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

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

Department of Informatics
TU München

January 22, 2016

Table of contents

Introduction
- Background
- Problem Statement

Methodology

Contribution

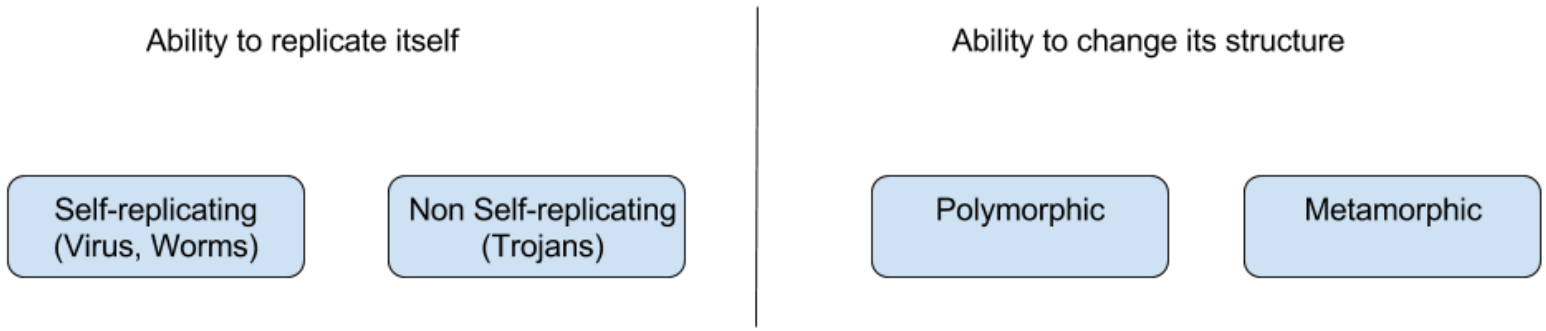
Evaluation
- Experiment
- Results

Threats to Validity

Conclusion and Future Work

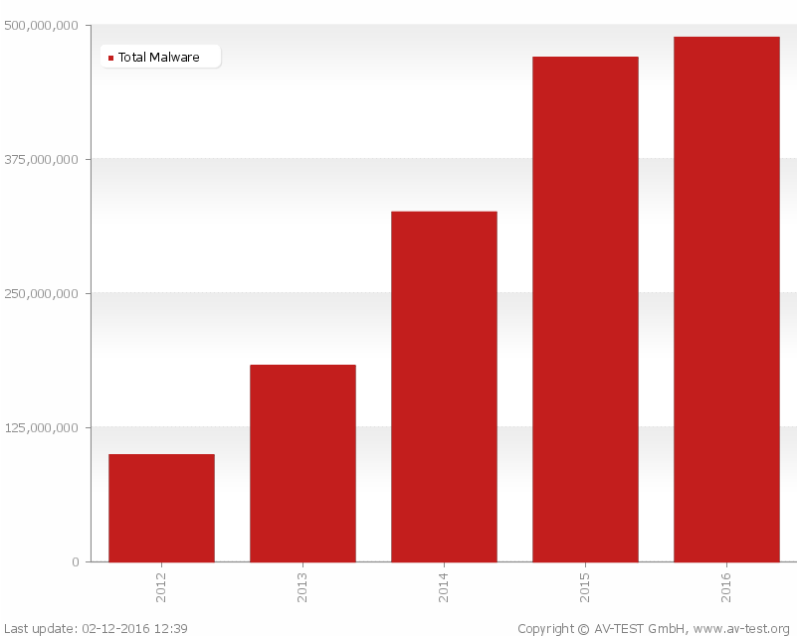
Malware

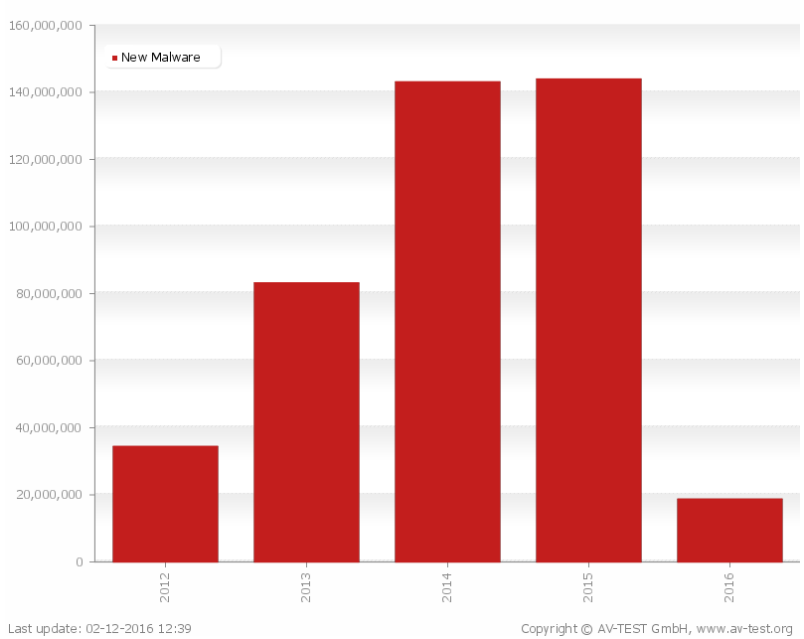
- Malicious software that corrupts or steals data, or disrupt operations with illegitimate access to computer or computer networks



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

Growth of Malware





- High rise, driven by monetary profit
- In 2006, 2.8 billion dollars in US and 9.3 billion euros in Europe

