
AtomShields Documentation

Release 1.2.0

ElevenPaths

Apr 04, 2018

Contents:

1	Getting started	1
1.1	What is AtomShields?	1
1.2	Installation	1
1.3	Basic usage	1
1.4	Run tests	2
1.5	Generate docs	2
2	Checkers	3
2.1	DS_STORE	3
2.2	RetireJS	3
2.3	Target-blank	3
3	Reports	5
3.1	Echo	5
3.2	Http Request	5
4	atomshields	7
4.1	atomshields package	7
5	What is AtomShields?	19
6	Installation	21
7	Basic usage	23
8	Indices and tables	25
	Python Module Index	27

1.1 What is AtomShields?

Security testing framework for repositories and source code.

This system has different modules that detect different vulnerabilities or files that may expose a risk, and the results obtained can be obtained or sent thanks to the reporting modules.

For developers: There is also the possibility to develop your own vulnerability detection (called checkers) or reporting modules. This tool offers certain facilities for those who want to implement their own modules, publish them and integrate them into the official ElevenPaths tool.

1.2 Installation

```
pip install atomshields
```

1.3 Basic usage

```
import atomshields

atoms = atomshields.AtomShieldsScanner('./MyRepo/')
atoms.project = "MyRepoName"

issues = atoms.executeCheckers()
```

1.4 Run tests

```
# If you are in AtomShields directory  
pip install -r requirements-dev.txt  
py.test tests/
```

1.5 Generate docs

```
# If you are in AtomShields directory  
pip install -r requirements-dev.txt  
cd docs  
make html
```

2.1 DS_STORE

DS_STORE is a Mac OS X file containing information about the system that created it. These files are rarely filtered in. gitignore, providing information about the system of the author of the repository.

```
atomshields.checkers.dsstore.  
DSStoreChecker()
```

2.2 RetireJS

The goal of Retire.js is to help you detect use of version of JavaScript libraries with known vulnerabilities. This checker finds js files with vulnerabilities. Also, the checker finds and download JS files linked via URL.

```
atomshields.checkers.retirejs.  
RetireJSChecker()
```

2.3 Target-blank

This checker helps you to detect the *target blank* vulnerability in your code files. For more details about the vulnerability please see [this link](#).

```
atomshields.checkers.targetblank.  
TargetBlankChecker()
```

3.1 Echo

This reports prints on screen a summary of all issues found. The fields printed are Name, Severity and File affected.

```
atomshields.reports.echo.  
EchoReport(*args, ...)
```

This module shows all the information about the vulnerabilities per screen (echo).

3.2 Http Request

Sends hte full information of issues via HTTP. The endpoint must be setted into the config file in the directory *.atomshields* into your home path.

```
atomshields.reports.http.  
HttpRequest(*args, ...)
```

This module sends all information about vulnerabilities to an endpoint via an http request.

4.1 atomshields package

4.1.1 Subpackages

atomshields.checkers package

Submodules

atomshields.checkers.base module

class atomshields.checkers.base.GenericChecker

Bases: object

dao

Getter for 'dao' property

Returns Instance of DAO class

Return type atomshield.helpers.DAO

path

Getter for 'path' property

Returns Absolute path to scan

Return type str

project

Getter for 'project' property

Returns Project's name

Return type str

issues

Getter for 'issues' property

Returns List of instances of Issue class

Return type list<atomshields.helpers.Issue>

config

Getter for 'config' property

Returns Dictionary which contains the current values for this report config

Return type dict

test ()

Check if the checker is OK to run.

This method should to run every test (requirements) in order to ensure the execution will not have errors.

Returns True if the checker has all the requirements installed. False else

Return type bool

run ()

Abstract method. This method will be executed for subclass which not implemented his own method

saveIssue (issue)

Stores an issue in 'issues' property

Parameters **issue** (*atomshields.helpers.Issue*) – Issue instance

classmethod parseConfig (value)

Parse the config values

Parameters **value** (*dict*) – Dictionary which contains the checker config

Returns The checker config with parsed values

Return type dict

static isInstalled ()

Check if a software is installed into machine.

