
Conpot Documentation

Release 0.6.0

MushMush Foundation

Dec 02, 2019

Contents

1	Installation	3
1.1	Quick Installation using Docker	3
1.2	Installation on host using Virtualenv	4
2	Conpot concepts	7
2.1	Databus	7
2.2	ConpotFS	7
2.3	Internal Interface	8
2.4	Protocols	8
2.5	Proxy Mode	8
2.6	Templates	8
3	Developmental guidelines	9
3.1	Development Guidelines	9
4	Usage and Frequently asked questions	13
4.1	Frequently Asked Questions	13
5	API reference	15
5.1	API Reference	15
	Python Module Index	75
	Index	77

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

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

1.1.1 Via a pre-built image

1. Install Docker
2. Run `docker pull honeynet/conpot`
3. Run

```
docker run -it -p 80:80 -p 102:102 -p 502:502 -p 161:161/udp
--network=bridge honeynet/conpot:latest /bin/sh
```
4. Finally run `conpot -f --template default`

Navigate to http://MY_IP_ADDRESS to confirm the setup.

1.1.2 Build docker image from source

1. Install Docker
2. Clone this repo with `git clone https://github.com/mushorg/conpot.git`
3. Run `sudo make run-docker`

Navigate to http://MY_IP_ADDRESS to confirm the setup.

Install the table version of Conpot from PyPI:

```
$ pip install conpot
```


<todo: add some data here>

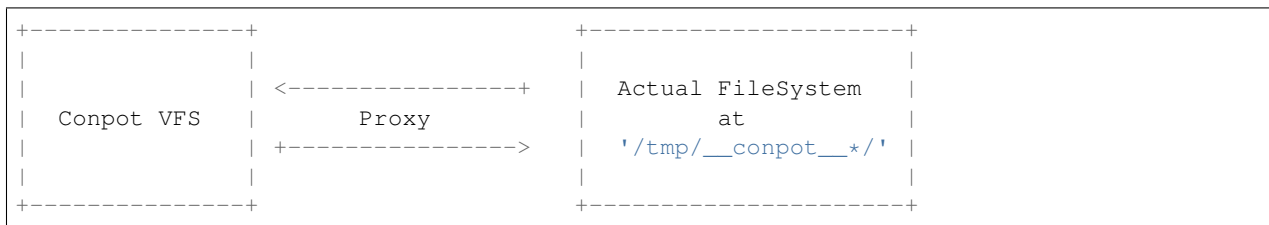
2.1 Databus

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.



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

Internal interface was a feature developed for advanced users. Basically every attribute of a class that is decorated by *@conpot_protocol* decorator can be accessed. This can be very powerful in case we want to emulate a system-wide phenomenon. Like for example we want to emulate a system restart (kamstrup management protocol ;-)) we can set a counter and freeze access to all protocols.

Some other uses include timing the last attack. This can be done by tracking the handle method for every protocol. Again can be easily done, without even touching the protocol implementation :-)

For more details refer to PR related to this issue: <https://github.com/mushorg/conpot/pull/375>

2.4 Protocols

2.5 Proxy Mode

2.6 Templates

<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.

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.

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.

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 than 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
<code>case_insensitive</code>	<i>True</i> if this filesystem is case insensitive.
<code>invalid_path_chars</code>	A string containing the characters that may not be used on this filesystem.
<code>max_path_length</code>	Maximum number of characters permitted in a path, or <i>None</i> for no limit.
<code>max_sys_path_length</code>	Maximum number of characters permitted in a sys path, or <i>None</i> for no limit.
<code>network</code>	<i>True</i> if this filesystem requires a network.
<code>read_only</code>	<i>True</i> if this filesystem is read only.
<code>supports_rename</code>	<i>True</i> if this filesystem supports an <i>os.rename</i> 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 belongs. 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 an 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, key-  
file=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.Packet'>]

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.Packet'>]

fields_desc = [<Field (BSI,asdu_infobj_7,asdu_infobj_8,asdu_infobj_51,asdu_infobj_64).I'>]

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

Bases: scapy.packet.Packet

aliastypes = [<class 'conpot.protocols.IEC104.frames.CP16Time'>, <class 'scapy.packet.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_infobj_20).I'>]

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

Bases: scapy.packet.Packet

aliastypes = [<class 'conpot.protocols.IEC104.frames.CP24Time'>, <class 'scapy.packet.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 (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.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 (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.Packet'>]
    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.Packet'>]
    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.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 (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.Packet'>]
    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.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 (SPE).Padding>, <Field (SPE).SRD>, <Field (SPE).SIE>, <Field (SPE).SIP>]
```

```
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.Packet'>]
```

```
fields_desc = [<Field (SVA, asdu_infobj_11, asdu_infobj_12, asdu_infobj_35, asdu_infobj_49).TypeID>, <Field (SVA, asdu_infobj_11, asdu_infobj_12, asdu_infobj_35, asdu_infobj_49).SQ>, <Field (SVA, asdu_infobj_11, asdu_infobj_12, asdu_infobj_35, asdu_infobj_49).SIP>]
```

```
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.Packet'>]
```

```
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.Packet'>]
```

```
fields_desc = [<Field (asdu_head).TypeID>, <Field (asdu_head).SQ>, <Field (asdu_head).SIP>]
```

```
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.packet.Packet'>]
```

```
fields_desc = [<Field (IOA, asdu_infobj_1, asdu_infobj_2, asdu_infobj_3, asdu_infobj_4, asdu_infobj_5).TypeID>, <Field (IOA, asdu_infobj_1, asdu_infobj_2, asdu_infobj_3, asdu_infobj_4, asdu_infobj_5).SQ>, <Field (IOA, asdu_infobj_1, asdu_infobj_2, asdu_infobj_3, asdu_infobj_4, asdu_infobj_5).SIP>]
```

```
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.packet.Packet'>]
```



```

    aliastypes = [<class 'conpot.protocols.IEC104.frames.asdu_infobj_13'>, <class 'scapy.p
    fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd
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.p
    fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd
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.p
    fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd
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.p
    fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd
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.p
    fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd
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.p
    fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd
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.p
    fields_desc = [<Field (IOA,asdu_infobj_1,asdu_infobj_2,asdu_infobj_3,asdu_infobj_4,asd
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.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_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.p
```

```
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.p
```

```
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.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_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.p
```

```
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.p
```

```
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.p
```

```
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.p
```

```
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.p
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.p
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.p
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.p
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.p
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.p
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.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_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.p
```

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

```
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.p
```

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

```
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.p
```

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

```
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.p
```

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

```
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.p
```

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

```
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.p
```

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

```
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.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_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.p
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.p
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.p
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.p
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.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_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.p
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.p
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.p
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.p
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.p
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.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_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.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_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.pa
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, _inter-
                                             nal=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).Ser
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.P
```

```
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.P
```

```
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 (YYYYMMDDHH-MMSS) 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 RNTTO command)
- do_RNTTO** (*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_entitytrailers (rqfilename, configuration)

get_status_headers (status, headers, configuration)

get_statustrailers (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,  
Re-  
questHandler-  
Class,  
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_
    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 for 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 p

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_ad...)
    Bases: object
```

```
class conpot.protocols.kamstrup.meter_protocol.messages.KamstrupRequestBase (communication_addr...)
    com-
    mand,
    mes-
    sage_bytes)

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

```
class conpot.protocols.kamstrup.meter_protocol.messages.KamstrupRequestGetRegisters (communi...)
    com-
    mand_byt...
    mes-
    sage_byte...

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

```
command_byte = 16
```

```
class conpot.protocols.kamstrup.meter_protocol.messages.KamstrupRequestUnknown (communication_...)
    com-
    mand_byte,
    mes-
    sage_bytes)

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

```
class conpot.protocols.kamstrup.meter_protocol.messages.KamstrupResponseBase (communication_...)
    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,  
un-  
known,  
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** (*databus*, *starting_address*, *ending_address*)

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.cleansse_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,      out-
                                                             put_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,  recur-
                                                           sive=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, mib-paths*)

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 be 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, time-
                                                out=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 (Actu-
        ator 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 (methodName='runTest')
    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_comm,  

                                                                                   kam-  

                                                                                   strup_manage)
```


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.TestMySQLlogger (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': '0110ffffffffffffffffffffffffffff'}}

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

C

- conpot, 74
- conpot.core, 26
- conpot.core.attack_session, 17
- conpot.core.databus, 17
- conpot.core.filesystem, 17
- conpot.core.fs_utils, 23
- conpot.core.internal_interface, 24
- conpot.core.loggers, 17
- conpot.core.loggers.helpers, 15
- conpot.core.loggers.hpfriends, 15
- conpot.core.loggers.json_log, 16
- conpot.core.loggers.log_worker, 16
- conpot.core.loggers.sqlite_log, 16
- conpot.core.loggers.stix_transform, 16
- conpot.core.loggers.syslog, 16
- conpot.core.loggers.taxii_log, 16
- conpot.core.protocol_wrapper, 25
- conpot.core.session_manager, 25
- conpot.core.virtual_fs, 25
- conpot.emulators, 27
- conpot.emulators.misc, 27
- conpot.emulators.misc.random, 26
- conpot.emulators.misc.uptime, 26
- conpot.emulators.proxy, 27
- conpot.emulators.sensors, 27
- conpot.helpers, 74
- conpot.protocols, 60
- conpot.protocols.bacnet, 42
- conpot.protocols.bacnet.bacnet_app, 41
- conpot.protocols.bacnet.bacnet_server, 42
- conpot.protocols.enip, 42
- conpot.protocols.enip.enip_server, 42
- conpot.protocols.ftp, 46
- conpot.protocols.ftp.ftp_base_handler, 42
- conpot.protocols.ftp.ftp_handler, 43
- conpot.protocols.ftp.ftp_server, 45
- conpot.protocols.ftp.ftp_utils, 46
- conpot.protocols.guardian_ast, 46
- conpot.protocols.guardian_ast.guardian_ast_server, 46
- conpot.protocols.http, 48
- conpot.protocols.http.command_responder, 46
- conpot.protocols.http.web_server, 48
- conpot.protocols.IEC104, 41
- conpot.protocols.IEC104.DeviceDataController, 27
- conpot.protocols.IEC104.errors, 28
- conpot.protocols.IEC104.frames, 29
- conpot.protocols.IEC104.i_frames_check, 40
- conpot.protocols.IEC104.IEC104, 28
- conpot.protocols.IEC104.IEC104_server, 28
- conpot.protocols.IEC104.register, 41
- conpot.protocols.ipmi, 49
- conpot.protocols.ipmi.fakebmc, 48
- conpot.protocols.ipmi.fakesession, 48
- conpot.protocols.ipmi.ipmi_server, 49
- conpot.protocols.kamstrup, 54
- conpot.protocols.kamstrup.management_protocol, 52
- conpot.protocols.kamstrup.management_protocol.command_responder, 49
- conpot.protocols.kamstrup.management_protocol.command_responder, 49
- conpot.protocols.kamstrup.management_protocol.kamstrup, 52
- conpot.protocols.kamstrup.meter_protocol, 54
- conpot.protocols.kamstrup.meter_protocol.command_responder, 52
- conpot.protocols.kamstrup.meter_protocol.decoder_3, 52
- conpot.protocols.kamstrup.meter_protocol.kamstrup, 52

