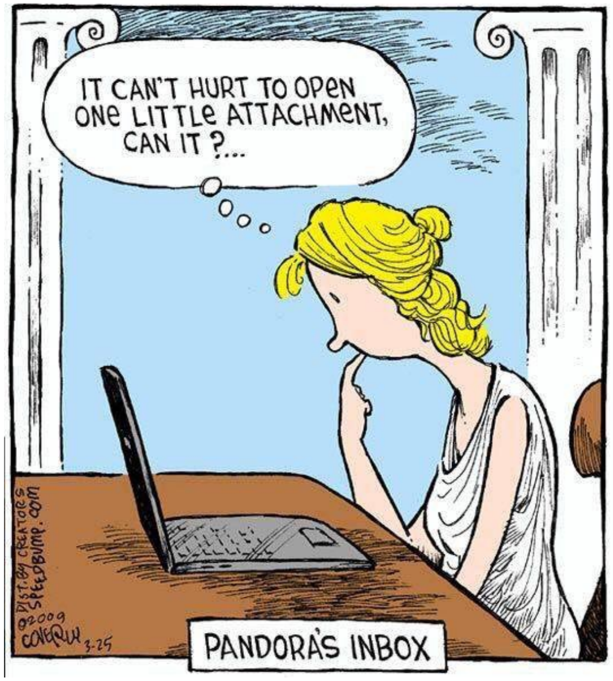
BareCloud: Bare-metal Analysis-based Evasive Malware Detection

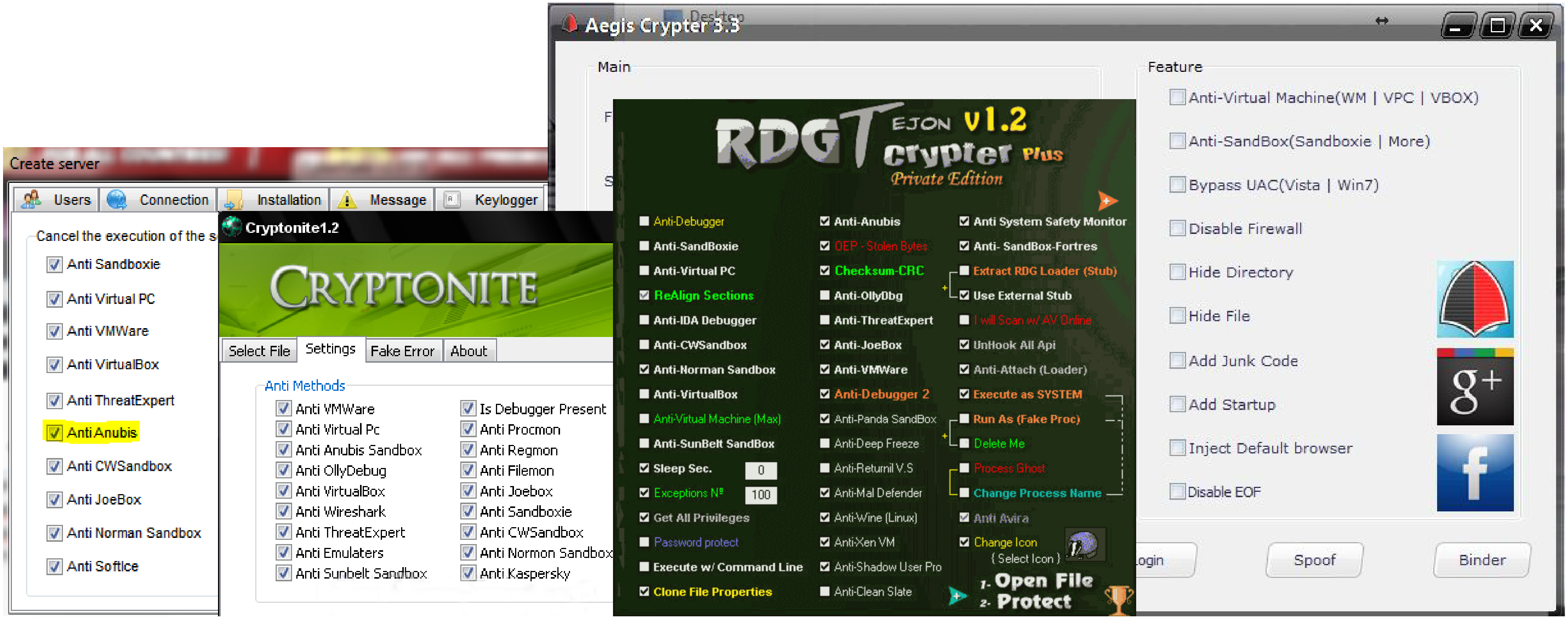
DHILUNG KIRAT, GIOVANNI VIGNA, AND CHRISTOPHER KRUEGEL, *UNIVERSITY OF CALIFORNIA, SANTA BARBARA*

PRESENTED BY: KEVIN COIA

CS563 – FALL 2018


IT CAN'T HURT TO OPEN ONE LITTLE ATTACHMENT, CAN IT?...
PANDORA'S INBOX
©2009 DIST. BY CREATORS SPEEDBUMP.COM
COVERLY 3-25

Evasive Malware



Create server

Users | Connection | Installation | Message | Keylogger

Cancel the execution of the s

☑ Anti Sandboxie
☑ Anti Virtual PC
☑ Anti VMWare
☑ Anti VirtualBox
☑ Anti ThreatExpert
☑ Anti Anubis
☑ Anti CWSandbox
☑ Anti JoeBox
☑ Anti Norman Sandbox
☑ Anti SoftIce

