Binary analysis: Malware Detection & Analysis

Yue Duan

Outline

- Malware basics
- Research papers:
  - Semantics-Aware Malware Detection
  - Panorama: Capturing System-wide Information Flow for Malware Detection and Analysis
  - BareCloud: Bare-metal Analysis-based Evasive Malware Detection

Malware Detection Basics

- Malware
  - Software intentionally designed to cause damage
- Malware Detection techniques
  - Static analysis
  - Dynamic analysis

Malware Detection Basics

- Static analysis
  - testing and evaluation of an application by examining the code without executing the application
  - Pros:
    - Good code coverage
    - Time efficiency
  - Cons:
    - False positives
    - code obfuscation
    - Encryption

Malware Detection Basics

- Dynamic analysis
  - testing and evaluation of an application during runtime
  - Pros:
    - Capture behaviors accurately
  - Cons:
    - Poor code coverage
    - High runtime overhead

Semantics-Aware Malware Detection

Mihai Christodorescu, Somesh Jha, Sanjit A. Seshia, Dawn Song, Randal E. Bryant

IEEE S&P 2005

Semantics-Aware Malware Detection

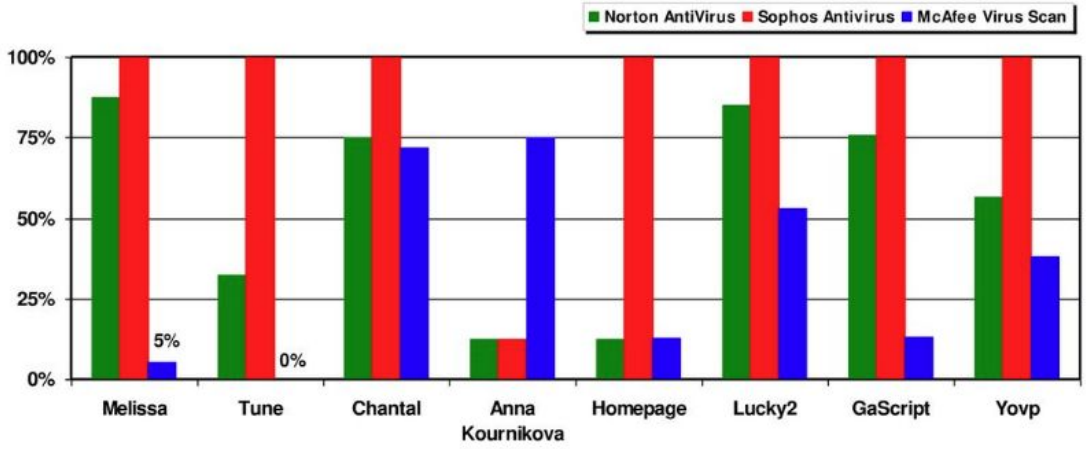
- State-of-the-art techniques - pattern matching
  - susceptible to obfuscations
  - purely syntactic
  - ignore the semantics of instructions

- Attacker’s goal - preserve behaviors
  - Transformation of code and data
  - Addition of new code and data

Semantics-Aware Malware Detection

No Resilience to Obfuscations

False Negative Rate for Obfuscated Worms



Source: "Testing Malware Detectors" (ISSTA 2004)

Semantics-Aware Malware Detection

- Major contributions
  - Introduce semantic signatures
    - Combine syntactic and semantic information
  - Develop a prototype based on the signatures
  - Empirical study shows that one semantic signature can detection a malware family

Semantics-Aware Malware Detection

- Example: detect mass-mailing virus
  - Detect email capability
  - Detect self-propagation

```
s = socket (...);
connect (s);

…

sprintf (buf, “EHLO %S”, dnsname);
send (s, buf);
```

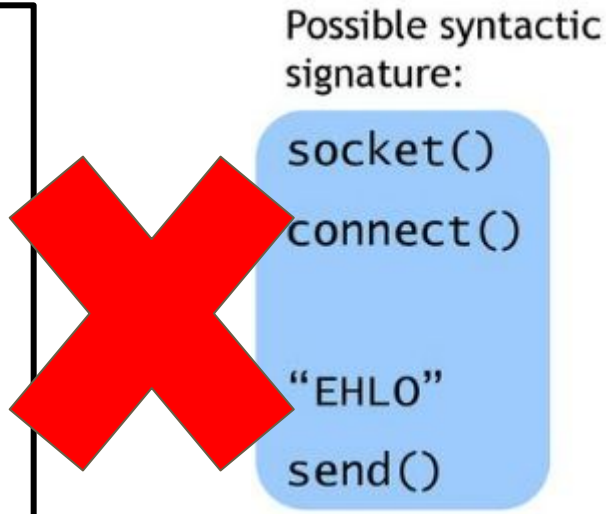
Possible syntactic signature:

```
socket()
connect()

“EHLO”
send()
```

Semantics-Aware Malware Detection

- Variant 1: string manipulation
  - Hide known constants
  - Syntactic signature does not match