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
- remove/prevent the infection of another malware
- Increase their own profit?

Problem Statement

- The purpose of our research is to identify the existence of aforementioned behavioral interference between the malware families
- Dynamic aspect of modern malware, the inter-family relations, and their associated underground economy
- Environment-sensitive malware

Contribution

Our research will provide the following contributions:

- 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

QUESTIONS???

Reverse Index

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

Inter and Intra Distance

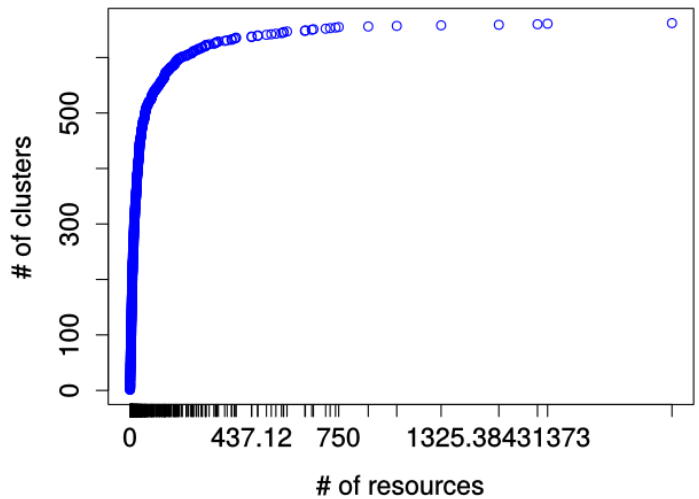


Figure 2 : Graph showing cdf distribution of common resource between same family topic