Cryptonite1.2

CRYPTONITE

Select File | Settings | Fake Error | About

Anti Methods

☑ Anti VMWare
☑ Anti Virtual Pc
☑ Anti Anubis Sandbox
☑ Anti OllyDebug
☑ Anti VirtualBox
☑ Anti Wireshark
☑ Anti ThreatExpert
☑ Anti Emulaters
☑ Anti Sunbelt Sandbox

☑ Is Debugger Present
☑ Anti Procmon
☑ Anti Regmon
☑ Anti Filemon
☑ Anti Joebox
☑ Anti Sandboxie
☑ Anti CWSandbox
☑ Anti Normon Sandbox
☑ Anti Kaspersky

Aegis Crypter 3.3

Main

Feature

□ Anti-Virtual Machine(WM | VPC | VBOX)
□ Anti-SandBox(Sandboxie | More)
□ Bypass UAC(Vista | Win7)
□ Disable Firewall
□ Hide Directory
□ Hide File
□ Add Junk Code
□ Add Startup
□ Inject Default browser
□ Disable EOF

ogin | Spoof | Binder

RDG TEJON v1.2 Crypter Plus
Private Edition

□ Anti-Debugger
□ Anti-SandBoxie
□ Anti-Virtual PC
☑ ReAlign Sections
□ Anti-IDA Debugger
□ Anti-CWSandbox
☑ Anti-Norman Sandbox
□ Anti-VirtualBox
□ Anti-Virtual Machine (Max)
□ Anti-SunBelt SandBox
☑ Sleep Sec. 0
☑ Exceptions Nº 100
☑ Get All Privileges
□ Password protect
□ Execute w/ Command Line
☑ Clone File Properties

☑ Anti-Anubis
☑ OEP - Stolen Bytes
☑ Checksum-CRC
□ Anti-OllyDbg
□ Anti-ThreatExpert
☑ Anti-JoeBox
☑ Anti-VMWare
☑ Anti-Debugger 2
☑ Anti-Panda SandBox
□ Anti-Deep Freeze
□ Anti-Returnil V.S
☑ Anti-Mal Defender
☑ Anti-Wine (Linux)
☑ Anti-Xen VM
☑ Anti-Shadow User Pro
□ Anti-Clean Slate

☑ Anti System Safety Monitor
☑ Anti- SandBox-Fortres
□ Extract RDG Loader (Stub)
☑ Use External Stub
□ I will Scan w/ AV Online
☑ UnHook All Api
☑ Anti-Attach (Loader)
☑ Execute as SYSTEM
□ Run As (Fake Proc)
□ Delete Me
□ Process Ghost
□ Change Process Name
☑ Anti Avira
☑ Change Icon
{ Select Icon }

1- Open File
2- Protect

Approach presented in paper

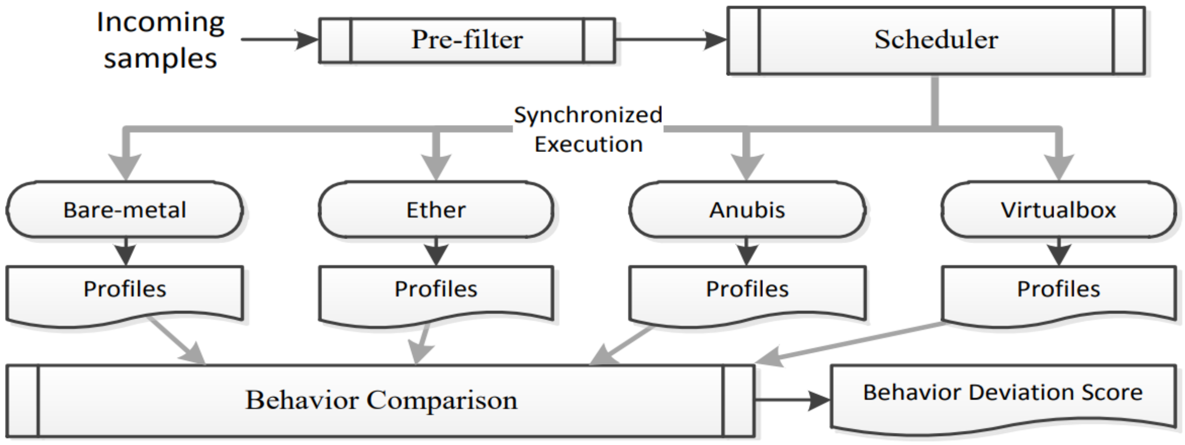
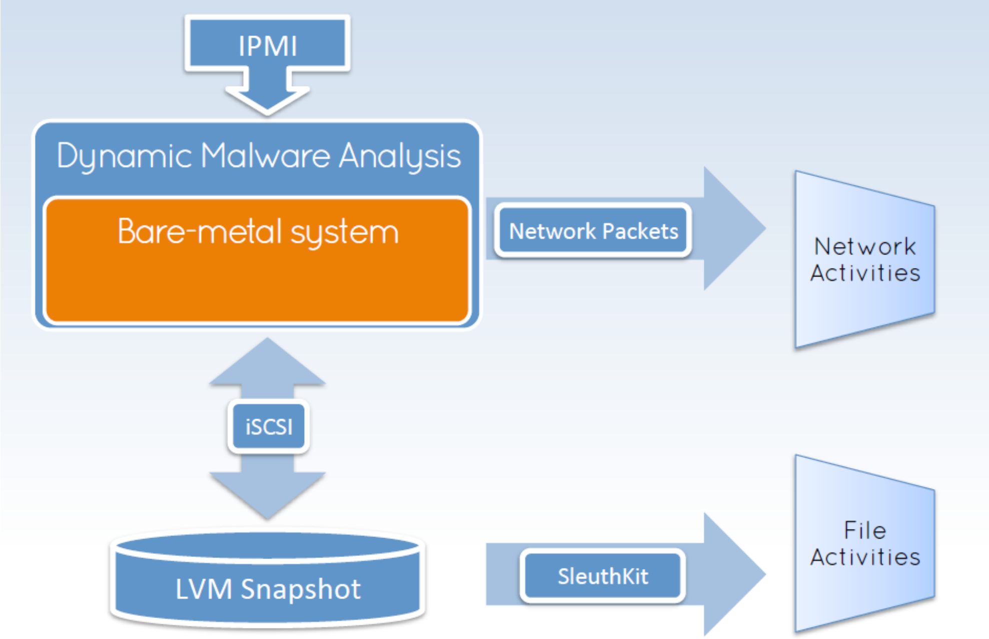