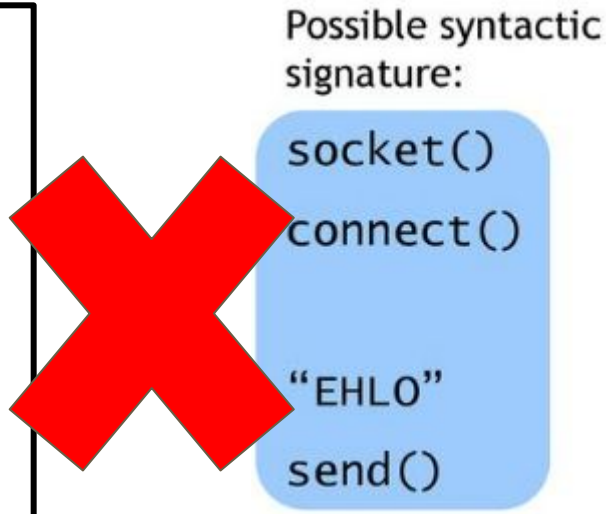
```
s = socket (...);
connect (s);

…
char str [80];
strcpy (str, “EH”);
strcat (str, “LO %S”);
sprintf (buf, str, dnsname);
send (s, buf);
```



Semantics-Aware Malware Detection

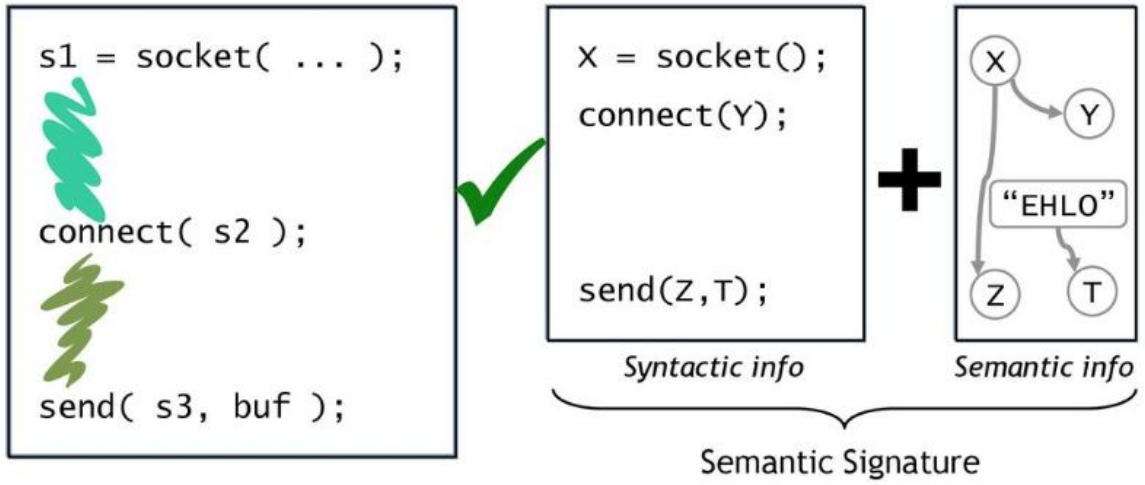
- Variant 2: string obfuscation
  - Hide known constants using simple encryption techniques

```
s = socket (...);
connect (s);

…
char* str = decrypt (encrypted string);
sprintf (buf, str, dnsname);
send (s, buf);
```



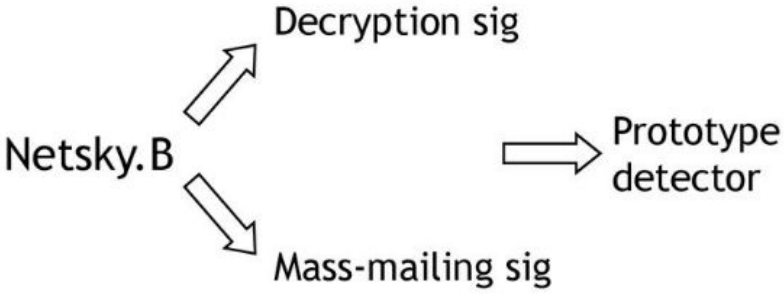
Semantics-Aware Malware Detection

- Attackers
  - Same behavior in different forms
  - contain same semantics
- Semantic signatures
  - Combine syntactic info and semantic info
  - Detect any variant



Semantics-Aware Malware Detection

- Evaluation



Netsky.C	✓
Netsky.D	✓
Netsky.O	✓
Netsky.P	✓
Netsky.T	✓
Netsky.W	✓

McAfee uses individual signatures for each worm.

Semantic signatures provide forward detection.

