Naftaly Avadiaev, Aviad Yifrah

naGuard End-Point Ransomware Protection

Project definition

Project Goals

i. Write an application, running standalone on the potential victim's machine, that will hook to core OS functions in order to detect ransomware activity
ii. You will need to globally hook specific system process to identify anomalies related to specific types of ransomware.
iii. Once hooks have been set, a detection mechanism should be implemented. See: https://www.cise.ufl.edu/~traynor/papers/scaife-icdcs16.pdf as an example of detecting ransomware activity
iv. In case the application has detected a ransomware infection on the victim's machine, it will respond by using the built-in Windows mechanisms, e.g. Shadow-Copy to recover one or more corrupted files.

Assumptions

1

Windows 10

2

Malware run in user mode (admin available)

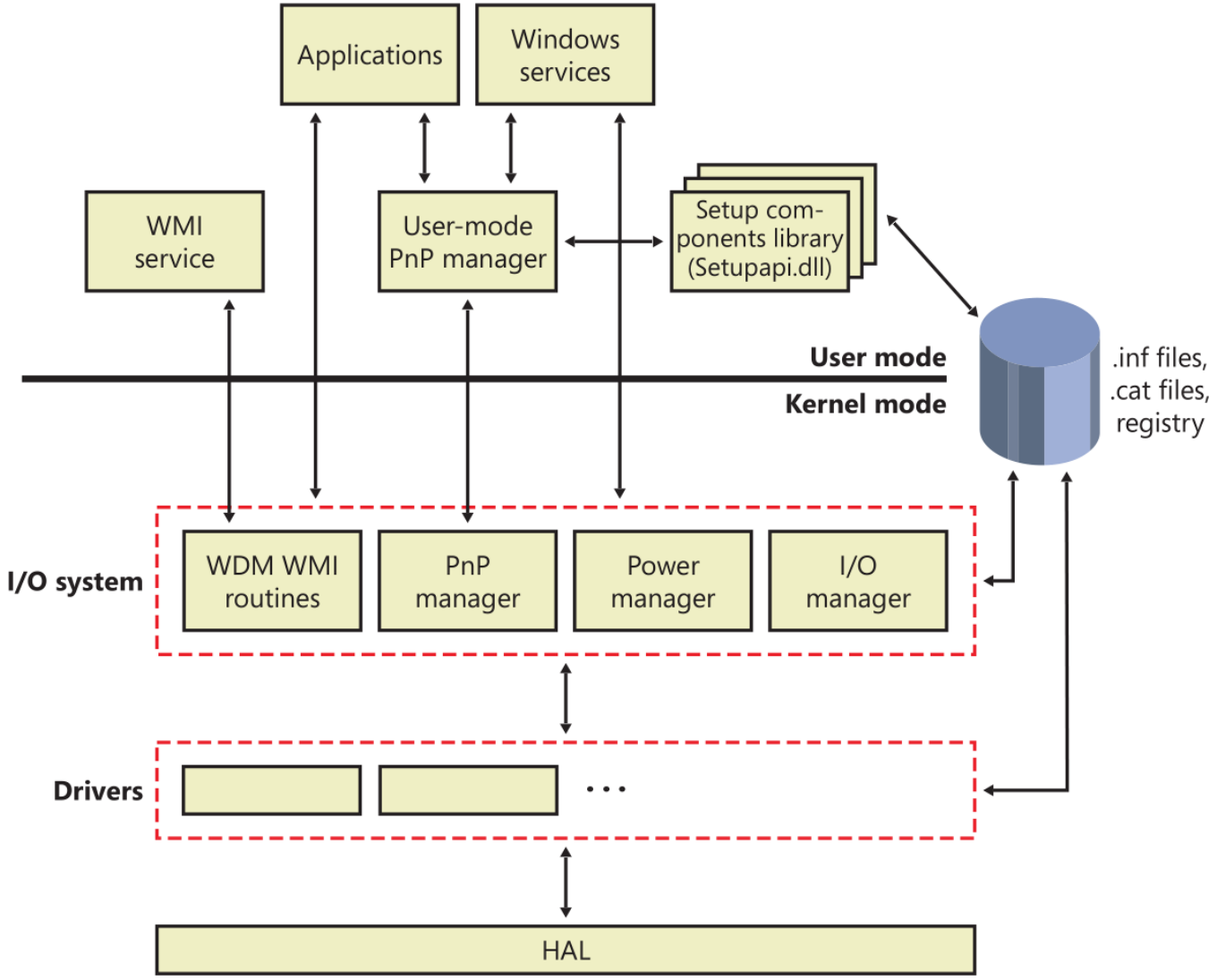
3

Malware main interest is in user important data files: doc, docx, xls, xlsx, ppt, pptx, pdf, txt.

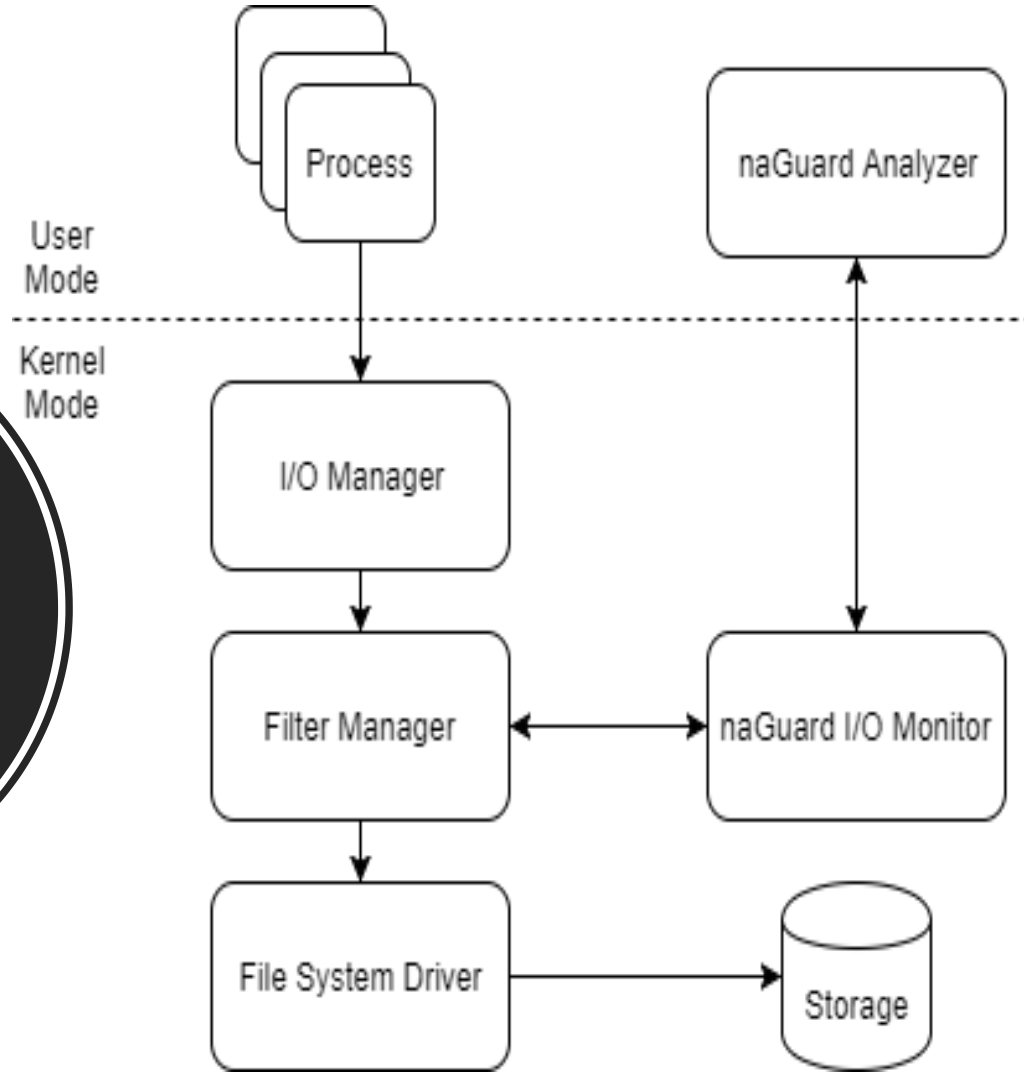
4

Malware probably will not be sign OR at least not by MS.