Figure 1: Overview of the system

High level approach:

- Identify interesting malware samples that they would like to run in the BareCloud system.
- Run the chosen malware sample at the same time with identical setups in several different environments to collect the behavioral profiles for each environment. The environments used were as follows:
  1. Bare-metal
  2. Ether (Xen hypervisor based analysis environment)
  3. Anubis (Emulated environment based on Qemu)
  4. VirtualBox (Type 2 hypervisor)
- Compare behavioral profiles to identify how similar each of the Virtualized / Emulated environments compared to the Bare-metal's profile, and if this difference is above a threshold classify it as Evasive Malware

Behavioral Profile Data Collection



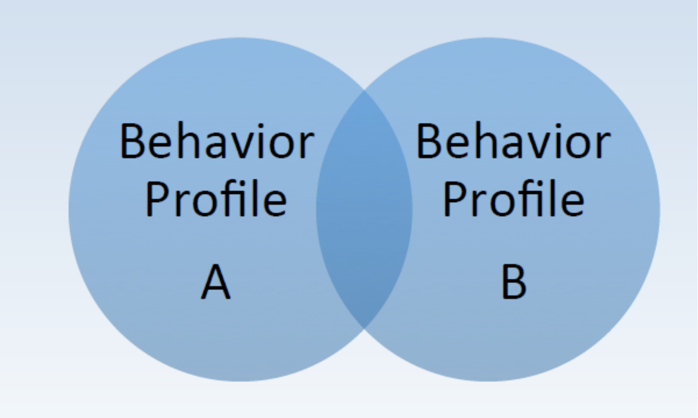
Behavioral Profile Data Collection

Will this data collection be sufficient to build an accurate profile of the malwares behavior?

Does this approach even prevent an adversary from fingerprinting the Bare metal system to detect that its being ran in this monitored environment?

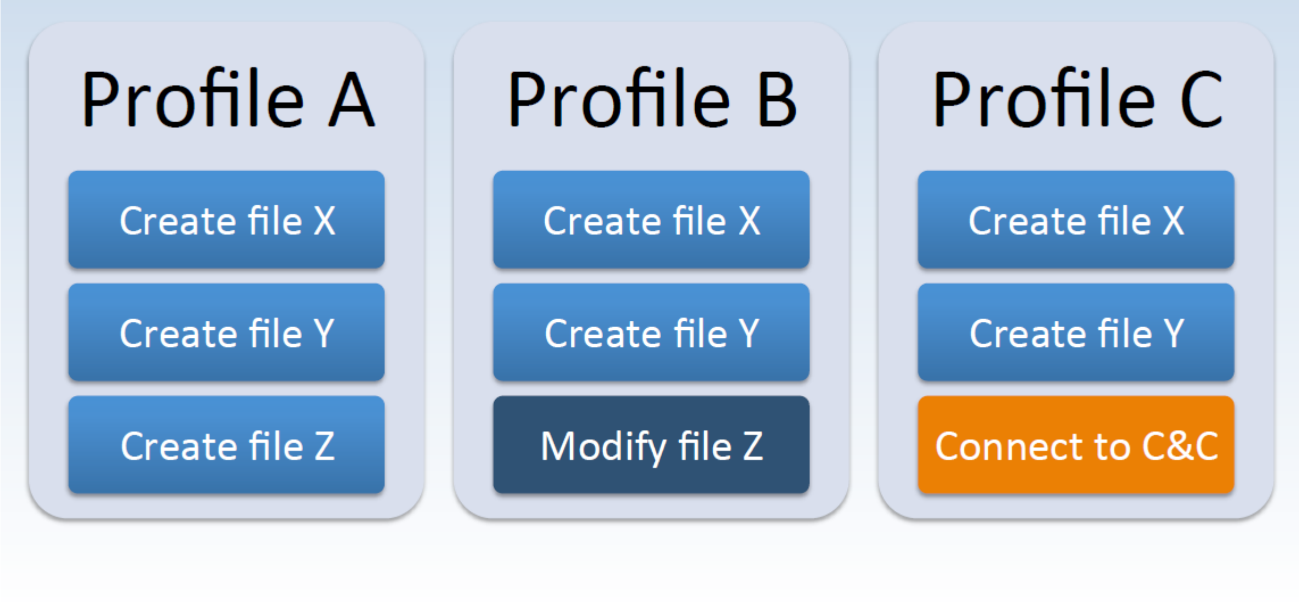
Can an adversary hide their persistent file operations in a way to look identical to the normal background file operations of the running OS?