Semantics-Aware Malware Detection

- Evaluation: Obfuscation resilient

Obfuscation Type	***Semantics-Aware Detection***		*McAfee*
	Average Time	***Detection Rate***	
Nop insertion	74.81 s	**100%**	75%
Stack op. insertion	159.10 s	**100%**	25%
Math op. insertion	186.50 s	**95%**	5%

Panorama: Capturing System-wide Information Flow for Malware Detection and Analysis

Heng Yin, Dawn Song, Manuel Egele, Christopher Kruegel, Engin Kirda

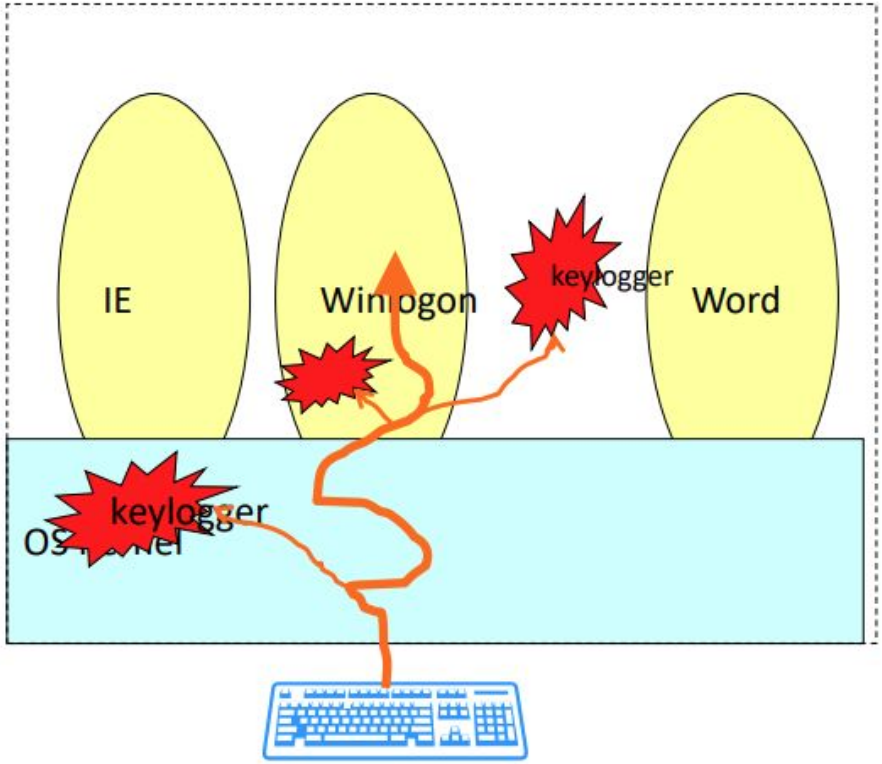
ACM CCS 2007

Panorama

- Malware detection
  - Signature based detection
    - Cannot detect new malware and variants
    - Semantic-aware signature can detect some
  - Behavior based
    - Heuristics: high false positives and false negatives
    - Hooking-based
- Malware analysis
  - Manual process

Panorama

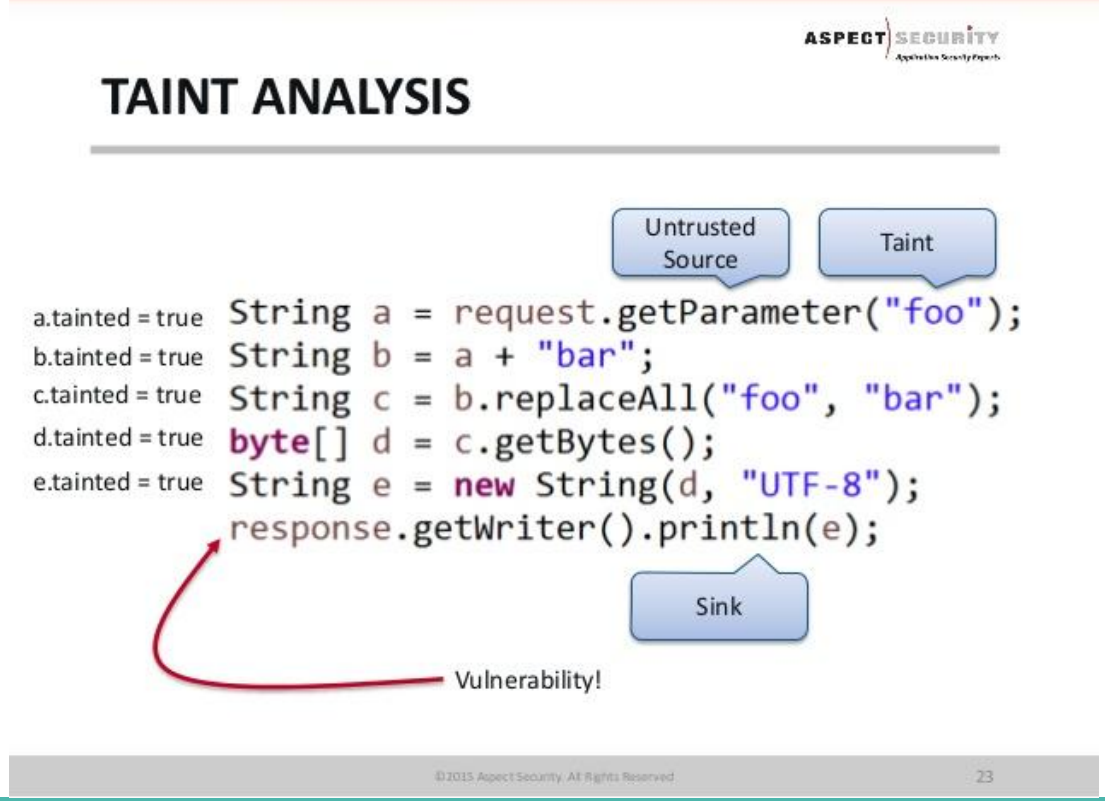
- Observation
  - Information access and processing (IAP) behavior
    - malicious/suspicious IAP behaviors as traces for malware detection and analysis
    - Steal, tamper, or leak sensitive information



