Bypassing Antivirus

With Understanding Comes Ease

Jeff McJunkin, Founder
Rogue Valley Information Security

Introduction and Agenda

- AV’s main approaches to detecting and blocking malware
- Fundamental limitations of the above
- Bypassing AV -- Different Approaches
- Live Demo
- Further discussions
- Q&A

Antivirus / Antimalware / EDR's tools

1. Static detections - searching each EXE/DLL on load for bad strings/patterns
2. Hooking runtime API calls and searching for bad strings / patterns
    - Anti-Malware Scanning Interface (AMSI)
        - Otherwise, all PowerShell / JavaScript, etc only has static detections
    - Userland hooks
    - Kernel-level hooks (Not used for modern operating systems due to PatchGuard, but replaced by mini-filter drivers instead)
    - This approach is more prone to false positives and has fewer signatures
3. Dynamic detection (behavioral analytics during brief emulation)
    - This approach is also more prone to false positives, and has fewer signatures

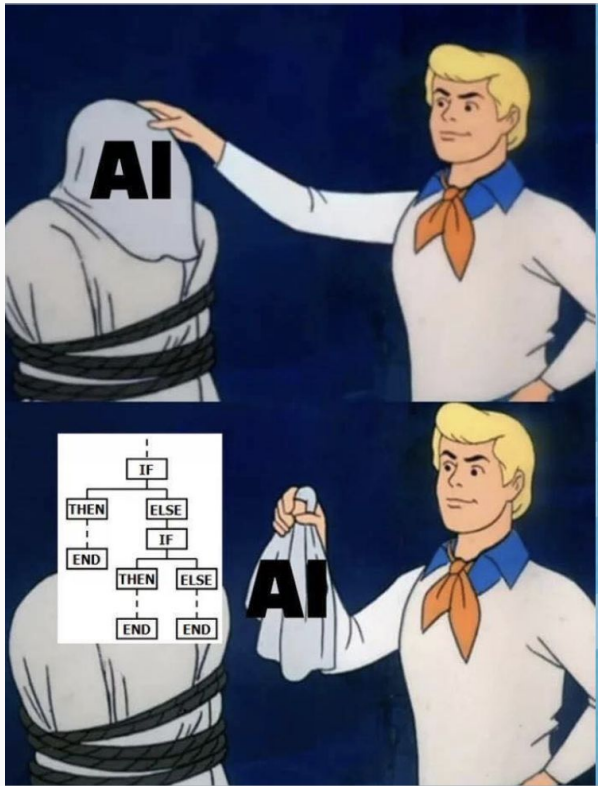
All of the techniques result in a yes/no decision using "badness" scores

Some strings (such as "Copyright Benjamin Delpy" or "Invoke-Mimikatz") are suspicious, whereas others (such as "Copyright Microsoft 2020") are not.

“Next-Gen” Antivirus

But what about machine learning? What about artificial intelligence?

AI is only as smart as it needs to be (finding bad strings)

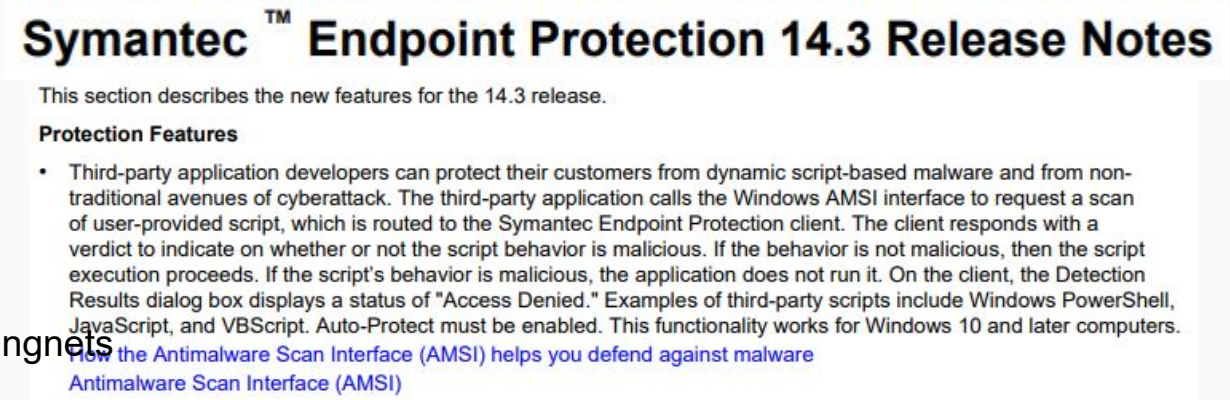


Researchers Easily Trick Cylance's AI-Based Antivirus Into Thinking Malware Is 'Goodware'

By taking strings from an online gaming program and appending them to malicious files, researchers were able to trick Cylance’s AI-based antivirus engine into thinking programs like WannaCry and other malware are benign.