Behavioral Profile Comparison



$$JaccardSimilarity = \frac{A \cap B}{A \cup B}$$

Behavioral Profile Comparison



Profile A	Profile B	Profile C
Create file X	Create file X	Create file X
Create file Y	Create file Y	Create file Y
Create file Z	Modify file Z	Connect to C&C

$$JaccardSimilarity(A, B) = 2/4 = JaccardSimilarity(A, C)$$

Hierarchical similarity

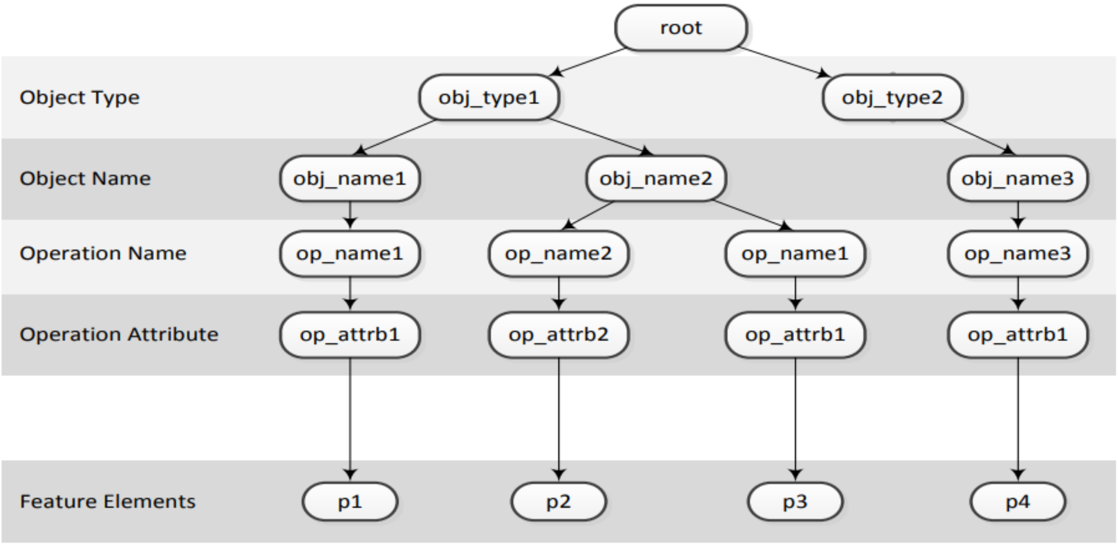
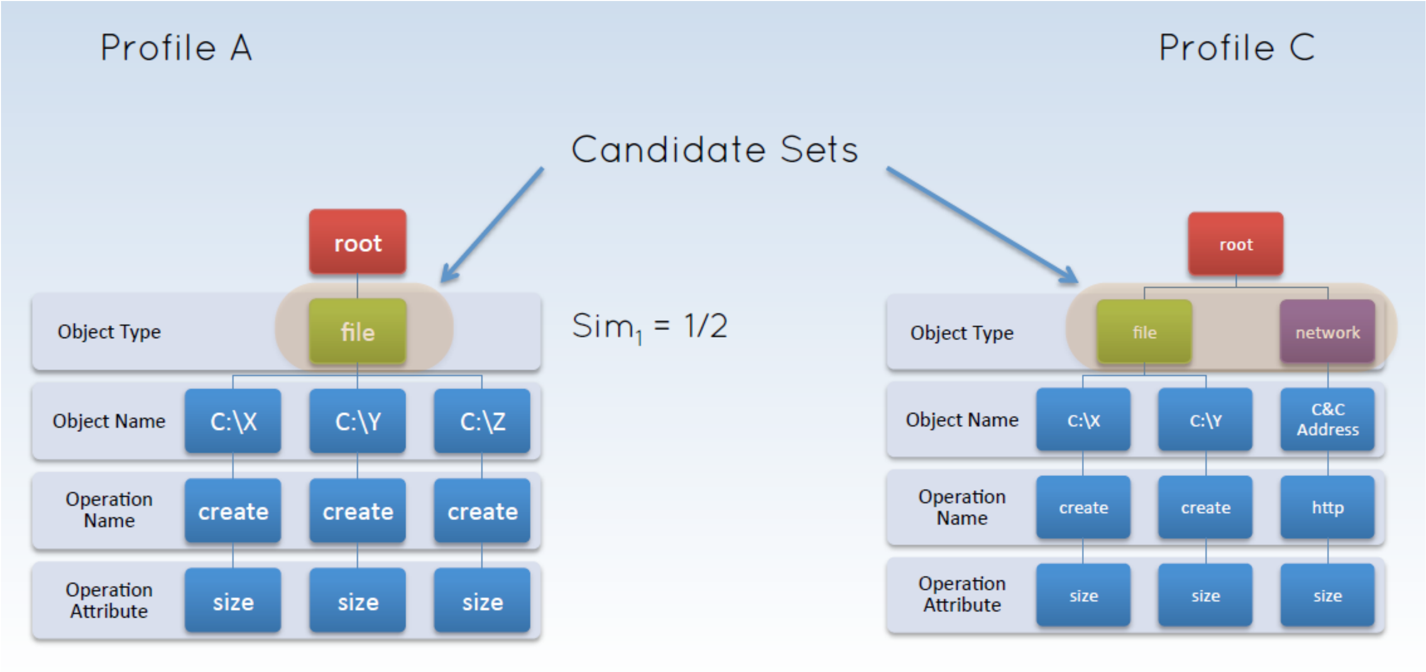
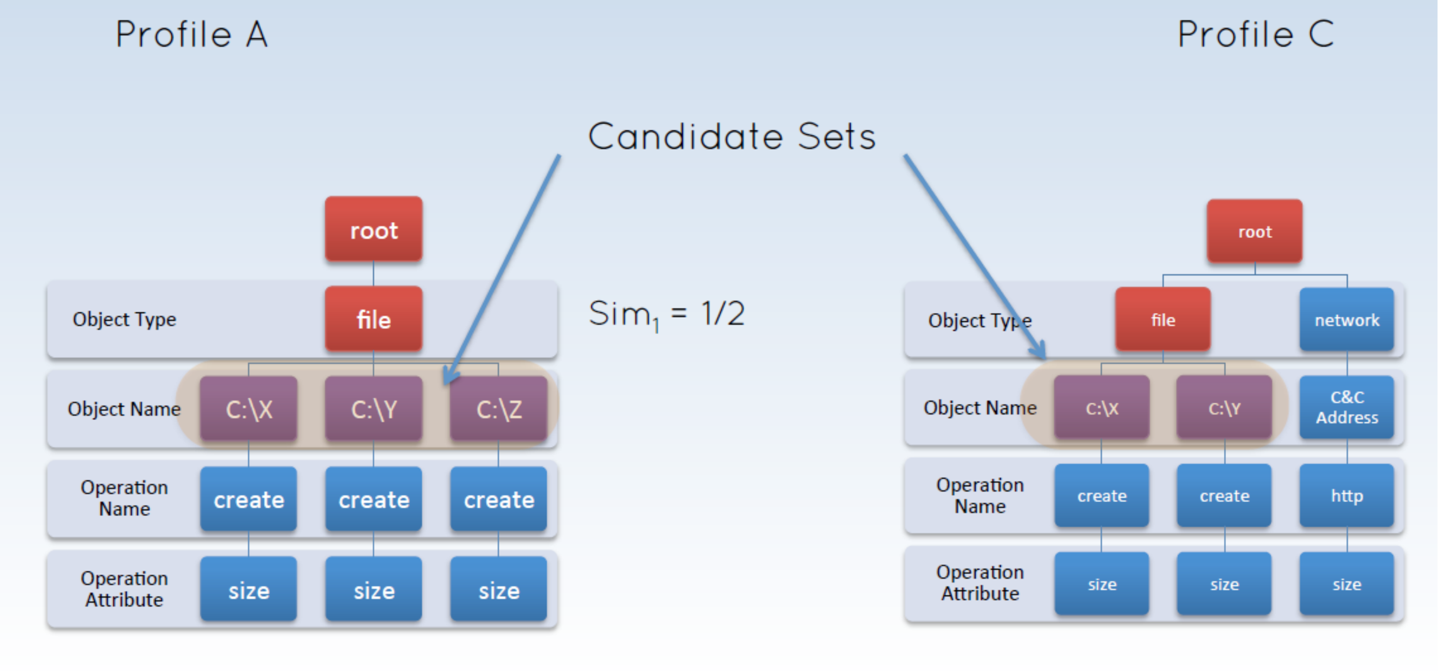


