Low-level Features for Multinomial Malware Classification

Sergii Banin

10.05.2017

Agenda

- Introduction (problem description).
- Previous research.
- Malware classification approaches.
- Related studies.
- Applying of low-level features for malware classification.

Introduction (problem description)

- Signature-based malware detection is not robust against simple obfuscation techniques.
- Static properties can be easily changed.
- Dynamic analysis can aid and outperform static analysis.
- Malware developers try to conceal malware's functionality.
- It is impossible to avoid execution on the hardware.
- Thus we propose to analyze hardware or low-level activity produced by malware.

Previous Research

- Memory access patterns were used for malware detection.
- Record **sequence** of memory **read** and **write** accesses (*memtraces*) with a help of dynamic binary instrumentation tool Intel Pin.
- Extract **n-grams** from memtrace sequence and use them as features.
- Select **best features**.
- **Train** Machine Learning models.
- Assess classification **accuracy** achieved by ML models.
- **kNN** and **ANN** achieved classification accuracy of **98.92%** for 1,000,000 memtraces, 800 features and 96-grams.

Next Step

- Apply low-level features for malware classification.

BUT

- Why do we need malware classification?
- How it is usually performed?
- Is it possible to apply low-level features for malware classification?

(Low-level Features for) Multinomial Malware Classification

Sergii Banin

10.05.2017

Agenda

- Problem description.
- Malware classification.
- Malware families and types.
- Reasons for separating malware by families and types.
- Related studies.

Problem description.

- Inconsistent terminology (family/type).
- Dozens of malware naming systems.
- CARO (Computer AntiVirus Researcher's Organization) naming system. [http://www.caro.org/articles/naming.html]
- Common Malware Enumeration (CME) initiative by MITRE. [https://cme.mitre.org/about/]
- Naming is usually made to describe malware's target platform, functionality, variation of a certain sample, etc. [http://security.di.unimi.it/~roberto/teaching/vigorelli/0607/malware/material/caro.pdf, https://zeltser.com/malware-naming-approaches/, https://www.microsoft.com/en-us/security/portal/mmpc/shared/malwarenaming.aspx]

Problem description (2)

CME-136	*Avira*: W2000M/Kukudro.C *Authentium*: W97M/Kukudro.C *CA*: W97M/Kukudro.B:trojan *ClamAV*: Trojan.Dropper.MSWord.MyNo-3 *ESET*: W97M/TrojanDropper.Lafool.NAA *Fortinet*: WM/Kukudro.C *GRISOFT*: W97M/Kukudro *H+BEDV*: W2000M/Kukudro.C *Kaspersky*: Trojan-Dropper.MSWord.Lafool.j *McAfee*: W97M/Kukudro.c *Microsoft*: W97M/Kukudro.C!CME-136 *Panda*: W97/Kukudro.C!CME-136 *Sophos*: WM97/Kukudr-Fam *Symantec*: W97M.Kukudro.A	CME-136 is a Microsoft Word macro virus that drops a trojan onto the infected host.	2006-06-29
CME-476	*Avira*: W2000M/Kukudro.B *Authentium*: W97M/Kukudro.B *CA*: W97M/Pricheck.B *ClamAV*: Trojan.Dropper.MSWord.MyNo-2 *ESET*: W97M/TrojanDropper.Lafool.NAA *Fortinet*: WM/Kukudro.B *GRISOFT*: W97M/Kukudro *H+BEDV*: W2000M/Kukudro.B *Kaspersky*: Trojan-Dropper.MSWord.Lafool.j *McAfee*: W97M/Kukudro.b!CME-476 *Microsoft*: W97M/Kukudro.B!CME-476 *Panda*: W97/Kukudro.A *Sophos*: WM97/Kukudro-B *Symantec*: W97M.Kukudro.A *Trend Micro*: W97M_DLOADER.BVS	CME-476 is a Microsoft Word macro virus that drops a trojan onto the infected host.	2006-06-28

Malware classification

With classification we can:

- assign threat level
- assess possible harm
- apply countermeasures
- perform post-attack actions

Malware classification (types)

- Malware types: Trojan, Virus, Hoax, Ransomware, Adware, Spyware.
- Malware type is assigned by general functionality.
  - E.g. Viruses are self-replicating malware, and Ransomware encrypts user data while asking for ransom to be paid.
- Certain type can have different subtypes assigned by actions performed on the victim system.
  - Trojan-Bankers are designed to steal account data from online banking.
  - Backdoor Trojans give malicious users remote control over the infected computer.

Malware classification (families)

- Malware families are the malware categorization based on the particular functionality.
- For describing malware families the following functionality could be used:
  - Which files are created/modified by a malware.
  - Which registry entries are affected by it.
  - Affected drivers.
  - Commands run by malware.
- E.g. Win32/Ursnif (Gozi) (according to Microsoft Malware Protection Center) can run from PDF, MSI or EXE file. Create files in system directories, change registry entries related to software protection, capture screenshots, steal cookies, download and install new executables etc.

