AVLeak:
Fingerprinting Antivirus Emulators Through Black-Box Testing

Jeremy Blackthorne, Alexei Bulazel, Andrew Fasano, Patrick Biernat, Bülent Yener

Alexei Bulazel and Andrew Fasano

@av_leak

Introduction

- Research group from Rensselaer Polytechnic Institute (RPI) under Dr. Bülent Yener

- Jeremy Blackthorne - PhD candidate
- Alexei Bulazel - recent MS graduate
- Andrew Fasano - undergraduate researcher (graduated)
- Patrick Biernat - undergraduate researcher
- Dr. Bülent Yener - advisor

Outline

Problem

- Modern AV software uses dynamic ("sandbox") analysis to scan the 1,000,000+ new malware binaries created every day
- Consumer AV emulators are *conceptually* easy to evade
- If emulation can be detected, malware can behave benignly to avoid detection
- There is not an efficient method to "fingerprint" consumer AV emulators

Motivation

- Existing methods to extract fingerprints from emulators are inefficient:
  - Reverse engineering
    - Too hard
  - Black-box dynamic analysis
    - Too slow

- Our goal: Automate and accelerate fingerprint discovery

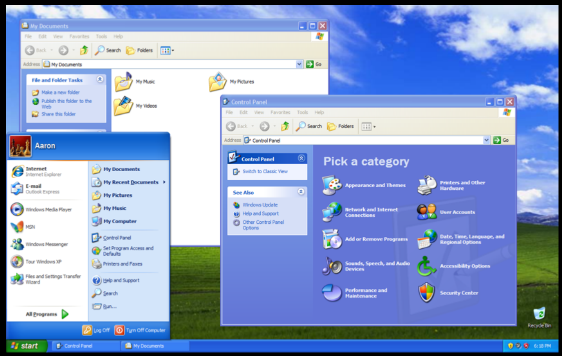
Outline

1. Introduction
2. Problem & Motivation
3. Background & Prior Work
4. AVLeak
5. Results & Demo
6. Conclusions

Background

- Packers can generate millions of unique binaries that behave identically while evading static signatures
- Dynamic (sandbox) analysis allows AV engines to identify known signatures or heuristically classify previously unknown malware
- Extensive prior research on detecting high-end emulators and VMs - QEMU, VMWare, Xen, Bochs, etc
- Little prior work on consumer AV emulators

Classes of *Consumer AV* Fingerprints



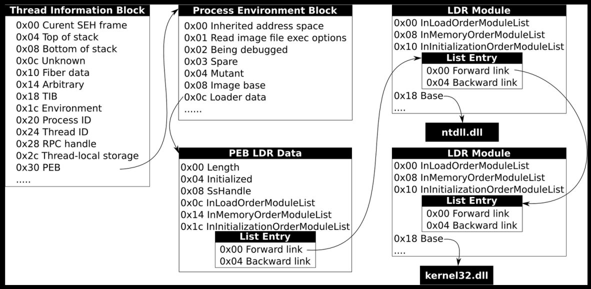
- Environmental artifacts
  - Hardcoded username, registry entries, processes names

- OS API inconsistency
  - Failures and incorrect return values
- Network emulation
  - Hardcoded responses and inconsistencies

Classes of *Consumer AV* Fingerprints

- Timing
  - Timing skews and dilation

- Process Introspection
  - Internal inconsistencies - PEB, heap allocations, etc

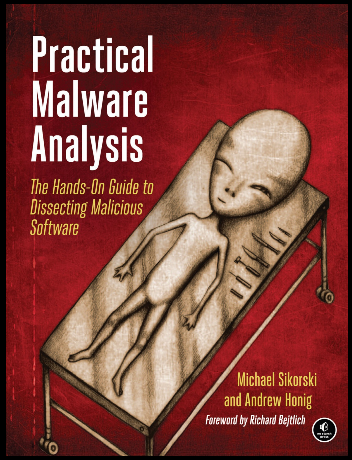


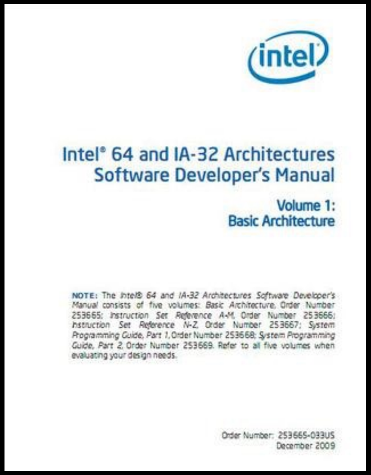
- CPU “Red Pills”
  - Instructions which behave differently on an emulated CPU

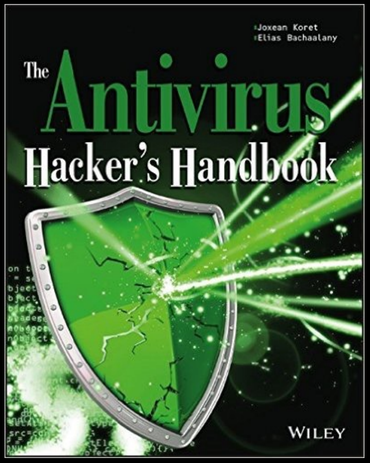
Reversing AV Emulators

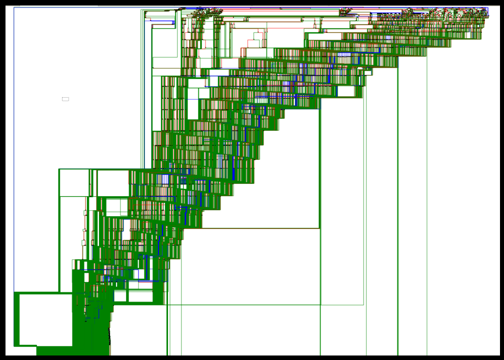
- Time consuming
- Expensive tools
- Expert knowledge
  – RE, AV, x86, Windows internals, malware behavior, anti-analysis
- Limited Lifespan - frequent updates