Figure 2: Behavior similarity hierarchy

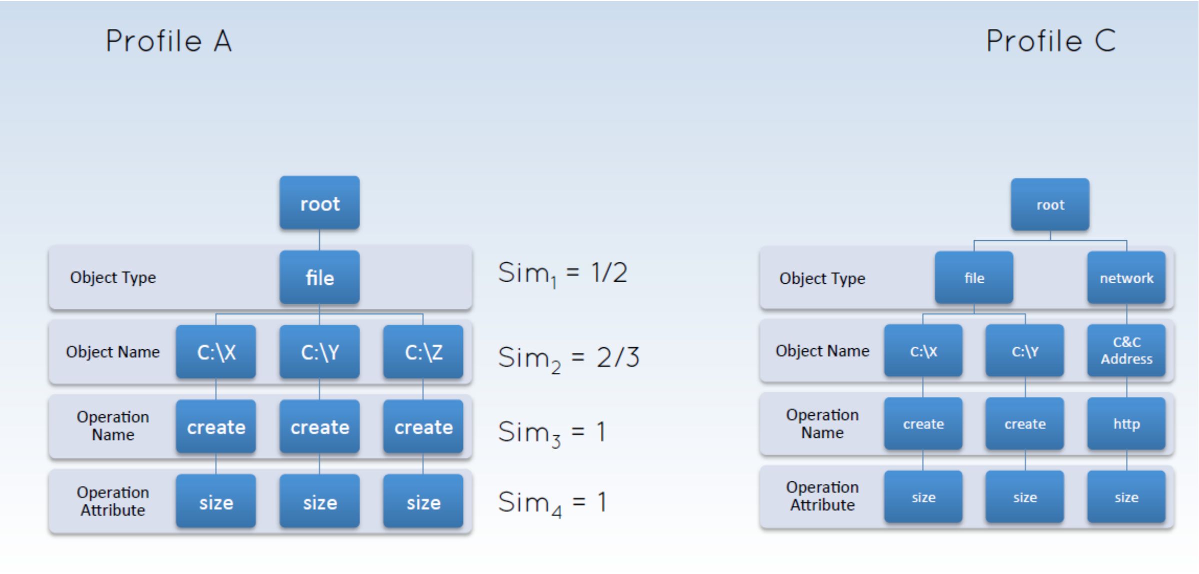
Hierarchical similarity



Hierarchical similarity

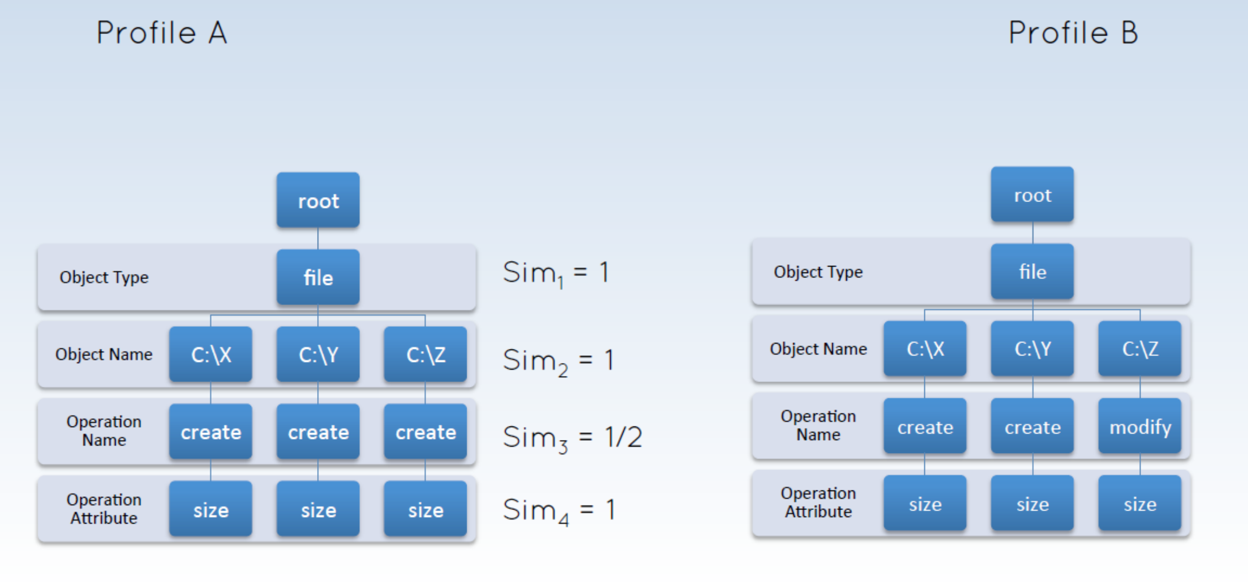


Hierarchical similarity



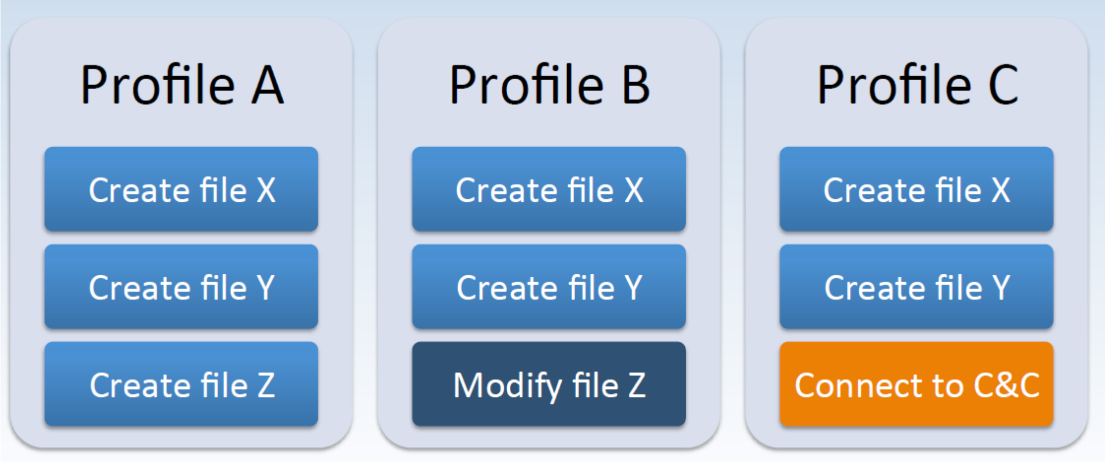
$$Sim(A, C) = AVG(Sim_1 \ldots Sim_4) = 0.79$$