conpot.protocols.kamstrup.meter_protocol, 53
conpot.protocols.kamstrup.meter_protocol_callbacks, 53
conpot.protocols.kamstrup.meter_protocol_message_tests.test_kamstrup_meter_protocol, 53
conpot.protocols.kamstrup.meter_protocol_callbacks_tests.test_logger_json, 54
conpot.protocols.kamstrup.meter_protocol_callbacks_tests.test_logger_mysql, 54
conpot.protocols.kamstrup.meter_protocol_callbacks_tests.test_mac_addr, 54
conpot.protocols.kamstrup.usage_simulator, 54
conpot.protocols.misc, 55
conpot.protocols.misc.ascii_decoder, 54
conpot.protocols.modbus, 55
conpot.protocols.modbus.modbus_block_data_response_decoder, 55
conpot.protocols.modbus.modbus_server, 55
conpot.protocols.modbus.slave, 55
conpot.protocols.modbus.slave_db, 55
conpot.protocols.s7comm, 57
conpot.protocols.s7comm.cotp, 56
conpot.protocols.s7comm.exceptions, 56
conpot.protocols.s7comm.s7, 56
conpot.protocols.s7comm.s7_server, 57
conpot.protocols.s7comm.tpkt, 57
conpot.protocols.snmp, 59
conpot.protocols.snmp.build_pysnmp_mib_wrapper, 57
conpot.protocols.snmp.command_responder, 58
conpot.protocols.snmp.conpot_cmdrsp, 58
conpot.protocols.snmp.databus_mediator, 59
conpot.protocols.snmp.snmp_server, 59
conpot.protocols.tftp, 60
conpot.protocols.tftp.tftp_handler, 59
conpot.protocols.tftp.tftp_server, 60
conpot.tests, 73
conpot.tests.helpers, 62
conpot.tests.helpers.s7comm_client, 61
conpot.tests.helpers.snmp_client, 62
conpot.tests.test_bacnet_server, 62
conpot.tests.test_base, 63
conpot.tests.test_docs, 63
conpot.tests.test_enip_server, 63
conpot.tests.test_ext_ip_util, 64
conpot.tests.test_ftp, 64
conpot.tests.test_guardian_ast, 65
conpot.tests.test_hpfriends, 66
conpot.tests.test_http_server, 66
conpot.tests.test_iec104_server, 67
conpot.tests.test_ipmi_server, 67
conpot.tests.test_kamstrup_decoder, 68
conpot.tests.test_kamstrup_management_protocol, 68
conpot.tests.test_modbus_server, 69
conpot.tests.test_proxy, 70
conpot.tests.test_pysnmp_wrapper, 70
conpot.tests.test_s7_server, 71
conpot.tests.test_snmp_server, 71
conpot.tests.test_taxii, 71
conpot.tests.test_tftp, 71
conpot.tests.test_vfs, 72
conpot.utils, 74
conpot.utils.ext_ip, 74
conpot.utils.mac_addr, 74

A

- AbstractFS (class in conpot.core.filesystem), 17
- access() (conpot.core.filesystem.AbstractFS method), 17
- access() (conpot.core.fs_utils.SubAbstractFS method), 23
- AccessControlCommand (class in conpot.protocols.kamstrup.management_protocol.commands), 49
- add_block() (conpot.protocols.modbus.slave.MBSlave method), 55
- add_byte() (conpot.protocols.kamstrup.meter_protocol.request_parser.KamstrupRequestParser method), 54
- add_event() (conpot.core.attack_session.AttackSession method), 17
- add_object() (conpot.protocols.bacnet.bacnet_app.BACnetApp method), 41
- add_property() (conpot.protocols.bacnet.bacnet_app.BACnetApp method), 41
- add_protocol() (conpot.core.virtual_fs.VirtualFS method), 25
- add_protocol() (in module conpot.core), 26
- add_register() (conpot.protocols.kamstrup.meter_protocol.messages.KamstrupResponseRegister method), 53
- add_slave() (conpot.protocols.modbus.slave_db.SlaveBase method), 55
- add_users_to_group() (conpot.core.filesystem.AbstractFS method), 18
- AddOptions() (in module conpot.tests.helpers.s7comm_client), 61
- addr_in_hex() (in module conpot.protocols.IEC104.DeviceDataController), 27
- addSocketTransport() (conpot.protocols.snmp.command_responder.CommandResponder method), 58
- AlarmServerCommand (class in conpot.protocols.kamstrup.management_protocol.commands), 50
- aliastypes (conpot.protocols.IEC104.frames.asdu_head attribute), 32
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_1 attribute), 32
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_10 attribute), 32
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_100 attribute), 33
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_101 attribute), 33
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_102 attribute), 33
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_103 attribute), 33
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_11 attribute), 33
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_12 attribute), 33
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_13 attribute), 33
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_14 attribute), 34
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_15 attribute), 34
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_16 attribute), 34
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_17 attribute), 34
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_18 attribute), 34
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_19 attribute), 34
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_2 attribute), 34
- aliastypes (conpot.protocols.IEC104.frames.asdu_infobj_20 attribute), 35

AsciiDecoder	(class in con-	asdu_infobj_36	(class in con-
<i>pot.protocols.misc.ascii_decoder</i>), 54		<i>pot.protocols.IEC104.frames</i>), 36	
asdu_head	(class in conpot.protocols.IEC104.frames), 32	asdu_infobj_37	(class in con-
		<i>pot.protocols.IEC104.frames</i>), 36	
asdu_infobj_1	(class in con-	asdu_infobj_38	(class in con-
<i>pot.protocols.IEC104.frames</i>), 32		<i>pot.protocols.IEC104.frames</i>), 36	
asdu_infobj_10	(class in con-	asdu_infobj_39	(class in con-
<i>pot.protocols.IEC104.frames</i>), 32		<i>pot.protocols.IEC104.frames</i>), 36	
asdu_infobj_100	(class in con-	asdu_infobj_4	(class in con-
<i>pot.protocols.IEC104.frames</i>), 33		<i>pot.protocols.IEC104.frames</i>), 36	
asdu_infobj_101	(class in con-	asdu_infobj_40	(class in con-
<i>pot.protocols.IEC104.frames</i>), 33		<i>pot.protocols.IEC104.frames</i>), 36	
asdu_infobj_102	(class in con-	asdu_infobj_45	(class in con-
<i>pot.protocols.IEC104.frames</i>), 33		<i>pot.protocols.IEC104.frames</i>), 37	
asdu_infobj_103	(class in con-	asdu_infobj_46	(class in con-
<i>pot.protocols.IEC104.frames</i>), 33		<i>pot.protocols.IEC104.frames</i>), 37	
asdu_infobj_11	(class in con-	asdu_infobj_47	(class in con-
<i>pot.protocols.IEC104.frames</i>), 33		<i>pot.protocols.IEC104.frames</i>), 37	
asdu_infobj_12	(class in con-	asdu_infobj_48	(class in con-
<i>pot.protocols.IEC104.frames</i>), 33		<i>pot.protocols.IEC104.frames</i>), 37	
asdu_infobj_13	(class in con-	asdu_infobj_49	(class in con-
<i>pot.protocols.IEC104.frames</i>), 33		<i>pot.protocols.IEC104.frames</i>), 37	
asdu_infobj_14	(class in con-	asdu_infobj_5	(class in con-
<i>pot.protocols.IEC104.frames</i>), 34		<i>pot.protocols.IEC104.frames</i>), 37	
asdu_infobj_15	(class in con-	asdu_infobj_50	(class in con-
<i>pot.protocols.IEC104.frames</i>), 34		<i>pot.protocols.IEC104.frames</i>), 37	
asdu_infobj_16	(class in con-	asdu_infobj_51	(class in con-
<i>pot.protocols.IEC104.frames</i>), 34		<i>pot.protocols.IEC104.frames</i>), 38	
asdu_infobj_17	(class in con-	asdu_infobj_58	(class in con-
<i>pot.protocols.IEC104.frames</i>), 34		<i>pot.protocols.IEC104.frames</i>), 38	
asdu_infobj_18	(class in con-	asdu_infobj_59	(class in con-
<i>pot.protocols.IEC104.frames</i>), 34		<i>pot.protocols.IEC104.frames</i>), 38	
asdu_infobj_19	(class in con-	asdu_infobj_6	(class in con-
<i>pot.protocols.IEC104.frames</i>), 34		<i>pot.protocols.IEC104.frames</i>), 38	
asdu_infobj_2	(class in con-	asdu_infobj_60	(class in con-
<i>pot.protocols.IEC104.frames</i>), 34		<i>pot.protocols.IEC104.frames</i>), 38	
asdu_infobj_20	(class in con-	asdu_infobj_61	(class in con-
<i>pot.protocols.IEC104.frames</i>), 34		<i>pot.protocols.IEC104.frames</i>), 38	
asdu_infobj_21	(class in con-	asdu_infobj_62	(class in con-
<i>pot.protocols.IEC104.frames</i>), 35		<i>pot.protocols.IEC104.frames</i>), 38	
asdu_infobj_3	(class in con-	asdu_infobj_63	(class in con-
<i>pot.protocols.IEC104.frames</i>), 35		<i>pot.protocols.IEC104.frames</i>), 39	
asdu_infobj_30	(class in con-	asdu_infobj_64	(class in con-
<i>pot.protocols.IEC104.frames</i>), 35		<i>pot.protocols.IEC104.frames</i>), 39	
asdu_infobj_31	(class in con-	asdu_infobj_7	(class in con-
<i>pot.protocols.IEC104.frames</i>), 35		<i>pot.protocols.IEC104.frames</i>), 39	
asdu_infobj_32	(class in con-	asdu_infobj_8	(class in con-
<i>pot.protocols.IEC104.frames</i>), 35		<i>pot.protocols.IEC104.frames</i>), 39	
asdu_infobj_33	(class in con-	asdu_infobj_9	(class in con-
<i>pot.protocols.IEC104.frames</i>), 35		<i>pot.protocols.IEC104.frames</i>), 39	
asdu_infobj_34	(class in con-	assemble ()	(conpot.protocols.s7comm.cotp.COTP_ConnectionConfirm method), 56
<i>pot.protocols.IEC104.frames</i>), 35			
asdu_infobj_35	(class in con-	assemble ()	(conpot.protocols.s7comm.cotp.COTP_ConnectionRequest method), 56
<i>pot.protocols.IEC104.frames</i>), 36			

AssembleException, 56

AttackSession (class in *conpot.core.attack_session*), 17

attribute_operations () (con-
pot.tests.test_enip_server.TestENIPServer
method), 63

authentication_ok () (con-
pot.protocols.ftp.ftp_base_handler.FTPHandlerBase
method), 42

B

BACnetApp (class in *con-*
pot.protocols.bacnet.bacnet_app), 41

BaseCommand (class in *con-*
pot.protocols.kamstrup.management_protocol.commands), 50

