()

**test_read_tags**()

**test_write_tags**()

### conpot.tests.test_ext_ip_util module

**class** conpot.tests.test_ext_ip_util.**TestExtIPUtil**(*methodName='runTest'*)

Bases: unittest.case.TestCase

**setUp**()

Hook method for setting up the test fixture before exercising it.

**tearDown**()

Hook method for deconstructing the test fixture after testing it.

**test_ext_util**()

**test_fetch_ext_ip**()

**test_ip_verify**()

### conpot.tests.test_ftp module

**class** conpot.tests.test_ftp.**TestFTPServer**(*methodName='runTest'*)

Bases: unittest.case.TestCase

All tests are executed in a similar way. We run a valid/invalid FTP request/command and check for valid response. Testing is done by sending/receiving files in data channel related commands. Implementation Note: There are no explicit tests for active/passive mode. These are covered in list and nlst tests

**refresh_client**()

Disconnect and reconnect a client

**setUp**()

Hook method for setting up the test fixture before exercising it.

**tearDown**()

Hook method for deconstructing the test fixture after testing it.

**test_abor**()

**test_allo**()

**test_appe**()

**test_auth**()

Test for user, pass and quit commands.

**test_cwd**()

**test_dele**()

**test_file_rename**()

**test_help**()

**test_list**()

**test_max_retries**()
client should raise an error when max retries are reached.

**test_mdtm**()

**test_mkd**()

**test_mode**()

**test_nlist**()

**test_noop**()

**test_pwd**()

**test_rein**()

**test_rest**()

**test_retr**()
Test retr or downloading a file from the server.

**test_rmd**()

**test_site**()

**test_site_chmod**()

**test_site_help**()

**test_size**()

**test_stat**()

**test_stor**()

**test_stou**()

**test_stru**()

**test_syst**()

**test_type**()

### conpot.tests.test_guardian_ast module

**class** conpot.tests.test_guardian_ast.**TestGuardianAST**(*methodName='runTest'*)
Bases: unittest.case.TestCase

**setUp**()
Hook method for setting up the test fixture before exercising it.

**tearDown**()
Hook method for deconstructing the test fixture after testing it.

**test_I20100**()

**test_I20200**()

**test_I20300**()

**test_I20400**()

**test_I20500**()

**test_S60200**()

**test_S60201**()

**test_S60202**()

**test_S60203**()

**test_S60204**()

**test_ast_error**()

### conpot.tests.test_hpfriends module

**class** conpot.tests.test_hpfriends.**Test_HPFriends**(*methodName='runTest'*)
Bases: unittest.case.TestCase

**test_hpfriends**()
Objective: Test if data can be published to hpfriends without errors.

### conpot.tests.test_http_server module

**class** conpot.tests.test_http_server.**TestHTTPServer**(*methodName='runTest'*)
Bases: unittest.case.TestCase

**setUp**()
Hook method for setting up the test fixture before exercising it.

**tearDown**()
Hook method for deconstructing the test fixture after testing it.

**test_do_HEAD**()
Objective: Test the web server by sending a HTTP HEAD request. Should be responded back by the valid HTTP headers

**test_do_OPTIONS**()
Objective: Test the web server by sending a valid OPTIONS HTTP request

**test_do_POST**()
Objective: send a POST request to a invalid URI. Should get a 404 response

**test_do_TRACE**()
Objective: Test the web server with a trace request

**test_http_backend_databus**()
Objective: Test if http backend is able to retrieve data from databus

**test_http_backend_tarpit**()
Objective: Test if http tarpit delays responses properly

**test_http_request_base**()
Objective: Test if http service delivers data on request

**test_http_subselect_trigger**()
Objective: Test if http subselect triggers work correctly

**test_not_implemented_method**()
Objective: PUT HTTP method is not implemented in Conpot, should raise 501

### conpot.tests.test_iec104_server module

**class** conpot.tests.test_iec104_server.**TestIEC104Server**(*methodName='runTest'*)

Bases: unittest.case.TestCase

**setUp**()

Hook method for setting up the test fixture before exercising it.

**tearDown**()

Hook method for deconstructing the test fixture after testing it.

**test_startdt**()

Objective: Test if answered correctly to STARTDT act

**test_testfr**()

Objective: Test if answered correctly to TESTFR act

**test_write_for_non_existing**()

Objective: Test answer for a command to a device that doesn't exist (Correct behaviour of the IEC104 protocol is not known exactly. Other case is test for no answer)

**test_write_no_relation_for_existing**()

Objective: Test answer for a correct command to a device that does exist and has no related sensor (Actuator 22_19 (Type 45: Single Command) will be tested, the corresponding(!) sensor is not existent)

**test_write_relation_for_existing**()

Objective: Test answer for a correct command to a device that does exist and has a related sensor (Actuator 22_20 (Type 45: Single Command) will be tested, the corresponding(!) sensor 13_20 (Type 1: Single Point Information) changes the value and the termination confirmation is returned)

**test_write_wrong_type_for_existing**()

Objective: Test answer for a command of wrong type to a device that does exist (Actuator 22_20 (Type 45: Single Command) will be tested, but a wrong command type (Double Commands instead of Single Command) is sent to device)

### conpot.tests.test_ipmi_server module

**class** conpot.tests.test_ipmi_server.**TestIPMI**(*methodName='runTest'*)

Bases: unittest.case.TestCase

**setUp**()

Hook method for setting up the test fixture before exercising it.

**tearDown**()

Hook method for deconstructing the test fixture after testing it.

**test_boot_device**()

Objective: test boot device get and set

**test_channel_get_access**()

**test_chassis_status**()

**test_misc**()

**test_power_state**()

Objective: test power on/off/reset/cycle/shutdown

**test_user_list**()

conpot.tests.test_ipmi_server.**run_cmd**(*cmd*, *port*)

### conpot.tests.test_kamstrup_decoder module

**class** conpot.tests.test_kamstrup_decoder.**TestKamstrupDecoder**(*methodName='runTest'*)

Bases: unittest.case.TestCase

**test_invalid_crc**()

**test_request_one**()

### conpot.tests.test_kamstrup_management_protocol module

**class** conpot.tests.test_kamstrup_management_protocol.**TestKamstrupManagementProtocol**(*methodNa*

Bases: unittest.case.TestCase

All tests work in similar way. We send a get command check for a valid reply. We send in set command and expect things to change in the databus.

**setUp**()

Hook method for setting up the test fixture before exercising it.

**tearDown**()

Hook method for deconstructing the test fixture after testing it.

**test_access_control_command**()

**test_alarm_server_command**()

**test_get_config_command**()

**test_get_software_version_command**()

**test_help_command**()

**test_request_connect_command**()

**test_set_config_command**()

**test_set_device_name_command**()

**test_set_ip_command**()

**test_set_kap1_command**()

**test_set_kap2_command**()

**test_set_lookup_command**()

**test_set_name_server_command**()

**test_set_ports_command**()

**test_set_serial_command**()

**test_set_watchdog_command**()

conpot.tests.test_kamstrup_management_protocol.**check_command_resp_help_message**(*packet_type*,
*help_msg_comm*
*packet_msg_com*
*kam-*
*strup_manageme*

### conpot.tests.test_kamstrup_meter_protocol module

**class** conpot.tests.test_kamstrup_meter_protocol.**TestKamstrup**(*methodName='runTest'*)
Bases: unittest.case.TestCase

**setUp**()
Hook method for setting up the test fixture before exercising it.

**tearDown**()
Hook method for deconstructing the test fixture after testing it.

**test_request_get_register**()

### conpot.tests.test_logger_json module

**class** conpot.tests.test_logger_json.**TestJsonLogger**(*methodName='runTest'*)
Bases: unittest.case.TestCase

**setUp**()
Hook method for setting up the test fixture before exercising it.

**tearDown**()
Hook method for deconstructing the test fixture after testing it.

**test_log_event**()

### conpot.tests.test_logger_mysql module

**class** conpot.tests.test_logger_mysql.**Test_MySQLlogger**(*methodName='runTest'*)
Bases: unittest.case.TestCase

**test_mysqllogger**()

### conpot.tests.test_mac_addr module

**class** conpot.tests.test_mac_addr.**TestMacAddrUtil**(*methodName='runTest'*)
Bases: unittest.case.TestCase

**setUp**()
Hook method for setting up the test fixture before exercising it.

**tearDown**()
Hook method for deconstructing the test fixture after testing it.

**test_mac**()
Objective: Test if the spoofer is able to change MAC address

### conpot.tests.test_modbus_server module

**class** conpot.tests.test_modbus_server.**TestModbusServer**(*methodName='runTest'*)
Bases: unittest.case.TestCase

**setUp**()
Hook method for setting up the test fixture before exercising it.

**tearDown**()

Hook method for deconstructing the test fixture after testing it.

**test_modbus_logging**()

Objective: Test if modbus generates log messages as expected. Expected output is a dictionary with the following structure: {'timestamp': datetime.datetime(2013, 4, 23, 18, 47, 38, 532960),

> 'remote': ('127.0.0.1', 60991), 'data_type': 'modbus', 'id': '01bd90d6-76f4-43cb-874f-5c8f254367f5', 'data': {'function_code': 1,
>
> > 'slave_id': 1, 'request': '0100010080', 'response': '0110ffffffffffffffffffffffffffffffff'}}

**test_read_coils**()

Objective: Test if we can extract the expected bits from a slave using the modbus protocol.

**test_read_nonexistent_slave**()

Objective: Test if the correct exception is raised when trying to read from nonexistent slave.

**test_report_slave_id**()

Objective: Test conpot for function code 17.

**test_response_function_43_device_info**()

**test_write_read_coils**()

Objective: Test if we can change values using the modbus protocol.

### conpot.tests.test_proxy module

**class** `conpot.tests.test_proxy.`**TestProxy**(*methodName='runTest'*)

Bases: `unittest.case.TestCase`

**echo_server**(*sock*, *address*)

**test_ascii_decoder**()

**test_proxy**()

**test_proxy_with_decoder**()

**test_ssl_proxy**()

**test_ssl_proxy_with_decoder**()

### conpot.tests.test_pysnmp_wrapper module

**class** `conpot.tests.test_pysnmp_wrapper.`**TestPySNMPWrapper**(*methodName='runTest'*)

Bases: `unittest.case.TestCase`

**setUp**()

Hook method for setting up the test fixture before exercising it.

**test_compile**()

Tests that the wrapper can output mib files.

**test_find**()

Tests that the wrapper can find mib files.

**test_wrapper_output**()

Tests that the wrapper generates output that can be consumed by the command responder.

**test_wrapper_processing**()

Tests that the wrapper can process a valid mib file without errors.

conpot.tests.test_pysnmp_wrapper.**check_content**(*pyfile*)

### conpot.tests.test_s7_server module

**class** conpot.tests.test_s7_server.**TestS7Server**(*methodName='runTest'*)

Bases: unittest.case.TestCase

**setUp**()

Hook method for setting up the test fixture before exercising it.

**tearDown**()

Hook method for deconstructing the test fixture after testing it.

**test_s7**()

Objective: Test if the S7 server returns the values expected.

### conpot.tests.test_snmp_server module

**class** conpot.tests.test_snmp_server.**TestSNMPServer**(*methodName='runTest'*)

Bases: unittest.case.TestCase

**mock_callback**(*sendRequestHandle*, *errorIndication*, *errorStatus*, *errorIndex*, *varBindTable*, *cbCtx*)

**setUp**()

Hook method for setting up the test fixture before exercising it.

**tearDown**()

Hook method for deconstructing the test fixture after testing it.

**test_snmp_get**()

Objective: Test if we can get data via snmp_get

**test_snmp_set**()

Objective: Test if we can set data via snmp_set

### conpot.tests.test_taxii module

**class** conpot.tests.test_taxii.**TestLoggers**(*methodName='runTest'*)

Bases: unittest.case.TestCase

**test_stix_transform**()

Objective: Test if our STIX xml can be validated.

**test_taxii**()

Objective: Test if we can transmit data to MITRE's TAXII test server. Note: This actually also tests the StixTransformer since the event is parsed by the transformer before transmission.

### conpot.tests.test_tftp module

**class** conpot.tests.test_tftp.**TestTFTPServer**(*methodName='runTest'*)

Bases: unittest.case.TestCase

**setUp**()
Hook method for setting up the test fixture before exercising it.

**tearDown**()
Hook method for deconstructing the test fixture after testing it.

**test_mkdir_upload**()
Testing TFTP upload files - while recursively making directories as per the TFTP path.

**test_tftp_download**()

**test_tftp_upload**()
Testing TFTP upload files.

### conpot.tests.test_vfs module

Test core features for Conpot's virtual file system

**class** conpot.tests.test_vfs.**TestFileSystem**(*methodName='runTest'*)
Bases: unittest.case.TestCase

Tests related to Conpot's virtual file system.

