Exploiting Temporal Consistency to Reduce False Positives in Host-Based Collaborative Detection of Worms

WORM 2006

David J. Malan and Michael D. Smith
Division of Engineering and Applied Sciences
Harvard University

Contact
malan@eecs.harvard.edu
Motivation

The speed of today’s worms demands automated detection, but avoiding false positives is difficult.
Prior Work

At WORM ’05, we proposed a host-based intrusion-detection system for worms that leveraged collaboration among peers to lower its risk of false positives.
Prior Work

We simulated a system with two peers using traces of actual worms.

We focused on true positives. We detected 100% of the worms in our study.
This Work

We have since deployed a prototype of our vision “in the wild.”

This time we focused on false positives.
Implementing Our Vision

**snapshots:** lists of syscalls executed during an 30-sec window

**anomalous behavior:** similarity among snapshots
Implementing Our Vision

temporal consistency: similarity in behavior over time

![Graph showing temporal consistency of I-Worm/Sasser.B](image)
False Positives

They present two problems.

1) If we mistake a popular non-worm for a worm, we might declare an outbreak when there is none.

2) If we confuse a non-worm on one host with a worm on another, we might overstate an outbreak’s severity.
Research Questions

Avoiding False Positives

- Can we avoid mistaking popular non-worms for worms?
  - `explorer.exe` is not a worm
- Are non-worms, like worms, temporally consistent?
  - If so, what properties distinguish one from the other?
- Can we detect processes with similar behavior on multiple hosts?
  - If so, we can detect a worm’s outbreak.
Methodology

Wormboy 2.0: A Prototype of Our Vision

- Deployed WORMBOY.{EXE, SYS} on 30 real-world hosts running Windows XP with Service Pack 2
- Deployed WormboyD to one snapshot server.
- Monitored and analyzed 10,776 processes, including 511 unique non-worms (873 unique versions)

Source code to be available for download:
http://www.eecs.harvard.edu/~malan/
Defining Worm-Like Behavior

In prior work, we identified $\tau$ and $r$.

$\tau$ = degree (%) of temporal consistency ($\geq 76\%$ for worms)
$r$ = rate (syscalls/sec) of syscalls’ execution ($\geq 64$ for worms)

- All worms in our prior work boasted $\tau \geq 76\%$ and $r \geq 64$.
- 17% of our non-worms (85 of 511) also boast $\tau \geq 76\%$ and $r \geq 64$. 
Can we detect worm-like processes on multiple hosts?

For $\tau \geq 65\%$, we detect common processes at non-negligible rates. These rates of recognition ($m/n$) are not rates of infection ($\iota$)!
Reducing the False Positives

We now also filter by $r'$.

\[ \tau = \text{degree} \, (\%) \, \text{of temporal consistency} \, (\geq 76\% \, \text{for worms}) \]
\[ r = \text{rate} \, (\text{syscalls/sec}) \, \text{of syscalls’ execution} \, (\geq 64 \, \text{for worms}) \]
\[ r' = \text{rate} \, (\text{syscalls/sec}) \, \text{of network activity} \, (\geq \delta \, \text{for worms}) \]

- All worms in our prior work boasted $\tau \geq 76\%$, $r \geq 64$, and $r' > \delta$.
- 2.9\% of our non-worms (15 of 511) pass this improved filter, down from 17\% (85 of 511) previously.
  - But only 3 (1\%) of those 15 are worrisome.
When do we suffer a false positive?

An apparent rate of infection of $i > 13\%$ is a red flag. This is not the same as our rate of recognition.

We suffer a false positive when we detect some non-worm on $i > 13\%$ of peers during a window.
Fewer than 1% (3 of 511) of our non-worms remain worrisome

We see high $\tau$, $r$, $r'$, and $m/n$ for \{ApntEx, explorer, OUTLOOK\}.exe.
Conclusions

Collaboration among peers discourages false positives.

- High $\tau$ lends itself to high rate of recognition.
- Filtration by $\tau$, $r$, and $r'$ avoids most false positives.
- Future Work:
  - Combat high $i$ for remaining 1% of non-worms.
  - Responses for true positives.
- Threats are discussed in paper.