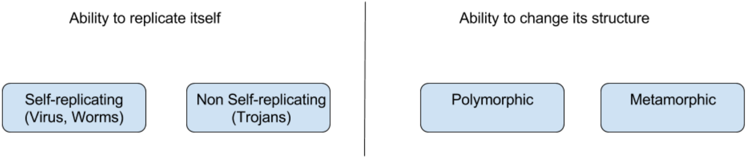
“Botnet Battlefield”: A Structured Study of Behavioral Interference Between Different Malware Families

Bishwa Hang Rai
Supervisor: Prof. Dr. Alexander Pretschner
Advisor: Mr. Tobias Wüchner

Table of contents

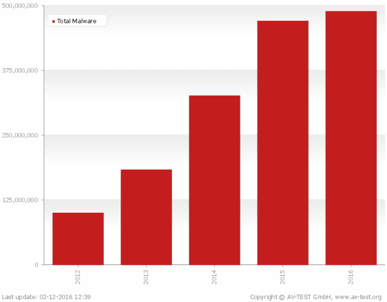
Malware

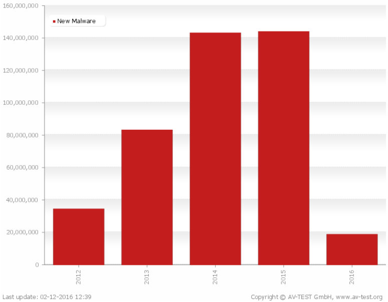
- Malicious software that corrupts or steals data, or disrupt operations with illegitimate access to computer or computer networks



- Different variants of same malware and hard to detect with signature based

Growth of Malware





- Annual loss caused by malware in 2006, 2.8 billion dollars in US and 9.3 billion euros in Europe
- Driven by monetary profit, high rise in numbers of new malware with 140 million new malware introduced in 2015 alone

Interference Between Malware Families

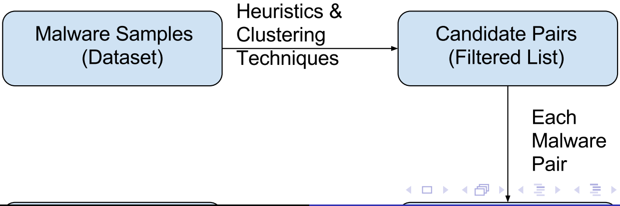
- There has been some anecdotal evidences of feud between the malware families
- In 2004, NetSky vs Bagle and MyDoom trying to remove each other along with message of profanity
- In 2010, SpyEye vs Zbot with KillZeus feature
- In 2015, Shifu malware family with AV like feature
- All of these interferences were to remove/prevent the infection of another malware
- Increase their own profit taking control of larger share of economy

Problem Statement

- The purpose of our research is to identify the existence of aforementioned behavioral interference between the malware families
- The study will provide novel knowledge for understanding the dynamic aspect of modern malware, the inter-family relations, and their associated underground economy
- This behavior is also a case for environment-sensitive malware
- That is to say malware changing their behavior depending on different factors of their running environment, such as presence or absence of files, programs, or running services

Research Process

- Get wide variety of malware samples
- Use heuristics and clustering to get the candidate pair list
- Run each candidate pair in malware analysis system (Anubis in our case)
- Analyze the log of analysis run to detect behavioral interference



Heuristics

- Millions of malware samples collected over time by Anubis
- Need to find candidate pairs from the dataset with probability of behavioral interference
- Choose based on common resource such that one malware created the resource and another malware tried to delete the same resource, but with failure
- Parsed behavioral profiles of malware to get the failed delete/access operations
- Reverse index of resource to malware samples
- Large number of possible pairs and inconsistent AntiVirus label lead to Clustering of malware sample

Latent Dirichlet Allocation

- Clustering of malware with respect to their behavioral profile
- LDA: General probabilistic model for collection of discrete data such as text corpora
- Does not depend on number of documents and its memory footprint is $O(words \times topics)$
- Each malware is represented as document and their activities as words in document
- Resource types, operations, and resource name were represented by numeric id
- We had a good clustering result with high intra-distance and low inter-distance of clusters

Candidate Selection

R = Set of all interesting resource
A_r = Set of malware that creates a particular resource 'r'
B_r = Set of malware that delete/access (failed) particular resource 'r'
N = Maximum number of families to consider
E = Set of all probable candidate
function C (j)
    c_j = cluster id that malware j belongs to
    Return c_j
end function
for all $r \in R$ **do**
    if $|C(x_r) : x \in A_r| > N \vee |C(y_r) : y \in B_r| > N$ **then**
        continue
    end if
    for all $(x_r, y_r) \in A_r \times B_r$ **do**
        if $C(x_r) \neq (y_r)$ **then**
            $E \leftarrow (x_r, y_r)$
        end if
    end for
end for

Packer/Unpacker

- The candidate pairs had to be run together inside the Anubis system
- We used the fact that addition of extraneous data would not affect the binary execution
- ‘Unpacker’ binary read itself from the behind
- The ‘Packer’ packs the candidate pair along with meta-information such as its size and time delay to Unpacker
- The Unpacker then would create the malware sample and execute it with specified time delay inside Anubis

Contribution

Our research will provide the following contributions:

- To the best of our knowledge, we are the first to perform a systematic study of interferences between malware families
- A novel approach to malware clustering based on malware behavior profiles
- An automated system that detects interfering malware samples on a large scale

