# Race to Zero with online scanners

Boris Lau, SophosLabs

Virus Bulletin 2008

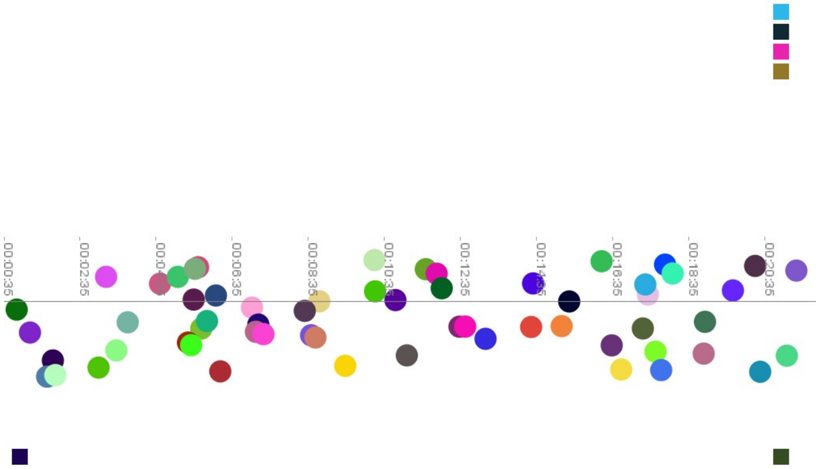
# Racing in the wild

- First half: Methodology of our work
  - Brief introduction
  - Difficulties in generating the data
- Second half: Case studies
  - High level visual demo of some cases
  - Some stats

# Brief introduction

- Why do it?
  - Malware writers want to avoid detection.
- How they do it?
  - One of the cheapest way is to use online scanners
- What to observe?
  - VirusTotal incoming samples
- When?
  - Last week's data (22/9/2008 + 7)

# (Demo: Order from chaos)



Explaining work required to filter the sample stream

- Properties (meta and real)
- Scoring of properties
- Grouping via meta data
- Grouping via real data

# (Demo: spotting the races)

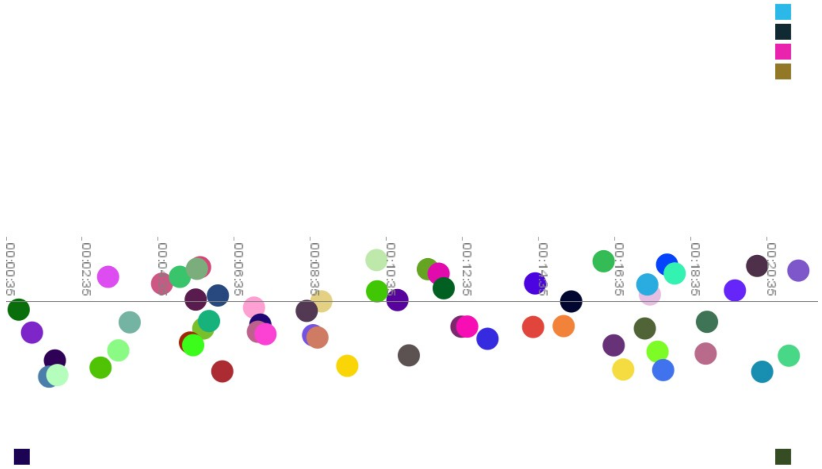
- Lots of different reasons that groupings are submitted (e.g. outbreak, multiple infection on same computer)

- Using meta-data to discover the races

1. signs of progression (e.g. filename, timestamp)

2. reducing number of detected products

# (Demo: techniques used)



How they will try to modify the binary to do the work

- manual modification
- recompilation
- code morphing

(These does not include repackaging techniques such as droppers/packers)

# Statistics

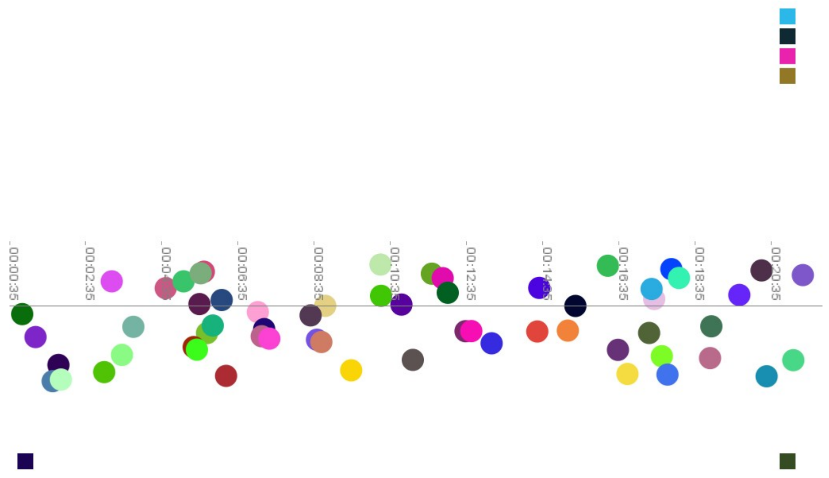
- Taking samples form 7 days period of 22/9
- About 74 attempts to submit samples which have signs of “progression”
- 251 samples – length of race is about 3.3 samples
- Average speed of about 72 minutes per sample

# (case study: example race)

- A real demo follows

- (showing modifications

  made to existing packers)



# Race result

- “Is it really that easy to beat the AV scanner?”
- Looking back at the 74 races that we had
  - Only 5 races shows clear sign of reducing detected count
  - Scorecard: AV 69, Malware writers 5?
- Difficult to say who wins
  - limited sample set
  - limited visibility to the real zero (only race to epsilon?!)

# Thank you

- VirusTotal.com is by Hispasec
  - http://www.hispasec.com
- Visualization is done using the processing framework
  - http://processing.org

# Appendix: VirusTotal explained

- Investigation with VirusTotal.com
  - One of the largest online scanning service
- Based on samples which are detected by >=1 vendors
- Only about 5% of samples we are interested in
  - See definition about “interested” later

# Appendix:
# Type of sample that was “raced”

- Bifrose / Backdoor / bots
- Online Game password stealers / trainers
- Exploits (Doc/SWF – generated by kits)
- Droppers
- Maybe an indication that these are more 'hobbyist' malware writers?

# Appendix: Why visualize?

- “Why don't you just have an automated classifier instead of looking at it manually?”
- To implement a good classifier, one needs to identify possible heuristic from complex information
- Also need to check how well behaved are those classifier
- That's where visualization could help – to create and debug automations

# Appendix:
# How good is our classifier?

- There will be changes that are too drastic to identify
- Packers – based on Meta information from the stream and the linker
- Dropper – difficult if we cant see through the archiver
  - Meta information might helps
  - Might need to “work” the sample dynamically

# Appendix: Scoring of properties

- Grouping algorithm to find related samples
  - Each files f have a set of properties P(f) = {p1,p2...pn}
- Using idea from Term-Frequency/Inverse Document Frequency (tf-idf) scoring from Information Retrieval
- roughly ~ (Number of appearance of property in a group / Number of files that have the properties)