**setUp**()
Hook method for setting up the test fixture before exercising it.

**tearDown**()
Hook method for deconstructing the test fixture after testing it.

**test_access**()

**test_chmod**()

**test_chown**()

**test_copydir**()

**test_copyfile**()

**test_format_list**()

**test_get_cwd**()

**test_get_permissions**()

**test_getmtime**()

**test_jail**()
Test for checking chroot jail a subfilesystem

**test_listdir**()

**test_mkdir**()

**test_mkdirs**()

**test_movedir**()

**test_movefile**()

**test_open_dir**()

**test_open_file**()

**test_openbin_file**()

**test_readlink**()

**test_remove**()

**test_removedir**()

**test_snapshot**()

**test_stat**()

**test_utime**()

**class** conpot.tests.test_vfs.**TestSubFileSystem**(*methodName='runTest'*)

Bases: unittest.case.TestCase

Tests related to Conpot's virtual sub file system. This would test fs generated folders for each and every protocol.

**setUp**()

Hook method for setting up the test fixture before exercising it.

**tearDown**()

Hook method for deconstructing the test fixture after testing it.

**test_access**()

**test_chmod**()

**test_chown**()

**test_format_list**()

**test_get_cwd**()

**test_get_permissions**()

**test_listdir**()

**test_mkdir**()

**test_mkdirs**()

**test_open_file**()

**test_readlink**()

**test_remove**()

**test_removedir**()

**test_set_time**()

Test for changing time in the file system.

**test_stat**()

**test_utime**()

### Module contents

### conpot.utils package

### Submodules

### conpot.utils.ext_ip module

conpot.utils.ext_ip.**get_ext_ip**(*config=None*, *urls=None*)

conpot.utils.ext_ip.**get_interface_ip**(*destination_ip: str*)

### conpot.utils.mac_addr module

conpot.utils.mac_addr.**change_mac**(*iface=None*, *mac=None*, *config=None*, *revert=None*)

conpot.utils.mac_addr.**revert_mac**(*iface*)

### Module contents

### Submodules

### conpot.helpers module

Some python3 fixtures - helper methods for handy conversions + fix ssl

conpot.helpers.**chr_py3**(*x*)

conpot.helpers.**fix_sslwrap**()

conpot.helpers.**number_to_bytes**(*x*)

conpot.helpers.**pack_short_int**(*x*)

conpot.helpers.**sanitize_file_name**(*name*, *host*, *port*)

Ensure that file_name is legal. Slug the filename and store it onto the server. This would ensure that there are no duplicates as far as writing a file is concerned. Also client addresses are noted so that one can verify which client uploaded the file. :param name: Name of the file :param host: host/client address :param port port/client port :type name: str

conpot.helpers.**str_to_bytes**(*x*)

conpot.helpers.**unpack_short_int**(*x*)

### Module contents

# Python Module Index

C

conpot, 72
conpot.core, 24
conpot.core.attack_session, 15
conpot.core.databus, 15
conpot.core.filesystem, 15
conpot.core.fs_utils, 21
conpot.core.internal_interface, 22
conpot.core.loggers, 15
conpot.core.loggers.helpers, 13
conpot.core.loggers.hpfriends, 13
conpot.core.loggers.json_log, 14
conpot.core.loggers.log_worker, 14
conpot.core.loggers.sqlite_log, 14
conpot.core.loggers.stix_transform, 14
conpot.core.loggers.syslog, 14
conpot.core.loggers.taxii_log, 14
conpot.core.protocol_wrapper, 23
conpot.core.session_manager, 23
conpot.core.virtual_fs, 23
conpot.emulators, 25
conpot.emulators.misc, 25
conpot.emulators.misc.random, 24
conpot.emulators.misc.uptime, 24
conpot.emulators.proxy, 25
conpot.emulators.sensors, 25
conpot.helpers, 72
conpot.protocols, 58
conpot.protocols.bacnet, 40
conpot.protocols.bacnet.bacnet_app, 39
conpot.protocols.bacnet.bacnet_server, 40
conpot.protocols.enip, 40
conpot.protocols.enip.enip_server, 40
conpot.protocols.ftp, 44
conpot.protocols.ftp.ftp_base_handler, 40
conpot.protocols.ftp.ftp_handler, 41
conpot.protocols.ftp.ftp_server, 43
conpot.protocols.ftp.ftp_utils, 44
conpot.protocols.guardian_ast, 44
conpot.protocols.guardian_ast.guardian_ast_server, 44
conpot.protocols.http, 46
conpot.protocols.http.command_responder, 44
conpot.protocols.http.web_server, 46
conpot.protocols.IEC104, 39
conpot.protocols.IEC104.DeviceDataController, 25
conpot.protocols.IEC104.errors, 26
conpot.protocols.IEC104.frames, 27
conpot.protocols.IEC104.i_frames_check, 38
conpot.protocols.IEC104.IEC104, 26
conpot.protocols.IEC104.IEC104_server, 26
conpot.protocols.IEC104.register, 39
conpot.protocols.ipmi, 47
conpot.protocols.ipmi.fakebmc, 46
conpot.protocols.ipmi.fakesession, 46
conpot.protocols.ipmi.ipmi_server, 47
conpot.protocols.kamstrup, 52
conpot.protocols.kamstrup.management_protocol, 50
conpot.protocols.kamstrup.management_protocol.comma 47
conpot.protocols.kamstrup.management_protocol.comma 47
conpot.protocols.kamstrup.management_protocol.kamst 50
conpot.protocols.kamstrup.meter_protocol, 52
conpot.protocols.kamstrup.meter_protocol.command_re 50
conpot.protocols.kamstrup.meter_protocol.decoder_38 50
conpot.protocols.kamstrup.meter_protocol.kamstrup_c 50

conpot.protocols.kamstrup.meter_protocol.kamstrup_server, 51
conpot.protocols.kamstrup.meter_protocol.messages, 51
conpot.protocols.kamstrup.meter_protocol.register, 52
conpot.protocols.kamstrup.meter_protocol.request_parser, 52
conpot.protocols.kamstrup.usage_simulator, 52
conpot.protocols.misc, 53
conpot.protocols.misc.ascii_decoder, 52
conpot.protocols.modbus, 53
conpot.protocols.modbus.modbus_block_databus_mediator, 53
conpot.protocols.modbus.modbus_server, 53
conpot.protocols.modbus.slave, 53
conpot.protocols.modbus.slave_db, 53
conpot.protocols.s7comm, 55
conpot.protocols.s7comm.cotp, 54
conpot.protocols.s7comm.exceptions, 54
conpot.protocols.s7comm.s7, 54
conpot.protocols.s7comm.s7_server, 55
conpot.protocols.s7comm.tpkt, 55
conpot.protocols.snmp, 57
conpot.protocols.snmp.build_pysnmp_mib_wrapper, 55
conpot.protocols.snmp.command_responder, 56
conpot.protocols.snmp.conpot_cmdrsp, 56
conpot.protocols.snmp.databus_mediator, 57
conpot.protocols.snmp.snmp_server, 57
conpot.protocols.tftp, 58
conpot.protocols.tftp.tftp_handler, 57
conpot.protocols.tftp.tftp_server, 58
conpot.tests, 71
conpot.tests.helpers, 60
conpot.tests.helpers.s7comm_client, 59
conpot.tests.helpers.snmp_client, 60
conpot.tests.test_bacnet_server, 60
conpot.tests.test_base, 61
conpot.tests.test_docs, 61
conpot.tests.test_enip_server, 61
conpot.tests.test_ext_ip_util, 62
conpot.tests.test_ftp, 62
conpot.tests.test_guardian_ast, 63
conpot.tests.test_hpfriends, 64
conpot.tests.test_http_server, 64
conpot.tests.test_iec104_server, 65
conpot.tests.test_ipmi_server, 65
conpot.tests.test_kamstrup_decoder, 66
conpot.tests.test_kamstrup_management_protocol, 66
conpot.tests.test_kamstrup_meter_protocol, 67
conpot.tests.test_logger_json, 67
conpot.tests.test_logger_mysql, 67
conpot.tests.test_mac_addr, 67
conpot.tests.test_modbus_server, 67
conpot.tests.test_proxy, 68
conpot.tests.test_pysnmp_wrapper, 68
conpot.tests.test_s7_server, 69
conpot.tests.test_snmp_server, 69
conpot.tests.test_taxii, 69
conpot.tests.test_tftp, 69
conpot.tests.test_vfs, 70
conpot.utils, 72
conpot.utils.ext_ip, 72
conpot.utils.mac_addr, 72

# Index

## A

AbstractFS (*class in conpot.core.filesystem*), 15
access() (*conpot.core.filesystem.AbstractFS method*), 15
access() (*conpot.core.fs_utils.SubAbstractFS method*), 21
AccessControlCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 47
add_block() (*conpot.protocols.modbus.slave.MBSlave method*), 53
add_byte() (*conpot.protocols.kamstrup.meter_protocol.request_parser.KamstrupRequestParser method*), 52
add_event() (*conpot.core.attack_session.AttackSession method*), 15
add_object() (*conpot.protocols.bacnet.bacnet_app.BACnetApp method*), 39
add_property() (*conpot.protocols.bacnet.bacnet_app.BACnetApp method*), 39
add_protocol() (*conpot.core.virtual_fs.VirtualFS method*), 23
add_protocol() (*in module conpot.core*), 24
add_register() (*conpot.protocols.kamstrup.meter_protocol.messages.KamstrupResponseRegister method*), 51
add_slave() (*conpot.protocols.modbus.slave_db.SlaveBase method*), 53
add_users_to_group() (*conpot.core.filesystem.AbstractFS method*), 16
AddOptions() (*in module conpot.tests.helpers.s7comm_client*), 59
addr_in_hex() (*in module conpot.protocols.IEC104.DeviceDataController*), 25
addSocketTransport() (*conpot.protocols.snmp.command_responder.CommandResponder method*), 56
AlarmServerCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 48
aliastypes (*conpot.protocols.IEC104.frames.asdu_head attribute*), 30
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_1 attribute*), 30
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_10 attribute*), 30
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_100 attribute*), 31
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_101 attribute*), 31
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_102 attribute*), 31
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_103 attribute*), 31
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_11 attribute*), 31
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_12 attribute*), 31
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_13 attribute*), 31
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_14 attribute*), 32
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_15 attribute*), 32
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_16 attribute*), 32
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_17 attribute*), 32
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_18 attribute*), 32
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_19 attribute*), 32
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_2 attribute*), 32
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_20 attribute*), 33

aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_21 attribute*), 33
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_3 attribute*), 33
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_30 attribute*), 33
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_31 attribute*), 33
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_32 attribute*), 33
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_33 attribute*), 33
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_34 attribute*), 34
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_35 attribute*), 34
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_36 attribute*), 34
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_37 attribute*), 34
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_38 attribute*), 34
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_39 attribute*), 34
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_4 attribute*), 34
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_40 attribute*), 35
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_45 attribute*), 35
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_46 attribute*), 35
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_47 attribute*), 35
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_48 attribute*), 35
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_49 attribute*), 35
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_5 attribute*), 35
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_50 attribute*), 36
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_51 attribute*), 36
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_58 attribute*), 36
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_59 attribute*), 36
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_6 attribute*), 36
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_60 attribute*), 36
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_61 attribute*), 36
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_62 attribute*), 37
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_63 attribute*), 37
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_64 attribute*), 37
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_7 attribute*), 37
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_8 attribute*), 37
aliastypes (*conpot.protocols.IEC104.frames.asdu_infobj_9 attribute*), 37
aliastypes (*conpot.protocols.IEC104.frames.BCR attribute*), 27
aliastypes (*conpot.protocols.IEC104.frames.BSI attribute*), 27
aliastypes (*conpot.protocols.IEC104.frames.CP16Time attribute*), 27
aliastypes (*conpot.protocols.IEC104.frames.CP24Time attribute*), 27
aliastypes (*conpot.protocols.IEC104.frames.CP56Time attribute*), 28
aliastypes (*conpot.protocols.IEC104.frames.DIQ attribute*), 28
aliastypes (*conpot.protocols.IEC104.frames.i_frame attribute*), 37
aliastypes (*conpot.protocols.IEC104.frames.IOA attribute*), 28
aliastypes (*conpot.protocols.IEC104.frames.NVA attribute*), 28
aliastypes (*conpot.protocols.IEC104.frames.OCI attribute*), 28
aliastypes (*conpot.protocols.IEC104.frames.QDP attribute*), 28
aliastypes (*conpot.protocols.IEC104.frames.QDS attribute*), 29
aliastypes (*conpot.protocols.IEC104.frames.QOS attribute*), 29
aliastypes (*conpot.protocols.IEC104.frames.s_frame attribute*), 38
aliastypes (*conpot.protocols.IEC104.frames.SCD attribute*), 29
aliastypes (*conpot.protocols.IEC104.frames.SEP attribute*), 29
aliastypes (*conpot.protocols.IEC104.frames.SIQ attribute*), 29
aliastypes (*conpot.protocols.IEC104.frames.SPE attribute*), 30
aliastypes (*conpot.protocols.IEC104.frames.SVA attribute*), 30
aliastypes (*conpot.protocols.IEC104.frames.u_frame attribute*), 38
aliastypes (*conpot.protocols.IEC104.frames.VTI attribute*), 30