AMSI

- The Antimalware Scan Interface allows AV vendors to “see” PowerShell, JavaScript, VBScript, Office macros, and a few other scripting formats, after de-obfuscation and before execution
  - AV then gets to make a yes/no decision
- Not all AV vendors support AMSI (introduced in 2015)
- Symantec didn’t support it until June 15, 2020 with SEP 14.3



Symantec™ Endpoint Protection 14.3 Release Notes

This section describes the new features for the 14.3 release.

Protection Features

- Third-party application developers can protect their customers from dynamic script-based malware and from non-traditional avenues of cyberattack. The third-party application calls the Windows AMSI interface to request a scan of user-provided script, which is routed to the Symantec Endpoint Protection client. The client responds with a verdict to indicate on whether or not the script behavior is malicious. If the behavior is not malicious, then the script execution proceeds. If the script's behavior is malicious, the application does not run it. On the client, the Detection Results dialog box displays a status of "Access Denied." Examples of third-party scripts include Windows PowerShell, JavaScript, and VBScript. Auto-Protect must be enabled. This functionality works for Windows 10 and later computers.

How the Antimalware Scan Interface (AMSI) helps you defend against malware

Antimalware Scan Interface (AMSI)

Which Vendors Support AMSI?

https://github.com/subat0mik/whoamsi

My hot take: if your AV/EDR vendor doesn't support AMSI, they have no business in this industry, and don't deserve your money.

Vendor/Product	AMSI	Date	Reference
Avast	Y	03/20/2016	https://forum.avast.com/index.php?topic=184491.msg1300884#msg13
AVG	Y	03/08/2016	https://support.avg.com/answers?id=906b00000008oUTAAY
Avira			
BitDefender Consumer	Y	09/20/2016	https://forum.bitdefender.com/index.php?/topic/72455-antimalware-sca service/
BitDefender Enterprise	N	04/05/2019	https://forum.bitdefender.com/index.php?/topic/79653-does-best-suppo antimalware-scan-interface-amsi/
Carbon Black Defense	Y	03/18/2020	https://www.carbonblack.com/2020/03/18/detecting-fileless-attacks-wit enterprise-edrs-amsi-visibility/
CrowdStrike Falcon			
Cybereason			
Cylance			
Dr. Web			
ESET	Y	04/12/2017	https://forum.eset.com/topic/11645-beta-eset-endpoint-security-66-is- available-for-evaluation
F-Secure			
Kaspersky	Y	10/10/2018	https://help.kaspersky.com/KIS/2019/en-US/119653.htm
McAfee	Y	06/25/2018	https://kc.mcafee.com/corporate/index?page=content&id=PD27443
Panda			
SentinelOne			
Sophos	N	02/20/19	https://community.sophos.com/products/endpoint-security-control/f/sop endpoint-software/88584/windows-10-support-for-amsi
Symantec	Y	07/15/2020	https://techdocs.broadcom.com/content/broadcom/techdocs/us/en/syn security-software/endpoint-security-and-management/endpoint-protection/all/release-notes/Whats-new-for-Symantec-Endpoint-Protec 14_3-.html Thanks, Jeff McJunkin!
ThreatTrack Vipre			
Trend Micro			
Windows Defender	Y	06/09/2015	https://www.microsoft.com/security/blog/2015/06/09/windows-10-to-off application-developers-new-malware-defenses/