Parameters **value** (*str*) – Software's name

Returns True if the software is installed. False else

Return type bool

`atomshields.checkers.base.checker` (*func*)

Decorator for method run. This method will be execute before the execution from the method with this decorator.

atomshields.checkers.dsstore module

class `atomshields.checkers.dsstore.DSStoreChecker`

Bases: `atomshields.checkers.base.GenericChecker`

NAME = 'dsstore'

DESCRIPTION = 'Busca y elimina los ficheros .DS_Store'

CONFIG = {'enabled': True, 'exclude_paths': [], 'remove': True}

run (*args, **kwargs)

atomshields.checkers.retirejs module

```

class atomshields.checkers.retirejs.RetireJSChecker
    Bases: atomshields.checkers.base.GenericChecker

    NAME = 'retirejs'

    DESCRIPTION = 'Detecta ficheros JavaScript vulnerables'

    CONFIG = {'enabled': True, 'exclude_paths': ['test/']}

    run (*args, **kwargs)

    scan (path, options="", tempfile=False)

    download (url, path)

    getLastVersion (component)

    getJSLinks (filepath)

    static install ()
        Install all the dependences

    test ()
        Check the dependences.

        Returns True if all dependences are installed. False else.

        Return type bool

```

atomshields.checkers.targetblank module

```

class atomshields.checkers.targetblank.TargetBlankChecker
    Bases: atomshields.checkers.base.GenericChecker

    NAME = 'targetblank'

    DESCRIPTION = "Detecta vulnerabilidades 'Target Blank' en ficheros HTML"

    CONFIG = {'enabled': True, 'exclude_paths': ['/test/', '/docs/']}

    REGEX = u'(<a (?=.*href=(["\']) (https?:)?\\/\\"
    run (*args, **kwargs)

```

Module contents

atomshields.models package

Submodules

atomshields.models.issue module

```

class atomshields.models.issue.Issue (name=None, file=None, details=None, severity=None,
    potential=None, checker_name=None)

    Bases: object

    Type of object returned by all checkers.

```

Este es el modelo genérico que será generado por cada vulnerabilidad encontrada por los checkers y tratada por los modulos de reporte.

SEVERITY_INFO

str – Name of the incidences of informatic value

SEVERITY_LOW

str – Name of incidents with low criticality

SEVERITY_MEDIUM

str – Name of incidents with medium criticality

SEVERITY_HIGH

str – Name of requests with high criticality

SEVERITY_CRITICAL

str – Name of the requests with very high criticality

_name

str – Name of the request

_file

str – File affected by the incident

_details

str – Other details about the incident

_severity

str – Criticism of the incidence

_potential

bool

_checker_name

str – Name of the module that detected the request

SEVERITY_INFO = 'Info'

SEVERITY_LOW = 'Low'

SEVERITY_MEDIUM = 'Medium'

SEVERITY_HIGH = 'High'

SEVERITY_CRITICAL = 'Critical'

name

Getter for 'name' property

Returns Issue's name

Return type string

file

Getter for 'file' property

Returns Issue's file

Return type string

severity

Getter for 'severity' property

Returns Issue's severity

Return type string

potential

Getter for 'potential' property

Returns potential is required?

Return type bool

details

Getter for 'details' property

Returns Issue's details

Return type string

checker

Getter for 'checker' property

Returns Issue's checker

Return type string

Module contents

class atomshields.models.**Issue** (*name=None, file=None, details=None, severity=None, potential=None, checker_name=None*)

Bases: object

Type of object returned by all checkers.

Este es el modelo genérico que será generado por cada vulnerabilidad encontrada por los checkers y tratada por los modulos de reporte.

SEVERITY_INFO

str – Name of the incidences of informatic value

SEVERITY_LOW

str – Name of incidents with low criticality

SEVERITY_MEDIUM

str – Name of incidents with medium criticality

SEVERITY_HIGH

str – Name of requests with high criticality

SEVERITY_CRITICAL

str – Name of the requests with very high criticality

_name

str – Name of the request

_file

str – File affected by the incident

_details

str – Other details about the incident

_severity

str – Criticism of the incidence

_potential

bool

_checker_name

str – Name of the module that detected the request

```
SEVERITY_INFO = 'Info'  
SEVERITY_LOW = 'Low'  
SEVERITY_MEDIUM = 'Medium'  
SEVERITY_HIGH = 'High'  
SEVERITY_CRITICAL = 'Critical'
```

name

Getter for 'name' property

Returns Issue's name

Return type string

file

Getter for 'file' property

Returns Issue's file

Return type string

severity

Getter for 'severity' property

Returns Issue's severity

Return type string

potential

Getter for 'potential' property

Returns potential is required?