BCR (class in *conpot.protocols.IEC104.frames*), 29

BruteTsap () (in module *con-*
pot.tests.helpers.s7comm_client), 61

BSI (class in *conpot.protocols.IEC104.frames*), 29

build () (*conpot.protocols.IEC104.IEC104.frame_object_with_time*
method), 28

C

c_BulkCommandResponder (class in *con-*
pot.protocols.snmp.conpot_cmdrsp), 58

c_GetCommandResponder (class in *con-*
pot.protocols.snmp.conpot_cmdrsp), 58

c_NextCommandResponder (class in *con-*
pot.protocols.snmp.conpot_cmdrsp), 58

c_SetCommandResponder (class in *con-*
pot.protocols.snmp.conpot_cmdrsp), 58

calctime () (in module *con-*
pot.protocols.IEC104.frames), 39

cancel_t1 () (*conpot.protocols.IEC104.IEC104.frame_object_with_time*
method), 28

cbFun () (*conpot.tests.helpers.snmp_client.SNMPClient*
method), 62

change_mac () (in module *conpot.utils.mac_addr*), 74

check_access () (*conpot.core.filesystem.AbstractFS*
method), 18

check_access () (con-
pot.core.fs_utils.SubAbstractFS method), 23

check_asdu_1 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_100 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_11 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_12 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_13 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_14 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_2 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_3 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_30 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_31 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_35 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_36 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_4 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_45 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_46 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_47 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_48 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_49 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_50 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_asdu_51 () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_command () (in module *con-*
pot.protocols.IEC104.i_frames_check), 40

check_command_resp_help_message ()
(in module *con-*
pot.tests.test_kamstrup_management_protocol), 68

check_content () (in module *con-*
pot.tests.test_pysnmp_wrapper), 71

check_evasive () (con-
pot.protocols.snmp.conpot_cmdrsp.conpot_extension
method), 59

check_information_with_time () (in module
conpot.protocols.IEC104.i_frames_check), 41

check_information_without_time () (in mod-
ule *conpot.protocols.IEC104.i_frames_check*),
41

check_registers () (con-
pot.protocols.IEC104.DeviceDataController.DeviceDataController
method), 27

chmod () (*conpot.core.filesystem.AbstractFS* method),
18

chmod () (*conpot.core.fs_utils.SubAbstractFS* method),
23

chown () (*conpot.core.filesystem.AbstractFS* method),

18
 chown () (*conpot.core.fs_utils.SubAbstractFS method*), 23
 chr_py3 () (*in module conpot.helpers*), 74
 clean () (*conpot.core.filesystem.AbstractFS method*), 18
 cleanse_byte_string () (*in module conpot.protocols.s7comm.s7_server*), 57
 close () (*conpot.core.virtual_fs.VirtualFS method*), 25
 close_fs () (*in module conpot.core*), 26
 close_server_session () (*conpot.protocols.ipmi.ipmi_server.IpmiServer method*), 49
 CMD_OUTPUT (*conpot.protocols.kamstrup.management_protocol.commands.AccessConfigFileCommand attribute*), 49
 CMD_OUTPUT (*conpot.protocols.kamstrup.management_protocol.commands.AlertServerCommand attribute*), 50
 CMD_OUTPUT (*conpot.protocols.kamstrup.management_protocol.commands.BaseCommand attribute*), 50
 CMD_OUTPUT (*conpot.protocols.kamstrup.management_protocol.commands.GetConfigCommand attribute*), 50
 CMD_OUTPUT (*conpot.protocols.kamstrup.management_protocol.commands.HelpCommand attribute*), 50
 CMD_OUTPUT (*conpot.protocols.kamstrup.management_protocol.commands.SetConfigCommand attribute*), 50
 CMD_OUTPUT (*conpot.protocols.kamstrup.management_protocol.commands.SetIPConnCommand attribute*), 51
 CMD_OUTPUT (*conpot.protocols.kamstrup.management_protocol.commands.SetKap1Command attribute*), 51
 CMD_OUTPUT (*conpot.protocols.kamstrup.management_protocol.commands.SetPortsCommand attribute*), 51
 CMD_OUTPUT (*conpot.protocols.kamstrup.management_protocol.commands.SetWatchdogCommand attribute*), 51
 CMD_OUTPUT (*conpot.protocols.kamstrup.management_protocol.commands.SoftwareVersionCommand attribute*), 52
 CMD_OUTPUT (*conpot.protocols.kamstrup.management_protocol.commands.WinkModuleCommand attribute*), 52
 CMD_OUTPUT_DOUBLE (*conpot.protocols.kamstrup.management_protocol.commands.SetKap2Command attribute*), 51
 CMD_OUTPUT_SINGLE (*conpot.protocols.kamstrup.management_protocol.commands.SetKap2Command attribute*), 51
 cold_reset () (*conpot.protocols.ipmi.fakebmc.FakeBmc method*), 48
 command_byte (*conpot.protocols.kamstrup.meter_protocol.messages.KamstrupRequestGetRegisters attribute*), 53
 COMMAND_NOT_FOUND (*conpot.protocols.kamstrup.management_protocol.commands.CommandRule attribute*), 49
 CommandResponder (*class in conpot.protocols.http.command_responder*), 46
 CommandResponder (*class in conpot.protocols.kamstrup.management_protocol.command_responder*), 49
 CommandResponder (*class in conpot.protocols.kamstrup.meter_protocol.command_responder*), 52
 CommandResponder (*class in conpot.protocols.snmp.command_responder*), 58
 compile_mib () (*in module conpot.protocols.snmp.build_pysnmp_mib_wrapper*), 57
 Connect () (*conpot.tests.helpers.s7comm_client.s7conpot (module)*), 74
 conpot (*module*), 74
 conpot.core.attack_session (*module*), 17
 conpot.core.filesystem (*module*), 17
 conpot.core.internal_interface (*module*), 17
 conpot.core.loggers (*module*), 17
 conpot.core.loggers.hpfriends (*module*), 15
 conpot.core.loggers.log_worker (*module*), 16
 conpot.core.loggers.sqlite_log (*module*), 16
 conpot.core.loggers.stix_transform (*module*), 16
 conpot.core.loggers.syslog (*module*), 16
 conpot.core.loggers.taxii_log (*module*), 16
 conpot.core.protocol_wrapper (*module*), 25
 conpot.core.virtual_fs (*module*), 25
 conpot.emulators (*module*), 27
 conpot.emulators.misc (*module*), 27
 conpot.emulators.misc.random (*module*), 26
 conpot.emulators.misc.uptime (*module*), 26
 conpot.emulators.sensors (*module*), 27
 conpot.helpers (*module*), 74
 conpot.protocols.bacnet (*module*), 42
 conpot.protocols.bacnet.bacnet_app (*module*), 41

conpot.protocols.bacnet.bacnet_server (module), 42
 conpot.protocols.enip (module), 42
 conpot.protocols.enip.enip_server (module), 42
 conpot.protocols.ftp (module), 46
 conpot.protocols.ftp.ftp_base_handler (module), 42
 conpot.protocols.ftp.ftp_handler (module), 43
 conpot.protocols.ftp.ftp_server (module), 45
 conpot.protocols.ftp.ftp_utils (module), 46
 conpot.protocols.guardian_ast (module), 46
 conpot.protocols.guardian_ast.guardian_ast_server (module), 46
 conpot.protocols.http (module), 48
 conpot.protocols.http.command_responder (module), 46
 conpot.protocols.http.web_server (module), 48
 conpot.protocols.IEC104 (module), 41
 conpot.protocols.IEC104.DeviceDataController (module), 27
 conpot.protocols.IEC104.errors (module), 28
 conpot.protocols.IEC104.frames (module), 29
 conpot.protocols.IEC104.i_frames_check (module), 40
 conpot.protocols.IEC104.IEC104 (module), 28
 conpot.protocols.IEC104.IEC104_server (module), 28
 conpot.protocols.IEC104.register (module), 41
 conpot.protocols.ipmi (module), 49
 conpot.protocols.ipmi.fakebmc (module), 48
 conpot.protocols.ipmi.fakesession (module), 48
 conpot.protocols.ipmi.ipmi_server (module), 49
 conpot.protocols.kamstrup (module), 54
 conpot.protocols.kamstrup.management_protocol (module), 52
 conpot.protocols.kamstrup.management_protocol.command_responder (module), 49
 conpot.protocols.kamstrup.management_protocol.commands (module), 49
 conpot.protocols.kamstrup.management_protocol.kamstrup_management_server (module), 52
 conpot.protocols.kamstrup.meter_protocol (module), 54
 conpot.protocols.kamstrup.meter_protocol.command_responder (module), 52
 conpot.protocols.kamstrup.meter_protocol.decoder_3 (module), 52
 conpot.protocols.kamstrup.meter_protocol.kamstrup (module), 52
 conpot.protocols.kamstrup.meter_protocol.kamstrup_ (module), 53
 conpot.protocols.kamstrup.meter_protocol.messages (module), 53
 conpot.protocols.kamstrup.meter_protocol.register (module), 54
 conpot.protocols.kamstrup.meter_protocol.request_p (module), 54
 conpot.protocols.kamstrup.usage_simulator (module), 54
 conpot.protocols.misc (module), 55
 conpot.protocols.misc.ascii_decoder (module), 54
 conpot.protocols.modbus (module), 55
 conpot.protocols.modbus.modbus_block_databus_mediat (module), 55
 conpot.protocols.modbus.modbus_server (module), 55
 conpot.protocols.modbus.slave (module), 55
 conpot.protocols.modbus.slave_db (module), 55
 conpot.protocols.s7comm (module), 57
 conpot.protocols.s7comm.cotp (module), 56
 conpot.protocols.s7comm.exceptions (module), 56
 conpot.protocols.s7comm.s7 (module), 56
 conpot.protocols.s7comm.s7_server (module), 57
 conpot.protocols.s7comm.tpkt (module), 57
 conpot.protocols.snmp (module), 59
 conpot.protocols.snmp.build_pysnmp_mib_wrapper (module), 57
 conpot.protocols.snmp.command_responder (module), 58
 conpot.protocols.snmp.conpot_cmdrsp (module), 58
 conpot.protocols.snmp.databus_mediator (module), 59
 conpot.protocols.snmp.snmp_server (module), 59
 conpot.protocols.snmp.snmp_server (module), 60
 conpot.protocols.tftp.tftp_handler (module), 59
 conpot.protocols.tftp.tftp_server (module), 60
 conpot.protocols.tftp.tftp_server_management_server (module), 60
 conpot.tests (module), 73
 conpot.tests.helpers (module), 62
 conpot.tests.helpers.s7comm_client (mod-