AsciiDecoder (*class in conpot.protocols.misc.ascii_decoder*), 52

asdu_head (*class in conpot.protocols.IEC104.frames*), 30

asdu_infobj_1 (*class in conpot.protocols.IEC104.frames*), 30

asdu_infobj_10 (*class in conpot.protocols.IEC104.frames*), 30

asdu_infobj_100 (*class in conpot.protocols.IEC104.frames*), 31

asdu_infobj_101 (*class in conpot.protocols.IEC104.frames*), 31

asdu_infobj_102 (*class in conpot.protocols.IEC104.frames*), 31

asdu_infobj_103 (*class in conpot.protocols.IEC104.frames*), 31

asdu_infobj_11 (*class in conpot.protocols.IEC104.frames*), 31

asdu_infobj_12 (*class in conpot.protocols.IEC104.frames*), 31

asdu_infobj_13 (*class in conpot.protocols.IEC104.frames*), 31

asdu_infobj_14 (*class in conpot.protocols.IEC104.frames*), 32

asdu_infobj_15 (*class in conpot.protocols.IEC104.frames*), 32

asdu_infobj_16 (*class in conpot.protocols.IEC104.frames*), 32

asdu_infobj_17 (*class in conpot.protocols.IEC104.frames*), 32

asdu_infobj_18 (*class in conpot.protocols.IEC104.frames*), 32

asdu_infobj_19 (*class in conpot.protocols.IEC104.frames*), 32

asdu_infobj_2 (*class in conpot.protocols.IEC104.frames*), 32

asdu_infobj_20 (*class in conpot.protocols.IEC104.frames*), 32

asdu_infobj_21 (*class in conpot.protocols.IEC104.frames*), 33

asdu_infobj_3 (*class in conpot.protocols.IEC104.frames*), 33

asdu_infobj_30 (*class in conpot.protocols.IEC104.frames*), 33

asdu_infobj_31 (*class in conpot.protocols.IEC104.frames*), 33

asdu_infobj_32 (*class in conpot.protocols.IEC104.frames*), 33

asdu_infobj_33 (*class in conpot.protocols.IEC104.frames*), 33

asdu_infobj_34 (*class in conpot.protocols.IEC104.frames*), 33

asdu_infobj_35 (*class in conpot.protocols.IEC104.frames*), 34

asdu_infobj_36 (*class in conpot.protocols.IEC104.frames*), 34

asdu_infobj_37 (*class in conpot.protocols.IEC104.frames*), 34

asdu_infobj_38 (*class in conpot.protocols.IEC104.frames*), 34

asdu_infobj_39 (*class in conpot.protocols.IEC104.frames*), 34

asdu_infobj_4 (*class in conpot.protocols.IEC104.frames*), 34

asdu_infobj_40 (*class in conpot.protocols.IEC104.frames*), 34

asdu_infobj_45 (*class in conpot.protocols.IEC104.frames*), 35

asdu_infobj_46 (*class in conpot.protocols.IEC104.frames*), 35

asdu_infobj_47 (*class in conpot.protocols.IEC104.frames*), 35

asdu_infobj_48 (*class in conpot.protocols.IEC104.frames*), 35

asdu_infobj_49 (*class in conpot.protocols.IEC104.frames*), 35

asdu_infobj_5 (*class in conpot.protocols.IEC104.frames*), 35

asdu_infobj_50 (*class in conpot.protocols.IEC104.frames*), 35

asdu_infobj_51 (*class in conpot.protocols.IEC104.frames*), 36

asdu_infobj_58 (*class in conpot.protocols.IEC104.frames*), 36

asdu_infobj_59 (*class in conpot.protocols.IEC104.frames*), 36

asdu_infobj_6 (*class in conpot.protocols.IEC104.frames*), 36

asdu_infobj_60 (*class in conpot.protocols.IEC104.frames*), 36

asdu_infobj_61 (*class in conpot.protocols.IEC104.frames*), 36

asdu_infobj_62 (*class in conpot.protocols.IEC104.frames*), 36

asdu_infobj_63 (*class in conpot.protocols.IEC104.frames*), 37

asdu_infobj_64 (*class in conpot.protocols.IEC104.frames*), 37

asdu_infobj_7 (*class in conpot.protocols.IEC104.frames*), 37

asdu_infobj_8 (*class in conpot.protocols.IEC104.frames*), 37

asdu_infobj_9 (*class in conpot.protocols.IEC104.frames*), 37

assemble() (*conpot.protocols.s7comm.cotp.COTP_ConnectionConfirm method*), 54

assemble() (*conpot.protocols.s7comm.cotp.COTP_ConnectionRequest method*), 54

AssembleException, 54
AttackSession (*class in conpot.core.attack_session*), 15
attribute_operations() (*conpot.tests.test_enip_server.TestENIPServer method*), 61
authentication_ok() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 40

B

BACnetApp (*class in conpot.protocols.bacnet.bacnet_app*), 39
BaseCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 48
BCR (*class in conpot.protocols.IEC104.frames*), 27
BruteTsap() (*in module conpot.tests.helpers.s7comm_client*), 59
BSI (*class in conpot.protocols.IEC104.frames*), 27
build() (*conpot.protocols.IEC104.IEC104.frame_object_with_timer method*), 26

C

c_BulkCommandResponder (*class in conpot.protocols.snmp.conpot_cmdrsp*), 56
c_GetCommandResponder (*class in conpot.protocols.snmp.conpot_cmdrsp*), 56
c_NextCommandResponder (*class in conpot.protocols.snmp.conpot_cmdrsp*), 56
c_SetCommandResponder (*class in conpot.protocols.snmp.conpot_cmdrsp*), 56
calctime() (*in module conpot.protocols.IEC104.frames*), 37
cancel_t1() (*conpot.protocols.IEC104.IEC104.frame_object_with_timer method*), 26
cbFun() (*conpot.tests.helpers.snmp_client.SNMPClient method*), 60
change_mac() (*in module conpot.utils.mac_addr*), 72
check_access() (*conpot.core.filesystem.AbstractFS method*), 16
check_access() (*conpot.core.fs_utils.SubAbstractFS method*), 21
check_asdu_1() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_100() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_11() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_12() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_13() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_14() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_2() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_3() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_30() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_31() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_35() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_36() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_4() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_45() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_46() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_47() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_48() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_49() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_50() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_asdu_51() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_command() (*in module conpot.protocols.IEC104.i_frames_check*), 38
check_command_resp_help_message() (*in module conpot.tests.test_kamstrup_management_protocol*), 66
check_content() (*in module conpot.tests.test_pysnmp_wrapper*), 69
check_evasive() (*conpot.protocols.snmp.conpot_cmdrsp.conpot_extension method*), 57
check_information_with_time() (*in module conpot.protocols.IEC104.i_frames_check*), 39
check_information_without_time() (*in module conpot.protocols.IEC104.i_frames_check*), 39
check_registers() (*conpot.protocols.IEC104.DeviceDataController.DeviceDataControlle method*), 25
chmod() (*conpot.core.filesystem.AbstractFS method*), 16
chmod() (*conpot.core.fs_utils.SubAbstractFS method*), 21
chown() (*conpot.core.filesystem.AbstractFS method*),

16
chown() (conpot.core.fs_utils.SubAbstractFS method), 21
chr_py3() (in module conpot.helpers), 72
clean() (conpot.core.filesystem.AbstractFS method), 16
cleanse_byte_string() (in module conpot.protocols.s7comm.s7_server), 55
close() (conpot.core.virtual_fs.VirtualFS method), 23
close_fs() (in module conpot.core), 24
close_server_session() (conpot.protocols.ipmi.ipmi_server.IpmiServer method), 47
CMD_OUTPUT (conpot.protocols.kamstrup.management_protocol.commands.AccessControlCommand attribute), 47
CMD_OUTPUT (conpot.protocols.kamstrup.management_protocol.commands.AlarmServerCommand attribute), 48
CMD_OUTPUT (conpot.protocols.kamstrup.management_protocol.commands.BaseCommand attribute), 48
CMD_OUTPUT (conpot.protocols.kamstrup.management_protocol.commands.GetConfigCommand attribute), 48
CMD_OUTPUT (conpot.protocols.kamstrup.management_protocol.commands.HelpCommand attribute), 48
CMD_OUTPUT (conpot.protocols.kamstrup.management_protocol.commands.SetConfigCommand attribute), 48
CMD_OUTPUT (conpot.protocols.kamstrup.management_protocol.commands.SetIPCommand attribute), 49
CMD_OUTPUT (conpot.protocols.kamstrup.management_protocol.commands.SetKap1Command attribute), 49
CMD_OUTPUT (conpot.protocols.kamstrup.management_protocol.commands.SetPortsCommand attribute), 49
CMD_OUTPUT (conpot.protocols.kamstrup.management_protocol.commands.SetWatchlogCommand attribute), 49
CMD_OUTPUT (conpot.protocols.kamstrup.management_protocol.commands.SoftwareVersionCommand attribute), 50
CMD_OUTPUT (conpot.protocols.kamstrup.management_protocol.commands.WinkModuleCommand attribute), 50
CMD_OUTPUT_DOUBLE (conpot.protocols.kamstrup.management_protocol.commands.SetKap2Command attribute), 49
CMD_OUTPUT_SINGLE (conpot.protocols.kamstrup.management_protocol.commands.SetKap2Command attribute), 49
cold_reset() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 46
command_byte (conpot.protocols.kamstrup.meter_protocol.messages.KamstrupRequestGetRegisters attribute), 51
COMMAND_NOT_FOUND (conpot.protocols.kamstrup.management_protocol.command_responder.CommandResponder attribute), 47
CommandResponder (class in conpot.protocols.http.command_responder), 44
CommandResponder (class in conpot.protocols.kamstrup.management_protocol.command_responder), 47
CommandResponder (class in conpot.protocols.kamstrup.meter_protocol.command_responder), 50
CommandResponder (class in conpot.protocols.snmp.command_responder), 56
compile_mib() (in module conpot.protocols.snmp.build_pysnmp_mib_wrapper), 55
config (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase attribute), 40
config_sanitize_tarpit() (conpot.protocols.http.command_responder.SubHTTPServer method), 45
Connect() (conpot.tests.helpers.s7comm_client.s7 method), 60
conpot (module), 72
conpot.core (module), 24
conpot.core.attack_session (module), 15
conpot.core.databus (module), 15
conpot.core.filesystem (module), 15
conpot.core.fs_utils (module), 21
conpot.core.internal_interface (module), 22
conpot.core.loggers (module), 15
conpot.core.loggers.helpers (module), 13
conpot.core.loggers.hpfriends (module), 13
conpot.core.loggers.json_log (module), 14
conpot.core.loggers.log_worker (module), 14
conpot.core.loggers.sqlite_log (module), 14
conpot.core.loggers.stix_transform (module), 14
conpot.core.loggers.syslog (module), 14
conpot.core.loggers.taxii_log (module), 14
conpot.core.protocol_wrapper (module), 23
conpot.core.session_manager (module), 23
conpot.core.virtual_fs (module), 23
conpot.emulators (module), 25
conpot.emulators.misc (module), 25
conpot.emulators.misc.random (module), 24
conpot.emulators.misc.uptime (module), 24
conpot.emulators.proxy (module), 25
conpot.emulators.sensors (module), 25
conpot.helpers (module), 72
conpot.protocols (module), 58
conpot.protocols.bacnet (module), 40
conpot.protocols.bacnet.bacnet_app (module), 39