Return type bool

details

Getter for 'details' property

Returns Issue's details

Return type string

checker

Getter for 'checker' property

Returns Issue's checker

Return type string

atomshields.reports package

Submodules

atomshields.reports.base module

class atomshields.reports.base.**GenericReport** (*issues=None*)

Bases: object

Class inherited by all reporting modules.

issues

Getter for 'issues' property

Returns List of *Issue* instances

Return type list

config

Getter for 'config' property

Returns Dictionary which contains the current values for this report config

Return type dict

project

Getter for 'project' property

Returns Project's name

Return type str

atomshields.reports.base.**report** (*func*)

Decorator for method run. This method will be execute before the execution from the method with this decorator.

atomshields.reports.echo module

class atomshields.reports.echo.**EchoReport** (*args, **kwargs)

Bases: *atomshields.reports.base.GenericReport*

This module shows all the information about the vulnerabilities per screen (echo).

NAME

str – Name of the module.

DESCRIPTION

str – Description of the functionality of the module.

CONFIG

dict – Default values of the module configuration..

NAME = 'echo'

DESCRIPTION = 'Muestra las vulnerabilidades por pantalla'

CONFIG = {'enabled': True}

run (*args, **kwargs)

atomshields.reports.http module

class atomshields.reports.http.**HttpReport** (*args, **kwargs)

Bases: *atomshields.reports.base.GenericReport*

This module sends all information about vulnerabilities to an endpoint via an http request.

Attributes: **NAME** (str): Name of the module. **DESCRIPTION** (str): Description of the functionality of the module. **CONFIG** (dict): Default values of the module configuration.

NAME = 'http'

DESCRIPTION = 'Envia los datos de las vulnerabilidades a un endpoint HTTP'

CONFIG = {'enabled': False, 'method': 'post', 'proxy': 'http://127.0.0.1:8080', 'ur

run (*args, **kwargs)

Module contents

4.1.2 Submodules

4.1.3 atomshields.helpers module

class atomshields.helpers.**CommandHelper** (*command=None*)

Bases: object

Class used to execute commands in shell, and get the output and the errors.

OS_UBUNTU = 'Ubuntu'

OS_DEBIAN = 'Debian'

OS_CENTOS = 'CentOS'

OS_REDHAT = 'Redhat'

OS_KALI = 'Kali'

OS_LINUX = 'Linux'

OS_MAC = 'Darwin'

OS_WINDOWS = 'Windows'

command

Getter for 'command' property

Returns Command to execute

Return type str

output

Getter for 'output' property

Returns Stdout content

Return type str

errors

Getter for 'errors' property

Returns Stderr content

Return type str

getOSName()

Get the OS name. If OS is linux, returns the Linux distribution name

Returns OS name

Return type str

install (*software, uninstall=False*)

execute (*shell=True*)

Executes the command setted into class

Parameters **shell** (*boolean*) – Set True if command is a shell command. Default: True

4.1.4 atomshields.scanner module

class atomshields.scanner.**AtomShieldsScanner** (*path, verbose=False*)

Bases: object

Class in charge of orchestrating the execution of the checkers and the results.

HOME = '/usr/local/share/atomshields'

HOME_2 = '/home/docs/.atomshields'

CHECKERS_DIR = '/home/docs/.atomshields/checkers'

REPORTS_DIR = '/home/docs/.atomshields/reports'

CONFIG_PATH = '/home/docs/.atomshields/config'

debug (*message, color=None, attrs=None*)

path

Getter for 'path' property

Returns Absolute path to target directory

Return type string

project

Getter for 'project' property

Returns Projects's name

Return type string

configFile

Getter for 'configFile' property

Returns Path to config file

Return type str

config

Getter for 'config' property

Returns Path to config file

Return type str

issues

Getter for 'issues' property

Returns List of Issue instances

Return type list

static setup ()

Creates required directories and copy checkers and reports.

static writeConfig (*configFile=None*)

static installPlugin (*destination*)

static installChecker (*exclude=['__init__.py', 'base.py']*)

static installReport (*exclude=['__init__.py', 'base.py']*)

static uninstallPlugin (*name=None, classArgs=None*)

static uninstallChecker ()

```
static uninstallReport ()
static generateConfig ()
showScanProperties ()
checkProperties ()
getConfig (section=None)
    Returns a dictionary which contains the current config. If a section is setted, only will returns the section
    config
    Parameters section (str) – (Optional) Section name.
    Returns Representation of current config
    Return type dict

loadConfig ()
executeCheckers ()
executeReports ()
saveIssue (issue)
isStable ()
showSummary ()
run ()
    Run a scan in the path setted.
```