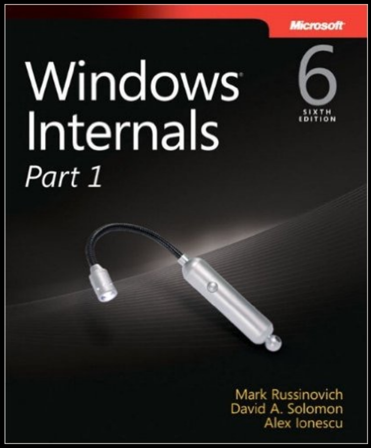
Line 20 of 13208

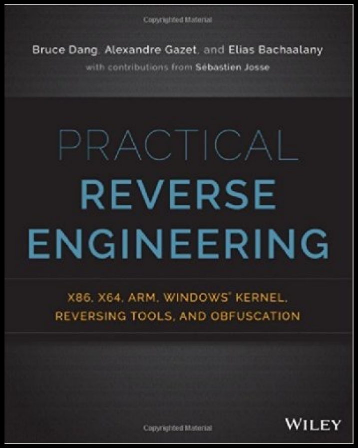




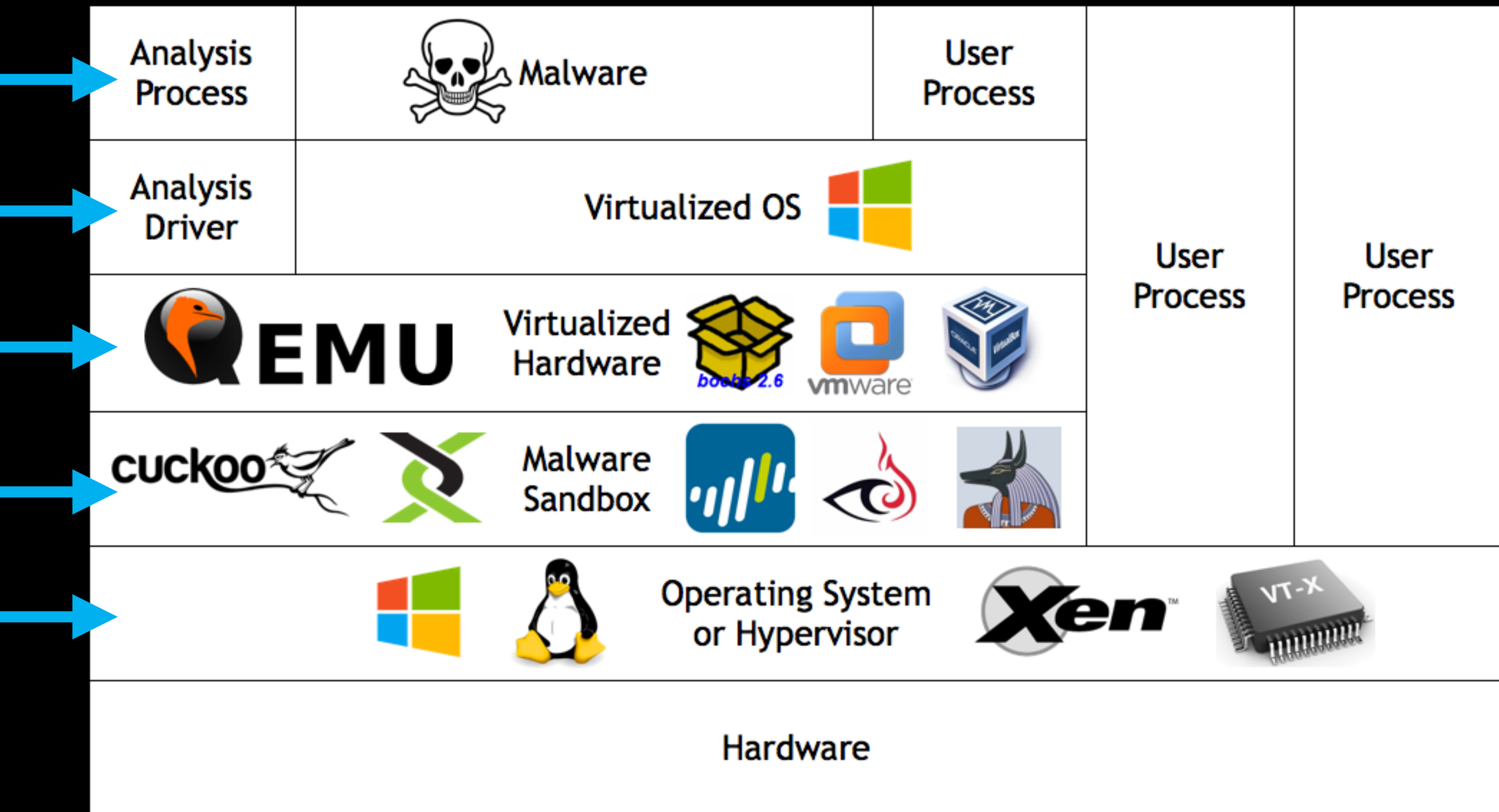






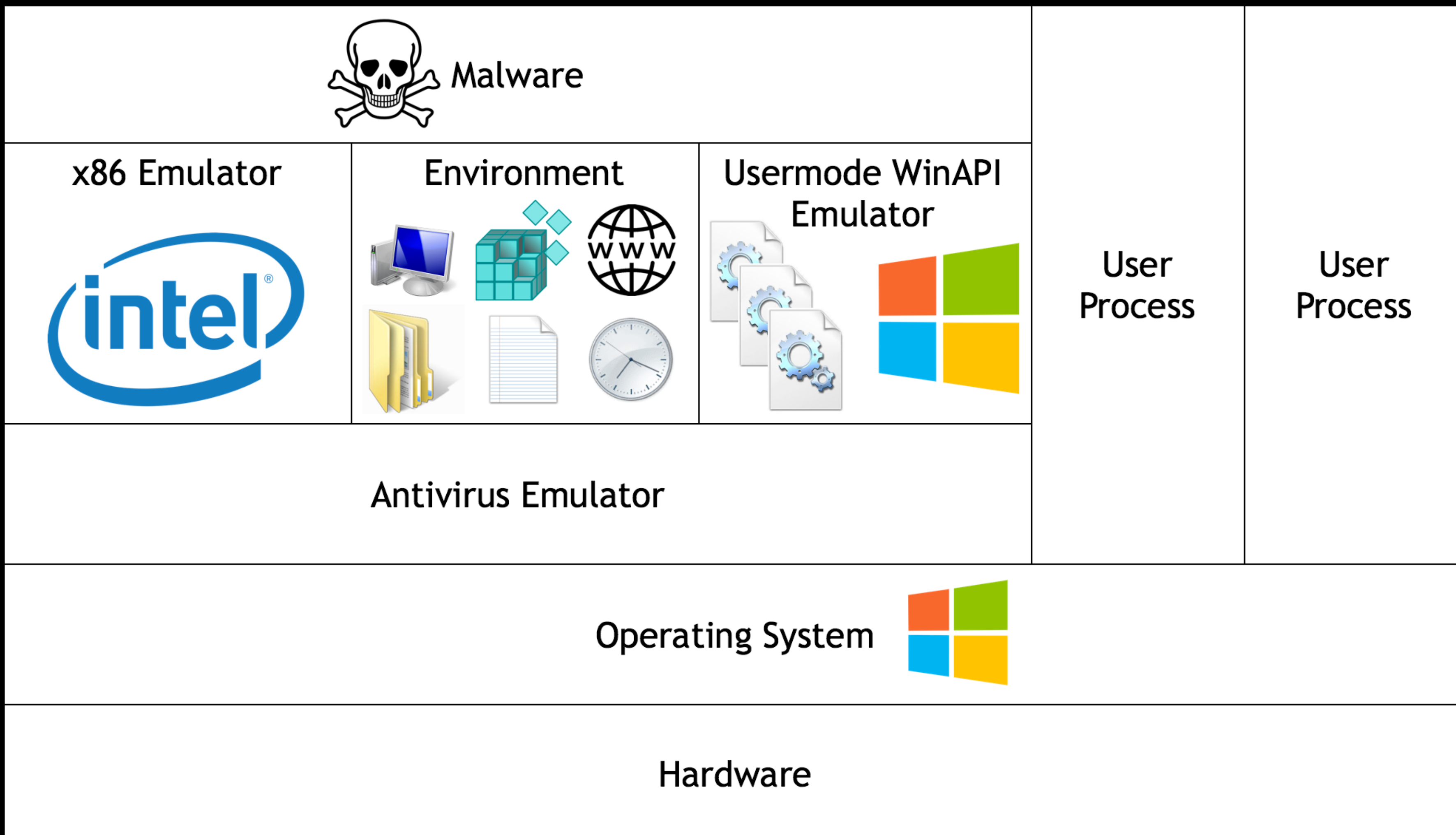


Traditional Malware Sandbox / Emulator Architecture



Many introspection points for fingerprint extraction

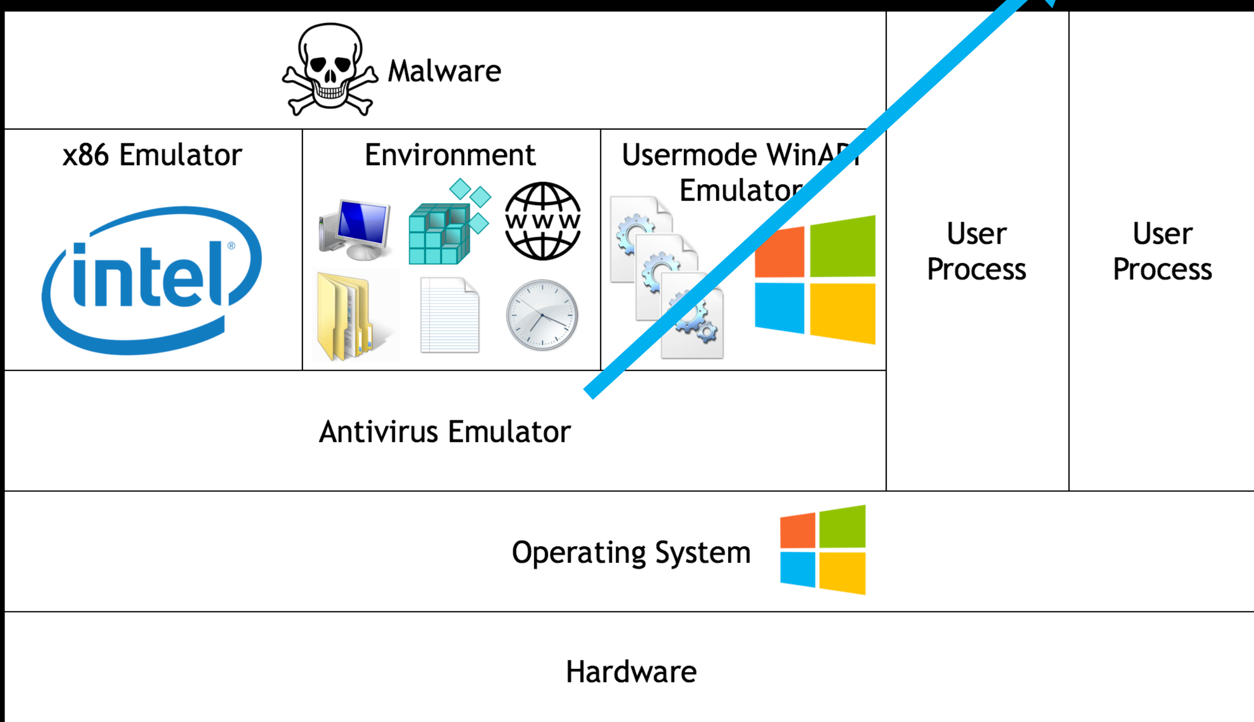
Consumer AV Emulator



Consumer AV Emulator

Single introspection point: analysis report for given input binary

```
Analysis report:
Dropped: Trojan.Infector.BAT.ABC123
Dropped: APT1337.Backdoor.2
Dropped: CryptoLocker.Downloader.K
```



Prior Approach: Black Box Testing

- Extract a single bit of data per run
  - Arne Swinnen & Alaeddine Mesbahi - One Packer To Rule Them All (Black Hat ’14)
  - Kyle Adams - Evading Code Emulation (BSidesLV ’14)
  - Daniel Sauder - Why Antivirus Software Fails (DeepSec ’14)
  - Emeric Nasi - Bypass Antivirus Dynamic Analysis (white paper ‘14)

Prior Approaches: Black Box Testing

True or False Question: Does the emulator emulate function_x() correctly?

AV Emulator

Prior Approaches: Black Box Testing

True or False Question: Does the emulator emulate function_x() correctly?

```
if function_x() != EXPECTED:
    DropMalware()
else:
    Exit()
```

Malware	TRUE	
No Malware	FALSE	

AV Emulator

Prior Approaches: Black Box Testing

True or False Question: Does the emulator emulate function_x() correctly?

```
if function_x() != EXPECTED:
    DropMalware()
else:
    Exit()
```