conpot.protocols.bacnet.bacnet_server (*module*), 40
conpot.protocols.enip (*module*), 40
conpot.protocols.enip.enip_server (*module*), 40
conpot.protocols.ftp (*module*), 44
conpot.protocols.ftp.ftp_base_handler (*module*), 40
conpot.protocols.ftp.ftp_handler (*module*), 41
conpot.protocols.ftp.ftp_server (*module*), 43
conpot.protocols.ftp.ftp_utils (*module*), 44
conpot.protocols.guardian_ast (*module*), 44
conpot.protocols.guardian_ast.guardian_ast_server (*module*), 44
conpot.protocols.http (*module*), 46
conpot.protocols.http.command_responder (*module*), 44
conpot.protocols.http.web_server (*module*), 46
conpot.protocols.IEC104 (*module*), 39
conpot.protocols.IEC104.DeviceDataController (*module*), 25
conpot.protocols.IEC104.errors (*module*), 26
conpot.protocols.IEC104.frames (*module*), 27
conpot.protocols.IEC104.i_frames_check (*module*), 38
conpot.protocols.IEC104.IEC104 (*module*), 26
conpot.protocols.IEC104.IEC104_server (*module*), 26
conpot.protocols.IEC104.register (*module*), 39
conpot.protocols.ipmi (*module*), 47
conpot.protocols.ipmi.fakebmc (*module*), 46
conpot.protocols.ipmi.fakesession (*module*), 46
conpot.protocols.ipmi.ipmi_server (*module*), 47
conpot.protocols.kamstrup (*module*), 52
conpot.protocols.kamstrup.management_protocol (*module*), 50
conpot.protocols.kamstrup.management_protocol.command_responder (*module*), 47
conpot.protocols.kamstrup.management_protocol.commands (*module*), 47
conpot.protocols.kamstrup.management_protocol.kamstrup_management_server (*module*), 50
conpot.protocols.kamstrup.meter_protocol (*module*), 52

conpot.protocols.kamstrup.meter_protocol.command_re
(*module*), 50
conpot.protocols.kamstrup.meter_protocol.decoder_38
(*module*), 50
conpot.protocols.kamstrup.meter_protocol.kamstrup_c
(*module*), 50
conpot.protocols.kamstrup.meter_protocol.kamstrup_s
(*module*), 51
conpot.protocols.kamstrup.meter_protocol.messages (*module*), 51
conpot.protocols.kamstrup.meter_protocol.register (*module*), 52
conpot.protocols.kamstrup.meter_protocol.request_pa
(*module*), 52
conpot.protocols.kamstrup.usage_simulator (*module*), 52
conpot.protocols.misc (*module*), 53
conpot.protocols.misc.ascii_decoder (*module*), 52
conpot.protocols.modbus (*module*), 53
conpot.protocols.modbus.modbus_block_databus_mediat
(*module*), 53
conpot.protocols.modbus.modbus_server (*module*), 53
conpot.protocols.modbus.slave (*module*), 53
conpot.protocols.modbus.slave_db (*module*), 53
conpot.protocols.s7comm (*module*), 55
conpot.protocols.s7comm.cotp (*module*), 54
conpot.protocols.s7comm.exceptions (*module*), 54
conpot.protocols.s7comm.s7 (*module*), 54
conpot.protocols.s7comm.s7_server (*module*), 55
conpot.protocols.s7comm.tpkt (*module*), 55
conpot.protocols.snmp (*module*), 57
conpot.protocols.snmp.build_pysnmp_mib_wrapper (*module*), 55
conpot.protocols.snmp.command_responder (*module*), 56
conpot.protocols.snmp.conpot_cmdrsp (*module*), 56
conpot.protocols.snmp.databus_mediator (*module*), 57
conpot.protocols.snmp.snmp_server (*module*), 57
conpot.protocols.tftp (*module*), 58
conpot.protocols.tftp.tftp_handler (*module*), 57
conpot.protocols.tftp.tftp_server (*module*), 58
conpot.tests (*module*), 71
conpot.tests.helpers (*module*), 60
conpot.tests.helpers.s7comm_client (*mod-*

*module*), 59

conpot.tests.helpers.snmp_client (*module*), 60

conpot.tests.test_bacnet_server (*module*), 60

conpot.tests.test_base (*module*), 61

conpot.tests.test_docs (*module*), 61

conpot.tests.test_enip_server (*module*), 61

conpot.tests.test_ext_ip_util (*module*), 62

conpot.tests.test_ftp (*module*), 62

conpot.tests.test_guardian_ast (*module*), 63

conpot.tests.test_hpfriends (*module*), 64

conpot.tests.test_http_server (*module*), 64

conpot.tests.test_iec104_server (*module*), 65

conpot.tests.test_ipmi_server (*module*), 65

conpot.tests.test_kamstrup_decoder (*module*), 66

conpot.tests.test_kamstrup_management_protocol (*module*), 66

conpot.tests.test_kamstrup_meter_protocol (*module*), 67

conpot.tests.test_logger_json (*module*), 67

conpot.tests.test_logger_mysql (*module*), 67

conpot.tests.test_mac_addr (*module*), 67

conpot.tests.test_modbus_server (*module*), 67

conpot.tests.test_proxy (*module*), 68

conpot.tests.test_pysnmp_wrapper (*module*), 68

conpot.tests.test_s7_server (*module*), 69

conpot.tests.test_snmp_server (*module*), 69

conpot.tests.test_taxii (*module*), 69

conpot.tests.test_tftp (*module*), 69

conpot.tests.test_vfs (*module*), 70

conpot.utils (*module*), 72

conpot.utils.ext_ip (*module*), 72

conpot.utils.mac_addr (*module*), 72

conpot_extension (*class in conpot.protocols.snmp.conpot_cmdrsp*), 57

conpot_protocol() (*in module conpot.core.protocol_wrapper*), 23

copy() (*conpot.core.filesystem.AbstractFS method*), 16

copy_files() (*in module conpot.core.fs_utils*), 22

COTP (*class in conpot.protocols.s7comm.cotp*), 54

COTP_ConnectionConfirm (*class in conpot.protocols.s7comm.cotp*), 54

COTP_ConnectionRequest (*class in conpot.protocols.s7comm.cotp*), 54

COTPConnectionPacket (*class in conpot.protocols.s7comm.cotp*), 54

COTPConnectionPacket (*class in conpot.tests.helpers.s7comm_client*), 59

COTPDataPacket (*class in conpot.tests.helpers.s7comm_client*), 59

CP16Time (*class in conpot.protocols.IEC104.frames*), 27

CP24Time (*class in conpot.protocols.IEC104.frames*), 27

CP56Time (*class in conpot.protocols.IEC104.frames*), 28

create_group() (*conpot.core.filesystem.AbstractFS method*), 16

create_jail() (*conpot.core.filesystem.AbstractFS method*), 16

## D

daemon_threads (*conpot.protocols.http.command_responder.SubHTTPServer attribute*), 45

data_fs (*conpot.protocols.tftp.tftp_handler.TFTPServerState attribute*), 58

Databus (*class in conpot.core.databus*), 15

DatabusMediator (*class in conpot.protocols.snmp.databus_mediator*), 57

decode_in() (*conpot.emulators.proxy.ProxyDecoder method*), 25

decode_in() (*conpot.protocols.kamstrup.meter_protocol.decoder_382. method*), 50

decode_in() (*conpot.protocols.misc.ascii_decoder.AsciiDecoder method*), 52

decode_out() (*conpot.emulators.proxy.ProxyDecoder method*), 25

decode_out() (*conpot.protocols.kamstrup.meter_protocol.decoder_382.Decoder382 method*), 50

decode_out() (*conpot.protocols.misc.ascii_decoder.AsciiDecoder method*), 52

Decoder382 (*class in conpot.protocols.kamstrup.meter_protocol.decoder_382*), 50

default_gid (*conpot.core.fs_utils.SubAbstractFS attribute*), 21

default_group (*conpot.core.fs_utils.SubAbstractFS attribute*), 21

default_perms (*conpot.core.fs_utils.SubAbstractFS attribute*), 21

default_uid (*conpot.core.fs_utils.SubAbstractFS attribute*), 21

default_user (*conpot.core.fs_utils.SubAbstractFS attribute*), 21

DeviceDataController (*class in conpot.protocols.IEC104.DeviceDataController*),

25
DIQ (class in conpot.protocols.IEC104.frames), 28
disconnect() (conpot.protocols.IEC104.IEC104.IEC104 method), 26
dissect() (conpot.protocols.s7comm.cotp.COTPConnectionPacket method), 54
do_ABOR() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 41
do_ALLO() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 41
do_APPE() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 41
do_BYE() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 41
do_CDUP() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_CWD() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_DELE() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_GET() (conpot.protocols.http.command_responder.HTTPServer method), 44
do_HEAD() (conpot.protocols.http.command_responder.HTTPServer method), 44
do_HELP() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_LIST() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_MDTM() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_MKD() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_MODE() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_NLST() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_NOOP() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_OPTIONS() (conpot.protocols.http.command_responder.HTTPServer method), 44
do_PASS() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_PASV() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_PORT() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_POST() (conpot.protocols.http.command_responder.HTTPServer method), 45
do_PWD() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_QUIT() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_REIN() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_REST() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_RETR() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_RMD() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_RNFR() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_RNTO() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_SITE_CHMOD() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 42
do_SITE_HELP() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_SIZE() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_STAT() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_STOR() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_STOU() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_STRU() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_SYST() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_tarpit() (conpot.protocols.http.command_responder.SubHTTPServe method), 45
do_tarpit() (conpot.protocols.snmp.conpot_cmdrsp.conpot_extension method), 57
do_TRACE() (conpot.protocols.http.command_responder.HTTPServer method), 45
do_TYPE() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_USER() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_XCUP() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_XCWD() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_XMKD() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_XPWD() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
do_XRMD() (conpot.protocols.ftp.ftp_handler.FTPCommandChannel method), 43
DotDict (class in conpot.core.internal_interface), 22
dump() (conpot.core.attack_session.AttackSession method), 15

E

echo_server() (*conpot.tests.test_proxy.TestProxy method*), 68

enabled (*conpot.core.internal_interface.Interface attribute*), 22

end() (*conpot.protocols.tftp.tftp_handler.TFTPContextServer method*), 57

EnipConfig (*class in conpot.protocols.enip.enip_server*), 40

EnipConfig.Tag (*class in conpot.protocols.enip.enip_server*), 40

escape() (*conpot.protocols.kamstrup.meter_protocol.messages.KamstrupResponseBase class method*), 51

extract_padding() (*conpot.protocols.IEC104.frames.CP16Time method*), 27

extract_padding() (*conpot.protocols.IEC104.frames.CP24Time method*), 27

extract_padding() (*conpot.protocols.IEC104.frames.QDP method*), 28

extract_padding() (*conpot.protocols.IEC104.frames.QDS method*), 29

extract_padding() (*conpot.protocols.IEC104.frames.SEP method*), 29

extract_padding() (*conpot.protocols.IEC104.frames.SPE method*), 30

F

FakeBmc (*class in conpot.protocols.ipmi.fakebmc*), 46

FakeSession (*class in conpot.protocols.ipmi.fakesession*), 46

fields_desc (*conpot.protocols.IEC104.frames.asdu_head attribute*), 30

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_1 attribute*), 30

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_10 attribute*), 30

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_100 attribute*), 31

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_101 attribute*), 31

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_102 attribute*), 31

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_103 attribute*), 31

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_11 attribute*), 31

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_12 attribute*), 31

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_13 attribute*), 32

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_14 attribute*), 32

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_15 attribute*), 32

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_16 attribute*), 32

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_17 attribute*), 32

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_18 attribute*), 32

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_19 attribute*), 32

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_2 attribute*), 32

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_20 attribute*), 33

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_21 attribute*), 33

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_3 attribute*), 33

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_30 attribute*), 33

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_31 attribute*), 33

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_32 attribute*), 33

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_33 attribute*), 33

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_34 attribute*), 34

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_35 attribute*), 34

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_36 attribute*), 34

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_37 attribute*), 34

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_38 attribute*), 34

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_39 attribute*), 34

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_4 attribute*), 34

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_40 attribute*), 35

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_45 attribute*), 35

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_46 attribute*), 35

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_47 attribute*), 35

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_48 attribute*), 35 fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_49 attribute*), 35