4.1.5 Module contents

```
class atomshields.AtomShieldsScanner (path, verbose=False)
    Bases: object

    Class in charge of orchestrating the execution of the cherckers and the results.

    HOME = '/usr/local/share/atomshields'
    HOME_2 = '/home/docs/.atomshields'
    CHECKERS_DIR = '/home/docs/.atomshields/checkers'
    REPORTS_DIR = '/home/docs/.atomshields/reports'
    CONFIG_PATH = '/home/docs/.atomshields/config'
    debug (message, color=None, attrs=None)

    path
        Getter for 'path' property
        Returns Absolute path to target directory
        Return type string

    project
        Getter for 'project' property
        Returns Projects's name
        Return type string
```

configFile

Getter for 'configFile' property

Returns Path to config file

Return type str

config

Getter for 'config' property

Returns Path to config file

Return type str

issues

Getter for 'issues' property

Returns List of Issue instances

Return type list

static setup()

Creates required directories and copy checkers and reports.

static writeConfig (*configFile=None*)**static installPlugin** (*destination*)**static installChecker** (*exclude=['__init__.py', 'base.py']*)**static installReport** (*exclude=['__init__.py', 'base.py']*)**static uninstallPlugin** (*name=None, classArgs=None*)**static uninstallChecker** ()**static uninstallReport** ()**static generateConfig** ()**showScanProperties** ()**checkProperties** ()**getConfig** (*section=None*)

Returns a dictionary which contains the current config. If a section is setted, only will returns the section config

Parameters **section** (*str*) – (Optional) Section name.

Returns Representation of current config

Return type dict

loadConfig ()**executeCheckers** ()**executeReports** ()**saveIssue** (*issue*)**isStable** ()**showSummary** ()**run** ()

Run a scan in the path setted.

class atomshields.CommandHelper (*command=None*)

Bases: object

Class used to execute commands in shell, and get the output and the errors.

OS_UBUNTU = 'Ubuntu'

OS_DEBIAN = 'Debian'

OS_CENTOS = 'CentOS'

OS_REDHAT = 'Redhat'

OS_KALI = 'Kali'

OS_LINUX = 'Linux'

OS_MAC = 'Darwin'

OS_WINDOWS = 'Windows'

command

Getter for 'command' property

Returns Command to execute

Return type str

output

Getter for 'output' property

Returns Stdout content

Return type str

errors

Getter for 'errors' property

Returns Stderr content

Return type str

getOSName ()

Get the OS name. If OS is linux, returns the Linux distribution name

Returns OS name

Return type str

install (*software, uninstall=False*)

execute (*shell=True*)

Executes the command setted into class

Parameters **shell** (*boolean*) – Set True if command is a shell command. Default: True

What is AtomShields?

Security testing framework for repositories and source code.

This system has different modules that detect different vulnerabilities or files that may expose a risk, and the results obtained can be obtained or sent thanks to the reporting modules.

For developers: There is also the possibility to develop your own vulnerability detection (called checkers) or reporting modules. This tool offers certain facilities for those who want to implement their own modules, publish them and integrate them into the official ElevenPaths tool.

CHAPTER 6

Installation

```
pip install atomshields
```

CHAPTER 7

Basic usage

```
import atomshields

atoms = atomshields.AtomShieldsScanner('./MyRepo/')
atoms.project = "MyRepoName"

issues = atoms.executeCheckers()
```

CHAPTER 8

Indices and tables

- `genindex`
- `modindex`
- `search`

a

- `atomshields`, 16
- `atomshields.checkers`, 9
 - `atomshields.checkers.base`, 7
 - `atomshields.checkers.dsstore`, 8
 - `atomshields.checkers.retirejs`, 9
 - `atomshields.checkers.targetblank`, 9
- `atomshields.helpers`, 14
- `atomshields.models`, 11
 - `atomshields.models.issue`, 9
- `atomshields.reports`, 14
 - `atomshields.reports.base`, 12
 - `atomshields.reports.echo`, 13
 - `atomshields.reports.http`, 13
- `atomshields.scanner`, 15