Fundamental Limitations of AV

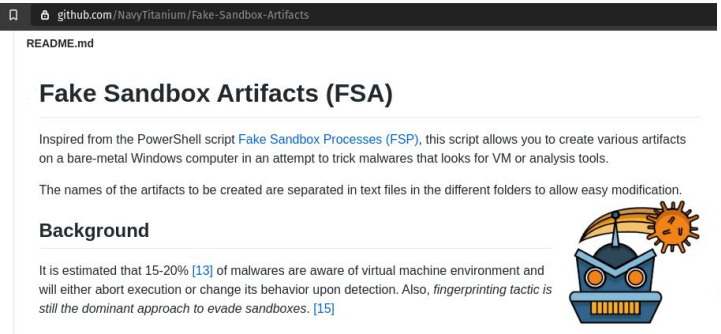
- AV isn't tracking everything, for performance reasons
  - Each API call interception adds latency, and there are strong financial incentives to avoid perceptibly slowing down computers due to AV
    - Emulation maximum time varies, but I've never seen it over 0.25 seconds (@taviso mentioned a "million instructions")
  - AV intercepts far more than just running EXE files from disk
    - Examples: Loading additional DLL's, writing to other process's memory, creating new threads, extracting compressed content, checking for debuggers, and more
- AV companies have a strong financial incentive to minimize false positives, which means A) accepting false negatives and B) brittle signatures

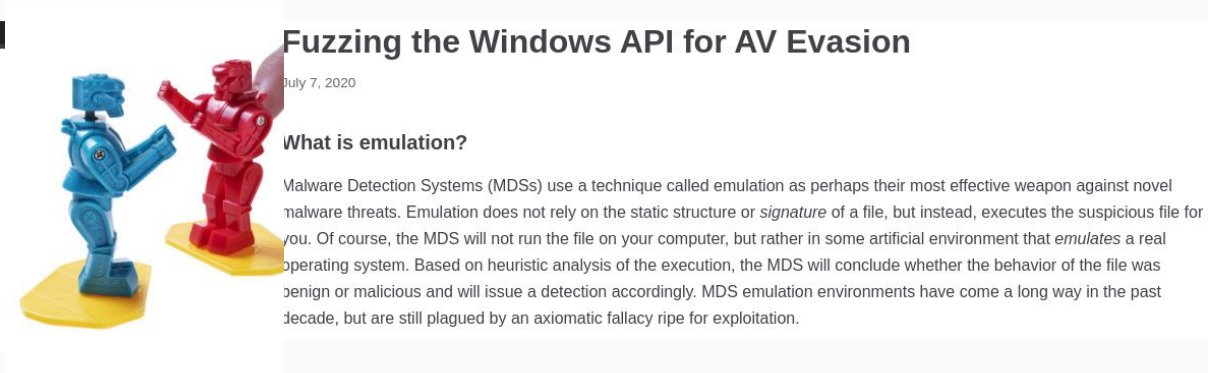
With AV, bots are fighting bots. Unfortunately, attackers can study AV's behavior, and they take the last move.

Rock ‘Em, Sock ‘Em

Left: https://github.com/NavyTitanium/Fake-Sandbox-Artifacts

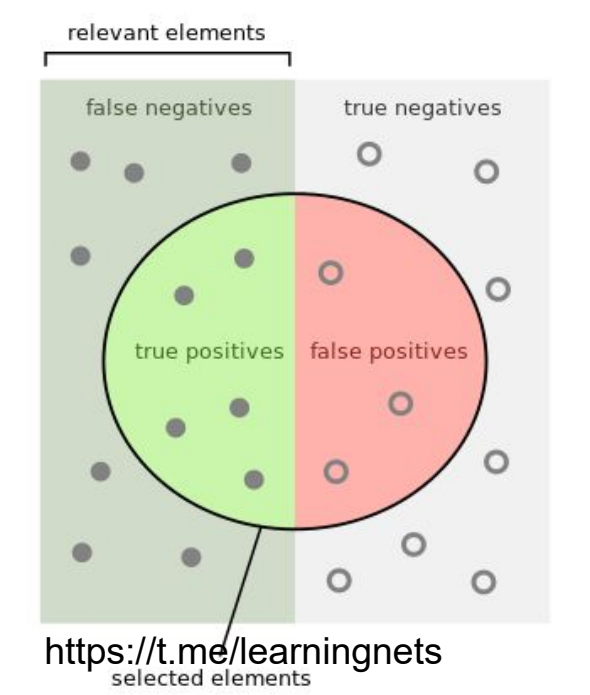
Right: https://winternl.com/fuzzing-the-windows-api-for-av-evasion/