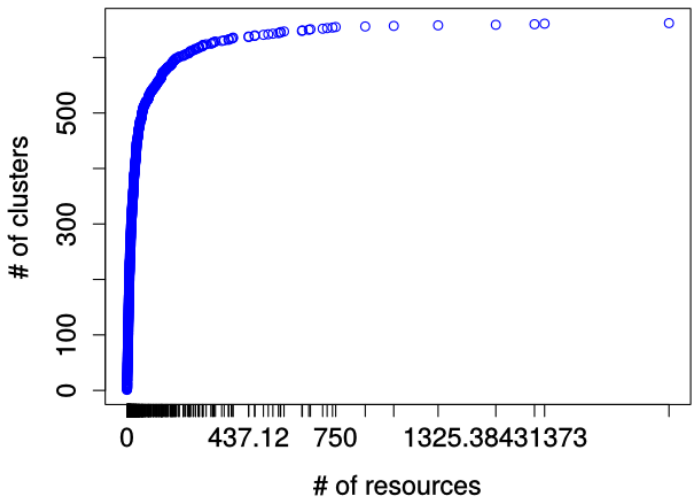


Figure 3 : Graph showing cdf distribution of common resource between same family topic

Max Flow

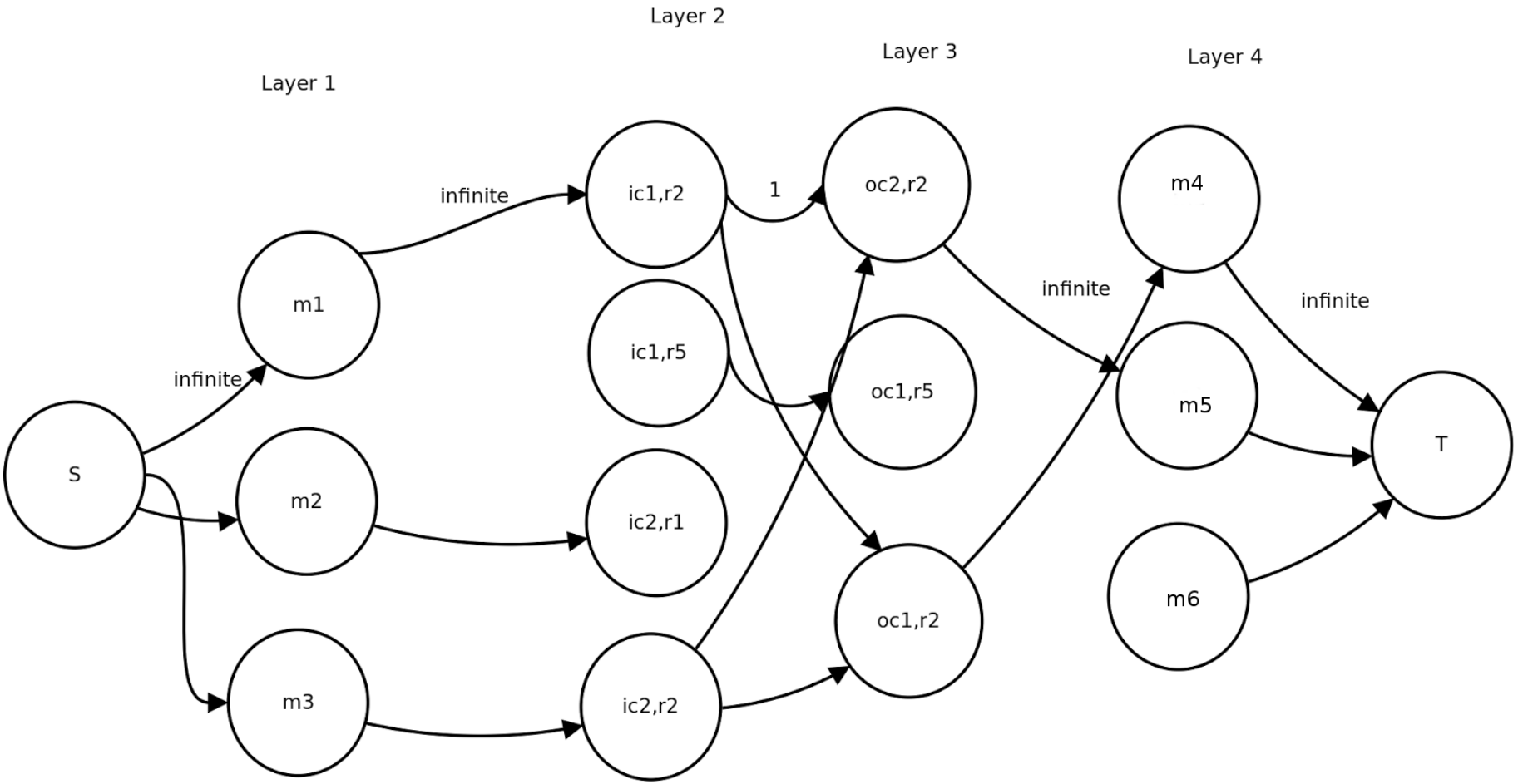