Symbols

_checker_name (atomshields.models.Issue attribute), 11
 _checker_name (atomshields.models.issue.Issue attribute), 10
 _details (atomshields.models.Issue attribute), 11
 _details (atomshields.models.issue.Issue attribute), 10
 _file (atomshields.models.Issue attribute), 11
 _file (atomshields.models.issue.Issue attribute), 10
 _name (atomshields.models.Issue attribute), 11
 _name (atomshields.models.issue.Issue attribute), 10
 _potential (atomshields.models.Issue attribute), 11
 _potential (atomshields.models.issue.Issue attribute), 10
 _severity (atomshields.models.Issue attribute), 11
 _severity (atomshields.models.issue.Issue attribute), 10

A

atomshields (module), 16
 atomshields.checkers (module), 9
 atomshields.checkers.base (module), 7
 atomshields.checkers.dsstore (module), 8
 atomshields.checkers.retirejs (module), 9
 atomshields.checkers.targetblank (module), 9
 atomshields.helpers (module), 14
 atomshields.models (module), 11
 atomshields.models.issue (module), 9
 atomshields.reports (module), 14
 atomshields.reports.base (module), 12
 atomshields.reports.echo (module), 13
 atomshields.reports.http (module), 13
 atomshields.scanner (module), 15
 AtomShieldsScanner (class in atomshields), 16
 AtomShieldsScanner (class in atomshields.scanner), 15

C

checker (atomshields.models.Issue attribute), 12
 checker (atomshields.models.issue.Issue attribute), 11
 checker() (in module atomshields.checkers.base), 8
 CHECKERS_DIR (atomshields.AtomShieldsScanner attribute), 16

CHECKERS_DIR (atomshields.scanner.AtomShieldsScanner attribute), 15
 checkProperties() (atomshields.AtomShieldsScanner method), 17
 checkProperties() (atomshields.scanner.AtomShieldsScanner method), 16
 command (atomshields.CommandHelper attribute), 18
 command (atomshields.helpers.CommandHelper attribute), 14
 CommandHelper (class in atomshields), 17
 CommandHelper (class in atomshields.helpers), 14
 config (atomshields.AtomShieldsScanner attribute), 17
 config (atomshields.checkers.base.GenericChecker attribute), 8
 CONFIG (atomshields.checkers.dsstore.DSStoreChecker attribute), 8
 CONFIG (atomshields.checkers.retirejs.RetireJSChecker attribute), 9
 CONFIG (atomshields.checkers.targetblank.TargetBlankChecker attribute), 9
 config (atomshields.reports.base.GenericReport attribute), 13
 CONFIG (atomshields.reports.echo.EchoReport attribute), 13
 CONFIG (atomshields.reports.http.HttpReport attribute), 13
 config (atomshields.scanner.AtomShieldsScanner attribute), 15
 CONFIG_PATH (atomshields.AtomShieldsScanner attribute), 16
 CONFIG_PATH (atomshields.scanner.AtomShieldsScanner attribute), 15
 configFile (atomshields.AtomShieldsScanner attribute), 16
 configFile (atomshields.scanner.AtomShieldsScanner attribute), 15