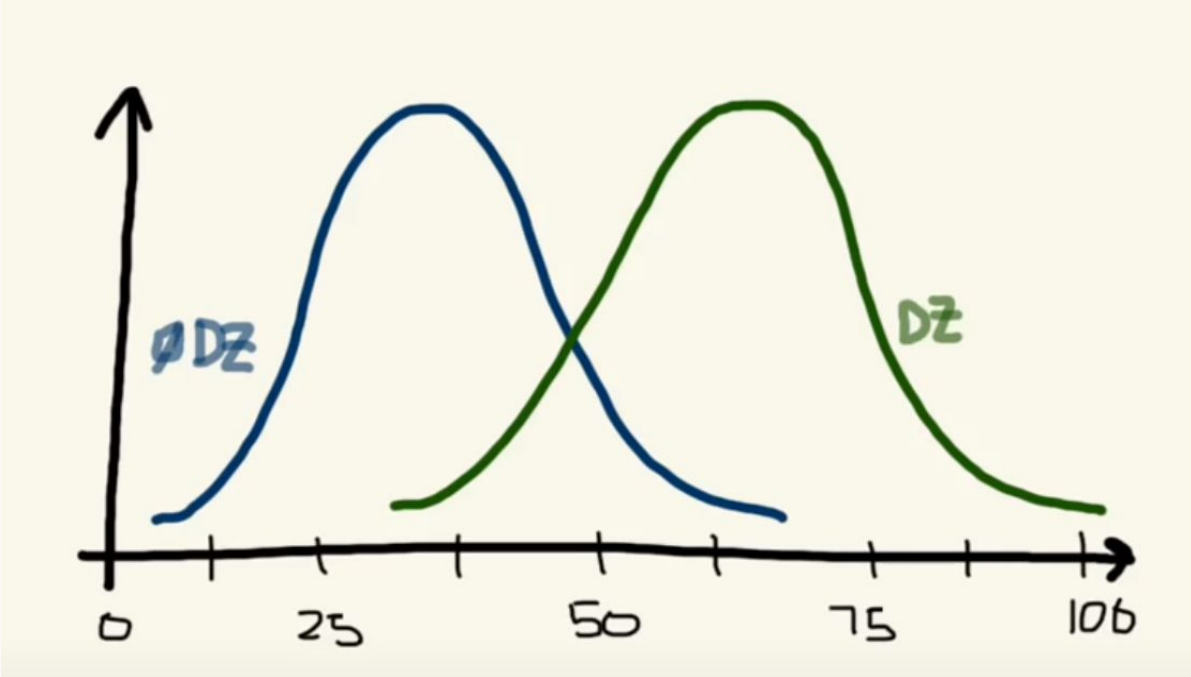




Relationship Between False Positives and False Negatives

Commonly discussed in testing theory, but terminology is often used elsewhere (categorization theory, AV/EDR, etc)