Introduction to Windows I/O System



Architecture



naGuard Analyzer: a user mode application which analyzes all the data received from the I/O Monitor and detects and stops malicious behavior.

naGuard I/O Monitor: A Windows kernel module which monitors and logs every single I/O operation in the system. The information is passed to naGuard Analyzer.

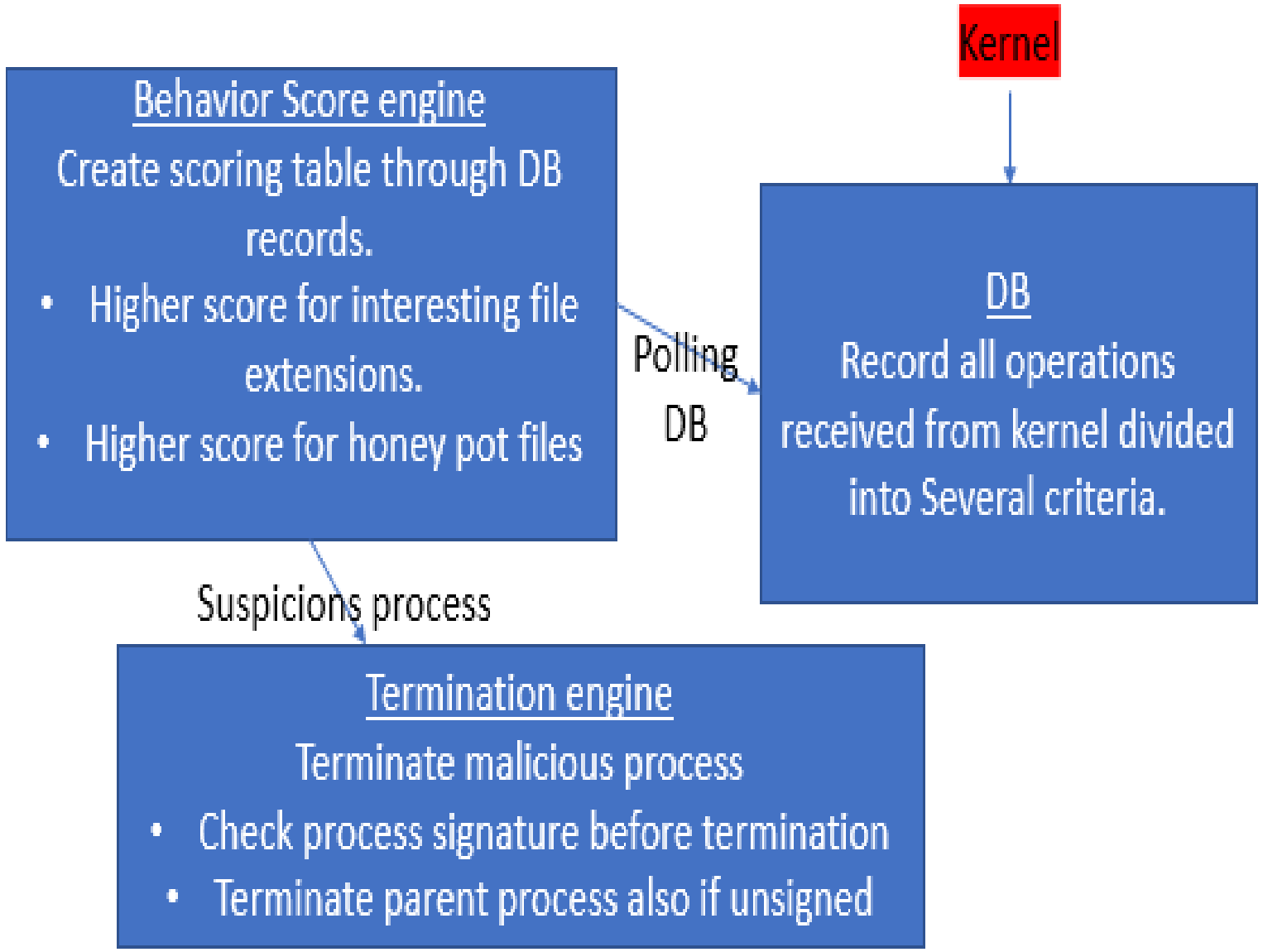
naGuard I\O monitor

Handle callbacks implemented: IRP_MJ_CREATE, IRP_MJ_WRITE, IRP_MJ_SET_INFORMATION.

```
typedef struct filter_message_t {
	int    opcode; // Operation {read, write, rename, delete}
	HANDLE process_id;
	float preop_entropy; // Entropy of a file at the time when the operation was
initiated
	float postop_entropy; // Entropy of a file at the time when the operation was
completed
	WCHAR preop_filename[4096]; // File name at the time a rename operation was
initiated
	WCHAR postop_filename[4096]; // File name at the time a rename operation was
completed

} NAGUARD_FMESSAGE, *PNAGUARD_FMESSAGE;
```

Kernel – User protocol

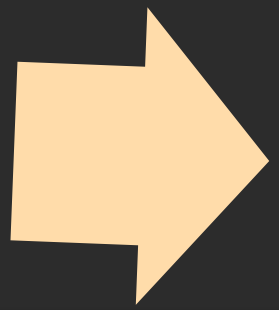


naGuard Analyzer

Multithreaded scoring table which grant higher scoring to malicious behavior processes.

naGuard Analyzer DB

unordered_map <HANDLE, ThreadInfo>



Criteria:

- I/O operations: Read, write(new file), rewrite, rename.
- short (3 sec) aggregation and long (process life) time aggregation
- Absolute and relative entropy.

ThreadInfo:

- M_score - current malicious score.
- honeyPotsCounter – counter of honey pots touched by process in the last X seconds (X=3).
- Write_end_entropy – sum of absolute entropy in rewrite operations.
- Write_delta_entropy – sum of delta entropy in rewrite operations.
- New_write_entropy – sum of entropy in new write operations.
- notExtInListWrite – number of all write operations (include all extensions) in the last X seconds (X=3) if absolute entropy higher then 3.5.