D

dao (atomshields.checkers.base.GenericChecker attribute), 7

debug() (atomshields.AtomShieldsScanner method), 16

debug() (atomshields.scanner.AtomShieldsScanner method), 15

DESCRIPTION (atomshields.checkers.dsstore.DSStoreChecker attribute), 8

DESCRIPTION (atomshields.checkers.retirejs.RetireJSChecker attribute), 9

DESCRIPTION (atomshields.checkers.targetblank.TargetBlankChecker attribute), 9

DESCRIPTION (atomshields.reports.echo.EchoReport attribute), 13

DESCRIPTION (atomshields.reports.http.HttpReport attribute), 13

details (atomshields.models.Issue attribute), 12

details (atomshields.models.issue.Issue attribute), 11

download() (atomshields.checkers.retirejs.RetireJSChecker method), 9

DSStoreChecker (class in atomshields.checkers.dsstore), 8

E

EchoReport (class in atomshields.reports.echo), 13

errors (atomshields.CommandHelper attribute), 18

errors (atomshields.helpers.CommandHelper attribute), 14

execute() (atomshields.CommandHelper method), 18

execute() (atomshields.helpers.CommandHelper method), 14

executeCheckers() (atomshields.AtomShieldsScanner method), 17

executeCheckers() (atomshields.scanner.AtomShieldsScanner method), 16

executeReports() (atomshields.AtomShieldsScanner method), 17

executeReports() (atomshields.scanner.AtomShieldsScanner method), 16

F

file (atomshields.models.Issue attribute), 12

file (atomshields.models.issue.Issue attribute), 10

G

generateConfig() (atomshields.AtomShieldsScanner static method), 17

generateConfig() (atomshields.scanner.AtomShieldsScanner static method), 16

GenericChecker (class in atomshields.checkers.base), 7

GenericReport (class in atomshields.reports.base), 12

getConfig() (atomshields.AtomShieldsScanner method), 17

getConfig() (atomshields.scanner.AtomShieldsScanner method), 16

getJSLinks() (atomshields.checkers.retirejs.RetireJSChecker method), 9

getLastVersion() (atomshields.checkers.retirejs.RetireJSChecker method), 9

getOSName() (atomshields.CommandHelper method), 18

getOSName() (atomshields.helpers.CommandHelper method), 14

H

HOME (atomshields.AtomShieldsScanner attribute), 16

HOME (atomshields.scanner.AtomShieldsScanner attribute), 15

HOME_2 (atomshields.AtomShieldsScanner attribute), 16

HOME_2 (atomshields.scanner.AtomShieldsScanner attribute), 15

HttpReport (class in atomshields.reports.http), 13

I

install() (atomshields.checkers.retirejs.RetireJSChecker static method), 9

install() (atomshields.CommandHelper method), 18

install() (atomshields.helpers.CommandHelper method), 14

installChecker() (atomshields.AtomShieldsScanner static method), 17

installChecker() (atomshields.scanner.AtomShieldsScanner static method), 15

installPlugin() (atomshields.AtomShieldsScanner static method), 17

installPlugin() (atomshields.scanner.AtomShieldsScanner static method), 15

installReport() (atomshields.AtomShieldsScanner static method), 17

installReport() (atomshields.scanner.AtomShieldsScanner static method), 15

isInstalled() (atomshields.checkers.base.GenericChecker static method), 8

isStable() (atomshields.AtomShieldsScanner method), 17

isStable() (atomshields.scanner.AtomShieldsScanner method), 16

Issue (class in atomshields.models), 11

Issue (class in atomshields.models.issue), 9

issues (atomshields.AtomShieldsScanner attribute), 17
 issues (atomshields.checkers.base.GenericChecker attribute), 7
 issues (atomshields.reports.base.GenericReport attribute), 12
 issues (atomshields.scanner.AtomShieldsScanner attribute), 15

L

loadConfig() (atomshields.AtomShieldsScanner method), 17
 loadConfig() (atomshields.scanner.AtomShieldsScanner method), 16

N

NAME (atomshields.checkers.dsstore.DSStoreChecker attribute), 8
 NAME (atomshields.checkers.retirejs.RetireJSChecker attribute), 9
 NAME (atomshields.checkers.targetblank.TargetBlankChecker attribute), 9
 name (atomshields.models.Issue attribute), 12
 name (atomshields.models.issue.Issue attribute), 10
 NAME (atomshields.reports.echo.EchoReport attribute), 13
 NAME (atomshields.reports.http.HttpReport attribute), 13

O

OS_CENTOS (atomshields.CommandHelper attribute), 18
 OS_CENTOS (atomshields.helpers.CommandHelper attribute), 14
 OS_DEBIAN (atomshields.CommandHelper attribute), 18
 OS_DEBIAN (atomshields.helpers.CommandHelper attribute), 14
 OS_KALI (atomshields.CommandHelper attribute), 18
 OS_KALI (atomshields.helpers.CommandHelper attribute), 14
 OS_LINUX (atomshields.CommandHelper attribute), 18
 OS_LINUX (atomshields.helpers.CommandHelper attribute), 14
 OS_MAC (atomshields.CommandHelper attribute), 18
 OS_MAC (atomshields.helpers.CommandHelper attribute), 14
 OS_REDHAT (atomshields.CommandHelper attribute), 18
 OS_REDHAT (atomshields.helpers.CommandHelper attribute), 14
 OS_UBUNTU (atomshields.CommandHelper attribute), 18
 OS_UBUNTU (atomshields.helpers.CommandHelper attribute), 14