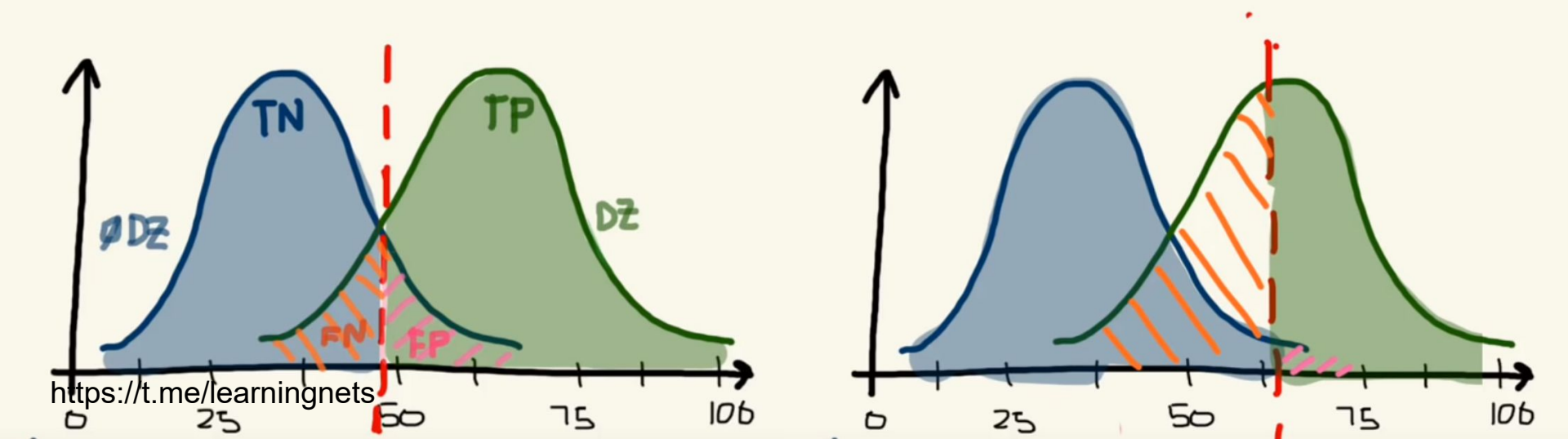


Relationship Between False Positives and False Negatives

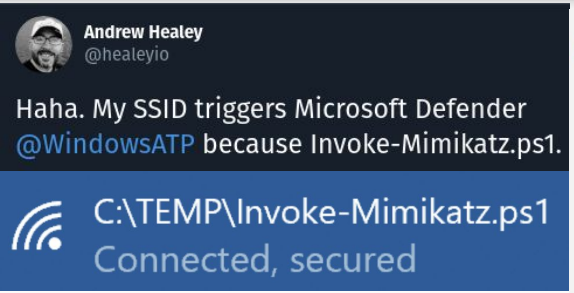
False Positive: Marking a benign binary (such as Notepad.exe) as malicious

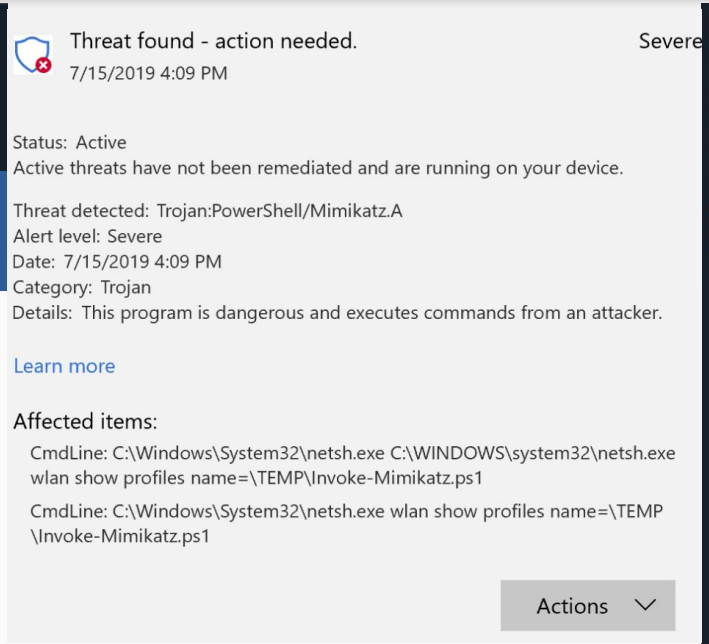
False Negative: Marking a malicious binary (such as Mimikatz.exe) as safe