- ule*), 61
 - conpot.tests.helpers.snmp_client (*module*), 62
 - conpot.tests.test_bacnet_server (*module*), 62
 - conpot.tests.test_base (*module*), 63
 - conpot.tests.test_docs (*module*), 63
 - conpot.tests.test_enip_server (*module*), 63
 - conpot.tests.test_ext_ip_util (*module*), 64
 - conpot.tests.test_ftp (*module*), 64
 - conpot.tests.test_guardian_ast (*module*), 65
 - conpot.tests.test_hpfriends (*module*), 66
 - conpot.tests.test_http_server (*module*), 66
 - conpot.tests.test_iec104_server (*module*), 67
 - conpot.tests.test_ipmi_server (*module*), 67
 - conpot.tests.test_kamstrup_decoder (*module*), 68
 - conpot.tests.test_kamstrup_management_protocols (*module*), 68
 - conpot.tests.test_kamstrup_meter_protocol (*module*), 69
 - conpot.tests.test_logger_json (*module*), 69
 - conpot.tests.test_logger_mysql (*module*), 69
 - conpot.tests.test_mac_addr (*module*), 69
 - conpot.tests.test_modbus_server (*module*), 69
 - conpot.tests.test_proxy (*module*), 70
 - conpot.tests.test_pysnmp_wrapper (*module*), 70
 - conpot.tests.test_s7_server (*module*), 71
 - conpot.tests.test_snmp_server (*module*), 71
 - conpot.tests.test_taxii (*module*), 71
 - conpot.tests.test_tftp (*module*), 71
 - conpot.tests.test_vfs (*module*), 72
 - conpot.utils (*module*), 74
 - conpot.utils.ext_ip (*module*), 74
 - conpot.utils.mac_addr (*module*), 74
 - conpot_extension (*class in conpot.protocols.snmp.conpot_cmdrsp*), 59
 - conpot_protocol () (*in module conpot.core.protocol_wrapper*), 25
 - copy () (*conpot.core.filesystem.AbstractFS method*), 18
 - copy_files () (*in module conpot.core.fs_utils*), 24
 - COTP (*class in conpot.protocols.s7comm.cotp*), 56
 - COTP_ConnectionConfirm (*class in conpot.protocols.s7comm.cotp*), 56
 - COTP_ConnectionRequest (*class in conpot.protocols.s7comm.cotp*), 56
 - COTPConnectionPacket (*class in conpot.protocols.s7comm.cotp*), 56
 - COTPConnectionPacket (*class in conpot.tests.helpers.s7comm_client*), 61
 - COTPDatapacket (*class in conpot.tests.helpers.s7comm_client*), 61
 - CP16Time (*class in conpot.protocols.IEC104.frames*), 29
 - CP24Time (*class in conpot.protocols.IEC104.frames*), 29
 - CP56Time (*class in conpot.protocols.IEC104.frames*), 30
 - create_group () (*conpot.core.filesystem.AbstractFS method*), 18
 - create_jail () (*conpot.core.filesystem.AbstractFS method*), 18
- ## D
- daemon_threads (*conpot.protocols.http.command_responder.SubHTTPServer attribute*), 47
 - daemons (*conpot.protocols.tftp.tftp_handler.TFTPServerState attribute*), 60
 - Databus (*class in conpot.core.databus*), 17
 - DatabusMediator (*class in conpot.protocols.snmp.databus_mediator*), 59
 - decode_in () (*conpot.emulators.proxy.ProxyDecoder method*), 27
 - decode_in () (*conpot.protocols.kamstrup.meter_protocol.decoder_382.L method*), 52
 - decode_in () (*conpot.protocols.misc.ascii_decoder.AsciiDecoder method*), 54
 - decode_out () (*conpot.emulators.proxy.ProxyDecoder method*), 27
 - decode_out () (*conpot.protocols.kamstrup.meter_protocol.decoder_382.Decoder382 method*), 52
 - decode_out () (*conpot.protocols.misc.ascii_decoder.AsciiDecoder method*), 54
 - Decoder382 (*class in conpot.protocols.kamstrup.meter_protocol.decoder_382*), 52
 - default_gid (*conpot.core.fs_utils.SubAbstractFS attribute*), 23
 - default_group (*conpot.core.fs_utils.SubAbstractFS attribute*), 23
 - default_perms (*conpot.core.fs_utils.SubAbstractFS attribute*), 23
 - default_uid (*conpot.core.fs_utils.SubAbstractFS attribute*), 23
 - default_user (*conpot.core.fs_utils.SubAbstractFS attribute*), 23
 - DeviceDataController (*class in conpot.protocols.IEC104.DeviceDataController*),

E

[echo_server\(\)](#) (*conpot.tests.test_proxy.TestProxy* method), 70
[enabled](#) (*conpot.core.internal_interface.Interface* attribute), 24
[end\(\)](#) (*conpot.protocols.tftp.tftp_handler.TFTPContextServer* method), 59
[EnipConfig](#) (class in *conpot.protocols.enip.enip_server*), 42
[EnipConfig.Tag](#) (class in *conpot.protocols.enip.enip_server*), 42
[escape\(\)](#) (*conpot.protocols.kamstrup.meter_protocol.messages.KamstrupResponseBase* class method), 53
[extract_padding\(\)](#) (*conpot.protocols.IEC104.frames.CP16Time* method), 29
[extract_padding\(\)](#) (*conpot.protocols.IEC104.frames.CP24Time* method), 29
[extract_padding\(\)](#) (*conpot.protocols.IEC104.frames.QDP* method), 30
[extract_padding\(\)](#) (*conpot.protocols.IEC104.frames.QDS* method), 31
[extract_padding\(\)](#) (*conpot.protocols.IEC104.frames.SEP* method), 31
[extract_padding\(\)](#) (*conpot.protocols.IEC104.frames.SPE* method), 32
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_13* attribute), 34
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_14* attribute), 34
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_15* attribute), 34
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_16* attribute), 34
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_17* attribute), 34
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_18* attribute), 34
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_19* attribute), 34
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_2* attribute), 34
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_20* attribute), 35
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_21* attribute), 35
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_3* attribute), 35
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_30* attribute), 35
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_31* attribute), 35
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_32* attribute), 35
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_33* attribute), 35
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_34* attribute), 36

F

[FakeBmc](#) (class in *conpot.protocols.ipmi.fakebmc*), 48
[FakeSession](#) (class in *conpot.protocols.ipmi.fakesession*), 48
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_1* attribute), 32
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_10* attribute), 32
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_100* attribute), 33
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_101* attribute), 33
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_102* attribute), 33
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_103* attribute), 33
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_11* attribute), 33
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_12* attribute), 33
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_35* attribute), 36
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_36* attribute), 36
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_37* attribute), 36
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_38* attribute), 36
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_39* attribute), 36
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_4* attribute), 36
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_40* attribute), 37
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_45* attribute), 37
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_46* attribute), 37
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_47* attribute), 37
[fields_desc](#) (*conpot.protocols.IEC104.frames.asdu_infobj_48* attribute), 37

Function() (*conpot.tests.helpers.s7comm_client.s7 method*), 62

G

generate_dependencies() (*in module conpot.protocols.snmp.build_pysnmp_mib_wrapper*), 57

get_boot_device() (*conpot.protocols.ipmi.fakebmc.FakeBmc method*), 48

get_command() (*conpot.tests.helpers.snmp_client.SNMPClient method*), 62

get_data_from_iter() (*in module conpot.protocols.ftp.ftp_utils*), 46

get_databus() (*in module conpot.core*), 26

get_elapsed_time() (*conpot.protocols.ftp.ftp_base_handler.FTPMetrics method*), 43

get_entity_headers() (*conpot.protocols.http.command_responder.HTTPServer method*), 47

get_entitytrailers() (*conpot.protocols.http.command_responder.HTTPServer method*), 47

get_ext_ip() (*in module conpot.utils.ext_ip*), 74

get_gid() (*conpot.protocols.ftp.ftp_server.FTPConfig method*), 45

get_infoobj_list() (*conpot.protocols.IEC104.IEC104.IEC104 static method*), 28

get_interface() (*in module conpot.core*), 26

get_interface_ip() (*in module conpot.utils.ext_ip*), 74

get_metrics() (*conpot.protocols.ftp.ftp_base_handler.FTPMetrics method*), 43

get_object_from_reg() (*conpot.protocols.IEC104.DeviceDataController.DeviceDataController method*), 27

get_objects_and_properties() (*conpot.protocols.bacnet.bacnet_app.BACnetApp method*), 41

get_permissions() (*conpot.core.filesystem.AbstractFS method*), 19

get_permissions() (*conpot.core.fs_utils.SubAbstractFS method*), 23

get_power_state() (*conpot.protocols.ipmi.fakebmc.FakeBmc method*), 48

get_registers() (*conpot.protocols.IEC104.DeviceDataController.DeviceDataController method*), 27

get_request() (*conpot.protocols.kamstrup.meter_protocol.request_parser.Kamstrup method*), 54

get_response() (*conpot.protocols.snmp.databus_mediator.DatabusMediator method*), 59

get_server() (*conpot.emulators.proxy.Proxy method*), 27

get_session() (*conpot.core.session_manager.SessionManager method*), 25

get_session() (*in module conpot.core*), 26

get_session_count() (*conpot.core.session_manager.SessionManager method*), 25

getSessionManager() (*in module conpot.core*), 26

get_shapshot() (*conpot.core.databus.Databus method*), 17

get_status_headers() (*conpot.protocols.http.command_responder.HTTPServer method*), 47

get_statustrailers() (*conpot.protocols.http.command_responder.HTTPServer method*), 47

get_trigger_appendix() (*conpot.protocols.http.command_responder.HTTPServer method*), 47

get_uid() (*conpot.protocols.ftp.ftp_server.FTPConfig method*), 45

get_value() (*conpot.core.databus.Databus method*), 17

get_value() (*conpot.emulators.misc.random.Random16bitRegister method*), 26

get_value() (*conpot.emulators.misc.random.Random8BitRegisters method*), 26

get_value() (*conpot.emulators.misc.uptime.Uptime method*), 26

get_vfs() (*in module conpot.core*), 26

GetConfigCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 50

getcwd() (*conpot.core.filesystem.AbstractFS method*), 19

getcwd() (*conpot.core.fs_utils.SubAbstractFS method*), 23

getFieldval() (*conpot.protocols.IEC104.IEC104.frame_object_with_timer method*), 28