Malware	TRUE	
No Malware	FALSE	

AV Emulator

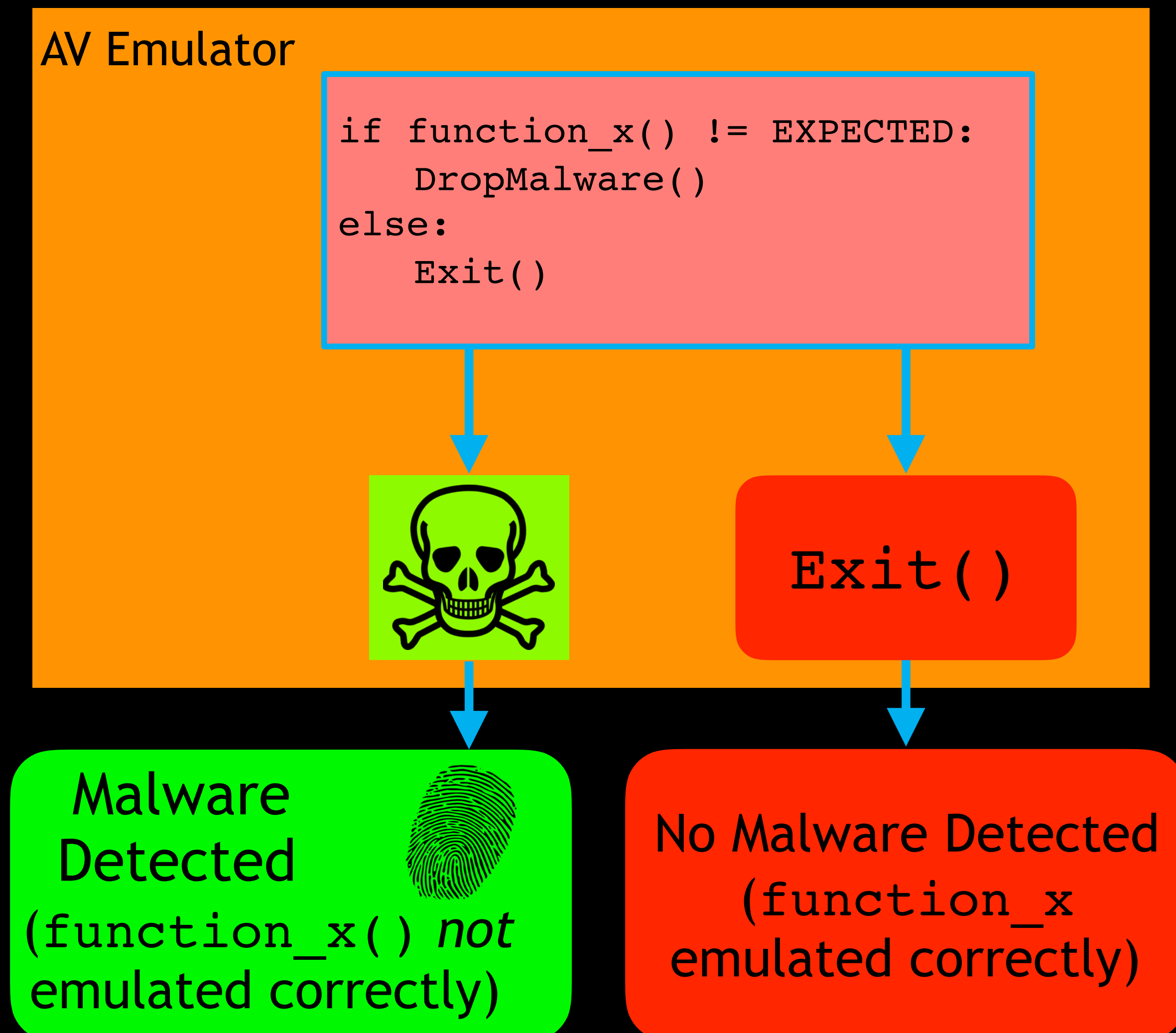
```
if function_x() != EXPECTED:
    DropMalware()
else:
    Exit()
```

Prior Approaches: Black Box Testing

True or False Question: Does the emulator emulate function_x() correctly?

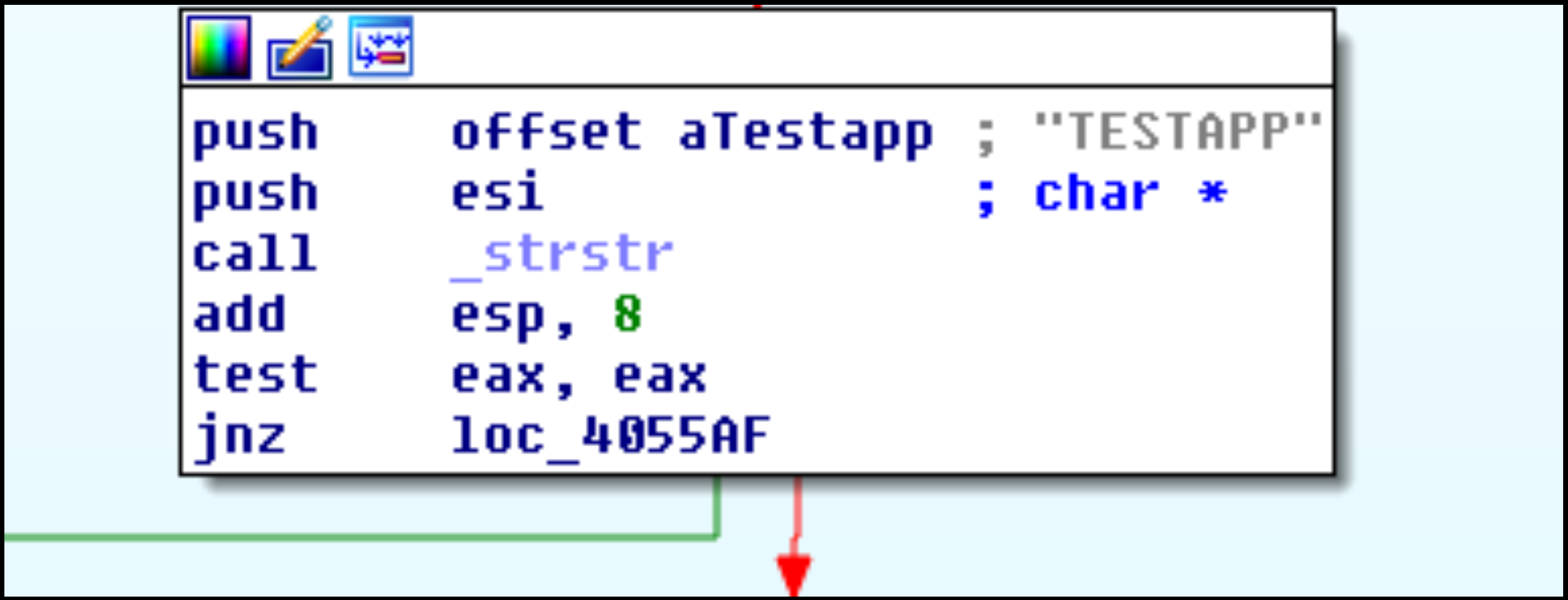
```
if function_x() != EXPECTED:
    DropMalware()
else:
    Exit()
```

Malware	TRUE	
No Malware	FALSE	



Evasive Malware: Case Study

- EvilBunny (Animal Farm APT) was using fingerprints to evade Bitdefender in 2011
- Bitdefender calls processes under analysis "TESTAPP"



EvilBunny doesn't run when when called "TESTAPP"

Outline

1. Introduction
2. Problem & Motivation
3. Background & Prior Work
4. AVLeak
5. Results & Demo
6. Conclusions

Introducing AVLeak

- Novel tool for researchers to easily and quickly extract fingerprints from consumer antivirus emulators in order to evade malware detection

- Design: Test cases in C, automated with Python, Python API

- Goals:
  - Fingerprint the AV itself
  - Ease of use
  - Abstract AV interaction from the programmer
  - Scriptable API
  - Find fingerprints in seconds not hours

Introducing AVLeak

- Novel approach to leak bytes values from inside AV emulators
- Map malware names to byte values
- Use malware detections to exfiltrate *specific* byte values per run

Virus Database	
A	Morris
B	Code Red
C	Zeus
…	
a	Conficker
…	
z	Brain

AVLeak's Innovation

Question: What is the username in the emulator?