Hierarchical similarity



$$Sim(A, B) = AVG(Sim_1 \ldots Sim_4) = 0.87$$

Profile Similarity Comparison



Profile A

Create file X

Create file Y

Create file Z

Profile B

Create file X

Create file Y

Modify file Z

Profile C

Create file X

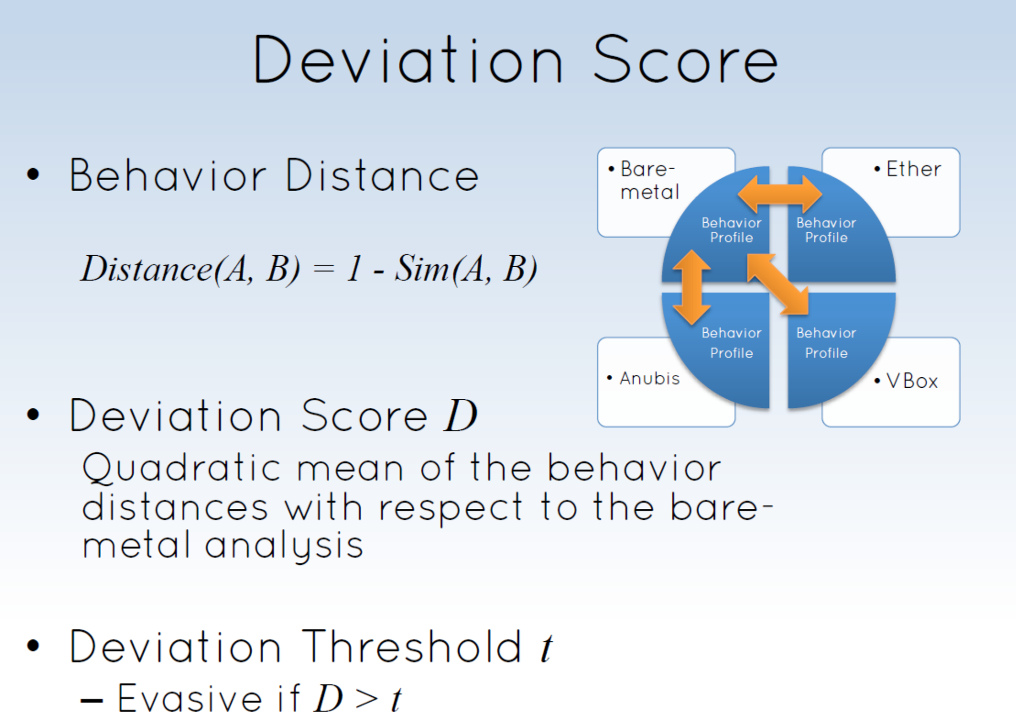
Create file Y

Connect to C&C

$$\mathit{JaccardSimilarity}(A, B) == \mathit{JaccardSimilarity}(A, C)$$

$$\begin{array}{ccc} \mathit{HierarchicalSim}(A, B) & > & \mathit{HierarchicalSim}(A, C) \\ 0.87 & > & 0.79 \end{array}$$

Scoring Deviation from Bare-Metal behavior



Deviation Score

- Behavior Distance

  Distance(A, B) = 1 - Sim(A, B)

- Deviation Score D

  Quadratic mean of the behavior distances with respect to the bare-metal analysis

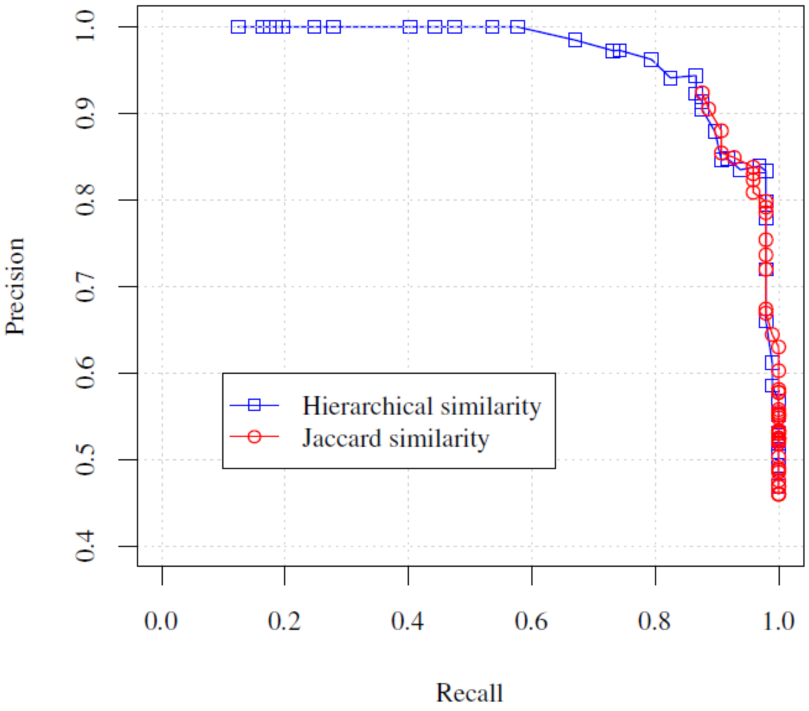
- Deviation Threshold t
  - Evasive if $D > t$

• Bare-metal
• Ether
Behavior Profile
Behavior Profile
Behavior Profile
Behavior Profile
• Anubis
• VBox

Hierarchical vs Jaccard similarity

Using a sample of 111 evasive and 119 non-evasive samples

Ultimately led to them concluding that their Hierarchical similarity method is better at quantizing the similarity between two behavioral profiles

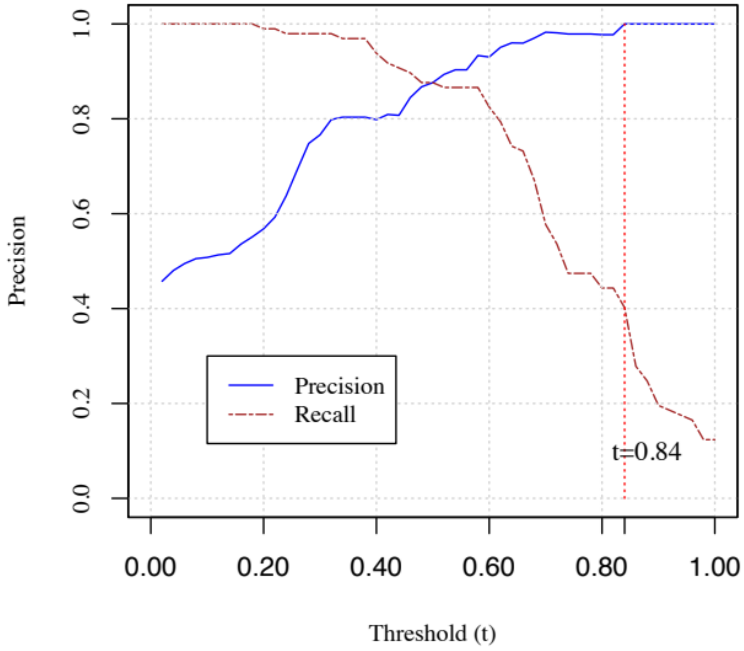


Hierarchical similarity

Is Hierarchical similarity a good approach to quantizing how similar two program behaviors were from their behavior profile?