getFile() (*conpot.core.filesystem.AbstractFS method*), 19

GetIdentity() (*in module conpot.tests.helpers.s7comm_client*), 61

getinfo() (*conpot.core.filesystem.AbstractFS method*), 19
 getinfo() (*conpot.core.fs_utils.SubAbstractFS method*), 23
 getmeta() (*conpot.core.filesystem.AbstractFS method*), 19
 getmtime() (*conpot.core.filesystem.AbstractFS method*), 20
 getmtime() (*conpot.core.fs_utils.SubAbstractFS method*), 23
 getTimerResolution() (*conpot.protocols.snmp.command_responder.SNMPDispatcher method*), 58
 groups (*conpot.core.filesystem.AbstractFS attribute*), 20
 guess_payload_class() (*conpot.protocols.IEC104.frames.asdu_head method*), 32
H
 handle() (*conpot.emulators.proxy.Proxy method*), 27
 handle() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 42
 handle() (*conpot.protocols.ipmi.ipmi_server.IpmiServer method*), 49
 handle() (*conpot.protocols.s7comm.s7.S7 method*), 56
 handle() (*conpot.protocols.snmp.command_responder.SNMPDispatcher method*), 58
 handle() (*conpot.protocols.tftp.tftp_handler.TFTPServerState method*), 60
 handle() (*conpot.protocols.tftp.tftp_handler.TFTPState method*), 60
 handle() (*conpot.protocols.tftp.tftp_handler.TFTPStateServerRecvMRO method*), 60
 handle() (*conpot.protocols.tftp.tftp_handler.TFTPStateServerRecvMRO method*), 60
 handle() (*conpot.protocols.tftp.tftp_handler.TFTPStateServerStartMRO method*), 60
 handle_client_request() (*conpot.protocols.ipmi.ipmi_server.IpmiServer method*), 49
 handle_cmd_channel() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 42
 handle_data_channel() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 43
 handle_double_command46() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 28
 handle_i_frame() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 28
 handle_in_data() (*conpot.emulators.proxy.Proxy method*), 27
 handle_inro_command100() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 28
 handle_out_data() (*conpot.emulators.proxy.Proxy method*), 27
 handle_request() (*conpot.protocols.modbus.slave.MBSlave method*), 55
 handle_request() (*conpot.protocols.modbus.slave_db.SlaveBase method*), 55
 handle_s_frame() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 28
 handle_setpointfloatpoint_command50() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 28
 handle_setpointscaled_command49() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 28
 handle_single_command45() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 28
 handle_startendtag() (*conpot.protocols.http.command_responder.TemplateParser method*), 47
 handle_u_frame() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 28
 handleMgmtOperation() (*conpot.protocols.snmp.conpot_cmdrsp.c_BulkCommandResponder method*), 58
 handleMgmtOperation() (*conpot.protocols.snmp.conpot_cmdrsp.c_GetCommandResponder method*), 58
 handleMgmtOperation() (*conpot.protocols.snmp.conpot_cmdrsp.c_NextCommandResponder method*), 58
 handleMgmtOperation() (*conpot.protocols.snmp.conpot_cmdrsp.c_SetCommandResponder method*), 59
 has_mib() (*conpot.protocols.snmp.command_responder.CommandResponder method*), 58
 help() (*conpot.protocols.kamstrup.management_protocol.commands.Base method*), 50
 HELP_MESSAGE (*conpot.protocols.kamstrup.management_protocol.commands.AccessControl attribute*), 49
 HELP_MESSAGE (*conpot.protocols.kamstrup.management_protocol.commands.AlarmStatus attribute*), 50
 HELP_MESSAGE (*conpot.protocols.kamstrup.management_protocol.commands.AlarmStatus attribute*), 50

pot.protocols.kamstrup.management_protocol.commands.BaseCommand (class in *conpot.protocols.kamstrup.management_protocol.commands*), 50
 HELP_MESSAGE (con- *HPFriendsLogger* (class in *conpot.core.loggers.hpfriends*), 15
pot.protocols.kamstrup.management_protocol.commands.HTTPSGetConfigCommand (class in *conpot.protocols.http.command_responder*),
 HELP_MESSAGE (con- 46
pot.protocols.kamstrup.management_protocol.commands.RequestConnectCommand
 attribute), 50
 HELP_MESSAGE (con- *i2repr()* (*conpot.protocols.IEC104.frames.NormValueField*
pot.protocols.kamstrup.management_protocol.commands.RequestRestartCommand
 attribute), 50 *i_frame* (class in *conpot.protocols.IEC104.frames*), 39
 HELP_MESSAGE (con- *iAm()* (*conpot.protocols.bacnet.bacnet_app.BACnetApp*
pot.protocols.kamstrup.management_protocol.commands.SetConfigCommand
 attribute), 50 *IEC104* (class in *conpot.protocols.IEC104.IEC104*), 28
 HELP_MESSAGE (con- *IEC104Register* (class in *conpot.protocols.kamstrup.management_protocol.commands.SetDeviceNameIEC104Register*), 41
 attribute), 50 *iHave()* (*conpot.protocols.bacnet.bacnet_app.BACnetApp*
 HELP_MESSAGE (con- method), 41
pot.protocols.kamstrup.management_protocol.commands.SetIPCommand (con-
 attribute), 51 *method_req()* (*conpot.protocols.IEC104.IEC104.IEC104*
 HELP_MESSAGE (con- method), 28
pot.protocols.kamstrup.management_protocol.commands.SetKup1Command (con-
 attribute), 51 *pot.protocols.bacnet.bacnet_app.BACnetApp*
 HELP_MESSAGE (con- method), 41
pot.protocols.kamstrup.management_protocol.commands.SetKup2Command (*conpot.core.databus.Databus*
 attribute), 51 *method*), 17
 HELP_MESSAGE (con- *initialize()* (*conpot.protocols.kamstrup.usage_simulator.UsageSimulator*
pot.protocols.kamstrup.management_protocol.commands.SetLookupCommand
 attribute), 51 *method*), 54
 HELP_MESSAGE (con- *initialize_databus()* (*conpot.protocols.kamstrup.management_protocol.commands.SetNumberSeriesCommand*
 attribute), 51 *manager.SessionManager*
 HELP_MESSAGE (con- *initialize_vfs()* (*conpot.protocols.kamstrup.management_protocol.commands.SetPortsCommand*
 attribute), 51 *VirtualFS* method), 26
 HELP_MESSAGE (con- *initialize_vfs()* (in module *conpot.core*), 26
pot.protocols.kamstrup.management_protocol.commands.SetSerialControl (con-
 attribute), 51 *pot.protocols.ipmi.ipmi_server.IpmiServer*
 HELP_MESSAGE (con- *method*), 49
pot.protocols.kamstrup.management_protocol.commands.SetWatchdogIEC104.DeviceDataController),
 attribute), 51 28
 HELP_MESSAGE (con- *Interface* (class in *conpot.core.internal_interface*),
pot.protocols.kamstrup.management_protocol.commands.SoftwareVersionCommand
 attribute), 52
 HELP_MESSAGE (con- *INVALID_PARAMETER* (con-
pot.protocols.kamstrup.management_protocol.commands.WipeMemoryCommand
 attribute), 52 *InvalidFieldValueException*, 28
 HelpCommand (class in *conpot.protocols.IEC104.frames*), 30
pot.protocols.kamstrup.management_protocol.commands.Server (class in *conpot.protocols.ipmi.ipmi_server*), 49
 50
 hex_in_addr() (in module *conpot.protocols.IEC104.DeviceDataController*),
 28 *is_in()* (*conpot.protocols.modbus.modbus_block_databus_mediator.Mediator*
 method), 55
 host (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase*

J

json_default() (in module conpot.core.loggers.helpers), 15
 JsonLogger (class in conpot.core.loggers.json_log), 16

K

K162M (conpot.protocols.kamstrup.meter_protocol.kamstrup_constants.MeterTypes attribute), 53
 K351C (conpot.protocols.kamstrup.meter_protocol.kamstrup_constants.MeterTypes attribute), 53
 K382M (conpot.protocols.kamstrup.meter_protocol.kamstrup_constants.MeterTypes attribute), 53
 KamstrupProtocolBase (class in conpot.protocols.kamstrup.meter_protocol.messages), 53
 KamstrupRegister (class in conpot.protocols.kamstrup.meter_protocol.register), 54
 KamstrupRequestBase (class in conpot.protocols.kamstrup.meter_protocol.messages), 53
 KamstrupRequestGetRegisters (class in conpot.protocols.kamstrup.meter_protocol.messages), 53
 KamstrupRequestParser (class in conpot.protocols.kamstrup.meter_protocol.request_parser), 54
 KamstrupRequestUnknown (class in conpot.protocols.kamstrup.meter_protocol.messages), 53
 KamstrupResponseBase (class in conpot.protocols.kamstrup.meter_protocol.messages), 53
 KamstrupResponseRegister (class in conpot.protocols.kamstrup.meter_protocol.messages), 53

log() (conpot.core.loggers.sqlite_log.SQLiteLogger method), 16
 log() (conpot.core.loggers.syslog.SysLogger method), 16
 log() (conpot.core.loggers.taxii_log.TaxiiLogger method), 16
 log() (conpot.protocols.http.command_responder.HTTPServer method), 47
 log() (conpot.protocols.snmp.conpot_cmdrsp.conpot_extension method), 59
 log_session() (conpot.core.loggers.json_log.JsonLogger method), 16
 log_session() (conpot.core.loggers.sqlite_log.SQLiteLogger method), 16
 LogWorker (class in conpot.core.loggers.log_worker), 16

M

make_subdirs() (conpot.protocols.tftp.tftp_handler.TFTPStateServerRecvWRQ method), 60
 mkdir() (conpot.core.filesystem.AbstractFS method), 20
 MBSlave (class in conpot.protocols.modbus.slave), 55
 MeterTypes (class in conpot.protocols.kamstrup.meter_protocol.kamstrup_constants), 52
 mib2pysnmp() (in module conpot.protocols.snmp.build_pysnmp_mib_wrapper), 57
 mock_callback() (conpot.tests.test_snmp_server.TestSNMPServer method), 71
 ModbusBlockDatabusMediator (class in conpot.protocols.modbus.modbus_block_databus_mediator), 55

L

LESignedShortField (class in conpot.protocols.IEC104.frames), 30
 listdir() (conpot.core.filesystem.AbstractFS method), 20
 load_entity() (conpot.protocols.http.command_responder.HTTPServer method), 47
 load_status() (conpot.protocols.http.command_responder.HTTPServer method), 47
 log() (conpot.core.loggers.hpfriends.HPFriendsLogger method), 15
 log() (conpot.core.loggers.json_log.JsonLogger method), 16

mount_fs() (conpot.core.filesystem.AbstractFS method), 20
 move() (conpot.core.filesystem.AbstractFS method), 21
 move() (conpot.core.fs_utils.SubAbstractFS method), 24

N

NegotiatePDU() (conpot.tests.helpers.s7comm_client.s7 method), 62
 Network (class in conpot.core.internal_interface), 24
 norm_path() (conpot.core.filesystem.AbstractFS method), 21
 NormValueField (class in conpot.protocols.IEC104.frames), 30

notify_observers() (conpot.core.databus.Databus method), 17
 number_to_bytes() (in module conpot.helpers), 74
 NVA (class in conpot.protocols.IEC104.frames), 30

O

observe_value() (conpot.core.databus.Databus method), 17
 OCI (class in conpot.protocols.IEC104.frames), 30
 OMNIA (conpot.protocols.kamstrup.meter_protocol.kamstrup_constants.MeterTypes attribute), 53
 open() (conpot.core.filesystem.AbstractFS method), 21
 openbin() (conpot.core.filesystem.AbstractFS method), 21
 opendir() (conpot.core.filesystem.AbstractFS method), 21