List of Candidate Pairs

- Value of N (maximum family cutoff) in algorithm chosen to be 10
- File with the highest number of candidate pair and Process the lowest
- No candidate pair from resource type Job, Device, Driver

Resource types	#candidate pairs
File	213,171
Registry	39,899
Sync	7,781
Section	2,786
Process	54
Total	263,691

Experiment Setup

- 7 Anubis instance
- Each instance emulates entire running PC with Windows XP Service Pack 3 as OS
- Uses Qemu and monitors process by invoking callback routine for every basic block executed in virtual processor
- Unpacker and Packer used to run the candidate pair
- 10 minutes as total run time of each candidate pair experiment
- 4 minute for each malware, and 2 minute to boot system

Result of Candidate Run

Resource types	# tested pairs	# true positive	prediction accuracy
File	5,000	1032	20.64%
Registry	5,000	731	14.62%
Sync	1,000	119	11.9%
Section	1,000	93	9.3%
Process	54	6	11.11%

- Highest Accuracy for File and Registry
- Lowest for Process
- Average accuracy rate 14.25%

Some Examples

- Artemis! vs Cosmu on resource `C:\Old.exe`
- VB.CB vs Startpage.AI on resource `C:\WINDOWS\window.exe`
- KeyLogger vs OnlineGames on resource `C:\windows\system32\svrchost.exe`

Threats to Validity

- Different values of N would give different candidate pairs and different results
- Didn’t deal with random resource name
- Total execution time 10 minutes
- Sequence of execution
- Semantics of Malware

Conclusion

- Behavioral interference between malware families exists
- Malware checks for the presence of resource created by other malware and deletes it
- Our system could detect such interfering malware with average accuracy rate of 14.25%
- In our dataset, Files and Registries were the most interfered resource and Process was the least

Future Work

- Make the experiment more efficient to run multiple times with different parameters
- Research on other possible approaches to clustering
- In depth analysis (static) of positive pair to know the true semantics of malware

QUESTIONS???

Listing 1 : Sort and join the reverse index

```
LANG=en_EN sort -t, -k 1,1 $file_name
LANG=en_EN join -t , -a1 -a2 $fn1 $fn2
```

Listing 2 : Sample of reverse index created for File activity

```
C:\mbr.exe,189524063,184501719,87504631,86763863
C:/DOCUME~1/ADMINI~1/LOCALS~1/Temp/telnet.exe
    ,178046895,174206059,183601891,89650247
C:/DOCUME~1/ADMINI~1/LOCALS~1/Temp/1.jpg
    ,161552035,116241803
```

Unpacker

Figure : Structure of the Unpacker binary that would create the candidate pair and run them with delay.

Inter and Intra Distance

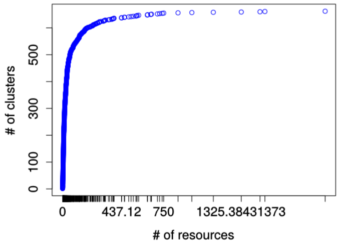


Figure : Graph showing cdf distribution of common resource between same family topic

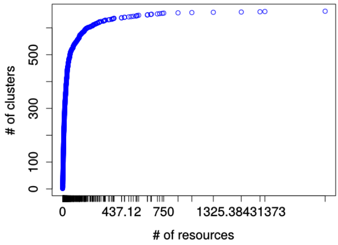


Figure : Graph showing cdf distribution of common resource between same family topic

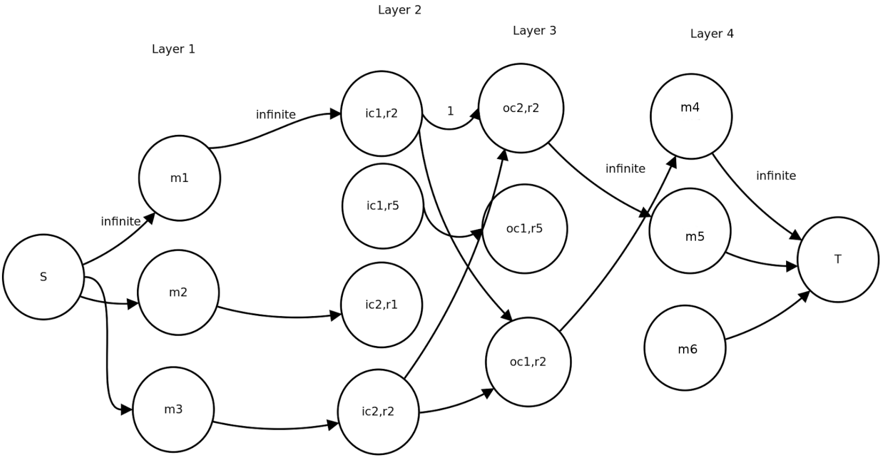


Figure : Graph representing the max flow implementation

Heuristics

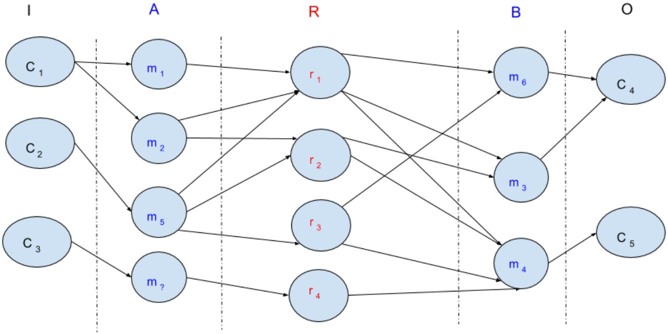


Figure : Heuristics approach to optimal malware pair selection