Application-Level Reconnaissance: Timing Channel Attacks Against Antivirus Software

Mohammed I. Al-Saleh and Jedidiah R. Crandall
Client Reconnaissance

Hmmm, what can I get about you?!!
Client Reconnaissance

• Browser identification
  – https://panopticlick.eff.org/

• AV related info
  – AV fingerprinting
  – Up-to-date?

• Timing channels
  – AV performance tradeoff
  – Make the common case fast
  – Updated?
Threat Model

Connect

Measure scanning time

Updated??

Server

Client
Basic Idea

• Antivirus (AV) scans data against sigs
• Sigs are stored somehow in AV’s data structures
• Scanning time
  – Based on scanning path
• Hitting the newly added sigs
ClamAV

• ClamAV
  – http://www.clamav.net
  – http://www.clamxav.com/
  – http://www.clamwin.com/

• Scanning steps:
  – File type filtering
  – Filtering step
  – Boyer-Moore algorithm
  – Aho-Corasick algorithm
File Type Filtering

File to scan

File Type Filtering

[Diagram of a process flow with a file to scan being filtered]
Filtering Step

Input → Filter → Yes/No
Boyer-Moore
Aho-Corasick
Methodology

• **Question #1**: Is there a timing channel in the way ClamAV scans data?

• **Question #2**: If the first question is confirmed, how could the attacker create the timing channel?
Methodology/Q1

• Collect viruses in (name,date) pairs and remove their sigs from current DB
Two Kinds of Experiments

- Whole-day sig experiment
- Single sig experiment
Whole-Day

DB before DateX

Becomes

DB after DateX

Sigs of DateX

Old

New

Scan
BuffSize = 256 KB

(content)

File Size
Single Signature

DB before SigX

Becomes

DB after SigX

Old

New

Scan
Whole-Day

Frequency

Time Difference (seconds)
Single

Time Difference (seconds)

Frequency
Methodology/Q2

Diagram showing the process:
- Start
- Connect
- Embed ActiveX
- Send Results
- Done
Methodology/Q2

1. Create file
2. Close the file
3. Start CPU Time Sampling
4. Determine CPU Busy Time
Possible Timing Channels in Modern AVs

- Pattern matching
- Algorithmic scanning
  - Zmist virus needs to execute at least 2 million p-code-based iterations
- Code emulation
  - Significantly slows scanning
- Heuristics
  - Extra work when triggered
Related Work

• Network discovery
  – Port scanning

• Timing channel attacks
  – Secret keys in cryptographic systems
  – Virtual machines detection
  – Others

• Antivirus research
  – Signature extraction
  – Detection evasion
Conclusion and Future Work

• Application-level reconnaissance through timing channels
• Running example: ClamAV
• Currently, we are exploring performance issues in commercial antiviruses
Acknowledgements

• Török Edwin
• LEET reviewers
• U.S. National Science Foundation (CNS-0905177)
Thanks