Disk Level Virus Detection
Adrienne Felt
The Project

• Group members:
  – David Evans, Sudhanva Gurumurthi
  – Nate Paul

• Goal: a better way to catch viruses
  – Using virus behavior
  – Using the disk processor
How Norton AV works

- **String scanning**
  - Compare files against a database of known viruses

- **All files stored as bits on a disk**
  - `MOVE.W D4 D5 0011101000000100`

- **Signatures are strings of bits**
  - `011010100111010000000100100110`
Do virus scanners work?

- Norton Anti-Virus detection rates
  - WildList viruses: 100%
  - Zoo threads: 97%
  - Heuristic detection:
    - 1-month-old signatures: 22%
    - 2-month-old signatures: 8%
  - Outbreak response time: 10-12 hrs

*From AV-test.org, an independent testing agency. Published in PC Mag.*
"Morphing" viruses

- Change their own code between generations
- For example:
  - $x = x+1$
  - $a = x$
  - $a = a + 1$
  - $x = a$
- Now it won’t match the signature!
  - $\ldots00110011000001000111011000100\ldots$
  - $\ldots00100001011001001000011000010\ldots$
Our solution

• Behavior-based detection
  – Static vs. dynamic approach
  – Harder to change actions than code

• Watch behavior using disk processor
  – Viruses access files
  – Disk processor sees all reads/writes
1. User opens a file
2. Operating system asks for data
3. Disk processor retrieves the data
4. Data given to the OS
5. Anti-virus scanner
1. User opens a file

2. Operating system asks for data

3. Disk processor retrieves the data

3.5 Disk processor watches for viruses

4. Data given to the OS

Adrienne Felt • Disk Level Virus Detection
My research

• **Goal**: show that idea is feasible
  – Create “low-level” behavioral signatures
  – Difficult-to-detect viruses

• Experimental overview
  – Run the viruses
  – Record their behavior
  – Look for patterns
Experimental model

- **Future disk processors**
  - File names, opens, closes, offsets
- **Current disk processors**
  - Reads, writes, block numbers

1. Virus accesses a file
2. OS asks for data
3. Disk processor retrieves the data
4. Data given to the OS
Disk requests -> signatures

READ  1636.1672  14:27:20:984  <NO NAME>
   block=  530  ## 5É~Éƒ  <E_RtlCreateActivationContextSXS:
          %s…

WRITE  1636.1672  14:27:20:984  EFISHNC.EXE
   block=  15  ## <@KERNEL32.dllUSER32.dllExitProcessWriteProce…

• Can see behavior from these requests!
  – Use “goat” files to make it clearer
  – Run lots of traces
  – Patterns emerge
What makes a virus a virus?

- Self-replicating program
- Adds its own code to the host’s programs
- Destroys data
- Annoys the user

- Can we tell this apart from user behavior?
Two types of signatures

- General behavior signatures
  - Viruses like executables
  - Change header information

- Virus-specific signatures
  - Characteristic virus behavior
  - Meant for a single or small number of viruses
Testing & refining signatures

• False positives
  – Detecting a user application as a virus
  – This is really bad

• False negatives
  – Not detecting a virus
  – This is bad too
My current work

- Looking for patterns in virus string databases
- There are many similar viruses
  - Can we take advantage of this?
  - aaa and aaaaaaaaa
Questions?

- Disk level virus detection
  - Behavioral signatures composed of disk requests
  - Based on intrinsic virus properties
  - General and specific signatures

- My thesis
  - Finding patterns in virus signatures