False Positives cost more than False Negatives, in PR and market share



Brittle Signatures, Demonstrated



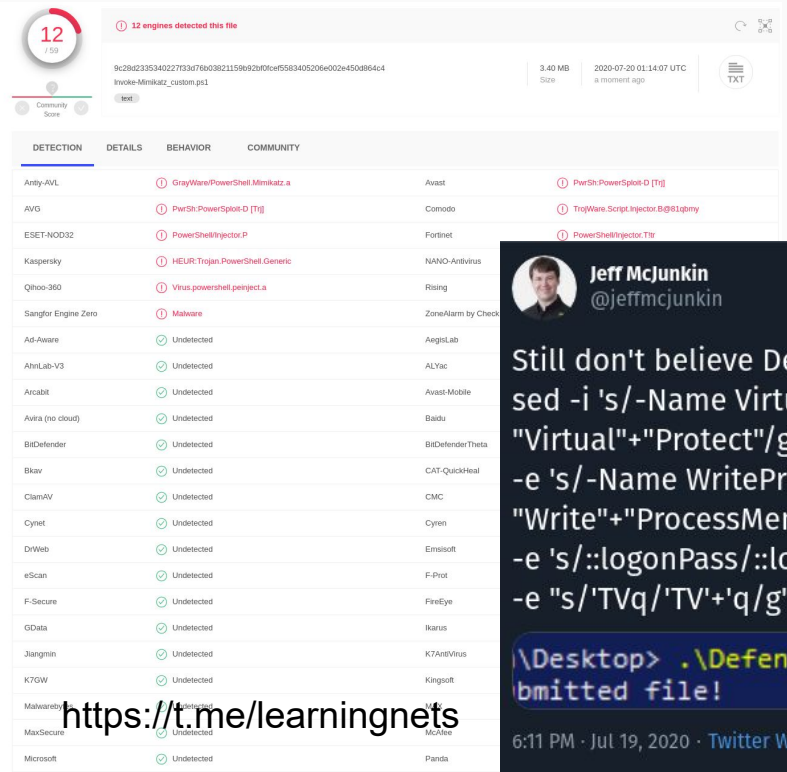


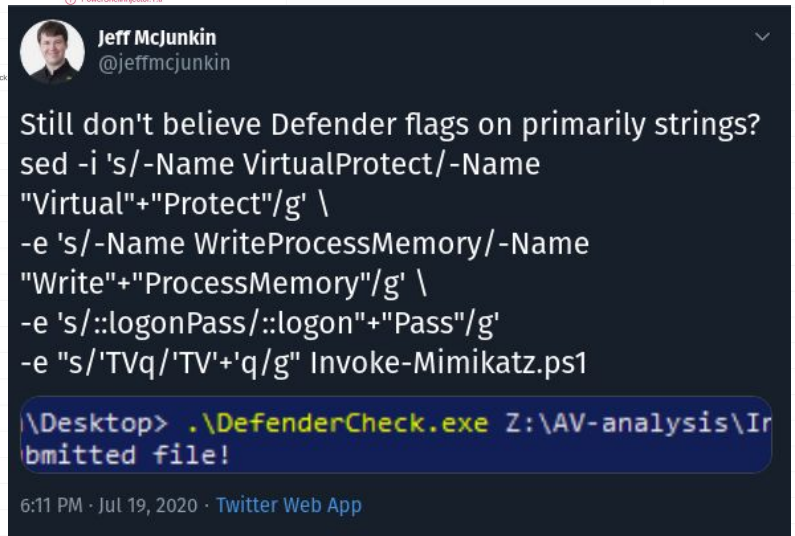
"Copyright Benjamin Delpy": unlikely to have false positives, but all sorts of legitimate software accesses LSASS as well.

DEMO:
Bypassing Static Signatures To Use Mimikatz

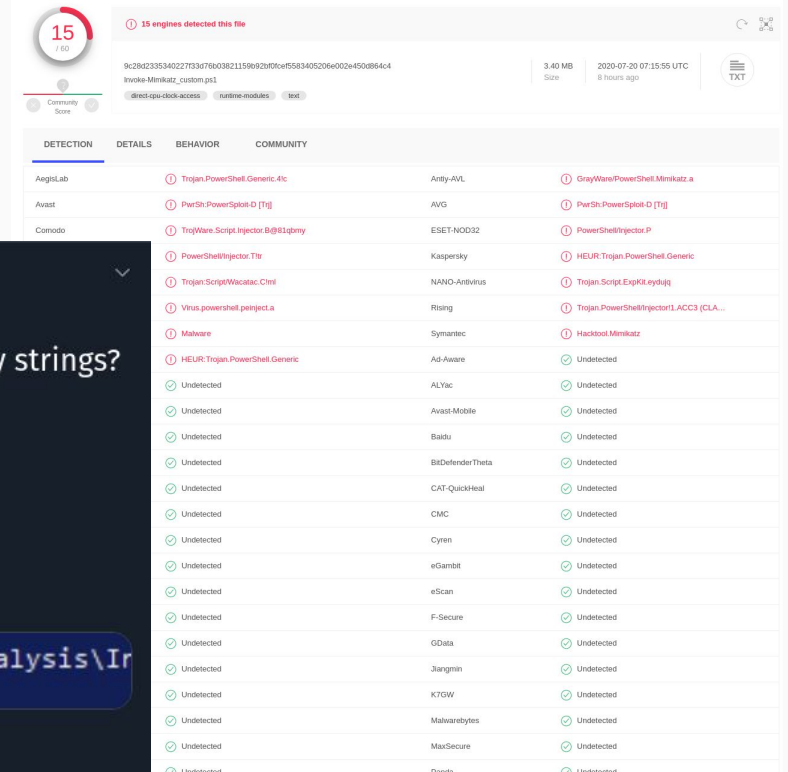
Disadvantages for Security Professionals

Before Tweet





After Tweet



Disadvantages for Security Professionals

Security professionals tend to…

1. Talk about their bypass methods
    - As shown on the prior slide, this tends to result in quickly-updated vendor signatures
2. Use public toolsets (Metasploit, Empire, Covenant, Veil-Evasion)
    - AV vendors spend a disproportionate amount of time making signatures for public tools
    - Security professionals often make their own non-public toolset to take advantage of this, to good effect (BHIS included)
3. Upload to VirusTotal to test their payloads
    - ...which results in every single AV vendor getting a copy of the payload

Methods of Bypassing Antivirus

1. **Use non-malicious software in malicious ways (preferred)**
    - Instead of Metasploit's psexec implementation, use PsExec.exe from Microsoft
    - Instead of Mimikatz.exe, dump LSASS memory with Task Manager and extract passwords elsewhere
    - Instead of hashdump, save out registry hives and extract hashes elsewhere
    - Instead of meterpreter (at first), use Remote Desktop, mRemote-NG, TeamViewer, etc.