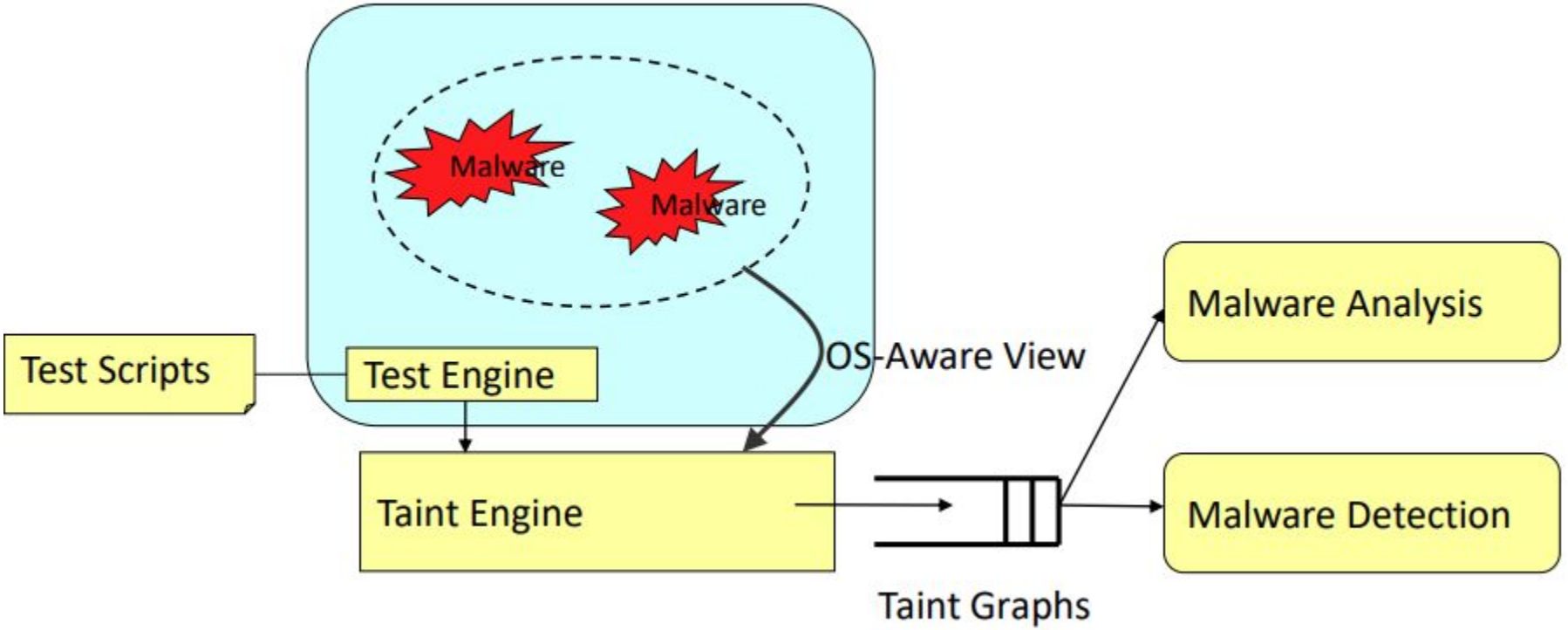
Panorama

- Approach: Whole-system dynamic taint analysis
  - Run the system in an emulator
  - Selectively mark data as tainted
  - Monitor taint propagation
  - Extract OS-level knowledge
  - Generate taint graphs
  - Graph based detection and analysis