AV Emulator

`username="emu"`

AVLeak's Innovation

Question: What is the username in the emulator?

GetUserName()	
A	Morris
B	Code Red
C	Zeus
…	
a	Conficker
…	
z	Brain

AV Emulator

username="emu"

AVLeak's Innovation

Question: What is the username in the emulator?

GetUserName()	
A	Morris
B	Code Red
C	Zeus
…	
a	Conficker
…	
z	Brain

AV Emulator

username="emu"

```
for c in GetUserName():
    Drop(MalwareArray[c])
```

AVLeak’s Innovation

Question: What is the username in the emulator?

GetUserName()	
A	Morris
B	Code Red
C	Zeus
…	
a	Conficker
…	
z	Brain

AV Emulator

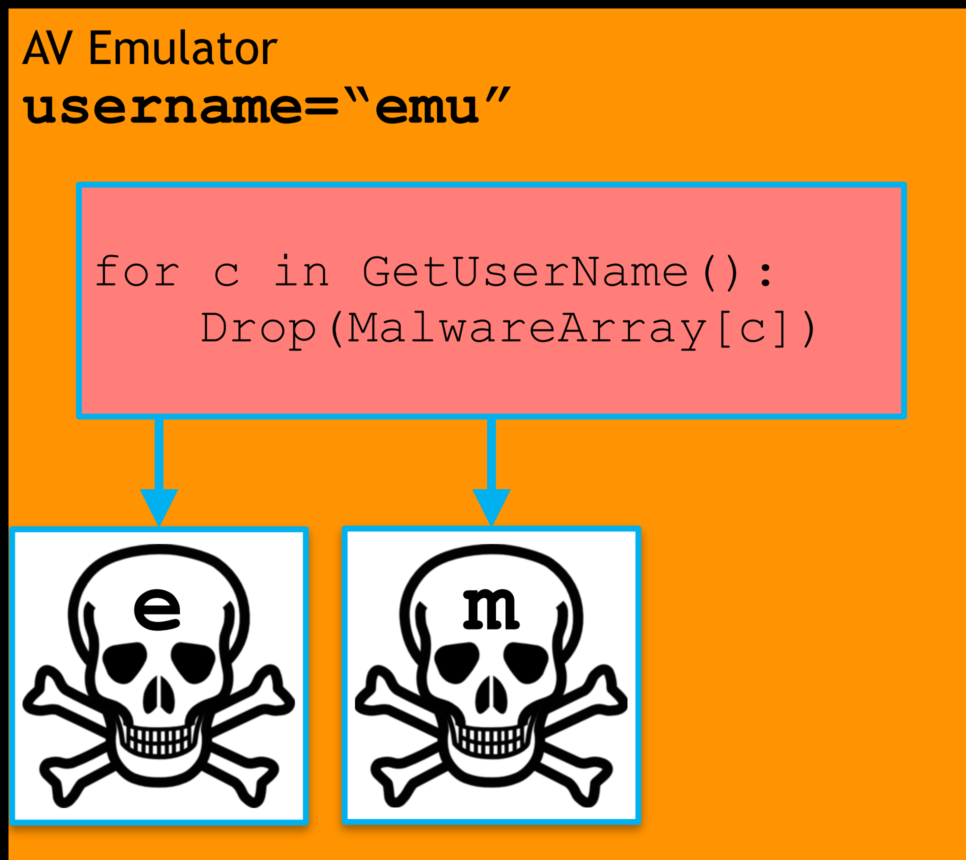
username=“emu”

```
for c in GetUserName():
    Drop(MalwareArray[c])
```

AVLeak's Innovation

Question: What is the username in the emulator?

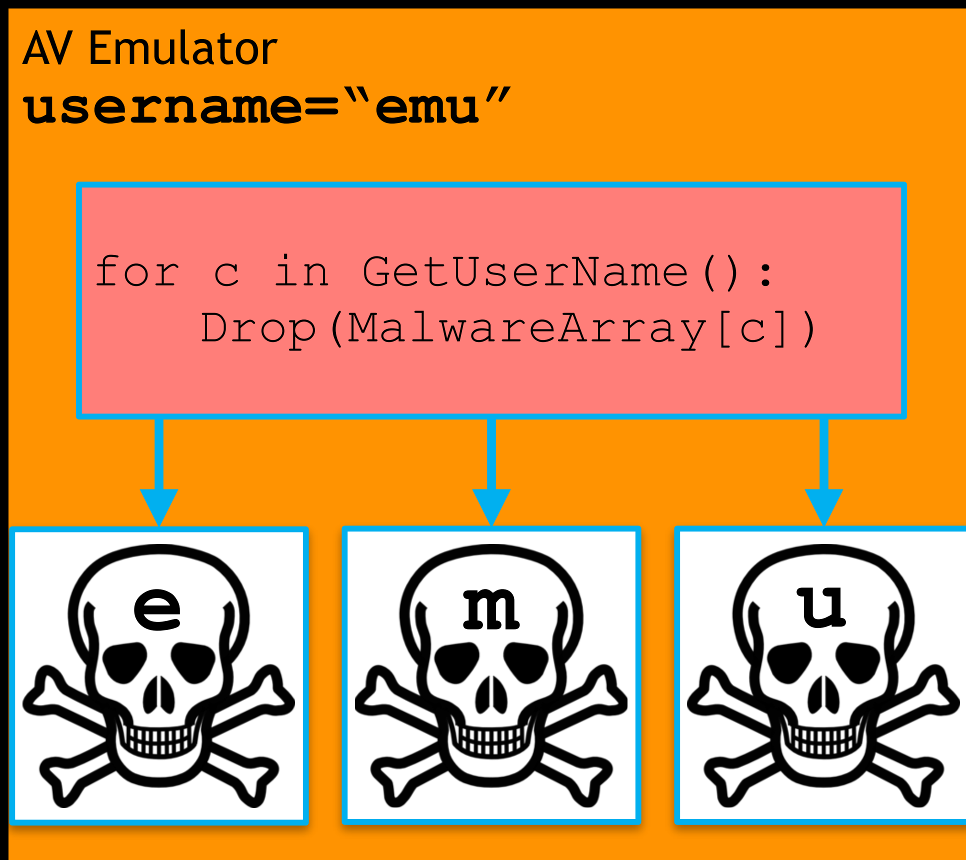
GetUserName()	
A	Morris
B	Code Red
C	Zeus
…	
a	Conficker
…	
z	Brain



AVLeak's Innovation

Question: What is the username in the emulator?

