Alternative scalable HIDS with investigation capability

FIDS - Forensic-based Intrusion Detection System

Julian Stampfli

June 21, 2019

Table of Contents

Table of Contents

Intrusions

- What?
  - Malware
  - Hacker
  - Insider Threats

Intrusions

- What?
  - Malware
  - Hacker
  - Insider Threats
- Protection
  - Firewals
  - Least privilege
  - ...
- Secure

Intrusions

- What?
  - Malware
  - Hacker
  - Insider Threats
- Protection
  - Firewals
  - Least privilege
  - ...
- Secure?
  - Open Ports
  - Weak Passwords
  - Insecure Applications
  - ...

Intrusion Detection

Network-Based

- Central scanning
- Uses
  - Traffic Load
  - Connections
  - Inspection
- Mainly pattern driven

Host-Based

- Distributed Scanning
- Uses
  - Processes
  - Files
  - Network Configuration
- Change driven

HIDS - FIM

Finding Changes

- Hashing to the rescue!

HIDS - FIM

Finding Changes

- Hashing to the rescue! or not?

Hashing

- Highly reliable
- Used for cryptographic use cases
- Fast?

Hashing

- Highly reliable
- Used for cryptographic use cases
- Fast? - Yes but not really.
- FIM using Hashing?
  - Tripwire - 1992 - Gone comercial
  - Aide / Samhain - Current Opensource alternatives

HIDS - FIM

Finding Changes

- Hashing
- Filesystem attributes to the rescue

TSK

The Sleuth Kit.

- Opensource
- Disk analyzis utility
- Used in forensics

Table of Contents

FIDS - Architecture

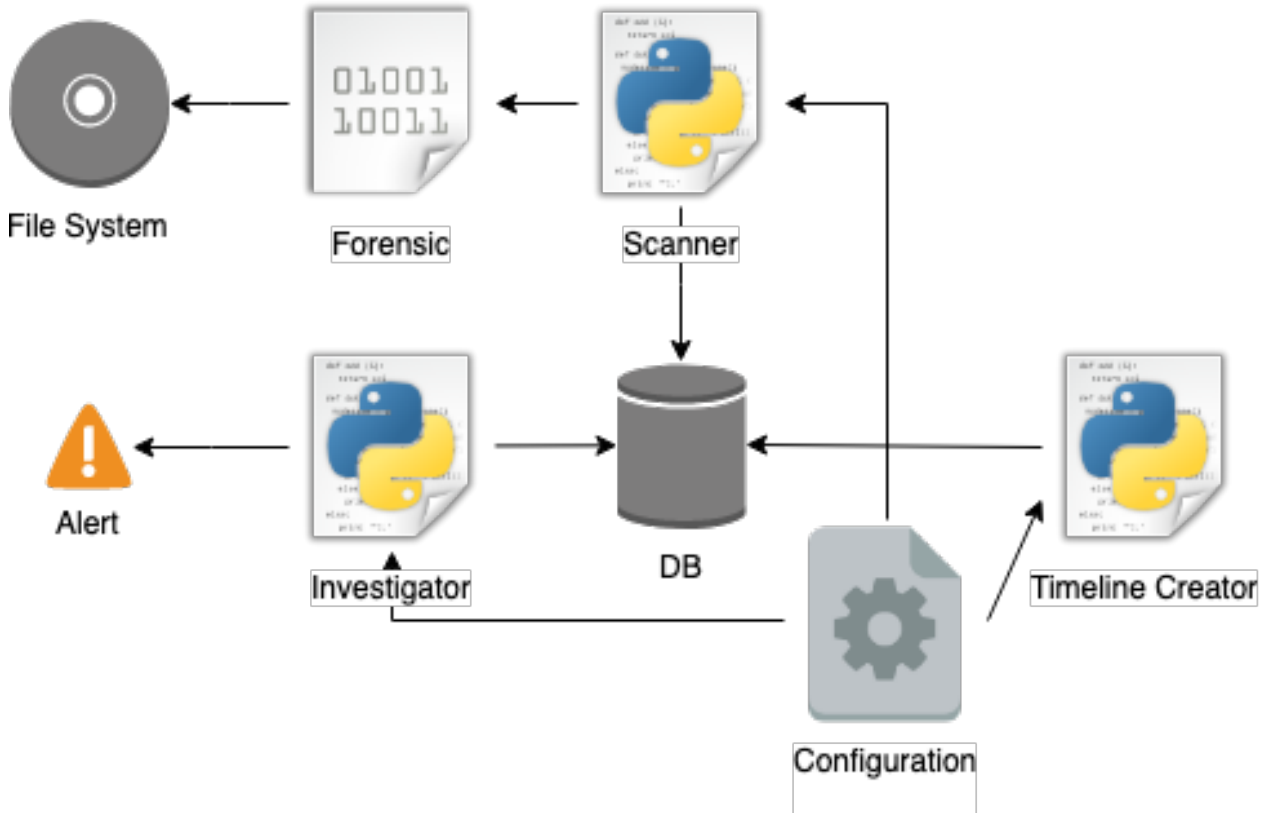

Figure: System Architecture

Search Functionality

```
1  self.stack = []
2  for path in self.paths:
3    open_dir_rec(get_entry(path))
4
5  def open_dir_rec(curDir):
6    ...
7    for entry in curDir:
8      if entry.isDir and inode not in self.stack:
9        self.open_directory_rec(entry)
10     else:
11       ...
```

Run comparison query

```
1   SELECT first.*, second.*
2   FROM FIDS_FILE first
3     LEFT JOIN FIDS_FILE second
4       on
5         (first.meta_addr=second.meta_addr
6           or first.path=second.path
7             and first.name_name=second.name_name)
8   WHERE first.run_id = ?
9     and second.run_id = ?
10    or first.run_id is null;
```

Investigator Configuration

```
1   investigator:
2     same_config: True
3     rules:
4       - name: all_files
5         rules:
6         equal: [meta_size]
7         greater:
8     investigation:
9       - paths:
10          - '/'
11        rules:
12          - all_files
```

Timeline Architecture

MD5|name|inode|mode|UID|GID|size|atime|mtime|ctime|crtime

```
1  print(
2          '0|'
3          f'{f.path}{f.name_name}|'
4          f'{f.meta_addr}|'
5          f'{mode}|'
6          f'{f.meta_uid}|'
7          f'{f.meta_gid}|'
8          f'{f.meta_size}|'
9          f'{f.meta_access_time}|'
10         f'{f.meta_modification_time}|'
11         f'{f.meta_changed_time}|'
12         f'{f.meta_creation_time}')
```

Table of Contents

Challenges

- Pytsk and TSK API Documentation
- Scan
- Thesis Document

Conclusion

Summary

- Fast Scans
- Intrusion Detection Possible
- Support for Forensic Investigation

Conclusion

Summary

- Fast Scans
- Intrusion Detection Possible
- Support for Forensic Investigation

Conclusion

- Risk-Based Approach
- Speed vs Reliability

Further Research

- Extension beyond Files
- Extensive Testing in a Live Environment
- Extend support for FS specific attributes
- Use more sophisticated storage solution

Thank you for listening / Demo

Code: `https://github.com/Tartori/fids`