Panorama

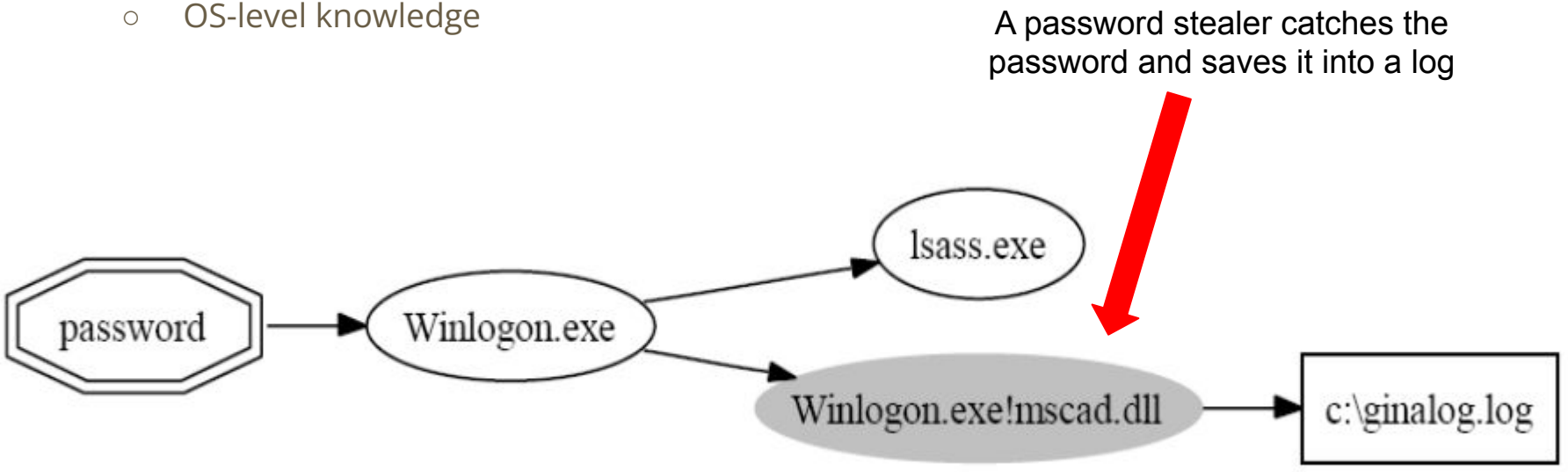

ASPECT SECURITY
TAINT ANALYSIS
Untrusted Source
Taint
a.tainted = true
b.tainted = true
c.tainted = true
d.tainted = true
e.tainted = true
String a = request.getParameter("foo");
String b = a + "bar";
String c = b.replaceAll("foo", "bar");
byte[] d = c.getBytes();
String e = new String(d, "UTF-8");
response.getWriter().println(e);
Sink
Vulnerability!
©2015 Aspect Security. All Rights Reserved
23

Panorama


Malware
Malware
Test Scripts
Test Engine
OS-Aware View
Taint Engine
Taint Graphs
Malware Analysis
Malware Detection

Panorama

- Taint Graph
  - Instruction and hardware level raw events
  - OS-level knowledge



Panorama

- Graph-based Detection and Analysis
  - Anomalous information access
    - Keyloggers:
      - Text: when sending keystrokes to a text editor
      - Password: when sending password to a web form
    - Backdoors:
      - ICMP: when pinging a remote host
      - FTP: when logging into a server
      - etc
  - Anomalous information leakage
    - URL: the keystrokes sent to the address bar
    - HTTP: the incoming HTTP traffic

Panorama

- Evaluation: effectiveness

- Evaluation: performance
  - Curl, scp, gzip: 20x slowdown on average
  - Test cases: 10-15 mins

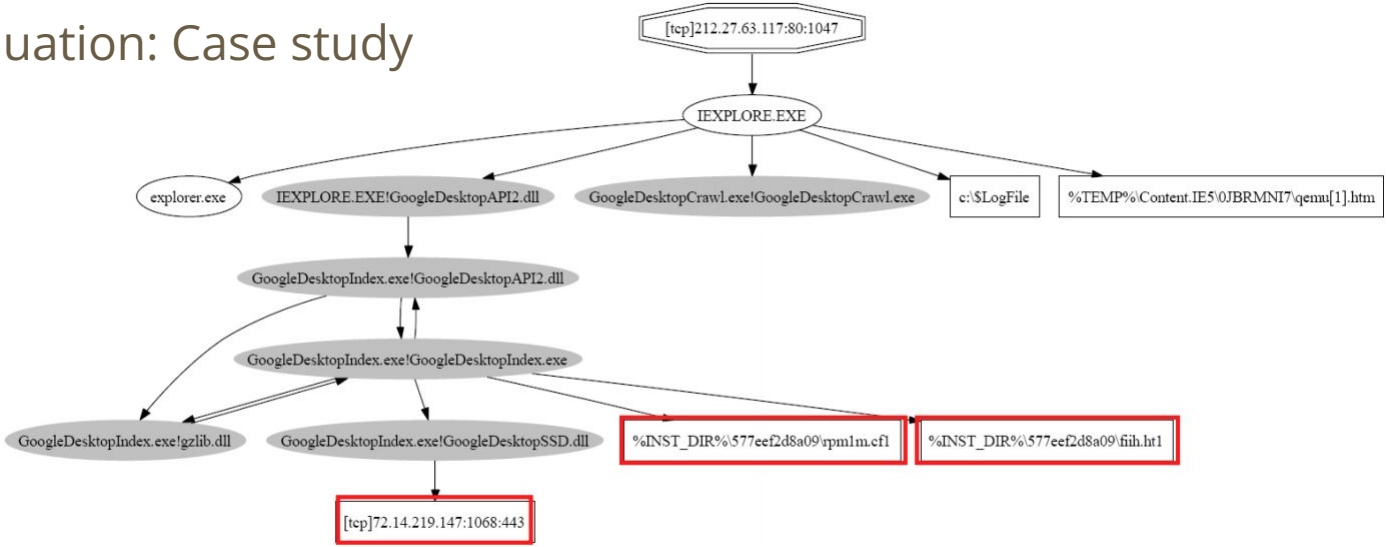
Category	Total	FNs	FPs
Keyloggers	5	0	-
Password thieves	2	0	-
Network sniffers	2	0	-
Stealth backdoors	3	0	-
Spyware/adware	22	0	-
Rootkits	8	0	-
Browser plugins	16	-	1
Multi-media	9	-	0
Security	10	-	2
System utilities	9	-	0
Office productivity	4	-	0
Games	4	-	0
Others	4	-	0
Sum	98	0	3