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_5 attribute*), 35

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_50 attribute*), 36

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_51 attribute*), 36

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_58 attribute*), 36

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_59 attribute*), 36

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_6 attribute*), 36

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_60 attribute*), 36

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_61 attribute*), 36

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_62 attribute*), 37

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_63 attribute*), 37

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_64 attribute*), 37

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_7 attribute*), 37

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_8 attribute*), 37

fields_desc (*conpot.protocols.IEC104.frames.asdu_infobj_9 attribute*), 37

fields_desc (*conpot.protocols.IEC104.frames.BCR attribute*), 27

fields_desc (*conpot.protocols.IEC104.frames.BSI attribute*), 27

fields_desc (*conpot.protocols.IEC104.frames.CP16Time attribute*), 27

fields_desc (*conpot.protocols.IEC104.frames.CP24Time attribute*), 28

fields_desc (*conpot.protocols.IEC104.frames.CP56Time attribute*), 28

fields_desc (*conpot.protocols.IEC104.frames.DIQ attribute*), 28

fields_desc (*conpot.protocols.IEC104.frames.i_frame attribute*), 37

fields_desc (*conpot.protocols.IEC104.frames.IOA attribute*), 28

fields_desc (*conpot.protocols.IEC104.frames.NVA attribute*), 28

fields_desc (*conpot.protocols.IEC104.frames.OCI attribute*), 28

fields_desc (*conpot.protocols.IEC104.frames.QDP attribute*), 29

fields_desc (*conpot.protocols.IEC104.frames.QDS attribute*), 29

fields_desc (*conpot.protocols.IEC104.frames.QOS attribute*), 29

fields_desc (*conpot.protocols.IEC104.frames.s_frame attribute*), 38

fields_desc (*conpot.protocols.IEC104.frames.SCD attribute*), 29

fields_desc (*conpot.protocols.IEC104.frames.SEP attribute*), 29

fields_desc (*conpot.protocols.IEC104.frames.SIQ attribute*), 29

fields_desc (*conpot.protocols.IEC104.frames.SPE attribute*), 30

fields_desc (*conpot.protocols.IEC104.frames.SVA attribute*), 30

fields_desc (*conpot.protocols.IEC104.frames.u_frame attribute*), 38

fields_desc (*conpot.protocols.IEC104.frames.VTI attribute*), 30

file_path (*conpot.protocols.tftp.tftp_handler.TFTPContextServer attribute*), 57

FilesystemError, 21

find_mibs() (*in module conpot.protocols.snmp.build_pysnmp_mib_wrapper*), 55

finish() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 40

fix_sslwrap() (*in module conpot.helpers*), 72

FloatField (*class in conpot.protocols.IEC104.frames*), 28

format_list() (*conpot.core.filesystem.AbstractFS method*), 17

format_list() (*conpot.core.fs_utils.SubAbstractFS method*), 21

frame_object_with_timer (*class in conpot.protocols.IEC104.IEC104*), 26

FrameError, 26

FSOperationNotPermitted, 21

ftp_path() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 40

FTPCommandChannel (*class in conpot.protocols.ftp.ftp_handler*), 41

FTPConfig (*class in conpot.protocols.ftp.ftp_server*), 43

FTPException, 44

FTPHandlerBase (*class in conpot.protocols.ftp.ftp_base_handler*), 40

FTPHandlerBase.false_request (*class in conpot.protocols.ftp.ftp_base_handler*), 40

FTPMaxLoginAttemptsExceeded, 44

FTPMetrics (*class in conpot.protocols.ftp.ftp_base_handler*), 41

FTPPrivilegeException, 44

full_path (*conpot.protocols.tftp.tftp_handler.TFTPServerState attribute*), 58

Function() (conpot.tests.helpers.s7comm_client.s7 method), 60

G

generate_dependencies() (in module conpot.protocols.snmp.build_pysnmp_mib_wrapper), 55
get_boot_device() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 46
get_command() (conpot.tests.helpers.snmp_client.SNMPClient method), 60
get_data_from_iter() (in module conpot.protocols.ftp.ftp_utils), 44
get_databus() (in module conpot.core), 24
get_elapsed_time() (conpot.protocols.ftp.ftp_base_handler.FTPMetrics method), 41
get_entity_headers() (conpot.protocols.http.command_responder.HTTPServer method), 45
get_entity_trailers() (conpot.protocols.http.command_responder.HTTPServer method), 45
get_ext_ip() (in module conpot.utils.ext_ip), 72
get_gid() (conpot.protocols.ftp.ftp_server.FTPConfig method), 43
get_infoobj_list() (conpot.protocols.IEC104.IEC104.IEC104 static method), 26
get_interface() (in module conpot.core), 24
get_interface_ip() (in module conpot.utils.ext_ip), 72
get_metrics() (conpot.protocols.ftp.ftp_base_handler.FTPMetrics method), 41
get_object_from_reg() (conpot.protocols.IEC104.DeviceDataController.DeviceDataController method), 25
get_objects_and_properties() (conpot.protocols.bacnet.bacnet_app.BACnetApp method), 39
get_permissions() (conpot.core.filesystem.AbstractFS method), 17
get_permissions() (conpot.core.fs_utils.SubAbstractFS method), 21
get_power_state() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 46
get_registers() (conpot.protocols.IEC104.DeviceDataController.DeviceDataController method), 25
get_request() (conpot.protocols.kamstrup.meter_protocol.request_parser.Kamstrup method), 52
get_response() (conpot.protocols.snmp.databus_mediator.DatabusMediator method), 57
get_server() (conpot.emulators.proxy.Proxy method), 25
get_session() (conpot.core.session_manager.SessionManager method), 23
get_session() (in module conpot.core), 24
get_session_count() (conpot.core.session_manager.SessionManager method), 23
get_sessionManager() (in module conpot.core), 24
get_shapshot() (conpot.core.databus.Databus method), 15
get_status_headers() (conpot.protocols.http.command_responder.HTTPServer method), 45
get_status_trailers() (conpot.protocols.http.command_responder.HTTPServer method), 45
get_trigger_appendix() (conpot.protocols.http.command_responder.HTTPServer method), 45
get_uid() (conpot.protocols.ftp.ftp_server.FTPConfig method), 43
get_value() (conpot.core.databus.Databus method), 15
get_value() (conpot.emulators.misc.random.Random16bitRegister method), 24
get_value() (conpot.emulators.misc.random.Random8BitRegisters method), 24
get_value() (conpot.emulators.misc.uptime.Uptime method), 24
get_vfs() (in module conpot.core), 24
GetConfigCommand (class in conpot.protocols.kamstrup.management_protocol.commands), 48
getcwd() (conpot.core.filesystem.AbstractFS method), 17
getcwd() (conpot.core.fs_utils.SubAbstractFS method), 21
getfieldval() (conpot.protocols.IEC104.IEC104.frame_object_with_timer method), 26
getfile() (conpot.core.filesystem.AbstractFS method), 17
GetIdentity() (in module conpot.tests.helpers.s7comm_client), 59

getinfo() (*conpot.core.filesystem.AbstractFS method*), 17

getinfo() (*conpot.core.fs_utils.SubAbstractFS method*), 21

getmeta() (*conpot.core.filesystem.AbstractFS method*), 17

getmtime() (*conpot.core.filesystem.AbstractFS method*), 18

getmtime() (*conpot.core.fs_utils.SubAbstractFS method*), 21

getTimerResolution() (*conpot.protocols.snmp.command_responder.SNMPDispatcher method*), 56

groups (*conpot.core.filesystem.AbstractFS attribute*), 18

guess_payload_class() (*conpot.protocols.IEC104.frames.asdu_head method*), 30

## H

handle() (*conpot.emulators.proxy.Proxy method*), 25

handle() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 40

handle() (*conpot.protocols.ipmi.ipmi_server.IpmiServer method*), 47

handle() (*conpot.protocols.s7comm.s7.S7 method*), 54

handle() (*conpot.protocols.snmp.command_responder.SNMPDispatcher method*), 56

handle() (*conpot.protocols.tftp.tftp_handler.TFTPServerState method*), 58

handle() (*conpot.protocols.tftp.tftp_handler.TFTPState method*), 58

handle() (*conpot.protocols.tftp.tftp_handler.TFTPStateServerRecvRRQ method*), 58

handle() (*conpot.protocols.tftp.tftp_handler.TFTPStateServerRecvWRQ method*), 58

handle() (*conpot.protocols.tftp.tftp_handler.TFTPStateServerStart method*), 58

handle_client_request() (*conpot.protocols.ipmi.ipmi_server.IpmiServer method*), 47

handle_cmd_channel() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 40

handle_data_channel() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 41

handle_double_command46() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26

handle_i_frame() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26

handle_in_data() (*conpot.emulators.proxy.Proxy method*), 25

handle_inro_command100() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26

handle_out_data() (*conpot.emulators.proxy.Proxy method*), 25

handle_request() (*conpot.protocols.modbus.slave.MBSlave method*), 53

handle_request() (*conpot.protocols.modbus.slave_db.SlaveBase method*), 53

handle_s_frame() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26

handle_setpointfloatpoint_command50() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26

handle_setpointscaled_command49() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26

handle_single_command45() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26

handle_startendtag() (*conpot.protocols.http.command_responder.TemplateParser method*), 45

handle_u_frame() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26

handleMgmtOperation() (*conpot.protocols.snmp.conpot_cmdrsp.c_BulkCommandResponder method*), 56

handleMgmtOperation() (*conpot.protocols.snmp.conpot_cmdrsp.c_GetCommandResponder method*), 56

handleMgmtOperation() (*conpot.protocols.snmp.conpot_cmdrsp.c_NextCommandResponder method*), 56

handleMgmtOperation() (*conpot.protocols.snmp.conpot_cmdrsp.c_SetCommandResponder method*), 57

has_mib() (*conpot.protocols.snmp.command_responder.CommandRespo method*), 56

help() (*conpot.protocols.kamstrup.management_protocol.commands.Bas method*), 48

HELP_MESSAGE (*conpot.protocols.kamstrup.management_protocol.commands.AccessC attribute*), 47

HELP_MESSAGE (*conpot.protocols.kamstrup.management_protocol.commands.AlarmS attribute*), 48

HELP_MESSAGE (*con-*

pot.protocols.kamstrup.management_protocol.commands.BaseCommand attribute), 48

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.GetConfigCommand attribute), 48

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.RequestConnectCommand attribute), 48

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.RequestRestartCommand attribute), 48

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.SetConfigCommand attribute), 48

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.SetDeviceNameCommand attribute), 48

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.SetIPCommand attribute), 49

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.SetKap1Command attribute), 49

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.SetKap2Command attribute), 49

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.SetLookupCommand attribute), 49

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.SetNameserverCommand attribute), 49

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.SetPortsCommand attribute), 49

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.SetSerialCommand attribute), 49

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.SetWatchdogCommand attribute), 49

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.SoftwareVersionCommand attribute), 50

HELP_MESSAGE (conpot.protocols.kamstrup.management_protocol.commands.WinkModuleCommand attribute), 50

HelpCommand (class in conpot.protocols.kamstrup.management_protocol.commands), 48

hex_in_addr() (in module conpot.protocols.IEC104.DeviceDataController), 26

host (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase attribute), 41

HPFriendsLogger (class in conpot.core.loggers.hpfriends), 13

HTTPServer (class in conpot.protocols.http.command_responder), 44

I

i2repr() (conpot.protocols.IEC104.frames.NormValueField method), 28

i_frame (class in conpot.protocols.IEC104.frames), 37

iAm() (conpot.protocols.bacnet.bacnet_app.BACnetApp method), 39

IEC104 (class in conpot.protocols.IEC104.IEC104), 26

IEC104Register (class in conpot.protocols.IEC104.register), 39

iHave() (conpot.protocols.bacnet.bacnet_app.BACnetApp method), 39

increment_sendseq() (conpot.protocols.IEC104.IEC104.IEC104 method), 26

indication() (conpot.protocols.bacnet.bacnet_app.BACnetApp method), 39

initialize() (conpot.core.databus.Databus method), 15

initialize() (conpot.protocols.kamstrup.usage_simulator.UsageSimulator method), 52

initialize_databus() (conpot.core.session_manager.SessionManager method), 23

initialize_vfs() (conpot.core.virtual_fs.VirtualFS method), 24

initialize_vfs() (in module conpot.core), 24

initiate_session() (conpot.protocols.ipmi.ipmi_server.IpmiServer method), 47

inro_response() (in module conpot.protocols.IEC104.DeviceDataController), 26

Interface (class in conpot.core.internal_interface), 22

INVALID_PARAMETER (conpot.protocols.kamstrup.management_protocol.commands.BaseCo attribute), 48

InvalidFieldValueException, 26

IOA (class in conpot.protocols.IEC104.frames), 28

IpmiServer (class in conpot.protocols.ipmi.ipmi_server), 47

is_in() (conpot.protocols.modbus.modbus_block_databus_mediator.Mod method), 53

J

json_default() (in module conpot.core.loggers.helpers), 13

JsonLogger (class in conpot.core.loggers.json_log), 14

K

K162M (conpot.protocols.kamstrup.meter_protocol.kamstrup_constants.MeterTypes attribute), 51

K351C (conpot.protocols.kamstrup.meter_protocol.kamstrup_constants.MeterTypes attribute), 51

K382M (conpot.protocols.kamstrup.meter_protocol.kamstrup_constants.MeterTypes attribute), 51

KamstrupProtocolBase (class in conpot.protocols.kamstrup.meter_protocol.messages), 51

KamstrupRegister (class in conpot.protocols.kamstrup.meter_protocol.register), 52

KamstrupRequestBase (class in conpot.protocols.kamstrup.meter_protocol.messages), 51