P

pack() (conpot.protocols.s7comm.cotp.COTP method), 56
 pack() (conpot.protocols.s7comm.s7.S7 method), 56
 pack() (conpot.protocols.s7comm.tpkt.TPKT method), 57
 pack() (conpot.tests.helpers.s7comm_client.COTPConnectionPacket method), 61
 pack() (conpot.tests.helpers.s7comm_client.COTPDatapacket method), 61
 pack() (conpot.tests.helpers.s7comm_client.S7Packet method), 61
 pack() (conpot.tests.helpers.s7comm_client.TPKTPacket method), 62
 pack_short_int() (in module conpot.helpers), 74
 parse() (conpot.protocols.s7comm.cotp.COTP method), 56
 parse() (conpot.protocols.s7comm.s7.S7 method), 56
 parse() (conpot.protocols.s7comm.tpkt.TPKT method), 57
 parse_ip() (in module conpot.protocols.kamstrup.management_protocol.commands), 52
 parse_port() (in module conpot.protocols.kamstrup.management_protocol.commands), 52
 parse_template() (conpot.protocols.enip.enip_server.EnipConfig method), 42
 ParseException, 56
 payload_guess (conpot.protocols.IEC104.frames.asdu_head attribute), 32
 payload_guess (conpot.protocols.IEC104.frames.BCR attribute), 29
 payload_guess (conpot.protocols.IEC104.frames.DIQ attribute), 30
 payload_guess (conpot.protocols.IEC104.frames.i_frame attribute), 39
 payload_guess (conpot.protocols.IEC104.frames.QDP attribute), 31
 payload_guess (conpot.protocols.IEC104.frames.QDS attribute), 31
 payload_guess (conpot.protocols.IEC104.frames.QOS attribute), 31
 payload_guess (conpot.protocols.IEC104.frames.SEP attribute), 31
 payload_guess (conpot.protocols.IEC104.frames.SIQ attribute), 31
 payload_guess (conpot.protocols.IEC104.frames.SPE attribute), 32
 payload_guess (conpot.protocols.IEC104.frames.VTI attribute), 32
 plc_stop_function() (conpot.tests.helpers.s7comm_client.s7 method), 62
 plc_stop_signal() (conpot.protocols.s7comm.s7.S7 method), 56
 port (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase attribute), 43
 post_build() (conpot.protocols.IEC104.frames.i_frame method), 39
 power_cycle() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 48
 power_off() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 48
 power_on() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 48
 power_reset() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 48
 power_shutdown() (conpot.protocols.ipmi.fakebmc.FakeBmc method), 48
 process_ftp_command() (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method), 43
 process_ftp_command() (con-

- pot.protocols.ftp.ftp_handler.FTPCommandChannel.remove()* (*conpot.core.filesystem.AbstractFS method*), 45
 - Proxy (*class in conpot.emulators.proxy*), 27
 - ProxyDecoder (*class in conpot.emulators.proxy*), 27
 - purge_sessions() (*conpot.core.session_manager.SessionManager method*), 25
 - push_data() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 43
- Q**
- QDP (*class in conpot.protocols.IEC104.frames*), 30
 - QDS (*class in conpot.protocols.IEC104.frames*), 31
 - QOS (*class in conpot.protocols.IEC104.frames*), 31
- R**
- Random16bitRegister (*class in conpot.emulators.misc.random*), 26
 - Random8BitRegisters (*class in conpot.emulators.misc.random*), 26
 - readlink() (*conpot.core.filesystem.AbstractFS method*), 22
 - readlink() (*conpot.core.fs_utils.SubAbstractFS method*), 24
 - readProperty() (*conpot.protocols.bacnet.bacnet_app.BACnetApp method*), 41
 - ReadSZL() (*conpot.tests.helpers.s7comm_client.s7 method*), 62
 - recv_file() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 43
 - recvseq_increment() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 28
 - refresh_client() (*conpot.tests.test_ftp.TestFTPServer method*), 64
 - register() (*conpot.protocols.snmp.command_responder.CommandResponder method*), 58
 - register_user() (*conpot.core.filesystem.AbstractFS method*), 22
 - registerRecvCbFun() (*conpot.protocols.snmp.command_responder.SNMPDispatcher method*), 58
 - REGISTERS (*conpot.protocols.kamstrup.meter_protocol.decoder_382.Decoder382 attribute*), 52
 - registerTimerCbFun() (*conpot.protocols.snmp.command_responder.SNMPDispatcher method*), 58
 - registerTransport() (*conpot.protocols.snmp.command_responder.SNMPDispatcher method*), 58
 - remove() (*conpot.core.filesystem.AbstractFS method*), 22
 - remove() (*conpot.core.fs_utils.SubAbstractFS method*), 24
 - removedir() (*conpot.core.filesystem.AbstractFS method*), 22
 - removedir() (*conpot.core.fs_utils.SubAbstractFS method*), 24
 - Request() (*conpot.tests.helpers.s7comm_client.s7 method*), 62
 - request_diagnostics() (*conpot.protocols.s7comm.s7.S7 method*), 56
 - request_not_implemented() (*conpot.protocols.s7comm.s7.S7 method*), 56
 - request_ssl_17() (*conpot.protocols.s7comm.s7.S7 method*), 57
 - request_ssl_28() (*conpot.protocols.s7comm.s7.S7 method*), 57
 - RequestConnectCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 50
 - RequestRestartCommand (*class in conpot.protocols.kamstrup.management_protocol.commands*), 50
 - reset() (*conpot.core.databus.Databus method*), 17
 - respond() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method*), 43
 - respond() (*conpot.protocols.kamstrup.management_protocol.command_responder.CommandResponder method*), 49
 - respond() (*conpot.protocols.kamstrup.meter_protocol.command_responder.CommandResponder method*), 52
 - response() (*conpot.protocols.bacnet.bacnet_app.BACnetApp method*), 41
 - restart_t1() (*conpot.protocols.IEC104.IEC104.frame_object_with_timer method*), 28
 - restart_t1() (*conpot.protocols.IEC104.IEC104.IEC104 method*), 28
 - revert_mac() (*in module conpot.utils.mac_addr*), 74
 - root (*conpot.core.filesystem.AbstractFS attribute*), 22
 - root (*conpot.core.fs_utils.SubAbstractFS attribute*), 24
 - run() (*conpot.protocols.kamstrup.management_protocol.commands.Acceptance method*), 49
 - run() (*conpot.protocols.kamstrup.management_protocol.commands.Alarm method*), 50
 - run() (*conpot.protocols.kamstrup.management_protocol.commands.Base method*), 50
 - run() (*conpot.protocols.kamstrup.management_protocol.commands.GetC method*), 50
 - run() (*conpot.protocols.kamstrup.management_protocol.commands.Help method*), 50
 - run() (*conpot.protocols.kamstrup.management_protocol.commands.Requ method*), 50

run () (*conpot.protocols.kamstrup.management_protocol.commands.RestartCommand* (con-
method), 50 *pot.protocols.http.command_responder.HTTPServer*)

run () (*conpot.protocols.kamstrup.management_protocol.commands.SetConfigCommand*
method), 50 *send_file () (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase*)

run () (*conpot.protocols.kamstrup.management_protocol.commands.SetDeviceNameCommand*
method), 50 *send_frame_imm () (con-*

run () (*conpot.protocols.kamstrup.management_protocol.commands.SetIPControlIEC104.IEC104.IEC104*
method), 51 *method), 28*

run () (*conpot.protocols.kamstrup.management_protocol.commands.SetKeepAliveCommand* (con-
method), 51 *pot.protocols.ipmi.fakesession.FakeSession*)

run () (*conpot.protocols.kamstrup.management_protocol.commands.SetKeepAliveCommand*
method), 51 *send_payload () (con-*

run () (*conpot.protocols.kamstrup.management_protocol.commands.SetKeepAliveCommand*
method), 51 *send_payload () (con-*

run () (*conpot.protocols.kamstrup.management_protocol.commands.SetNameServerCommand* (con-
method), 51 *pot.protocols.http.command_responder.HTTPServer*)

run () (*conpot.protocols.kamstrup.management_protocol.commands.SetRouteCommand*
method), 51 *sendMessage () (con-*

run () (*conpot.protocols.kamstrup.management_protocol.commands.SetSerialConsoleCommand*
method), 51 *method), 58*

run () (*conpot.protocols.kamstrup.management_protocol.commands.SetWatchdogControlIEC104.frames*), 31
method), 51 *serialize () (conpot.protocols.kamstrup.meter_protocol.messages.Kam-*

run () (*conpot.protocols.kamstrup.management_protocol.commands.SetVersionCommand*
method), 52 *method), 53*

run_cmd () (*in module conpot.tests.test_ipmi_server*),
67 *serve_forever () (con-*

S *pot.protocols.http.command_responder.CommandResponder*
method), 46

S7 (*class in conpot.protocols.s7comm.s7*), 56 *serve_forever () (con-*

s7 (*class in conpot.tests.helpers.s7comm_client*), 62 *pot.protocols.snmp.command_responder.CommandResponder*
method), 58

S7Error, 61

S7Packet (*class in con-* *serverInitial () (con-*
pot.tests.helpers.s7comm_client), 61 *pot.protocols.tftp.tftp_handler.TFTPServerState*
method), 60

S7ProtocolError, 61

s_frame (*class in conpot.protocols.IEC104.frames*), 40

sanitize_file_name () (*in module con-* *SessionManager (class in con-*
pot.helpers), 74 *pot.core.session_manager*), 25

Scan () (*in module con-* *set_access_ip () (con-*
pot.tests.helpers.s7comm_client), 61 *pot.protocols.kamstrup.management_protocol.commands.Access*
method), 49

SCD (*class in conpot.protocols.IEC104.frames*), 31

select_data () (*con-* *set_boot_device () (con-*
pot.core.loggers.sqlite_log.SQLiteLogger *pot.protocols.ipmi.fakebmc.FakeBmc* *method*),
method), 16 48

send_104frame () (*con-* *set_command () (con-*
pot.protocols.IEC104.IEC104.IEC104 *pot.tests.helpers.snmp_client.SNMPCClient*
method), 28 method), 62

send_auth_cap () (*con-* *set_ended () (conpot.core.attack_session.AttackSession*
pot.protocols.ipmi.ipmi_server.IpmiServer *method*), 17

send_chunked () (*con-* *set_object_val () (con-*
pot.protocols.http.command_responder.HTTPServer *pot.protocols.IEC104.DeviceDataController.DeviceDataControll*
method), 47 method), 27

send_data () (*conpot.protocols.ipmi.fakesession.FakeSession* *set_val () (conpot.protocols.IEC104.register.IEC104Register*
method), 48 method), 41

set_value () (conpot.core.databus.Databus *method*),
17

set_value() (*conpot.protocols.snmp.databus_mediator.DatabusMediator* (class in *conpot.tests.test_ipmi_server.TestIPMI* method), 59)

setbinfile() (*conpot.core.filesystem.AbstractFS* method), 22

SetConfigCommand (class in *conpot.protocols.kamstrup.management_protocol.commands*), 50

SetDeviceNameCommand (class in *conpot.protocols.kamstrup.management_protocol.commands*), 50

setinfo() (*conpot.core.filesystem.AbstractFS* method), 22

SetIPCommand (class in *conpot.protocols.kamstrup.management_protocol.commands*), 50

SetKap1Command (class in *conpot.protocols.kamstrup.management_protocol.commands*), 51

SetKap2Command (class in *conpot.protocols.kamstrup.management_protocol.commands*), 51

SetLookupCommand (class in *conpot.protocols.kamstrup.management_protocol.commands*), 51

SetNameserverCommand (class in *conpot.protocols.kamstrup.management_protocol.commands*), 51

SetPortsCommand (class in *conpot.protocols.kamstrup.management_protocol.commands*), 51