In general, avoid fair fights (both against AV and in life)

Example of #1 Approach

Methods of Bypassing Antivirus

2. Make an unfair fight:
   - Run inside PowerShell version 2, which doesn't support AMSI, even on Windows 10
   - Use API calls that aren't intercepted
   - "Unhook" API calls so antivirus doesn't have any visibility
     - Read https://github.com/NtRaiseHardError/Antimalware-Research/ for more on this!
   - Detect AV's sandboxed environments and run differently there:
     - https://github.com/David-Reguera-Garcia-Dreg/anticuckoo
     - https://winternl.com/fuzzing-the-windows-api-for-av-evasion/
   - Encrypt the payload and only decrypt at runtime (Hyperion, bypasses static signatures and emulation)
   - Add extra strings (from legitimate software) to increase the "goodness" score
   - Add extra data to go above certain thresholds

```
#                                                                              #
# DESCRIPTION                                                                  #
# Hyperion is a runtime encrypter for 32/64 bit portable executables. It is a  #
# reference implementation and bases on the paper "Hyperion: Implementation    #
# of a PE-Crypter". The paper describes the implementation details which       #
# aren't in the scope of this readme file.                                     #
# The crypter is a C project and can be compiled with the corresponding        #
# makefile (tested with Mingw and Visual Studio). Afterwards it is started     #
# via the command line and encrypts an input executable with AES-128. The      #
# encrypted file decrypts itself on startup (bruteforcing the AES key which    #
# may take a few seconds) and generates a log file for debug purpose.          #
```

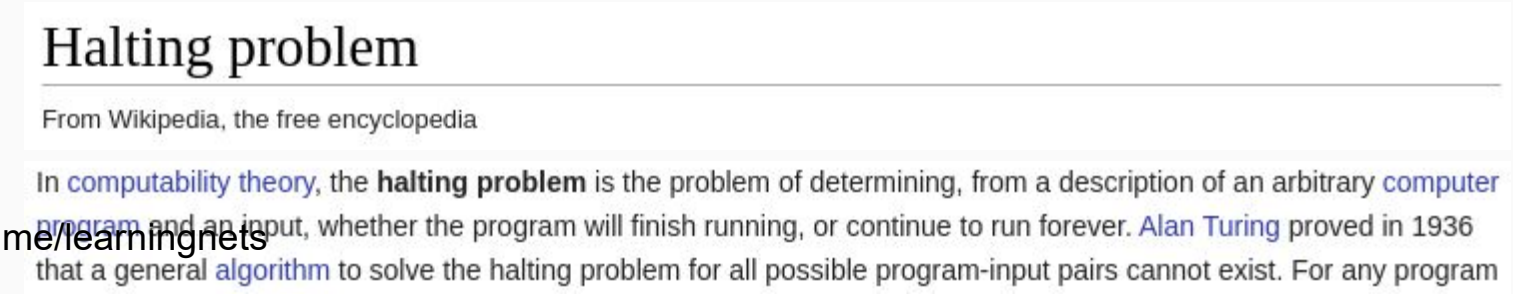
Methods of Bypassing Antivirus, Cont.

3. If necessary: Stack the odds in your favor before a fair fight through the following methodology:

- Install the AV in an isolated virtual machine
  - (which doesn't report to vendor or client)
- Update AV signatures
- Take a snapshot
- Disconnect network adapter
- Introduce malware
- If discovered, modify and repeat as necessary
- Revert to snapshot (no matter what)

Application Control

- AV (stopping malicious software while allowing legitimate software) has a literally impossible job
  - There are infinite ways to accomplish a given task, whether it's "Hello World", meterpreter, Mimikatz, or ransomware
  - Determining whether software is "malicious" is definitely a harder problem than the Halting Problem, which was proven impossible to solve in 1936
- Instead, focus on application control, only allowing known-good software, such as Microsoft-signed binaries, known vendor software, and internally-developed applications