KamstrupRequestGetRegisters (class in conpot.protocols.kamstrup.meter_protocol.messages), 51

KamstrupRequestParser (class in conpot.protocols.kamstrup.meter_protocol.request_parser), 52

KamstrupRequestUnknown (class in conpot.protocols.kamstrup.meter_protocol.messages), 51

KamstrupResponseBase (class in conpot.protocols.kamstrup.meter_protocol.messages), 51

KamstrupResponseRegister (class in conpot.protocols.kamstrup.meter_protocol.messages), 51

L

LESignedShortField (class in conpot.protocols.IEC104.frames), 28

listdir() (conpot.core.filesystem.AbstractFS method), 18

load_entity() (conpot.protocols.http.command_responder.HTTPServer method), 45

load_status() (conpot.protocols.http.command_responder.HTTPServer method), 45

log() (conpot.core.loggers.hpfriends.HPFriendsLogger method), 13

log() (conpot.core.loggers.json_log.JsonLogger method), 14

log() (conpot.core.loggers.sqlite_log.SQLiteLogger method), 14

log() (conpot.core.loggers.syslog.SysLogger method), 14

log() (conpot.core.loggers.taxii_log.TaxiiLogger method), 14

log() (conpot.protocols.http.command_responder.HTTPServer method), 45

log() (conpot.protocols.snmp.conpot_cmdrsp.conpot_extension method), 57

log_session() (conpot.core.loggers.json_log.JsonLogger method), 14

log_session() (conpot.core.loggers.sqlite_log.SQLiteLogger method), 14

LogWorker (class in conpot.core.loggers.log_worker), 14

M

make_subdirs() (conpot.protocols.tftp.tftp_handler.TFTPStateServerRecvWRQ method), 58

makedir() (conpot.core.filesystem.AbstractFS method), 18

MBSlave (class in conpot.protocols.modbus.slave), 53

MeterTypes (class in conpot.protocols.kamstrup.meter_protocol.kamstrup_constants), 50

mib2pysnmp() (in module conpot.protocols.snmp.build_pysnmp_mib_wrapper), 55

mock_callback() (conpot.tests.test_snmp_server.TestSNMPServer method), 69

ModbusBlockDatabusMediator (class in conpot.protocols.modbus.modbus_block_databus_mediator), 53

mount_fs() (conpot.core.filesystem.AbstractFS method), 18

move() (conpot.core.filesystem.AbstractFS method), 19

move() (conpot.core.fs_utils.SubAbstractFS method), 22

N

NegotiatePDU() (conpot.tests.helpers.s7comm_client.s7 method), 60

Network (class in conpot.core.internal_interface), 22

norm_path() (conpot.core.filesystem.AbstractFS method), 19

NormValueField (class in conpot.protocols.IEC104.frames), 28

notify_observers() (conpot.core.databus.Databus method), 15
number_to_bytes() (in module conpot.helpers), 72
NVA (class in conpot.protocols.IEC104.frames), 28

O

observe_value() (conpot.core.databus.Databus method), 15
OCI (class in conpot.protocols.IEC104.frames), 28
OMNIA (conpot.protocols.kamstrup.meter_protocol.kamstrup_constants.MeterTypes attribute), 51
open() (conpot.core.filesystem.AbstractFS method), 19
openbin() (conpot.core.filesystem.AbstractFS method), 19
opendir() (conpot.core.filesystem.AbstractFS method), 19

P

pack() (conpot.protocols.s7comm.cotp.COTP method), 54
pack() (conpot.protocols.s7comm.s7.S7 method), 54
pack() (conpot.protocols.s7comm.tpkt.TPKT method), 55
pack() (conpot.tests.helpers.s7comm_client.COTPConnectionPacket method), 59
pack() (conpot.tests.helpers.s7comm_client.COTPDataPacket method), 59
pack() (conpot.tests.helpers.s7comm_client.S7Packet method), 59
pack() (conpot.tests.helpers.s7comm_client.TPKTPacket method), 60
pack_short_int() (in module conpot.helpers), 72
parse() (conpot.protocols.s7comm.cotp.COTP method), 54
parse() (conpot.protocols.s7comm.s7.S7 method), 54
parse() (conpot.protocols.s7comm.tpkt.TPKT method), 55
parse_ip() (in module conpot.protocols.kamstrup.management_protocol.commands), 50
parse_port() (in module conpot.protocols.kamstrup.management_protocol.commands), 50
parse_template() (conpot.protocols.enip.enip_server.EnipConfig method), 40
ParseException, 54
payload_guess (conpot.protocols.IEC104.frames.asdu_head attribute), 30
payload_guess (conpot.protocols.IEC104.frames.BCR attribute), 27
payload_guess (conpot.protocols.IEC104.frames.DIQ attribute), 28
payload_guess (conpot.protocols.IEC104.frames.i_frame attribute), 37
payload_guess (conpot.protocols.IEC104.frames.QDP attribute), 29
payload_guess (conpot.protocols.IEC104.frames.QDS attribute), 29
payload_guess (conpot.protocols.IEC104.frames.QOS attribute), 29
payload_guess (conpot.protocols.IEC104.frames.SEP attribute), 29
payload_guess (conpot.protocols.IEC104.frames.SIQ attribute), 29
payload_guess (conpot.protocols.IEC104.frames.SPE attribute), 30
payload_guess (conpot.protocols.IEC104.frames.VTI attribute), 30
plc_stop_function() (conpot.tests.helpers.s7comm_client.s7 method), 60
plc_stop_signal() (conpot.protocols.s7comm.s7.S7 method), 54
port (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase attribute), 41
post_build() (conpot.protocols.IEC104.frames.i_frame method), 37
power_cycle() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 46
power_off() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 46
power_on() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 46
power_reset() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 46
power_shutdown() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 46
process_ftp_command() (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method), 41
process_ftp_command() (con-

*pot.protocols.ftp.ftp_handler.FTPCommandChannel method*), 43

Proxy (*class in conpot.emulators.proxy*), 25

ProxyDecoder (*class in conpot.emulators.proxy*), 25

purge_sessions() (*conpot.core.session_manager.SessionManager method*), 23

push_data() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 41

## Q

QDP (*class in conpot.protocols.IEC104.frames*), 28

QDS (*class in conpot.protocols.IEC104.frames*), 29

QOS (*class in conpot.protocols.IEC104.frames*), 29

## R

Random16bitRegister (*class in conpot.emulators.misc.random*), 24

Random8BitRegisters (*class in conpot.emulators.misc.random*), 24

readlink() (*conpot.core.filesystem.AbstractFS method*), 20

readlink() (*conpot.core.fs_utils.SubAbstractFS method*), 22

readProperty() (*conpot.protocols.bacnet.bacnet_app.BACnetApp method*), 39

ReadSZL() (*conpot.tests.helpers.s7comm_client.s7 method*), 60

recv_file() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 41

recvseq_increment() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26

refresh_client() (*conpot.tests.test_ftp.TestFTPServer method*), 62

register() (*conpot.protocols.snmp.command_responder.CommandResponder method*), 56

register_user() (*conpot.core.filesystem.AbstractFS method*), 20

registerRecvCbFun() (*conpot.protocols.snmp.command_responder.SNMPDispatcher method*), 56

REGISTERS (*conpot.protocols.kamstrup.meter_protocol.decoder_382.Decoder382 attribute*), 50

registerTimerCbFun() (*conpot.protocols.snmp.command_responder.SNMPDispatcher method*), 56

registerTransport() (*conpot.protocols.snmp.command_responder.SNMPDispatcher method*), 56

remove() (*conpot.core.filesystem.AbstractFS method*), 20

remove() (*conpot.core.fs_utils.SubAbstractFS method*), 22

removedir() (*conpot.core.filesystem.AbstractFS method*), 20

removedir() (*conpot.core.fs_utils.SubAbstractFS method*), 22

Request() (*conpot.tests.helpers.s7comm_client.s7 method*), 60

request_diagnostics() (*conpot.protocols.s7comm.s7.S7 method*), 54

request_not_implemented() (*conpot.protocols.s7comm.s7.S7 method*), 54

request_ssl_17() (*conpot.protocols.s7comm.s7.S7 method*), 55

request_ssl_28() (*conpot.protocols.s7comm.s7.S7 method*), 55

RequestConnectCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 48

RequestRestartCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 48

reset() (*conpot.core.databus.Databus method*), 15

respond() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 41

respond() (*conpot.protocols.kamstrup.management_protocol.command_ method*), 47

respond() (*conpot.protocols.kamstrup.meter_protocol.command_respon method*), 50

response() (*conpot.protocols.bacnet.bacnet_app.BACnetApp method*), 39

restart_t1() (*conpot.protocols.IEC104.IEC104.frame_object_with_timer method*), 26

restart_t1() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26

revert_mac() (*in module conpot.utils.mac_addr*), 72

root (*conpot.core.filesystem.AbstractFS attribute*), 20

root (*conpot.core.fs_utils.SubAbstractFS attribute*), 22

run() (*conpot.protocols.kamstrup.management_protocol.commands.Acce method*), 47

run() (*conpot.protocols.kamstrup.management_protocol.commands.Alar method*), 48

run() (*conpot.protocols.kamstrup.management_protocol.commands.Base method*), 48

run() (*conpot.protocols.kamstrup.management_protocol.commands.GetC method*), 48

run() (*conpot.protocols.kamstrup.management_protocol.commands.Help method*), 48

run() (*conpot.protocols.kamstrup.management_protocol.commands.Requ method*), 48

run() (*conpot.protocols.kamstrup.management_protocol.commands.RequestRestartCommand method*), 48
run() (*conpot.protocols.kamstrup.management_protocol.commands.SetConfigCommand method*), 48
run() (*conpot.protocols.kamstrup.management_protocol.commands.SetDeviceNameCommand method*), 48
run() (*conpot.protocols.kamstrup.management_protocol.commands.SetIPCommand method*), 49
run() (*conpot.protocols.kamstrup.management_protocol.commands.SetKap1Command method*), 49
run() (*conpot.protocols.kamstrup.management_protocol.commands.SetKap2Command method*), 49
run() (*conpot.protocols.kamstrup.management_protocol.commands.SetLookupCommand method*), 49
run() (*conpot.protocols.kamstrup.management_protocol.commands.SetNameserverCommand method*), 49
run() (*conpot.protocols.kamstrup.management_protocol.commands.SetPortsCommand method*), 49
run() (*conpot.protocols.kamstrup.management_protocol.commands.SetSerialCommand method*), 49
run() (*conpot.protocols.kamstrup.management_protocol.commands.SetWatchdogCommand method*), 49
run() (*conpot.protocols.kamstrup.management_protocol.commands.SoftwareVersionCommand method*), 50
run_cmd() (*in module conpot.tests.test_ipmi_server*), 65

S

S7 (*class in conpot.protocols.s7comm.s7*), 54
s7 (*class in conpot.tests.helpers.s7comm_client*), 60
S7Error, 59
S7Packet (*class in conpot.tests.helpers.s7comm_client*), 59
S7ProtocolError, 59
s_frame (*class in conpot.protocols.IEC104.frames*), 38
sanitize_file_name() (*in module conpot.helpers*), 72
Scan() (*in module conpot.tests.helpers.s7comm_client*), 59
SCD (*class in conpot.protocols.IEC104.frames*), 29
select_data() (*conpot.core.loggers.sqlite_log.SQLiteLogger method*), 14
send_104frame() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26
send_auth_cap() (*conpot.protocols.ipmi.ipmi_server.IpmiServer method*), 47
send_chunked() (*conpot.protocols.http.command_responder.HTTPServer method*), 45
send_data() (*conpot.protocols.ipmi.fakesession.FakeSession method*), 46
send_error() (*conpot.protocols.http.command_responder.HTTPServer method*), 45
send_file() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 41
send_frame_imm() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26
send_ipmi_response() (*conpot.protocols.ipmi.fakesession.FakeSession method*), 46
send_payload() (*conpot.protocols.ipmi.fakesession.FakeSession method*), 46
send_response() (*conpot.protocols.http.command_responder.HTTPServer method*), 45
sendMessage() (*conpot.protocols.snmp.command_responder.SNMPDispatcher method*), 56
SEP (*class in conpot.protocols.IEC104.frames*), 29
serialize() (*conpot.protocols.kamstrup.meter_protocol.messages.Kam method*), 51
serialize() (*conpot.protocols.kamstrup.meter_protocol.messages.Kam method*), 51
serve_forever() (*conpot.protocols.http.command_responder.CommandResponder method*), 44
serve_forever() (*conpot.protocols.snmp.command_responder.CommandResponder method*), 56
serverInitial() (*conpot.protocols.tftp.tftp_handler.TFTPServerState method*), 58
SessionManager (*class in conpot.core.session_manager*), 23
set_access_ip() (*conpot.protocols.kamstrup.management_protocol.commands.AccessC method*), 47
set_boot_device() (*conpot.protocols.ipmi.fakebmc.FakeBmc method*), 46
set_command() (*conpot.tests.helpers.snmp_client.SNMPClient method*), 60
set_ended() (*conpot.core.attack_session.AttackSession method*), 15
set_object_val() (*conpot.protocols.IEC104.DeviceDataController.DeviceDataControlle method*), 25
set_val() (*conpot.protocols.IEC104.register.IEC104Register method*), 39
set_value() (*conpot.core.databus.Databus method*), 15

set_value() (*conpot.protocols.snmp.databus_mediator.DatabusMediator method*), 57

setbinfile() (*conpot.core.filesystem.AbstractFS method*), 20

SetConfigCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 48

SetDeviceNameCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 48

setinfo() (*conpot.core.filesystem.AbstractFS method*), 20

SetIPCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 48

SetKap1Command (*class in conpot.protocols.kamstrup.management_protocol.commands*), 49