Browser accelerator

Personal firewall

Panorama

- Evaluation: Case study



Google Desktop obtains the incoming HTTP traffic, saves it into two index files, and then sends it out though an HTTPS connection, to a remote Google Server

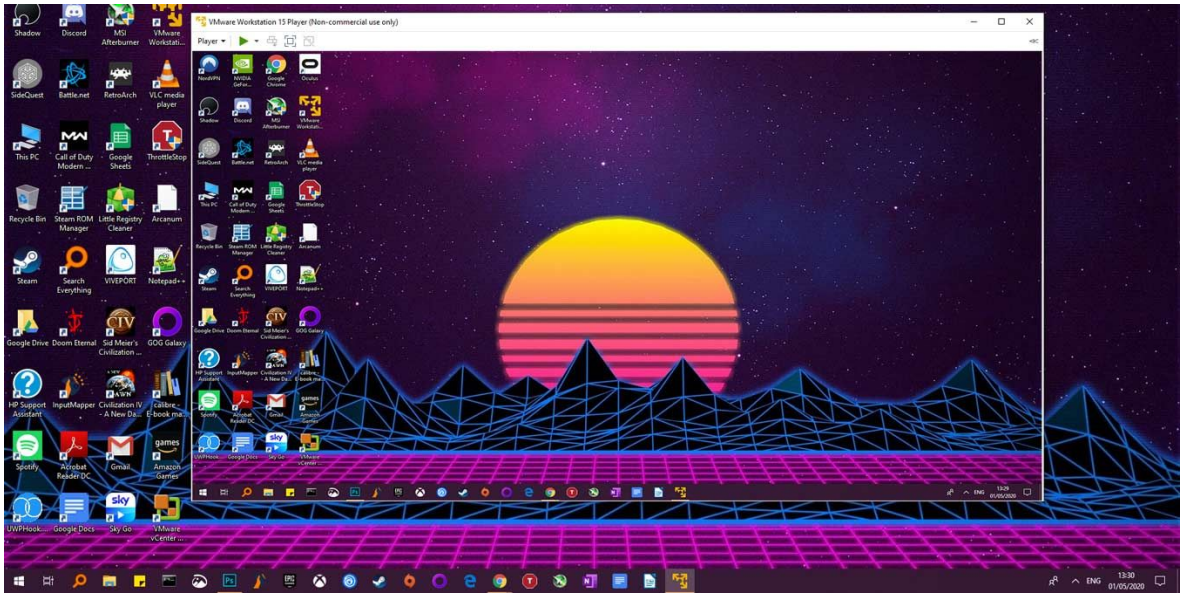
BareCloud: Bare-metal Analysis-based Evasive Malware Detection

Dhilung Kirat, Giovanni Vigna, and Christopher Kruegel

Usenix Security 2014

BareCloud

- VM environment is different from real machine.



BareCloud

- VM detection and evasion:
  - CPU instruction semantics
  - Timing attacks
  - VM bugs
  - Etc
- Motivation:
  - Can we automatically identify evasive malware while preserving transparency?
- Key idea:
  - Collect behavioral information on multiple platforms and compare the behaviors