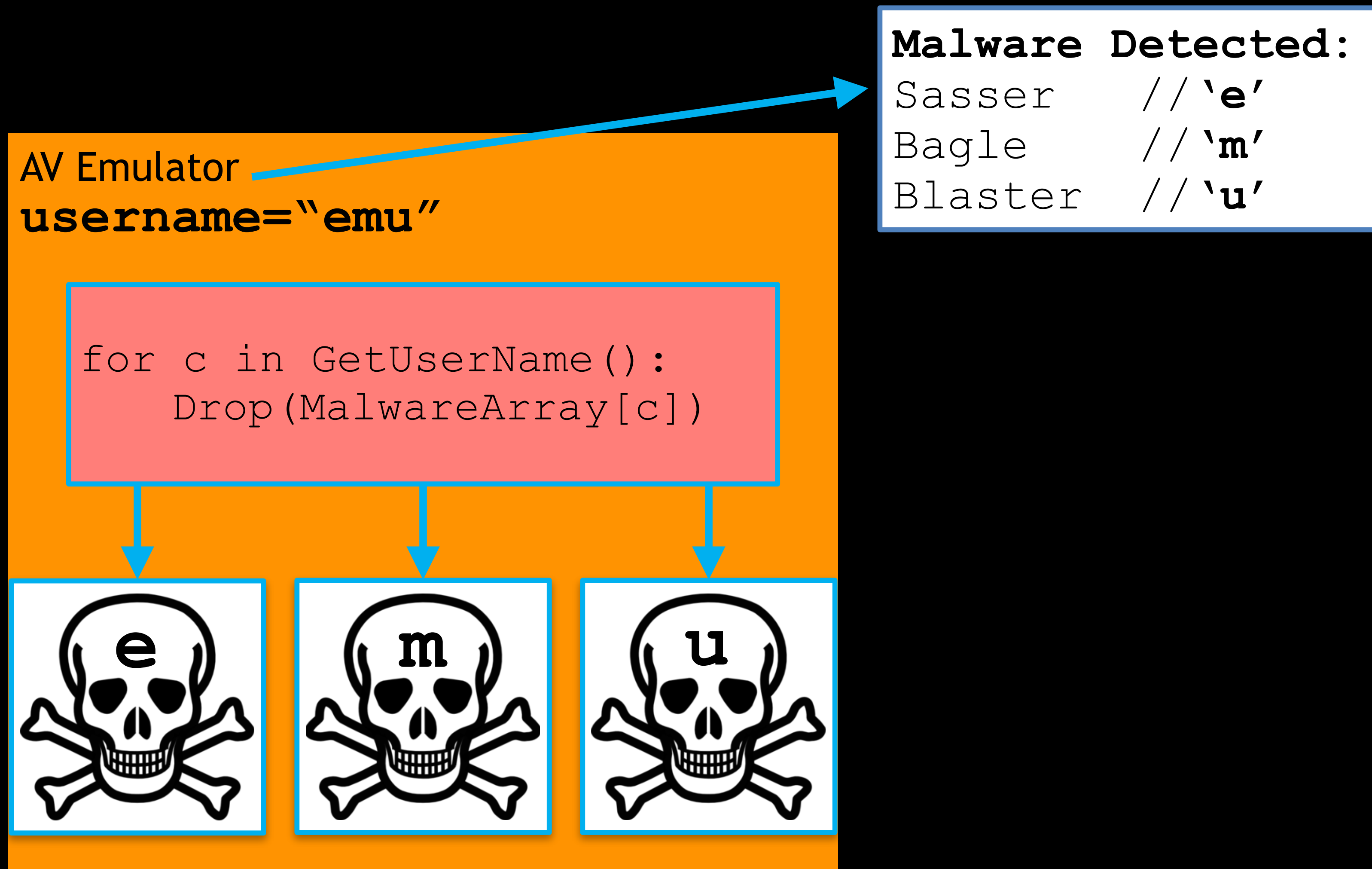
GetUserName()	
A	Morris
B	Code Red
C	Zeus
…	
a	Conficker
…	
z	Brain



AVLeak's Innovation

Question: What is the username in the emulator?

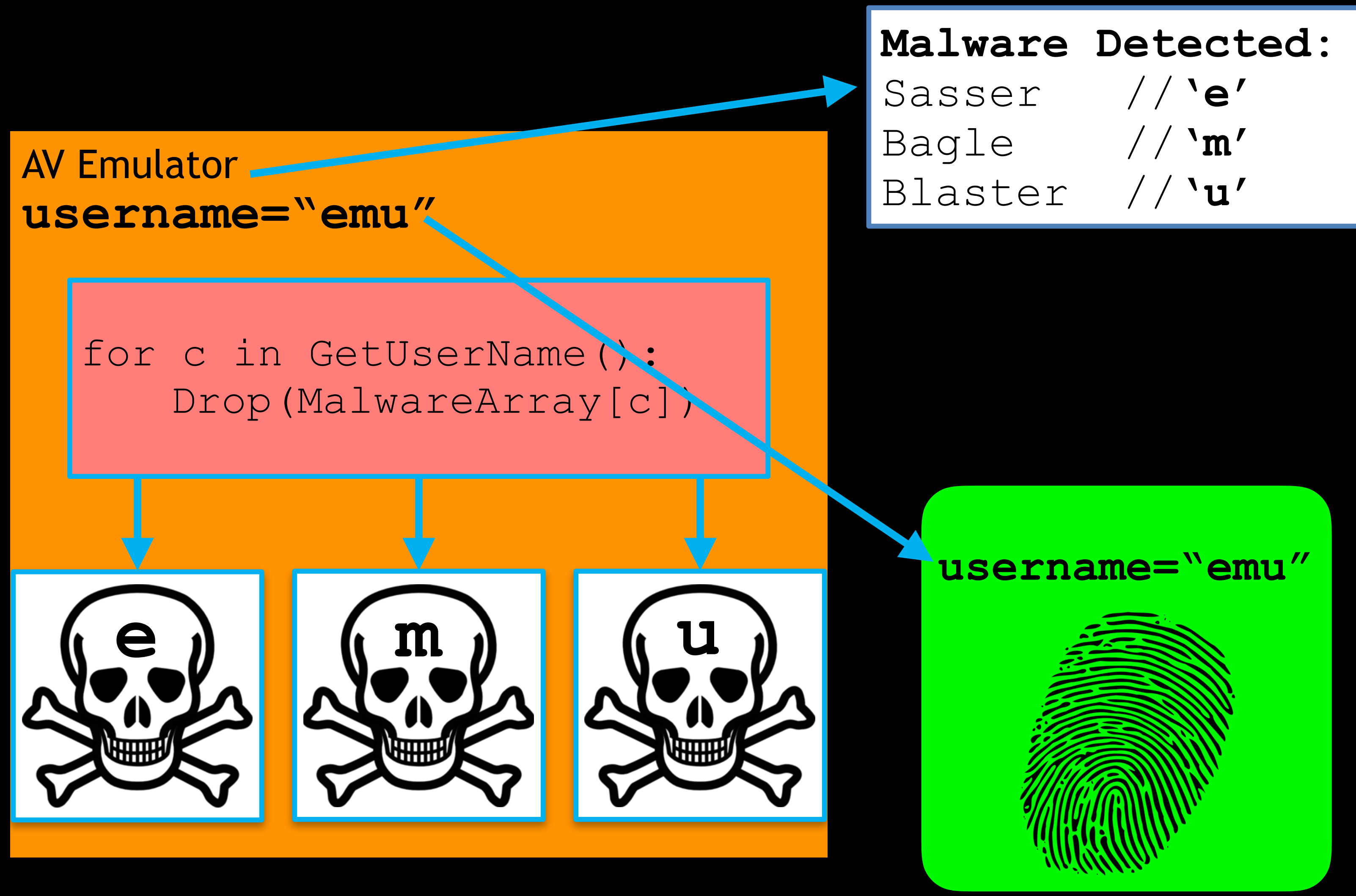
GetUserName()	
A	Morris
B	Code Red
C	Zeus
…	
a	Conficker
…	
z	Brain