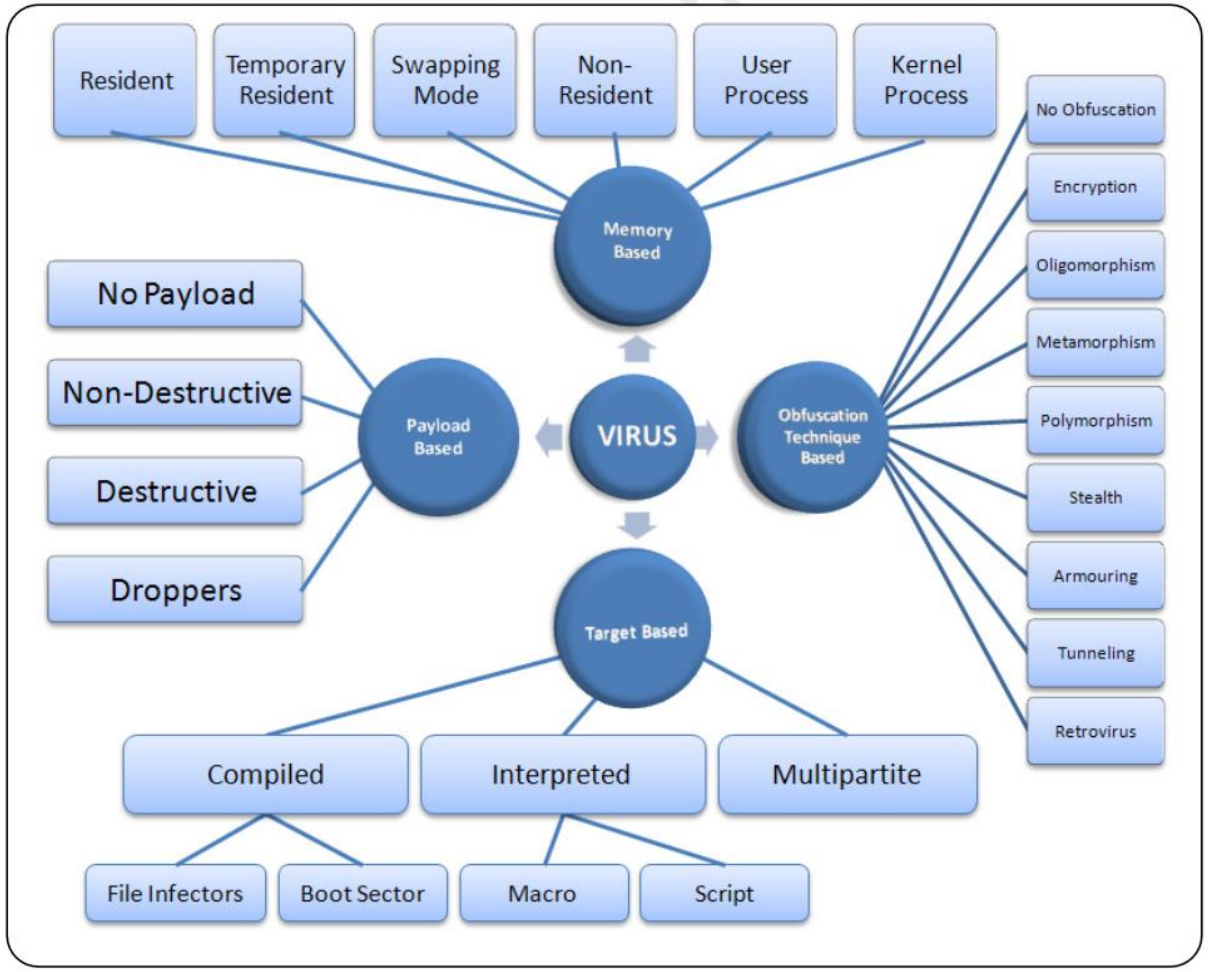
Reasons for separating malware by families and types.

- Classification allows to assign threat level and assess possible harm.
- Knowledge about the type of malware allows to apply particular counter-measures.
- Knowledge about the family of malware allows to perform exact actions for restoring and cleaning the system.

Related studies

- SANS in their paper (Malware 101 - Viruses) suggest the following virus classification strategy:
  - How malware stay in memory: resident, temporary resident, user process, kernel process.
  - Spreading methods: compiled, interpreted, multipartite.
  - Obfuscation techniques: no obfuscation, encryption, metamorphism, polymorphism, stealth.
  - By payload type: no payload, non-destructive, destructive, droppers.

Related studies



SANS

Ways to perform multinomial malware detection

- In the literature there are different ways of performing multinomial malware classification:
  - API calls, Byte and opcode n-grams, opcode frequencies.
  - API call sequences, control flow, autostart exstensibility points.
  - System state changes (through VM slices), call graph analysis, clustering via filesystem/registry/network activity. [Malware Analysis and Classification: A Survey Ekta Gandotra , Divya Bansal , Sanjeev Sofat]

Application of low-level features for multinomial classification.

(Based on SANS taxonomy)

- Memory activity can be analyzed on the level of single operations. (ongoing research)
- Malware obfuscation-related activity could be traced within memory and CPU.
- Interpreted viruses can be analyzed while interpreter is active.
- Boot sector infection can be traced via HDD operations.

Conclusions and Suggestions.

- Different ways of classification.
- Properly described clustering can work better than well-known taxonomies.
- Application of low-level activity can improve stealthy detection and new classification methods.

Thank you.