BareCloud

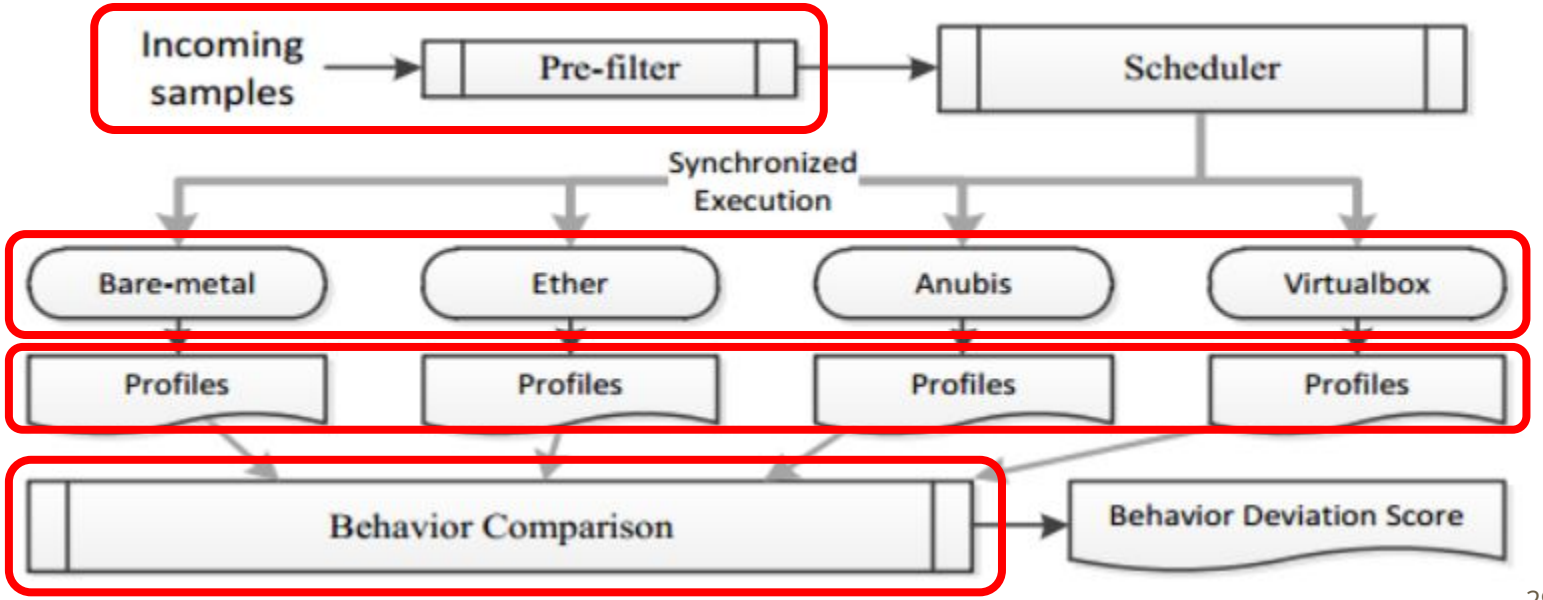
- System overview

prescreening

Execution

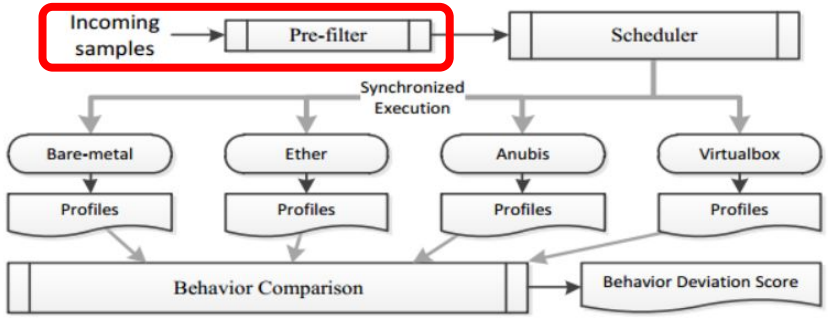
Behavior extraction

Behavior comparison



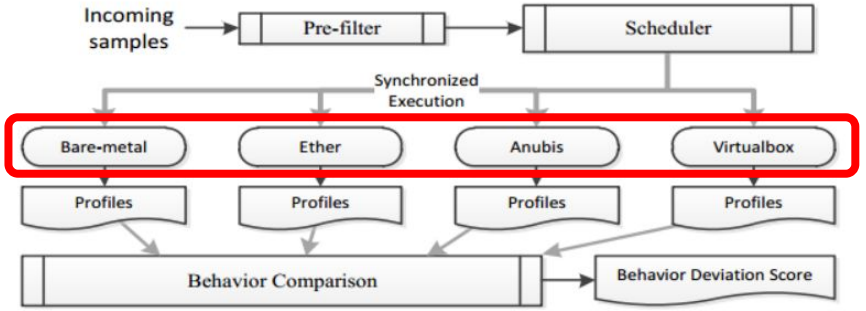
BareCloud

- Prescreening
  - Select interesting samples
    - Likely to have environment-sensitive behaviors
    - Use Anubis platform



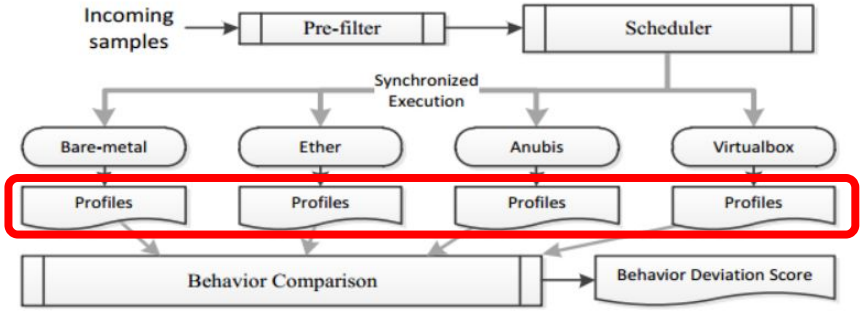
BareCloud

- Execution
  - Run malware sample on 4 platforms simultaneously
    - Bare-metal
    - Anubis (emulator)
    - Ether (Intel VT)
    - Virtualbox (Type2 hypervisor)