- honey_pots_touched – number of honey pots files touched (all OPS)
- m_total_ops[OPS_NUM] – number of all operations (interesting extensions only) in the last X seconds (X=3).
- m_ops[OPS_NUM] – number of all operations (interesting extensions only) in the process life.

Malware behavior analysis

Ransomware can behave in one of 3 ways:

- Read file -> encrypt -> write content to file in same path with different extensions -> delete original file.
- Read file -> encrypt -> write content to same file-> rename original file.
- Read file -> encrypt -> write content to same.

Score engine

- The algorithm based on malware behavior analysis which was presented earlier.
- Increase score for suspicions behavior and decrease score for innocent behavior through time
- Higher score for “interesting” file extensions and higher for honey pots.
- The algorithm was tuned experimentally.
- If total score of process is higher than 100 and the process is not signed it would be terminated.

Tests and Results

Detection accuracy - Detect 5 known real ransomware

- WannaCry
- Jigsaw
- Satana
- Vipassana
- Cerber

False positive checks

- Windows update – no FP.
- Applications installation – no FP.
- 7z (zip and unzip) – no FP.

Various checks for System stability

Overhead [performance hit].

Future Work

- Analyzer based on machine learning.
- Check for bin file signer.
- Check for ADS.
- Protect boot sector.
- Recover encrypted files.
- Smarter honey pots (randomly generated)

Conclution

- The project include 3 distinct parts of research
  - Windows internals.
  - kernel development and debugging.
  - Ransomware research & behavior analysis.
- Despite the complexity we decided to carry out the project because we believed this is the right way to implement such mission.
- Bonus points implemented
  - **Kernel based (driver) solution.**
  - Honey pots.
  - Sign check