SetKap2Command (*class in conpot.protocols.kamstrup.management_protocol.commands*), 49

SetLookupCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 49

SetNameserverCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 49

SetPortsCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 49

SetSerialCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 49

settimes() (*conpot.core.filesystem.AbstractFS method*), 20

setup() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 41

setUp() (*conpot.tests.test_bacnet_server.TestBACnetServer method*), 60

setUp() (*conpot.tests.test_base.TestBase method*), 61

setUp() (*conpot.tests.test_docs.TestMakeDocs method*), 61

setUp() (*conpot.tests.test_enip_server.TestENIPServer method*), 61

setUp() (*conpot.tests.test_ext_ip_util.TestExtIPUtil method*), 62

setUp() (*conpot.tests.test_ftp.TestFTPServer method*), 62

setUp() (*conpot.tests.test_guardian_ast.TestGuardianAST method*), 63

setUp() (*conpot.tests.test_http_server.TestHTTPServer method*), 64

setUp() (*conpot.tests.test_iec104_server.TestIEC104Server method*), 65

setUp() (*conpot.tests.test_ipmi_server.TestIPMI method*), 65

setUp() (*conpot.tests.test_kamstrup_management_protocol.TestKamstrup method*), 66

setUp() (*conpot.tests.test_kamstrup_meter_protocol.TestKamstrup method*), 67

setUp() (*conpot.tests.test_logger_json.TestJsonLogger method*), 67

setUp() (*conpot.tests.test_mac_addr.TestMacAddrUtil method*), 67

setUp() (*conpot.tests.test_modbus_server.TestModbusServer method*), 67

setUp() (*conpot.tests.test_pysnmp_wrapper.TestPySNMPWrapper method*), 68

setUp() (*conpot.tests.test_s7_server.TestS7Server method*), 69

setUp() (*conpot.tests.test_snmp_server.TestSNMPServer method*), 69

setUp() (*conpot.tests.test_tftp.TestTFTPServer method*), 69

setUp() (*conpot.tests.test_vfs.TestFileSystem method*), 70

setUp() (*conpot.tests.test_vfs.TestSubFileSystem method*), 71

SetWatchdogCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 49

show_send_list() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 26

SIQ (*class in conpot.protocols.IEC104.frames*), 29

SlaveBase (*class in conpot.protocols.modbus.slave_db*), 53

SNMPClient (*class in conpot.tests.helpers.snmp_client*), 60

SNMPDispatcher (*class in conpot.protocols.snmp.command_responder*), 56

SoftwareVersionCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 49

SPE (*class in conpot.protocols.IEC104.frames*), 29

Split() (*in module conpot.tests.helpers.s7comm_client*), 59

SQLiteLogger (*class in conpot.core.loggers.sqlite_log*), 14

ssl_lists (*conpot.protocols.s7comm.s7.S7 attribute*), 55

start() (*conpot.core.loggers.log_worker.LogWorker method*), 14

start() (*conpot.protocols.ipmi.ipmi_server.IpmiServer method*), 47

start() (*conpot.protocols.tftp.tftp_handler.TFTPContextServer method*), 57 start_data_channel() (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method), 41

stat() (conpot.core.filesystem.AbstractFS method), 20

stat() (conpot.core.fs_utils.SubAbstractFS method), 22

StixTransformer (class in conpot.core.loggers.stix_transform), 14

stop() (conpot.core.loggers.log_worker.LogWorker method), 14

stop() (conpot.emulators.proxy.Proxy method), 25

stop() (conpot.protocols.http.command_responder.CommandResponder method), 44

stop() (conpot.protocols.ipmi.ipmi_server.IpmiServer method), 47

stop() (conpot.protocols.kamstrup.usage_simulator.UsageSimulator method), 52

stop() (conpot.protocols.snmp.command_responder.CommandResponder method), 56

stop_data_channel() (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method), 41

str_to_bytes() (in module conpot.helpers), 72

stream_server_handle() (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase class method), 41

StripUnprintable() (in module conpot.tests.helpers.s7comm_client), 59

SubAbstractFS (class in conpot.core.fs_utils), 21

SubHTTPServer (class in conpot.protocols.http.command_responder), 45

substitute_template_fields() (conpot.protocols.http.command_responder.HTTPServer method), 45

SVA (class in conpot.protocols.IEC104.frames), 30

SysLogger (class in conpot.core.loggers.syslog), 14

## T

take_snapshot() (conpot.core.filesystem.AbstractFS method), 21

TaxiiLogger (class in conpot.core.loggers.taxii_log), 14

tearDown() (conpot.tests.test_bacnet_server.TestBACnetServer method), 60

tearDown() (conpot.tests.test_base.TestBase method), 61

tearDown() (conpot.tests.test_docs.TestMakeDocs method), 61

tearDown() (conpot.tests.test_enip_server.TestENIPServer method), 61

tearDown() (conpot.tests.test_ext_ip_util.TestExtIPUtil method), 62

tearDown() (conpot.tests.test_ftp.TestFTPServer method), 62

tearDown() (conpot.tests.test_guardian_ast.TestGuardianAST method), 63

tearDown() (conpot.tests.test_http_server.TestHTTPServer method), 64

tearDown() (conpot.tests.test_iec104_server.TestIEC104Server method), 65

tearDown() (conpot.tests.test_ipmi_server.TestIPMI method), 65

tearDown() (conpot.tests.test_kamstrup_management_protocol.TestKam method), 66

tearDown() (conpot.tests.test_kamstrup_meter_protocol.TestKamstrup method), 67

tearDown() (conpot.tests.test_logger_json.TestJsonLogger method), 67

tearDown() (conpot.tests.test_mac_addr.TestMacAddrUtil method), 67

tearDown() (conpot.tests.test_modbus_server.TestModbusServer method), 67

tearDown() (conpot.tests.test_s7_server.TestS7Server method), 69

tearDown() (conpot.tests.test_snmp_server.TestSNMPServer method), 69

tearDown() (conpot.tests.test_tftp.TestTFTPServer method), 70

tearDown() (conpot.tests.test_vfs.TestFileSystem method), 70

tearDown() (conpot.tests.test_vfs.TestSubFileSystem method), 71

TemplateParser (class in conpot.protocols.http.command_responder), 45

test_abor() (conpot.tests.test_ftp.TestFTPServer method), 62

test_access() (conpot.tests.test_vfs.TestFileSystem method), 70

test_access() (conpot.tests.test_vfs.TestSubFileSystem method), 71

test_access_control_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupMan method), 66

test_alarm_server_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupMan method), 66

test_allo() (conpot.tests.test_ftp.TestFTPServer method), 62

test_appe() (conpot.tests.test_ftp.TestFTPServer method), 62

test_ascii_decoder() (conpot.tests.test_proxy.TestProxy method), 68

test_ast_error() (conpot.tests.test_guardian_ast.TestGuardianAST method), 64

test_auth() (conpot.tests.test_ftp.TestFTPServer method), 62

test_base() (conpot.tests.test_base.TestBase method), 61

test_boot_device() (conpot.tests.test_ipmi_server.TestIPMI method), 65

test_channel_get_access() (conpot.tests.test_ipmi_server.TestIPMI method), 65

test_chassis_status() (conpot.tests.test_ipmi_server.TestIPMI method), 65

test_chmod() (conpot.tests.test_vfs.TestFileSystem method), 70

test_chmod() (conpot.tests.test_vfs.TestSubFileSystem method), 71

test_chown() (conpot.tests.test_vfs.TestFileSystem method), 70

test_chown() (conpot.tests.test_vfs.TestSubFileSystem method), 71

test_compile() (conpot.tests.test_pysnmp_wrapper.TestPySNMPWrapper method), 68

test_copydir() (conpot.tests.test_vfs.TestFileSystem method), 70

test_copyfile() (conpot.tests.test_vfs.TestFileSystem method), 70

test_cwd() (conpot.tests.test_ftp.TestFTPServer method), 62

test_dele() (conpot.tests.test_ftp.TestFTPServer method), 62

test_do_HEAD() (conpot.tests.test_http_server.TestHTTPServer method), 64

test_do_OPTIONS() (conpot.tests.test_http_server.TestHTTPServer method), 64

test_do_POST() (conpot.tests.test_http_server.TestHTTPServer method), 64

test_do_TRACE() (conpot.tests.test_http_server.TestHTTPServer method), 64

test_ext_util() (conpot.tests.test_ext_ip_util.TestExtIPUtil method), 62

test_fetch_ext_ip() (conpot.tests.test_ext_ip_util.TestExtIPUtil method), 62

test_file_rename() (conpot.tests.test_ftp.TestFTPServer method), 62

test_find() (conpot.tests.test_pysnmp_wrapper.TestPySNMPWrapper method), 68

test_format_list() (conpot.tests.test_vfs.TestFileSystem method), 70

test_format_list() (conpot.tests.test_vfs.TestSubFileSystem method), 71

test_get_config_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupMan method), 66

test_get_cwd() (conpot.tests.test_vfs.TestFileSystem method), 70

test_get_cwd() (conpot.tests.test_vfs.TestSubFileSystem method), 71

test_get_permissions() (conpot.tests.test_vfs.TestFileSystem method), 70

test_get_permissions() (conpot.tests.test_vfs.TestSubFileSystem method), 71

test_get_software_version_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrup method), 66

test_getmtime() (conpot.tests.test_vfs.TestFileSystem method), 70

test_help() (conpot.tests.test_ftp.TestFTPServer method), 62

test_help_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupMan method), 66

Test_HPFriends (class in conpot.tests.test_hpfriends), 64

test_hpfriends() (conpot.tests.test_hpfriends.Test_HPFriends method), 64

test_http_backend_databus() (conpot.tests.test_http_server.TestHTTPServer method), 64

test_http_backend_tarpit() (conpot.tests.test_http_server.TestHTTPServer method), 64

test_http_request_base() (conpot.tests.test_http_server.TestHTTPServer method), 64

test_http_subselect_trigger() (conpot.tests.test_http_server.TestHTTPServer *method*), 64

test_I20100() (*conpot.tests.test_guardian_ast.TestGuardianAST method*), 63

test_I20200() (*conpot.tests.test_guardian_ast.TestGuardianAST method*), 63

test_I20300() (*conpot.tests.test_guardian_ast.TestGuardianAST method*), 63

test_I20400() (*conpot.tests.test_guardian_ast.TestGuardianAST method*), 63

test_I20500() (*conpot.tests.test_guardian_ast.TestGuardianAST method*), 63

test_invalid_crc() (*conpot.tests.test_kamstrup_decoder.TestKamstrupDecoder method*), 66

test_ip_verify() (*conpot.tests.test_ext_ip_util.TestExtIPUtil method*), 62

test_jail() (*conpot.tests.test_vfs.TestFileSystem method*), 70

test_list() (*conpot.tests.test_ftp.TestFTPServer method*), 62

test_list_identity_tcp() (*conpot.tests.test_enip_server.TestENIPServer method*), 61

test_list_identity_udp() (*conpot.tests.test_enip_server.TestENIPServer method*), 61

test_list_interfaces_tcp() (*conpot.tests.test_enip_server.TestENIPServer method*), 61

test_list_interfaces_udp() (*conpot.tests.test_enip_server.TestENIPServer method*), 61

test_list_services_tcp() (*conpot.tests.test_enip_server.TestENIPServer method*), 61

test_list_services_udp() (*conpot.tests.test_enip_server.TestENIPServer method*), 61