Figure 4 : Graph representing the max flow implementation

Heuristics

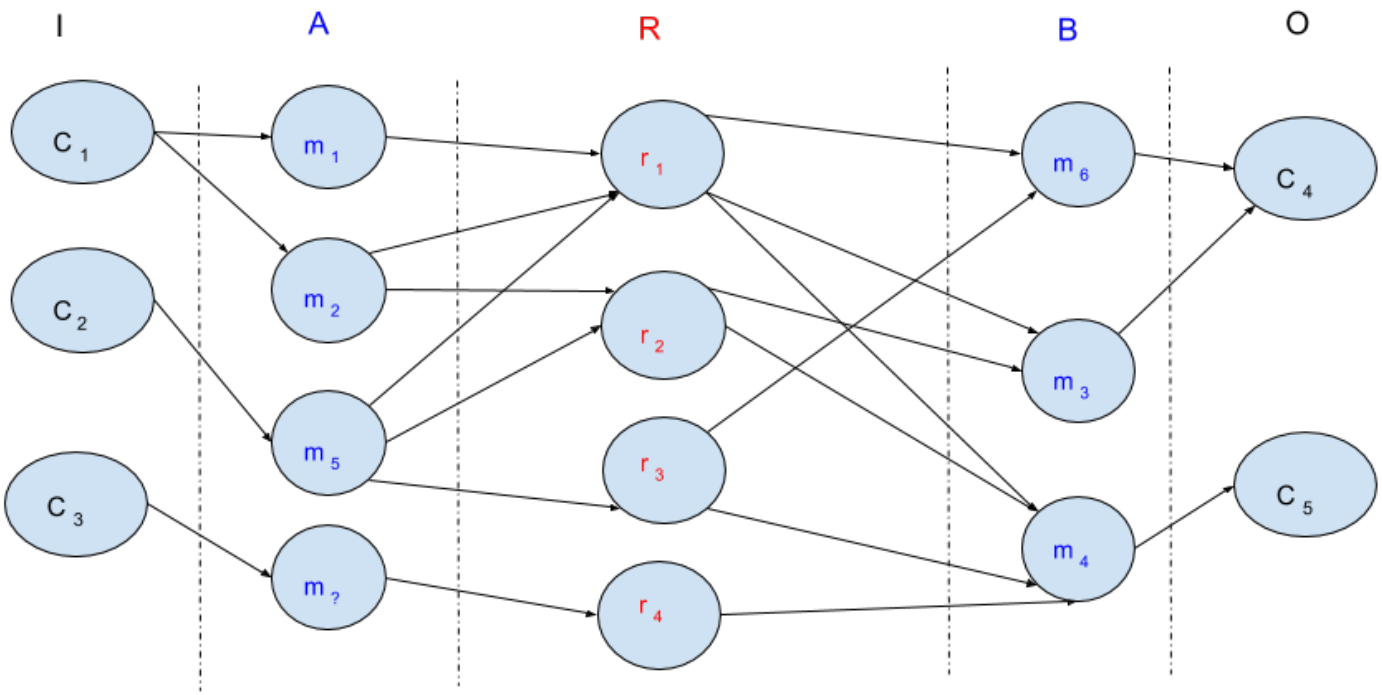


Figure 5 : Heuristics approach to optimal malware pair selection