AVLeak's Innovation

Question: What is the username in the emulator?

GetUserName()	
A	Morris
B	Code Red
C	Zeus
…	
a	Conficker
…	
z	Brain



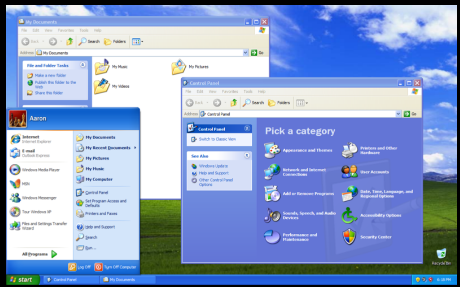
AVs Tested

- Tested four commercial AVs found on VirusTotal
  - Identified by uploading EICAR droppers
- Bitdefender emulator licensed to 20+ other AVs

Outline

Classes of *Consumer AV* Fingerprints

- Environmental artifacts
  - Hardcoded strings for username/computer name/ environment variables, file system, registry entries, processes
- OS API inconsistency
  - Functions that fail, return hardcoded values, generally don't behave correctly
- Network emulation
  - Inconsistencies with real network behavior, hardcoded responses to network traffic
- Timing
  - Timing skews, dilation, inconsistencies across observations
- Process Introspection
  - Internal process traits - uninitialized memory, data left on stack or in registers after function calls, PEB/TEB, DLLs in memory
- CPU "Red Pills"
  - Instructions which behave differently on an emulated CPU



DEMO

Environmental Artifacts

- argv[0]:
  - K: `C:\{random letters}.exe`
  - AVG: `C:\…\mwsmpl.exe`
  - BD: `C:\TESTAPP.EXE`
  - VBA: `C:\SELF.EXE`
- GetComputerName():
  - K: `NfZtFbPfH`
  - AVG: `ELICZ`
  - BD: `tz`
  - VBA: `MAIN`

- BD: `A_E_O_FANTOMA_DE_FISIER_CARE_VA_SA_ZICA_NU_EXISTA` (Romanian: "this is a ghost file which will tell you [that] it doesn't exist.bat"), `TZEAPA_A_LA_BATMAN.EXE` ("Batman's Spike.exe" [with Romanian keyboard specific misspelling]), `C:\\BATMAN`, `NOTHING.COM`
- Kaspersky FS (random flailing on a QWERTY keyboard): `C:\\Documents and Settings \Administrator\My Documents\{koio.mpg, muuo.mp3, qcse.xls, dvzrv.rar,…}`
  - `STD_OUTxe`, `Dummy.exebat`, `welcome.exe`, `Arquivos de programas`
- Kaspersky file headers: `<KL Autogenerated>`
- Fake installs of other AV products, file sharing clients, games
- AVG Product ID: "`76588-371-4839594-51979`"

- Far Manager installs in Kaspersky and VBA
  - "Far Manager … for former USSR countries … as freeware…"

Hardcoded Start Times

- Kaspersky: `11:01:19, July 13, 2012`
- AVG: `1:40:41.16, May 23, 2011`
- VBA: `1:31:12.123, November 3, 2014`
  - `GetSystemTimeAsFileTime: 0:0:0.00, 0/0/2000`
- Bitdefender:
  - `GetSystemTimeAsFileTime: 0:0:0.00 January 1, 2008`
  - `GetSystemTime` doesn't work!
  - `NtQuerySystemTime` doesn't work!

Fake Library Code

- Fake library code in all four AVs
- `GetProcAddress` – dump bytes at pointer
- Obscure instructions are used to trigger library function emulation

AVG:

```
mov edi, edi
push ebp
mov ebp, esp
nop
lock mov ebx,
    0xff(1b lib #)(2b func #)
pop ebp      ; epilogue
ret (size of args)
nop…
```

Outline

1. Introduction
2. Problem & Motivation
3. Background & Prior Work
4. AVLeak
5. Results & Demo
6. Conclusions

Common Themes

- Checking for simple fingerprints enables malware to evade detection
- Hardcoded environmental artifacts are clearly left by programmers as jokes, or as "bait" for malware
- AVs don't do heuristic malware classification based on emulation-detection behavior

Low Budget Malware Discovery

- Advanced malware authors are already using these artifacts

[58a5faf7f2928a7eb24d73b3059d2221e2acd83a - Analysis ...](https://totalhash.cymru.com/analysis/?...)
https://totalhash.cymru.com/analysis/?...
Jan 24, 2014 - BAT CCCIMceg CCf!4Ch4 CCFFf9 CCIMceg "cd#^Z ceeddbbaa``Y ... \ **A_E_O_FANTOMA_DE_FISIER_CARE_VA_SA_ZICA_NU_EXISTA.BAT** ...

Analysis | #totalhash - Team Cymru
https://totalhash.cymru.com/analysis/?...
Jan 2, 2014 - File type, PE32 executable for MS Windows (GUI) Intel 80386 32-bit. Language, 040904b0. Section .text md5: ...

4166c77a7f7891ce8756fb9784c46a2da2d511dd - Analysis ...
https://totalhash.cymru.com/analysis/?...
Jan 24, 2014 - File type, PE32 executable for MS Windows (GUI) Intel 80386 32-bit. Language, 040904B0. Section .text md5: ...

e094d944954303f06d769b89a46e650cc347dc4f - Analysis ...
https://totalhash.cymru.com/analysis/?...
Jan 1, 2014 - ... BMSx:TR B-`Q+= `bTs p~ bY/KB+G -,C8nQA c,ae) C:\ **A_E_O_FANTOMA_DE_FISIER_CARE_VA_SA_ZICA_NU_EXISTA.BAT** California1#0!

6 results (0.33 seconds)

Did you mean: "<kl ***auto generated***>"

Analysis - Malwr - Malware Analysis by Cuckoo Sandbox
https://malwr.com/.../ZmM0ZTg0Zjg5OTk0NGM1OGI0YmFkMTQ2ZjM2...
Apr 24, 2014 - EXE. wswhacker.dllMZ. This program cannot be run in DOS mode. **<KL Autogenerated>**. MSIMG32.dll. AlphaBlend. DllInitialize. GradientFill.