OS_WINDOWS (atomshields.CommandHelper attribute), 18
 OS_WINDOWS (atomshields.helpers.CommandHelper attribute), 14
 output (atomshields.CommandHelper attribute), 18
 output (atomshields.helpers.CommandHelper attribute), 14

P

parseConfig() (atomshields.checkers.base.GenericChecker class method), 8
 path (atomshields.AtomShieldsScanner attribute), 16
 path (atomshields.checkers.base.GenericChecker attribute), 7
 path (atomshields.scanner.AtomShieldsScanner attribute), 15
 potential (atomshields.models.Issue attribute), 12
 potential (atomshields.models.issue.Issue attribute), 10
 project (atomshields.AtomShieldsScanner attribute), 16
 project (atomshields.checkers.base.GenericChecker attribute), 7
 project (atomshields.reports.base.GenericReport attribute), 13
 project (atomshields.scanner.AtomShieldsScanner attribute), 15

R

REGEX (atomshields.checkers.targetblank.TargetBlankChecker attribute), 9
 report() (in module atomshields.reports.base), 13
 REPORTS_DIR (atomshields.AtomShieldsScanner attribute), 16
 REPORTS_DIR (atomshields.scanner.AtomShieldsScanner attribute), 15
 RetireJSChecker (class in atomshields.checkers.retirejs), 9
 run() (atomshields.AtomShieldsScanner method), 17
 run() (atomshields.checkers.base.GenericChecker method), 8
 run() (atomshields.checkers.dsstore.DSStoreChecker method), 8
 run() (atomshields.checkers.retirejs.RetireJSChecker method), 9
 run() (atomshields.checkers.targetblank.TargetBlankChecker method), 9
 run() (atomshields.reports.echo.EchoReport method), 13
 run() (atomshields.reports.http.HttpReport method), 13
 run() (atomshields.scanner.AtomShieldsScanner method), 16

S

saveIssue() (atomshields.AtomShieldsScanner method), 17

saveIssue() (atomshields.checkers.base.GenericChecker method), 8

saveIssue() (atomshields.scanner.AtomShieldsScanner method), 16

scan() (atomshields.checkers.retirejs.RetireJSChecker method), 9

setup() (atomshields.AtomShieldsScanner static method), 17

setup() (atomshields.scanner.AtomShieldsScanner static method), 15

severity (atomshields.models.Issue attribute), 12

severity (atomshields.models.issue.Issue attribute), 10

SEVERITY_CRITICAL (atomshields.models.Issue attribute), 11, 12

SEVERITY_CRITICAL (atomshields.models.issue.Issue attribute), 10

SEVERITY_HIGH (atomshields.models.Issue attribute), 11, 12

SEVERITY_HIGH (atomshields.models.issue.Issue attribute), 10

SEVERITY_INFO (atomshields.models.Issue attribute), 11, 12

SEVERITY_INFO (atomshields.models.issue.Issue attribute), 10

SEVERITY_LOW (atomshields.models.Issue attribute), 11, 12

SEVERITY_LOW (atomshields.models.issue.Issue attribute), 10

SEVERITY_MEDIUM (atomshields.models.Issue attribute), 11, 12

SEVERITY_MEDIUM (atomshields.models.issue.Issue attribute), 10

showScanProperties() (atomshields.AtomShieldsScanner method), 17

showScanProperties() (atomshields.scanner.AtomShieldsScanner method), 16

showSummary() (atomshields.AtomShieldsScanner method), 17

showSummary() (atomshields.scanner.AtomShieldsScanner method), 16

T

TargetBlankChecker (class in atomshields.checkers.targetblank), 9

test() (atomshields.checkers.base.GenericChecker method), 8

test() (atomshields.checkers.retirejs.RetireJSChecker method), 9

U

uninstallChecker() (atomshields.AtomShieldsScanner static method), 17

uninstallChecker() (atomshields.scanner.AtomShieldsScanner static method), 15

uninstallPlugin() (atomshields.AtomShieldsScanner static method), 17

uninstallPlugin() (atomshields.scanner.AtomShieldsScanner static method), 15

uninstallReport() (atomshields.AtomShieldsScanner static method), 17

uninstallReport() (atomshields.scanner.AtomShieldsScanner static method), 15

W

writeConfig() (atomshields.AtomShieldsScanner static method), 17

writeConfig() (atomshields.scanner.AtomShieldsScanner static method), 15