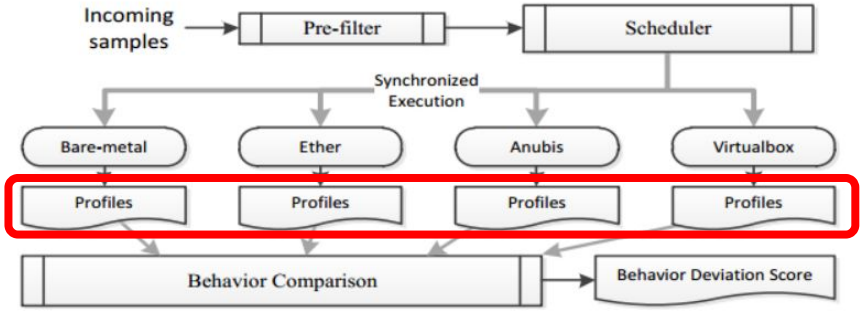
BareCloud

- Behavior extraction
  - Two common ways
    - VMI based approach
    - In-guest monitoring
  - Problem:
    - Not transparent enough



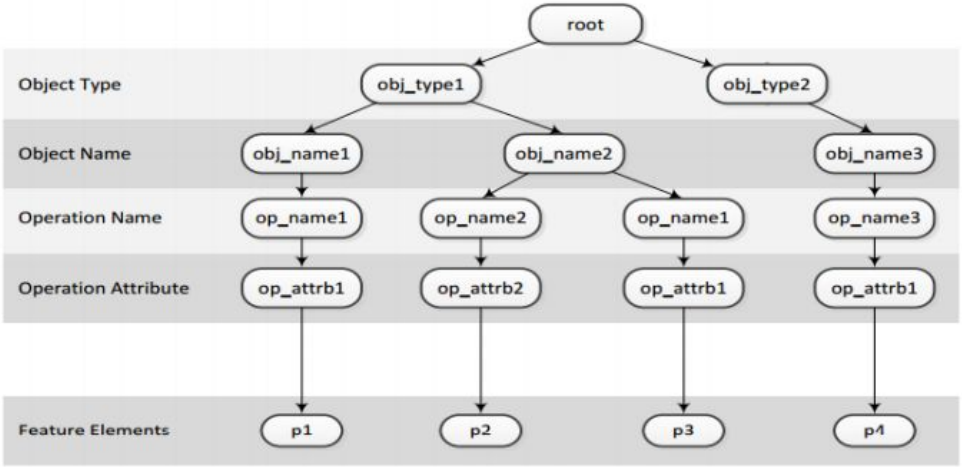
BareCloud

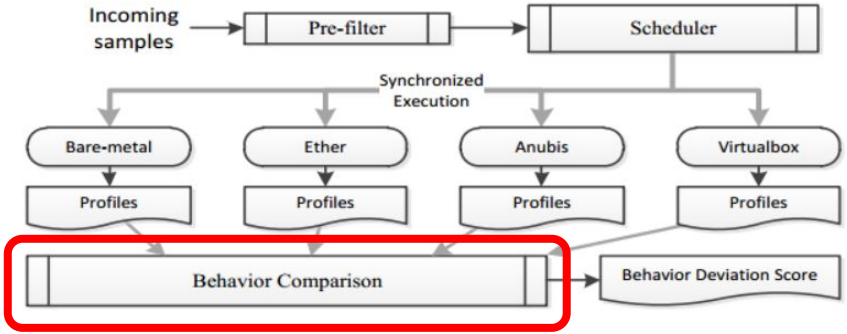
- Behavior extraction
  - BareCloud extracts file system behaviors and network behaviors
    - File system: compare the disk contents from before and after the malware execution
    - Network: use an external traffic capture component



BareCloud

- Behavior comparison
  - Compare the behavior profiles in a hierarchical way





BareCloud

- Evaluation
  - 110,000+ malware samples
  - 5835 evasive malware samples are found

Environment	Detection count	Percentage
Anubis	4,947	84.78
Ether	4,562	78.18
VirtualBox	3,576	61.28
All	2,530	43.35
Total	5,835	

Summary

- Malware detection and analysis
  - Static analysis
  - Dynamic analysis
- Semantics-Aware Malware Detection, *IEEE S&P 2005*
- Panorama: Capturing System-wide Information Flow for Malware Detection and Analysis, *ACM CCS 2007*
- BareCloud: Bare-metal Analysis-based Evasive Malware Detection, *Usenix Security 2015*

Thank you!

Question?