Could this approach be improved further to better handle similar operations, for example creating a randomly named temp file?

Threshold Determination



Large-scale Evaluation

Ran the BareCloud system on 110,005 samples during a 4 months period starting in July 2013

Had the following selection criteria to decide which samples to test their system with:

- Low system and low network activity
- High system and high network activity
- High system but low network activity
- Low system but high network activity

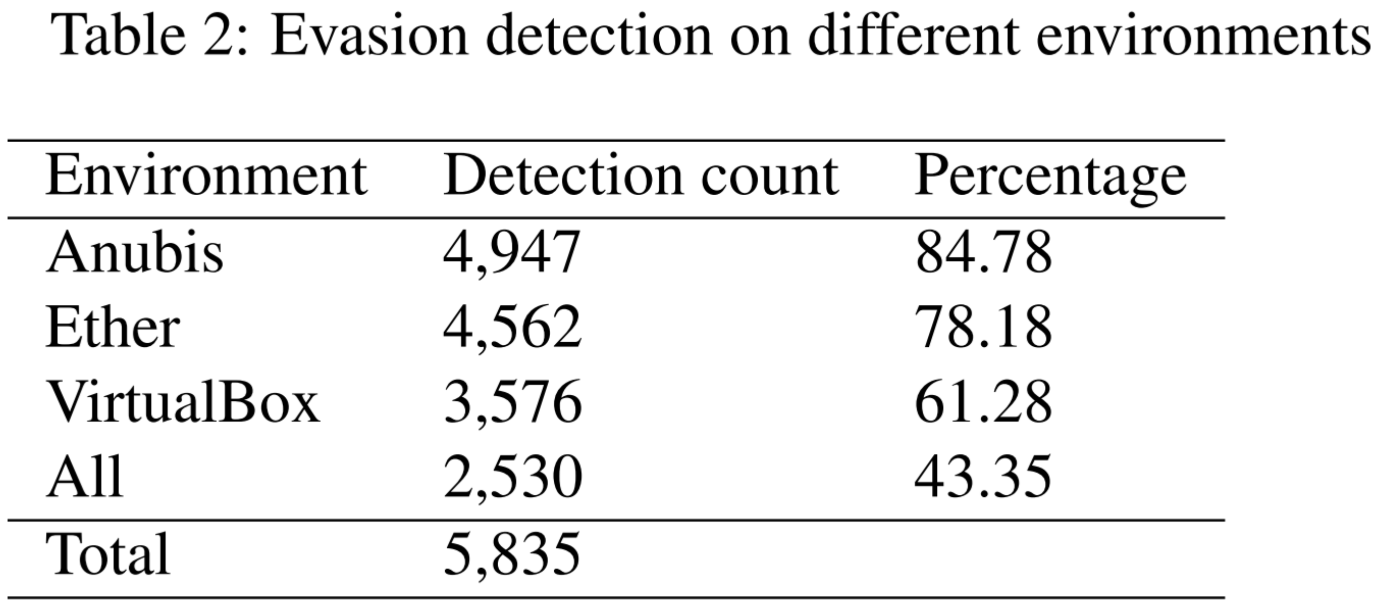


Table 2: Evasion detection on different environments

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	

What can we actually pull from this?

- Did they check that these detected evasive malwares, truly were evasive malware? Was there any misidentification?
- With 110,005 samples, what does this mean if anything since there is no ground truth for this data set?
- What would happen if you applied this to non malware? Can we verify that normal applications (or simply normal non-evasive malware) wouldn’t be mis-identified as “Evasive Malware”?

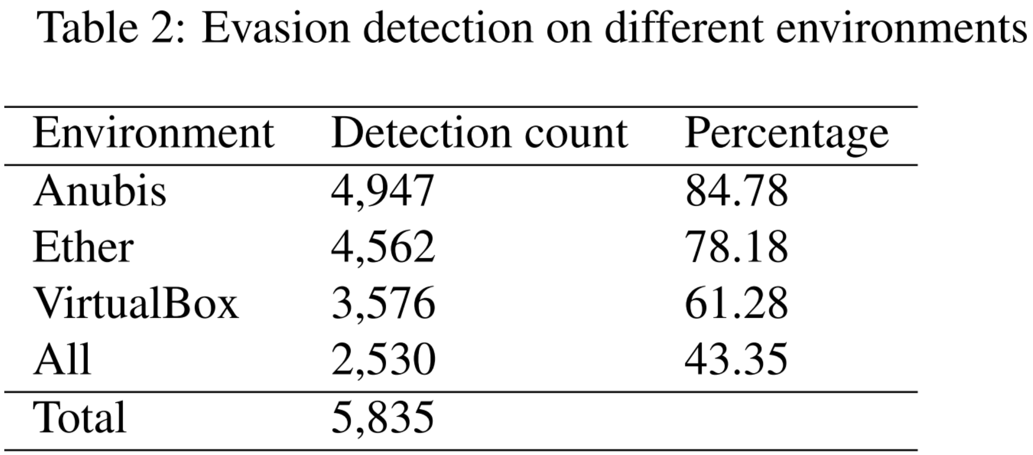


Table 2: Evasion detection on different environments

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	

Questions