SetSerialCommand (class in *conpot.protocols.kamstrup.management_protocol.commands*), 51

settimes() (*conpot.core.filesystem.AbstractFS* method), 22

setup() (*conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase* method), 43

setUp() (*conpot.tests.test_bacnet_server.TestBACnetServer* method), 62

setUp() (*conpot.tests.test_base.TestBase* method), 63

setUp() (*conpot.tests.test_docs.TestMakeDocs* method), 63

setUp() (*conpot.tests.test_enip_server.TestENIPServer* method), 63

setUp() (*conpot.tests.test_ext_ip_util.TestExtIPUtil* method), 64

setUp() (*conpot.tests.test_ftp.TestFTPServer* method), 64

setUp() (*conpot.tests.test_guardian_ast.TestGuardianAST* method), 65

setUp() (*conpot.tests.test_http_server.TestHTTPServer* method), 66

setUp() (*conpot.tests.test_iec104_server.TestIEC104Server* method), 67

setUp() (*conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol* method), 68

setUp() (*conpot.tests.test_kamstrup_meter_protocol.TestKamstrupMeterProtocol* method), 69

setUp() (*conpot.tests.test_logger_json.TestJsonLogger* method), 69

setUp() (*conpot.tests.test_mac_addr.TestMacAddrUtil* method), 69

setUp() (*conpot.tests.test_modbus_server.TestModbusServer* method), 69

setUp() (*conpot.tests.test_pysnmp_wrapper.TestPySNMPWrapper* method), 70

setUp() (*conpot.tests.test_s7_server.TestS7Server* method), 71

setUp() (*conpot.tests.test_snmp_server.TestSNMPServer* method), 71

setUp() (*conpot.tests.test_tftp.TestTFTPServer* method), 71

setUp() (*conpot.tests.test_vfs.TestFileSystem* method), 72

setUp() (*conpot.tests.test_vfs.TestSubFileSystem* method), 73

SetWatchdogCommand (class in *conpot.protocols.kamstrup.management_protocol.commands*), 51

show_send_list() (*conpot.protocols.IEC104.IEC104.IEC104Frames* method), 28

SIQ (class in *conpot.protocols.IEC104.frames*), 31

SlaveDB (class in *conpot.protocols.modbus.slave_db*), 55

SNMPClient (class in *conpot.tests.helpers.snmp_client*), 62

SNMPDispatcher (class in *conpot.protocols.snmp.command_responder*), 58

SoftwareVersionCommand (class in *conpot.protocols.kamstrup.management_protocol.commands*), 51

SPE (class in *conpot.protocols.IEC104.frames*), 31

Split() (in module *conpot.tests.helpers.s7comm_client*), 61

SQLiteLogger (class in *conpot.core.loggers.sqlite_log*), 16

ssl_lists (*conpot.protocols.s7comm.s7.S7* attribute), 57

start() (*conpot.core.loggers.log_worker.LogWorker* method), 16

start() (*conpot.protocols.ipmi.ipmi_server.IpmiServer* method), 49

start() (*conpot.protocols.tftp.tftp_handler.TFTPContextServer* method), 59

start_data_channel() (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method), 43
stat() (conpot.core.filesystem.AbstractFS method), 22
stat() (conpot.core.fs_utils.SubAbstractFS method), 24
StixTransformer (class in conpot.core.loggers.stix_transform), 16
stop() (conpot.core.loggers.log_worker.LogWorker method), 16
stop() (conpot.emulators.proxy.Proxy method), 27
stop() (conpot.protocols.http.command_responder.CommandResponder method), 46
stop() (conpot.protocols.ipmi.ipmi_server.IpmiServer method), 49
stop() (conpot.protocols.kamstrup.usage_simulator.UsageSimulator method), 54
stop() (conpot.protocols.snmp.command_responder.CommandResponder method), 58
stop_data_channel() (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase method), 43
str_to_bytes() (in module conpot.helpers), 74
stream_server_handle() (conpot.protocols.ftp.ftp_base_handler.FTPHandlerBase class method), 43
StripUnprintable() (in module conpot.tests.helpers.s7comm_client), 61
SubAbstractFS (class in conpot.core.fs_utils), 23
SubHTTPServer (class in conpot.protocols.http.command_responder), 47
substitute_template_fields() (conpot.protocols.http.command_responder.HTTPServer method), 47
SVA (class in conpot.protocols.IEC104.frames), 32
SysLogger (class in conpot.core.loggers.syslog), 16

T

take_snapshot() (conpot.core.filesystem.AbstractFS method), 23
TaxiiLogger (class in conpot.core.loggers.taxii_log), 16
tearDown() (conpot.tests.test_bacnet_server.TestBACnetServer method), 62
tearDown() (conpot.tests.test_base.TestBase method), 63
tearDown() (conpot.tests.test_docs.TestMakeDocs method), 63
tearDown() (conpot.tests.test_enip_server.TestENIPServer method), 63
tearDown() (conpot.tests.test_ext_ip_util.TestExtIPUtil method), 64
tearDown() (conpot.tests.test_ftp.TestFTPServer method), 64
tearDown() (conpot.tests.test_guardian_ast.TestGuardianAST method), 65
tearDown() (conpot.tests.test_http_server.TestHTTPServer method), 66
tearDown() (conpot.tests.test_iec104_server.TestIEC104Server method), 67
tearDown() (conpot.tests.test_ipmi_server.TestIPMI method), 67
tearDown() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol method), 68
tearDown() (conpot.tests.test_kamstrup_meter_protocol.TestKamstrupMeterProtocol method), 69
tearDown() (conpot.tests.test_logger_json.TestJsonLogger method), 69
tearDown() (conpot.tests.test_mac_addr.TestMacAddrUtil method), 69
tearDown() (conpot.tests.test_modbus_server.TestModbusServer method), 69
tearDown() (conpot.tests.test_s7_server.TestS7Server method), 71
tearDown() (conpot.tests.test_snmp_server.TestSNMPServer method), 71
tearDown() (conpot.tests.test_tftp.TestTFTPServer method), 72
tearDown() (conpot.tests.test_vfs.TestFileSystem method), 72
tearDown() (conpot.tests.test_vfs.TestSubFileSystem method), 73
TemplateParser (class in conpot.protocols.http.command_responder), 47
test_abort() (conpot.tests.test_ftp.TestFTPServer method), 64
test_access() (conpot.tests.test_vfs.TestFileSystem method), 72
test_access() (conpot.tests.test_vfs.TestSubFileSystem method), 73
test_access_control_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol method), 68
test_alarm_server_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol method), 68
test_allo() (conpot.tests.test_ftp.TestFTPServer method), 64
test_appe() (conpot.tests.test_ftp.TestFTPServer method), 64
test_ascii_decoder() (conpot.tests.test_proxy.TestProxy method), 70
test_ast_error() (conpot.tests.test_guardian_ast.TestGuardianAST method), 65

method), 66
 test_auth() (*conpot.tests.test_ftp.TestFTPServer method*), 64
 test_base() (*conpot.tests.test_base.TestBase method*), 63
 test_boot_device() (*conpot.tests.test_ipmi_server.TestIPMI method*), 67
 test_channel_get_access() (*conpot.tests.test_ipmi_server.TestIPMI method*), 67
 test_chassis_status() (*conpot.tests.test_ipmi_server.TestIPMI method*), 67
 test_chmod() (*conpot.tests.test_vfs.TestFileSystem method*), 72
 test_chmod() (*conpot.tests.test_vfs.TestSubFileSystem method*), 73
 test_chown() (*conpot.tests.test_vfs.TestFileSystem method*), 72
 test_chown() (*conpot.tests.test_vfs.TestSubFileSystem method*), 73
 test_compile() (*conpot.tests.test_pysnmp_wrapper.TestPySNMPWrapper method*), 70
 test_copydir() (*conpot.tests.test_vfs.TestFileSystem method*), 72
 test_copyfile() (*conpot.tests.test_vfs.TestFileSystem method*), 72
 test_cwd() (*conpot.tests.test_ftp.TestFTPServer method*), 64
 test_delete() (*conpot.tests.test_ftp.TestFTPServer method*), 64
 test_do_HEAD() (*conpot.tests.test_http_server.TestHTTPServer method*), 66
 test_do_OPTIONS() (*conpot.tests.test_http_server.TestHTTPServer method*), 66
 test_do_POST() (*conpot.tests.test_http_server.TestHTTPServer method*), 66
 test_do_TRACE() (*conpot.tests.test_http_server.TestHTTPServer method*), 66
 test_ext_util() (*conpot.tests.test_ext_ip_util.TestExtIPUtil method*), 64
 test_fetch_ext_ip() (*conpot.tests.test_ext_ip_util.TestExtIPUtil method*), 64
 test_file_rename() (*conpot.tests.test_ftp.TestFTPServer method*), 64
 test_find() (*conpot.tests.test_pysnmp_wrapper.TestPySNMPWrapper method*), 70
 test_format_list() (*conpot.tests.test_vfs.TestFileSystem method*), 72
 test_format_list() (*conpot.tests.test_vfs.TestSubFileSystem method*), 73
 test_get_config_command() (*conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol method*), 68
 test_get_cwd() (*conpot.tests.test_vfs.TestFileSystem method*), 72
 test_get_cwd() (*conpot.tests.test_vfs.TestSubFileSystem method*), 73
 test_get_permissions() (*conpot.tests.test_vfs.TestFileSystem method*), 72
 test_get_permissions() (*conpot.tests.test_vfs.TestSubFileSystem method*), 73
 test_get_software_version_command() (*conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol method*), 68
 test_getmtime() (*conpot.tests.test_vfs.TestFileSystem method*), 72
 test_help() (*conpot.tests.test_ftp.TestFTPServer method*), 64
 test_help_command() (*conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol method*), 68
 Test_HPFriends (*class in conpot.tests.test_hpfriends*), 66
 test_hpfriends() (*conpot.tests.test_hpfriends.Test_HPFriends method*), 66
 test_http_backend_databus() (*conpot.tests.test_http_server.TestHTTPServer method*), 66
 test_http_backend_tarpit() (*conpot.tests.test_http_server.TestHTTPServer method*), 66
 test_http_request_base() (*conpot.tests.test_http_server.TestHTTPServer method*), 66
 test_http_subselect_trigger() (*conpot.tests.test_http_server.TestHTTPServer method*), 66

