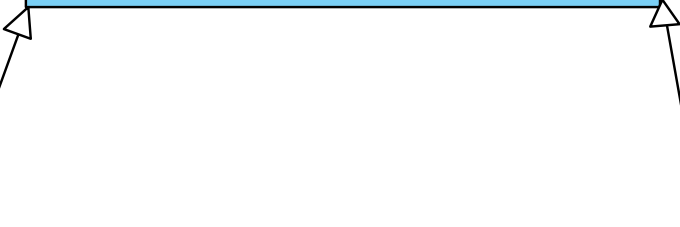**ExploitAnalyzer**

-__metaclass__ = ABCMeta
-parser : ConfigParser
-comment_dir : str
-code_dir : str
-result : dict
-result_counter : dict

+__init__() : ExploitAnalyzer
+read_file(abs_filename) : str
-general_analyze(exploit, filename) : None
+generate_stats() : None
+*analyze(exploit, filename, exploit_id) : None*

**CommentAnalyzer**

-comment_tags : list
-comment_group_tags : list

+__init__() : ExploitAnalyzer
-_merge_similar_tags(filename) : None
-_result_check(result) : str
-_hash_handling(result) : str
+analyze(filename, exploit_id) : None

**CodeAnalyzer**

-code_tags : list

+__init__() : ExploitAnalyzer
-_assign_node(node, filename) : None
-_assign_node_helper(node, filename, target_id) : None
-_augassign_node(node, filename) : None
-_expr_node(node, filename) : None
-_dict_node(node, filename) : None
-_result_add(filename, tag, value) : None
+analyze(filename, exploit_id) : None

| MongoDBConnection |
|---|
| -parser : ConfigParser<br>-client : MongoClient<br>-collection : Collection<br>-logger : Logger |
| +__init__(logger, db_name = "exploitdb", collection = "exploits") : MongoDBConnection<br>+get_all_exploits() : list<br>+insert_exploits(data) : None<br>+get_exploits_by_file_extension(extension) : list<br>+get_exploits_by_file_extension_and_id(extension, ids) : list<br>+insert_or_update_exploits_by_id(ins_up_exploits) : None<br>+close() : None |

| MISPIntegrator |
| --- |
| -parser : ConfigParser<br>-misp_parser : ConfigParser<br>-misp : PyMISP<br>-distribution : int<br>-local_git : str<br>-remote_git : str<br>-attributes : dict<br>-logger : Logger<br>-db : MongoDBConnection |
| +__init__(logger, db) : MISPIntegrator<br>-_set_tag_id(tag_name, tag_colour) : list<br>-_set_taxonomy_tag_ids(taxonomies, all_tags) : None<br>-_create_event(date, description) : str<br>-_add_attributes_to_event(uuid, misp_type, values, category, new_event = True) : None<br>-_add_exploit_poc_template(uuid, exploit) : None<br>+integrate(exploits) : None |

```
┌──────────────────────────────────────────────────────┐
│                      Separator                          │
├──────────────────────────────────────────────────────┤
│ -__metaclass__ = ABCMeta                                │
├──────────────────────────────────────────────────────┤
│ +read_file(abs_filename) : str                          │
│ +write_file(abs_filename, data) : None                  │
│ +create_separator(cls, sep_type, logger) : Separator    │
│ +separate(abs_filename) : None                          │
└──────────────────────────────────────────────────────┘
                           △
                           │
┌──────────────────────────────────────────────────────────────┐
│                      PythonSeparator                            │
├──────────────────────────────────────────────────────────────┤
│ -logger : Logger                                                │
│ -exploit_dir : str                                              │
│ -comment_dir : str                                              │
│ -code_dir : str                                                 │
│ -qurantine_dir : str                                            │
│ -total_error_count : int                                        │
├──────────────────────────────────────────────────────────────┤
│ +__init__(logger, exploit_dir, comment_dir, code_dir, quarantine_dir) : Separator │
│ -_version_check(exploit) : Module                               │
│ -_handle_command(child) : tuple                                 │
│ -_walk(child, line_num) : tuple                                 │
│ -_check_nodes(child) : tuple                                    │
│ +separate(filename) : None                                      │
└──────────────────────────────────────────────────────────────┘
```

**Comment Analyzer**

**GitHub**

**ExploitDB Fetcher**

D

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

D

{"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>"]}

**Database**

D

{"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-arg": ["<string>"],
"py-imports": ["<string>"]}

**Code Analyzer**

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.