0b621aa5c4e63b3579eea52f0422bb9f - Malwr - Malware ...
https://malwr.com/.../ODc2ZDZlZjlkYWU2NGYzZjk0ZDc4OTczNWE3...
7 days ago - Error: Analysis failed: The package "modules.packages.exe" start function raised an error: Unable to execute the initial process, analysis ...

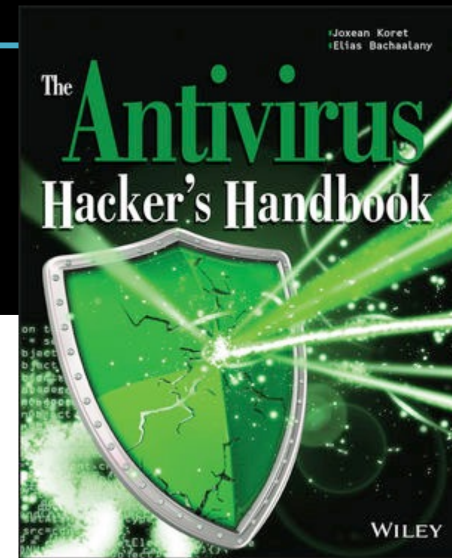
39fef96e2ef1a9cd27d96d16d4b55dda7d21112f - Analysis ...
https://totalhash.cymru.com/analysis/?...
Jan 22, 2015 - ... IsWow64Process KERNEL32.dll **<KL Autogenerated>** _lclose LoadLibraryA LockResource lstrcmpi lstrcpyA lstrcpynW LZStart MoveFileExA ...

Malware Analysis Database - totalhash
https://totalhash.com/analysis/?...
Aug 14, 2014 - DLL kfkS_)Y(W **<KL Autogenerated>** #k~nel %l0ra#j lAj78=V LCMapStringA _lcreat l g*Y'Y:S+R LoadLibraryA LoadLibraryExA LoadResource ...

Analysis | #totalhash
totalhash.com/analysis/f361693130dcaab81c08abeb2550f147b796745d
Nov 4, 2014 - Creates File, C:\Documents and Settings\Administrator\Local Settings\Temp\2445_appcompat.txt. Creates File, PIPE\lsarpc. Creates Process ...

Future Work

- More emulators, more tests
- Use AVLeak for vulnerability research against emulators (breakout exploits)
  - See Tavis Ormandy and Joxean Koret's work



Project Zero

News and updates from the Project Zero team at Google

Tuesday, June 23, 2015

Analysis and Exploitation of an ESET Vulnerability

Do we understand the risk vs. benefit trade-offs of security software?
Tavis Ormandy, June 2015

Introduction

Many antivirus products include emulation capabilities that are intended to allow unpackers to run for a few cycles before signatures are applied. ESET NOD32 uses a minifilter or kext to intercept all disk I/O, which is analyzed and then emulated if executable code is detected.

Attackers can cause I/O via Web Browsers, Email, IM, file sharing, network storage, USB, or hundreds of other vectors. Whenever a message, file, image or other data is received, it's likely some untrusted data passes through the disk. Because it's so easy for attackers to trigger emulation of untrusted code, it's critically important that the emulator is robust and isolated.

Unfortunately, analysis of ESET emulation reveals that is not the case and it can be trivially compromised. This report discusses the development of a remote root exploit for an ESET vulnerability and demonstrates how attackers could compromise ESET users. This is not a theoretical risk, recent evidence suggests a growing interest in anti-virus products from advanced attackers.

Conclusion

- Pushed the state of the art in emulator fingerprinting
- Presented a survey of emulator fingerprints across six categories
- Demonstrated real world examples of malware exploiting these fingerprints

Selected References

- ADAMS, K. Evading Code Emulation: Writing Ridiculously Obvious Malware That Bypasses AV, 2014. Talk at BSides Las Vegas 2014, Las Vegas, Nevada.
- CZUMAK, M. peCloak.py An Experiment in AV Evasion. http://www.securitysift.com/pecloak-py-an-experiment-in-av-evasion, 2015.
- FERRIE P. Attacks on Virtual Machine Emulators. Tech. rep., Symantec Advanced Threat Research, 2006.
- KORET, J., AND BACHAALANY, E. The Antivirus Hacker's Handbook. Wiley, Indianapolis, Indiana, 2015.
- MARSCHALEK, M. EvilBunny: Malware Instrumented By Lua. http://www.cyphort.com/evilbunny-malware-instrumented-lua, 2014.
- NASI, E. Bypass Antivirus Dynamic Analysis: Limitations of the AV model and how to exploit them. Tech. rep., Self-published, 2014.
- OBERHEIDE, J., BAILEY, M., AND JAHANIAN, F. PolyPack: An Automated Online Packing Service for Optimal Antivirus Evasion. In WOOT'09 Proceedings of the 3rd USENIX conference on Offensive technologies (2009).
- ORMANDY, T. Analysis and Exploitation of an ESET Vulnerability. http://googleprojectzero.blogspot.com/2015/06/analysis-and-exploitation-of-eset.html, 2015.
- PALEARI, R., MARTIGNONI, L., ROGLIA, G. F., AND BRUSCHI, D. A fistful of red-pills: How to automatically generate procedures to detect CPU emulators. In WOOT'09 Proceedings of the 3rd USENIX conference on Offensive technologies (2009).
- ROLLES, R. Detecting an emulator using the windows api. http://reverseengineering.stackexchange.com/questions/2805/detecting-an-emulator-using-the-windows-api, 2013.
- SAUDER, D. Why Antivirus Software Fails, 2014. Talk at DeepSec 2014, Vienna, Austria.
- SECOND PART TO HELL. Dynamic Anti-Emulation using Blackbox Analysis. http://vxheaven.org/lib/vsp42.html, 2011.
- SWINNEN, A., AND MESBAHI, A. One Packer to Rule Them All: Empirical Identification, Comparison and Circumvention of Current Antivirus Detection Techniques, 2014. Talk at Black Hat 2014, Las Vegas, Nevada.
- YOSHIOKA, K., HOSOBUCHI, Y., ORII, T., AND MATSUMOTO, T. Your Sandbox is Blinded: Impact of Decoy Injection to Public Malware Analysis Systems. Journal of Information Processing 19 (2011).

Thank You

- RPI Research Team:
  - Jeremy Blackthorne
  - Patrick Biernat
  - Dr. Bülent Yener
  - Dr. Greg Hughes
- Help & Inspiration:
  - Marion Marshalek
  - Rolf Rolles
  - Alex Ionescu
  - Bruce Dang
  - Dr. Sergey Bratus

Questions?

Kaspersky Lab - Packin' The K