method), 66
test_I20100 () (*conpot.tests.test_guardian_ast.TestGuardianAST method*), 65
test_I20200 () (*conpot.tests.test_guardian_ast.TestGuardianAST method*), 65
test_I20300 () (*conpot.tests.test_guardian_ast.TestGuardianAST method*), 65
test_I20400 () (*conpot.tests.test_guardian_ast.TestGuardianAST method*), 65
test_I20500 () (*conpot.tests.test_guardian_ast.TestGuardianAST method*), 65
test_invalid_crc () (*conpot.tests.test_kamstrup_decoder.TestKamstrupDecoder method*), 68
test_ip_verify () (*conpot.tests.test_ext_ip_util.TestExtIPUtil method*), 64
test_jail () (*conpot.tests.test_vfs.TestFileSystem method*), 72
test_list () (*conpot.tests.test_ftp.TestFTPServer method*), 64
test_list_identity_tcp () (*conpot.tests.test_enip_server.TestENIPServer method*), 63
test_list_identity_udp () (*conpot.tests.test_enip_server.TestENIPServer method*), 63
test_list_interfaces_tcp () (*conpot.tests.test_enip_server.TestENIPServer method*), 63
test_list_interfaces_udp () (*conpot.tests.test_enip_server.TestENIPServer method*), 63
test_list_services_tcp () (*conpot.tests.test_enip_server.TestENIPServer method*), 63
test_list_services_udp () (*conpot.tests.test_enip_server.TestENIPServer method*), 63
test_listdir () (*conpot.tests.test_vfs.TestFileSystem method*), 72
test_listdir () (*conpot.tests.test_vfs.TestSubFileSystem method*), 73
test_log_event () (*conpot.tests.test_logger_json.TestJsonLogger method*), 69
test_mac () (*conpot.tests.test_mac_addr.TestMacAddrUtil method*), 69
test_make_docs () (*conpot.tests.test_docs.TestMakeDocs method*), 63
test_malformend_request_tcp () (*conpot.tests.test_enip_server.TestENIPServer method*), 63
test_malformend_request_udp () (*conpot.tests.test_enip_server.TestENIPServer method*), 64
test_max_retries () (*conpot.tests.test_ftp.TestFTPServer method*), 64
test_mdtm () (*conpot.tests.test_ftp.TestFTPServer method*), 65
test_misc () (*conpot.tests.test_ipmi_server.TestIPMI method*), 67
test_mkcd () (*conpot.tests.test_ftp.TestFTPServer method*), 65
test_mkdir () (*conpot.tests.test_vfs.TestFileSystem method*), 72
test_mkdir () (*conpot.tests.test_vfs.TestSubFileSystem method*), 73
test_mkdir_upload () (*conpot.tests.test_tftp.TestTFTPServer method*), 72
test_mkdirs () (*conpot.tests.test_vfs.TestFileSystem method*), 72
test_mkdirs () (*conpot.tests.test_vfs.TestSubFileSystem method*), 73
test_modbus_logging () (*conpot.tests.test_modbus_server.TestModbusServer method*), 70
test_mode () (*conpot.tests.test_ftp.TestFTPServer method*), 65
test_movedir () (*conpot.tests.test_vfs.TestFileSystem method*), 72
test_movefile () (*conpot.tests.test_vfs.TestFileSystem method*), 72
TestMySQLlogger (*class in conpot.tests.test_logger_mysql*), 69
test_mysqllogger () (*conpot.tests.test_logger_mysql.TestMySQLlogger method*), 69
test_nlist () (*conpot.tests.test_ftp.TestFTPServer method*), 65
test_no_response_requests () (*conpot.tests.test_bacnet_server.TestBACnetServer method*), 62
test_noop () (*conpot.tests.test_ftp.TestFTPServer method*), 69

<i>method</i>), 68	test_stor() (conpot.tests.test_ftp.TestFTPServer
test_set_lookup_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol	method), 65
<i>method</i>), 68	test_stru() (conpot.tests.test_ftp.TestFTPServer
test_set_name_server_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol	method), 65
<i>method</i>), 68	test_syst() (conpot.tests.test_ftp.TestFTPServer
test_set_ports_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol	method), 71
<i>method</i>), 68	test_testfr() (conpot.tests.test_iec104_server.TestIEC104Server
test_set_serial_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol	method), 67
<i>method</i>), 68	test_tftp_download() (conpot.tests.test_tftp.TestTFTPServer
test_set_time() (conpot.tests.test_vfs.TestSubFileSystem	method), 72
73	test_tftp_upload() (conpot.tests.test_tftp.TestTFTPServer
test_set_watchdog_command() (conpot.tests.test_kamstrup_management_protocol.TestKamstrupManagementProtocol	method), 72
<i>method</i>), 68	test_type() (conpot.tests.test_ftp.TestFTPServer
test_site() (conpot.tests.test_ftp.TestFTPServer	method), 65
<i>method</i>), 65	test_user_list() (conpot.tests.test_ipmi_server.TestIPMI
test_site_chmod() (conpot.tests.test_ftp.TestFTPServer	method), 67
65	test_utime() (conpot.tests.test_vfs.TestFileSystem
test_site_help() (conpot.tests.test_ftp.TestFTPServer	method), 73
65	test_utime() (conpot.tests.test_vfs.TestSubFileSystem
test_size() (conpot.tests.test_ftp.TestFTPServer	method), 73
<i>method</i>), 65	test_whoHas() (conpot.tests.test_bacnet_server.TestBACnetServer
test_snapshot() (conpot.tests.test_vfs.TestFileSystem	method), 63
73	test_whoIs() (conpot.tests.test_bacnet_server.TestBACnetServer
test_snmp_get() (conpot.tests.test_snmp_server.TestSNMPServer	method), 63
<i>method</i>), 71	test_wrapper_output() (conpot.tests.test_pysnmp_wrapper.TestPySNMPWrapper
test_snmp_set() (conpot.tests.test_snmp_server.TestSNMPServer	method), 70
<i>method</i>), 71	test_wrapper_processing() (conpot.tests.test_pysnmp_wrapper.TestPySNMPWrapper
test_ssl_proxy() (conpot.tests.test_proxy.TestProxy	method), 70
<i>method</i>), 70	test_write_for_non_existing() (conpot.tests.test_iec104_server.TestIEC104Server
test_ssl_proxy_with_decoder() (conpot.tests.test_proxy.TestProxy	method), 67
<i>method</i>), 70	test_write_no_relation_for_existing() (conpot.tests.test_iec104_server.TestIEC104Server
test_startdt() (conpot.tests.test_iec104_server.TestIEC104Server	method), 67
<i>method</i>), 67	test_write_read_coils() (conpot.tests.test_modbus_server.TestModbusServer
test_stat() (conpot.tests.test_ftp.TestFTPServer	method), 70
<i>method</i>), 65	test_write_relation_for_existing() (conpot.tests.test_iec104_server.TestIEC104Server
test_stat() (conpot.tests.test_vfs.TestFileSystem	method), 67
<i>method</i>), 73	test_write_tags() (con-
test_stat() (conpot.tests.test_vfs.TestSubFileSystem	<i>method</i>), 71
<i>method</i>), 73	
test_stix_transform() (conpot.tests.test_taxii.TestLoggers	
<i>method</i>), 71	

pot.tests.test_enip_server.TestENIPServer (method), 64
test_write_wrong_type_for_existing() (*conpot.tests.test_iec104_server.TestIEC104Server* method), 67
TestBACnetServer (class in *conpot.tests.test_bacnet_server*), 62
TestBase (class in *conpot.tests.test_base*), 63
TestENIPServer (class in *conpot.tests.test_enip_server*), 63
TestExtIPUtil (class in *conpot.tests.test_ext_ip_util*), 64
TestFileSystem (class in *conpot.tests.test_vfs*), 72
TestFTPServer (class in *conpot.tests.test_ftp*), 64
TestGuardianAST (class in *conpot.tests.test_guardian_ast*), 65
TestHTTPServer (class in *conpot.tests.test_http_server*), 66
TestIEC104Server (class in *conpot.tests.test_iec104_server*), 67
TestIPMI (class in *conpot.tests.test_ipmi_server*), 67
TestJsonLogger (class in *conpot.tests.test_logger_json*), 69
TestKamstrup (class in *conpot.tests.test_kamstrup_meter_protocol*), 69
TestKamstrupDecoder (class in *conpot.tests.test_kamstrup_decoder*), 68
TestKamstrupManagementProtocol (class in *conpot.tests.test_kamstrup_management_protocol*), 68
TestLoggers (class in *conpot.tests.test_taxii*), 71
TestMacAddrUtil (class in *conpot.tests.test_mac_addr*), 69
TestMakeDocs (class in *conpot.tests.test_docs*), 63
TestModbusServer (class in *conpot.tests.test_modbus_server*), 69
TestProxy (class in *conpot.tests.test_proxy*), 70
TestPySNMPWrapper (class in *conpot.tests.test_pysnmp_wrapper*), 70
TestS7Server (class in *conpot.tests.test_s7_server*), 71
TestSNMPServer (class in *conpot.tests.test_snmp_server*), 71
TestSubFileSystem (class in *conpot.tests.test_vfs*), 73
TestTFTPServer (class in *conpot.tests.test_ftp*), 71
TFTPContextServer (class in *conpot.protocols.ftp.tftp_handler*), 59
TFTPServerState (class in *conpot.protocols.ftp.tftp_handler*), 59
TFTPState (class in *conpot.protocols.ftp.tftp_handler*), 60
TFTPStateServerRecvRRQ (class in *conpot.protocols.ftp.tftp_handler*), 60
TFTPStateServerRecvWRQ (class in *conpot.protocols.ftp.tftp_handler*), 60
TFTPStateServerStart (class in *conpot.protocols.ftp.tftp_handler*), 60
ThreadedHTTPServer (class in *conpot.protocols.http.command_responder*), 48
timeout (*conpot.protocols.ftp.ftp_base_handler.FTPMetrics* attribute), 43
Timeout_t1, 29
Timeout_t1_2nd, 29
Timeout_t3, 29
TPKT (class in *conpot.protocols.s7comm.tpkt*), 57
TPKTPacket (class in *conpot.tests.helpers.s7comm_client*), 61
transform() (*conpot.core.loggers.stix_transform.StixTransformer* method), 16
try_parse_uint() (in module *conpot.protocols.kamstrup.management_protocol.commands*), 52

U

u_frame (class in *conpot.protocols.IEC104.frames*), 40
Unknown (*conpot.protocols.kamstrup_meter_protocol.kamstrup_constants* attribute), 53
unpack() (*conpot.tests.helpers.s7comm_client.COTPCConnectionPacket* method), 61
unpack() (*conpot.tests.helpers.s7comm_client.COTPDDataPacket* method), 61
unpack() (*conpot.tests.helpers.s7comm_client.S7Packet* method), 61
unpack() (*conpot.tests.helpers.s7comm_client.TPKTPacket* method), 62
unpack_short_int() (in module *conpot.helpers*), 74
update_evasion_table() (*conpot.protocols.snmp.databases_mediator.DatabasesMediator* method), 59
Uptime (class in *conpot.emulators.misc.uptime*), 26
usage_counter() (*conpot.protocols.kamstrup.usage_simulator.UsageSimulator* method), 54
UsageSimulator (class in *conpot.protocols.kamstrup.usage_simulator*), 54
user_groups (*conpot.core.filesystem.AbstractFS* attribute), 23
users (*conpot.core.filesystem.AbstractFS* attribute), 23

V

valid_crc() (*conpot.protocols.kamstrup_meter_protocol.decoder_382.L* class method), 52

`valid_crc()` (*conpot.protocols.kamstrup.meter_protocol.request_parser.KamstrupRequestParser* class method), 54

`vfs` (*conpot.protocols.tftp.tftp_handler.TFTPServerState* attribute), 60

`VirtualFS` (*class in conpot.core.virtual_fs*), 25

`VTI` (*class in conpot.protocols.IEC104.frames*), 32

W

`walk_command()` (*conpot.tests.helpers.snmp_client.SNMPCClient* method), 62

`whoHas()` (*conpot.protocols.bacnet.bacnet_app.BACnetApp* method), 41

`whoIs()` (*conpot.protocols.bacnet.bacnet_app.BACnetApp* method), 41

`WinkModuleCommand` (*class in conpot.protocols.kamstrup.management_protocol.commands*), 52