ExploitAnalyzer

-__metaclass__ = ABCMeta
-parser
-comment_dir
-code_dir
-result
-result_counter

+__init__()
+read_file(abs_filename)
-general_analyze(exploit, filename)
+generate_stats()
+*analyze(exploit, filename)*

CommentAnalyzer

-comment_tags
-comment_group_tags

+__init__()
-_merge_similar_tags(filename)
-_result_check(result)
-_hash_handling(result)
+analyze(filename)

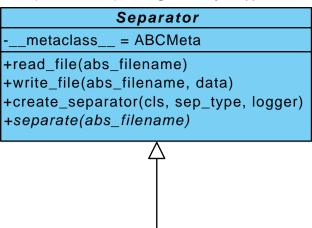**CodeAnalyzer**

-code_tags

+__init__()
-_assign_node(node, filename)
-_assign_node_helper(node, filename, target_id)
-_augassign_node(node, filename)
-_expr_node(node, filename)
-_dict_node(node, filename)
-_result_add(filename, tag, value)
+analyze(filename)

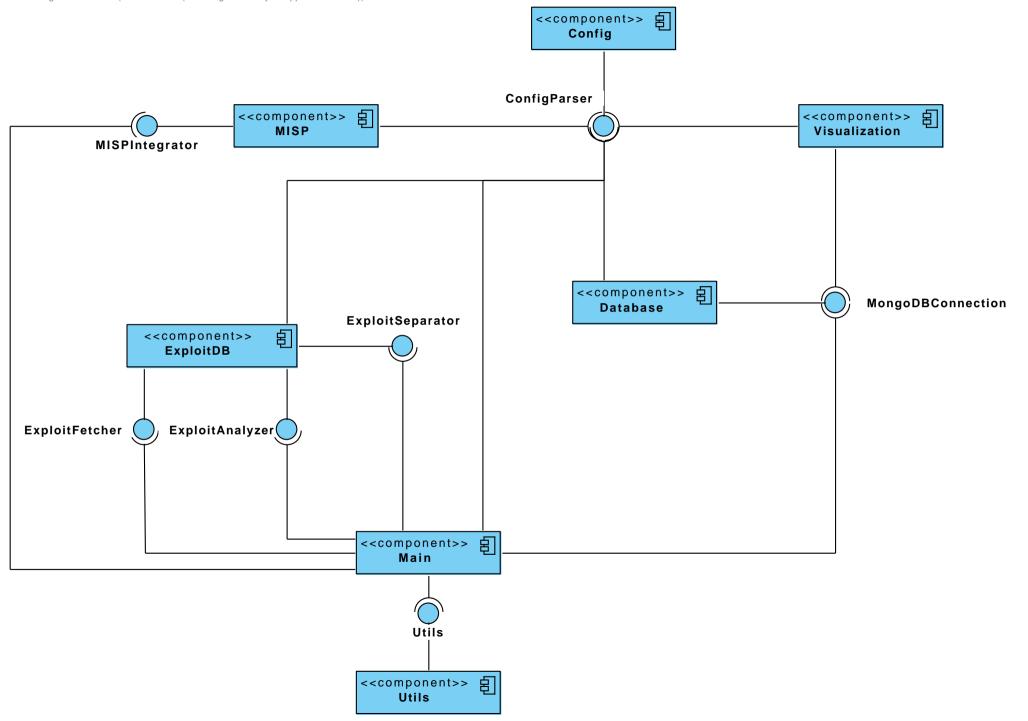| MongoDBConnection |
| --- |
| -parser |
| -client |
| -collection |
| -logger |
| +__init__(logger, db_name = "exploitdb", collection = "exploits") |
| +get_all_exploits() |
| +insert_exploits(data) |
| +get_exploits_by_file_extension(extension) |
| +get_exploits_by_file_extension_and_id(extension, ids) |
| +insert_or_update_exploits_by_id(ins_up_exploits) |
| +update_exploits_by_file(update_dict) |
| +close() |

Separator

-__metaclass__ = ABCMeta

+read_file(abs_filename)
+write_file(abs_filename, data)
+create_separator(cls, sep_type, logger)
+*separate(abs_filename)*

PythonSeparator

-logger
-exploit_dif
-comment_dir
-code_dir
-qurantine_dir
-total_error_count

+__init__(logger, exploit_dir, comment_dir, code_dir, quarantine_dir)
-_version_check(exploit)
-_handle_command(child)
-_walk(child, line_num)
-_check_nodes(child)
+separate(filename)

Comment Analyzer

GitHub

ExploitDB Fetcher

Code Analyzer

Database

D
```json
{"id": "<int>", "file": "<string>",
"description": "<string>", "date":
"<string>", "author": "<string>",
"type": "<string>", "platform":
"<string>", "port": "<int>",
"cve": ["<string>"], "ip":
["<string>"], "additional_urls":
["<string>"], "additional_emails":
["<string>"], "software link":
["<string>"], "vendor
homepage": ["<string>"],
"vendor": ["<string>"], "tested
version": ["<string>"], "version":
["<string>"], "tested on":
["<string>"], "steps to
reproduce": ["<string>"], "google
dork": ["<string>"], "usage":
["<string>"], "example":
["<string>"], "reference":
["<string>"], "md5":
["<string>"], "contact":
["<string>"], "github":
["<string>"], "website":
["<string>"], "email":
["<string>"], "irc": ["<string>"],
"group": ["<string>"]}
```

D
```json
{"id": "<int>", "file": "<string>",
"description": "<string>", "date":
"<string>", "author": "<string>",
"type": "<string>", "platform":
"<string>", "port": "<int>",
"cve": ["<string>"], "ip":
["<string>"], "additional_urls":
["<string>"], "additional_emails":
["<string>"], "software link":
["<string>"], "vendor
homepage": ["<string>"],
"vendor": ["<string>"], "tested
version": ["<string>"], "version":
["<string>"], "tested on":
["<string>"], "steps to
reproduce": ["<string>"], "google
dork": ["<string>"], "usage":
["<string>"], "example":
["<string>"], "reference":
["<string>"], "md5":
["<string>"], "contact":
["<string>"], "github":
["<string>"], "website":
["<string>"], "email":
["<string>"], "irc": ["<string>"],
"group": ["<string>"],
"analyzed_port": ["<string>"],
"user-agent": ["<string>"],
"payload": ["<string>"],
"content-type": ["<string>"],
"help_parser_args": ["<string>"],
"py_imports": ["<string>"]}
```

This component
also adds keys
with "port" in the
variable name, e.g.
"proxyport",
"target_port". It is
not possible to
document it,
because it varies
from exploit to
exploit. The value
is a list of strings.

D
```json
{"id": "<int>", "file":
"<string>",
"description":
"<string>", "date":
"<string>", "author":
"<string>", "type":
"<string>", "platform":
"<string>", "port":
"<int>"}
```