test_listdir() (*conpot.tests.test_vfs.TestFileSystem method*), 70

test_listdir() (*conpot.tests.test_vfs.TestSubFileSystem method*), 71

test_log_event() (*conpot.tests.test_logger_json.TestJsonLogger method*), 67

test_mac() (*conpot.tests.test_mac_addr.TestMacAddrUtil method*), 67

test_make_docs() (*conpot.tests.test_docs.TestMakeDocs method*), 61

test_malformend_request_tcp() (*conpot.tests.test_enip_server.TestENIPServer method*), 61

test_malformend_request_udp() (*conpot.tests.test_enip_server.TestENIPServer method*), 62

test_max_retries() (*conpot.tests.test_ftp.TestFTPServer method*), 62

test_mdtm() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_misc() (*conpot.tests.test_ipmi_server.TestIPMI method*), 65

test_mkd() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_mkdir() (*conpot.tests.test_vfs.TestFileSystem method*), 70

test_mkdir() (*conpot.tests.test_vfs.TestSubFileSystem method*), 71

test_mkdir_upload() (*conpot.tests.test_tftp.TestTFTPServer method*), 70

test_mkdirs() (*conpot.tests.test_vfs.TestFileSystem method*), 70

test_mkdirs() (*conpot.tests.test_vfs.TestSubFileSystem method*), 71

test_modbus_logging() (*conpot.tests.test_modbus_server.TestModbusServer method*), 68

test_mode() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_movedir() (*conpot.tests.test_vfs.TestFileSystem method*), 70

test_movefile() (*conpot.tests.test_vfs.TestFileSystem method*), 70

Test_MySQLlogger (*class in conpot.tests.test_logger_mysql*), 67

test_mysqllogger() (*conpot.tests.test_logger_mysql.Test_MySQLlogger method*), 67

test_nlist() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_no_response_requests() (*conpot.tests.test_bacnet_server.TestBACnetServer method*), 60

test_noop() (*conpot.tests.test_ftp.TestFTPServer*

method), 63
test_not_implemented_method() (conpot.tests.test_http_server.TestHTTPServer method), 64
test_open_dir() (conpot.tests.test_vfs.TestFileSystem method), 70
test_open_file() (conpot.tests.test_vfs.TestFileSystem method), 70
test_open_file() (conpot.tests.test_vfs.TestSubFileSystem method), 71
test_openbin_file() (conpot.tests.test_vfs.TestFileSystem method), 70
test_power_state() (conpot.tests.test_ipmi_server.TestIPMI method), 65
test_proxy() (conpot.tests.test_proxy.TestProxy method), 68
test_proxy_with_decoder() (conpot.tests.test_proxy.TestProxy method), 68
test_pwd() (conpot.tests.test_ftp.TestFTPServer method), 63
test_read_coils() (conpot.tests.test_modbus_server.TestModbusServer method), 68
test_read_nonexistent_slave() (conpot.tests.test_modbus_server.TestModbusServer method), 68
test_read_tags() (conpot.tests.test_enip_server.TestENIPServer method), 62
test_readlink() (conpot.tests.test_vfs.TestFileSystem method), 70
test_readlink() (conpot.tests.test_vfs.TestSubFileSystem method), 71
test_readProperty() (conpot.tests.test_bacnet_server.TestBACnetServer method), 61
test_rein() (conpot.tests.test_ftp.TestFTPServer method), 63
test_remove() (conpot.tests.test_vfs.TestFileSystem method), 71
test_remove() (conpot.tests.test_vfs.TestSubFileSystem method), 71
test_removedir() (conpot.tests.test_vfs.TestFileSystem method), 71
test_removedir() (conpot.tests.test_vfs.TestSubFileSystem method), 71
test_report_slave_id() (conpot.tests.test_modbus_server.TestModbusServer method), 68
test_request_connect_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupMan
method), 66
test_request_get_register() (conpot.tests.test_kamstrup_meter_protocol.TestKamstrup method), 67
test_request_one() (conpot.tests.test_kamstrup_decoder.TestKamstrupDecoder method), 66
test_response_function_43_device_info() (conpot.tests.test_modbus_server.TestModbusServer method), 68
test_rest() (conpot.tests.test_ftp.TestFTPServer method), 63
test_retr() (conpot.tests.test_ftp.TestFTPServer method), 63
test_rmd() (conpot.tests.test_ftp.TestFTPServer method), 63
test_S60200() (conpot.tests.test_guardian_ast.TestGuardianAST method), 63
test_S60201() (conpot.tests.test_guardian_ast.TestGuardianAST method), 63
test_S60202() (conpot.tests.test_guardian_ast.TestGuardianAST method), 64
test_S60203() (conpot.tests.test_guardian_ast.TestGuardianAST method), 64
test_S60204() (conpot.tests.test_guardian_ast.TestGuardianAST method), 64
test_s7() (conpot.tests.test_s7_server.TestS7Server method), 69
test_set_config_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupMan
method), 66
test_set_device_name_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupMan
method), 66
test_set_ip_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupMan
method), 66
test_set_kap1_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupMan
method), 66
test_set_kap2_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupMan

*method*), 66

test_set_lookup_command() (*conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol method*), 66

test_set_name_server_command() (*conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol method*), 66

test_set_ports_command() (*conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol method*), 66

test_set_serial_command() (*conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol method*), 66

test_set_time() (*conpot.tests.test_vfs.TestSubFileSystem method*), 71

test_set_watchdog_command() (*conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol method*), 66

test_site() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_site_chmod() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_site_help() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_size() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_snapshot() (*conpot.tests.test_vfs.TestFileSystem method*), 71

test_snmp_get() (*conpot.tests.test_snmp_server.TestSNMPServer method*), 69

test_snmp_set() (*conpot.tests.test_snmp_server.TestSNMPServer method*), 69

test_ssl_proxy() (*conpot.tests.test_proxy.TestProxy method*), 68

test_ssl_proxy_with_decoder() (*conpot.tests.test_proxy.TestProxy method*), 68

test_startdt() (*conpot.tests.test_iec104_server.TestIEC104Server method*), 65

test_stat() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_stat() (*conpot.tests.test_vfs.TestFileSystem method*), 71

test_stat() (*conpot.tests.test_vfs.TestSubFileSystem method*), 71

test_stix_transform() (*conpot.tests.test_taxii.TestLoggers method*), 69

test_stor() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_stou() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_stru() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_syst() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_taxii() (*conpot.tests.test_taxii.TestLoggers method*), 69

test_testfr() (*conpot.tests.test_iec104_server.TestIEC104Server method*), 65

test_tftp_download() (*conpot.tests.test_tftp.TestTFTPServer method*), 70

test_tftp_upload() (*conpot.tests.test_tftp.TestTFTPServer method*), 70

test_type() (*conpot.tests.test_ftp.TestFTPServer method*), 63

test_user_list() (*conpot.tests.test_ipmi_server.TestIPMI method*), 65

test_utime() (*conpot.tests.test_vfs.TestFileSystem method*), 71

test_utime() (*conpot.tests.test_vfs.TestSubFileSystem method*), 71

test_whoHas() (*conpot.tests.test_bacnet_server.TestBACnetServer method*), 61

test_whoIs() (*conpot.tests.test_bacnet_server.TestBACnetServer method*), 61

test_wrapper_output() (*conpot.tests.test_pysnmp_wrapper.TestPySNMPWrapper method*), 68

test_wrapper_processing() (*conpot.tests.test_pysnmp_wrapper.TestPySNMPWrapper method*), 68

test_write_for_non_existing() (*conpot.tests.test_iec104_server.TestIEC104Server method*), 65

test_write_no_relation_for_existing() (*conpot.tests.test_iec104_server.TestIEC104Server method*), 65

test_write_read_coils() (*conpot.tests.test_modbus_server.TestModbusServer method*), 68

test_write_relation_for_existing() (*conpot.tests.test_iec104_server.TestIEC104Server method*), 65

test_write_tags() (*con-*

*pot.tests.test_enip_server.TestENIPServer method*), 62

test_write_wrong_type_for_existing() (*conpot.tests.test_iec104_server.TestIEC104Server method*), 65

TestBACnetServer (*class in conpot.tests.test_bacnet_server*), 60

TestBase (*class in conpot.tests.test_base*), 61

TestENIPServer (*class in conpot.tests.test_enip_server*), 61

TestExtIPUtil (*class in conpot.tests.test_ext_ip_util*), 62

TestFileSystem (*class in conpot.tests.test_vfs*), 70

TestFTPServer (*class in conpot.tests.test_ftp*), 62

TestGuardianAST (*class in conpot.tests.test_guardian_ast*), 63

TestHTTPServer (*class in conpot.tests.test_http_server*), 64

TestIEC104Server (*class in conpot.tests.test_iec104_server*), 65

TestIPMI (*class in conpot.tests.test_ipmi_server*), 65

TestJsonLogger (*class in conpot.tests.test_logger_json*), 67

TestKamstrup (*class in conpot.tests.test_kamstrup_meter_protocol*), 67

TestKamstrupDecoder (*class in conpot.tests.test_kamstrup_decoder*), 66

TestKamstrupManagementProtocol (*class in conpot.tests.test_kamstrup_management_protocol*), 66

TestLoggers (*class in conpot.tests.test_taxii*), 69

TestMacAddrUtil (*class in conpot.tests.test_mac_addr*), 67

TestMakeDocs (*class in conpot.tests.test_docs*), 61

TestModbusServer (*class in conpot.tests.test_modbus_server*), 67

TestProxy (*class in conpot.tests.test_proxy*), 68

TestPySNMPWrapper (*class in conpot.tests.test_pysnmp_wrapper*), 68

TestS7Server (*class in conpot.tests.test_s7_server*), 69

TestSNMPServer (*class in conpot.tests.test_snmp_server*), 69

TestSubFileSystem (*class in conpot.tests.test_vfs*), 71

TestTFTPServer (*class in conpot.tests.test_tftp*), 69

TFTPContextServer (*class in conpot.protocols.tftp.tftp_handler*), 57

TFTPServerState (*class in conpot.protocols.tftp.tftp_handler*), 57

TFTPState (*class in conpot.protocols.tftp.tftp_handler*), 58

TFTPStateServerRecvRRQ (*class in conpot.protocols.tftp.tftp_handler*), 58

TFTPStateServerRecvWRQ (*class in conpot.protocols.tftp.tftp_handler*), 58

TFTPStateServerStart (*class in conpot.protocols.tftp.tftp_handler*), 58

ThreadedHTTPServer (*class in conpot.protocols.http.command_responder*), 46

timeout (*conpot.protocols.ftp.ftp_base_handler.FTPMetrics attribute*), 41

Timeout_t1, 27

Timeout_t1_2nd, 27

Timeout_t3, 27

TPKT (*class in conpot.protocols.s7comm.tpkt*), 55

TPKTPacket (*class in conpot.tests.helpers.s7comm_client*), 59

transform() (*conpot.core.loggers.stix_transform.StixTransformer method*), 14

try_parse_uint() (*in module conpot.protocols.kamstrup.management_protocol.commands*), 50

## U

u_frame (*class in conpot.protocols.IEC104.frames*), 38

Unknown (*conpot.protocols.kamstrup.meter_protocol.kamstrup_constants. attribute*), 51

unpack() (*conpot.tests.helpers.s7comm_client.COTPConnectionPacket method*), 59

unpack() (*conpot.tests.helpers.s7comm_client.COTPDataPacket method*), 59

unpack() (*conpot.tests.helpers.s7comm_client.S7Packet method*), 59

unpack() (*conpot.tests.helpers.s7comm_client.TPKTPacket method*), 60

unpack_short_int() (*in module conpot.helpers*), 72

update_evasion_table() (*conpot.protocols.snmp.databus_mediator.DatabusMediator method*), 57

Uptime (*class in conpot.emulators.misc.uptime*), 24

usage_counter() (*conpot.protocols.kamstrup.usage_simulator.UsageSimulator method*), 52

UsageSimulator (*class in conpot.protocols.kamstrup.usage_simulator*), 52

user_groups (*conpot.core.filesystem.AbstractFS attribute*), 21

users (*conpot.core.filesystem.AbstractFS attribute*), 21

## V

valid_crc() (*conpot.protocols.kamstrup.meter_protocol.decoder_382.1 class method*), 50

valid_crc() (*conpot.protocols.kamstrup.meter_protocol.request_parser.KamstrupRequestParser class method*), 52
vfs (*conpot.protocols.tftp.tftp_handler.TFTPServerState attribute*), 58
VirtualFS (*class in conpot.core.virtual_fs*), 23
VTI (*class in conpot.protocols.IEC104.frames*), 30

W

walk_command() (*conpot.tests.helpers.snmp_client.SNMPClient method*), 60
whoHas() (*conpot.protocols.bacnet.bacnet_app.BACnetApp method*), 39
whoIs() (*conpot.protocols.bacnet.bacnet_app.BACnetApp method*), 39
WinkModuleCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 50