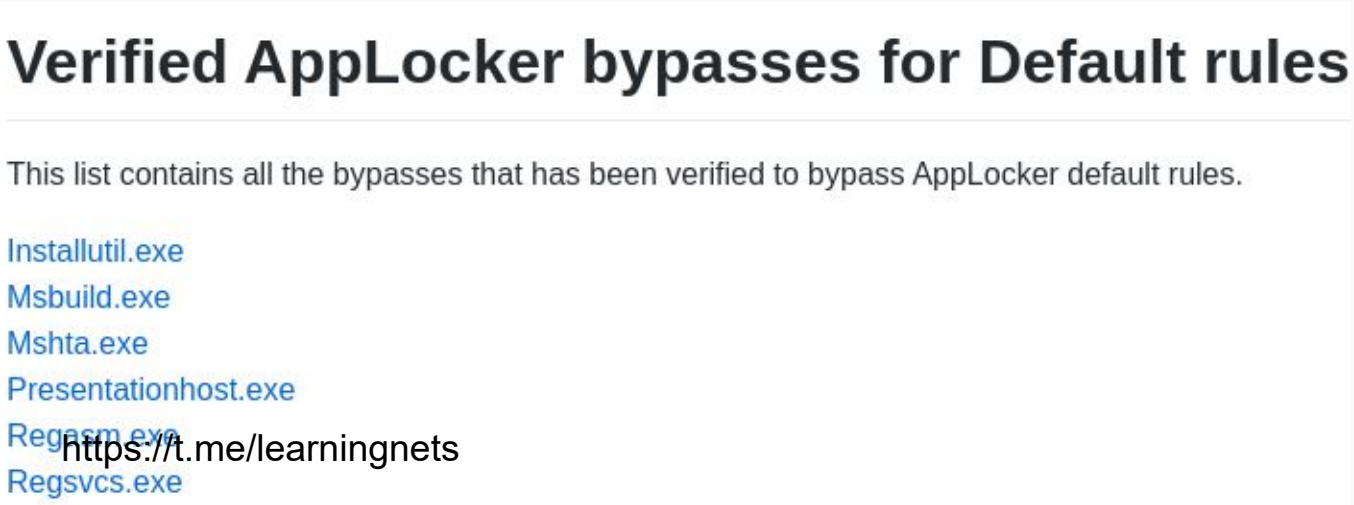
Halting problem

From Wikipedia, the free encyclopedia

In computability theory, the **halting problem** is the problem of determining, from a description of an arbitrary computer program and an input, whether the program will finish running, or continue to run forever. Alan Turing proved in 1936 that a general algorithm to solve the halting problem for all possible program-input pairs cannot exist. For any program

Application Control: Bypasses

- Allowing only signed Microsoft binaries isn't 100% effective
- Some signed binaries allow arbitrary code execution directly
  - Some refer to these as "LOLBINS" for Living Off the Land BINaries
  - Example: MSBuild.exe allows compiling applications and running arbitrary C# code
  - Many are documented at https://lolbas-project.github.io/ (Living Off the Land Binaries And Scripts)



A Better Approach with AV

1. Rely on detection in depth and rapid response, not solely preventive controls
2. Plug AV alerts into monitoring feeds for Security Operations Centers
    - Anecdote here about DC
3. React quickly to AV alerts <u>including root cause analysis</u>
4. Don't pick products based on the Gartner Magic Quadrant, but factor it into your own evaluations
    - Anecdotally, Defender is the toughest to bypass
5. Spend effort on application control as well, with EDR and AV as additional layers

AV's own attack surface

Some attributes of AV:

- Highly privileged
- Often not sandboxed
- Evaluates untrusted inputs and code
- Often not logged
- Sometimes has poor software protections

Project Zero

News and updates from the Project Zero team at Google

Tuesday, June 28, 2016

How to Compromise the Enterprise Endpoint

Posted by Tavis Ormandy.

Symantec is a popular vendor in the enterprise security market, their flagship product is Symantec Endpoint Protection. They sell various products using the same core engine in several markets, including a consumer version under the Norton brand.

Today we're publishing details of multiple critical vulnerabilities that we discovered, including many wormable remote code execution flaws.

These vulnerabilities are as bad as it gets. They don't require any user interaction, they affect the default configuration, and the software runs at the highest privilege levels possible. In certain cases on Windows, vulnerable code is even loaded into the kernel, resulting in remote kernel memory corruption.

```
Z:\AV-analysis>wincheckseс.exe Avast\aswhook-x86.dll
Dynamic Base     : "Present"
ASLR             : "Present"
High Entropy VA  : "NotPresent"
Force Integrity  : "NotPresent"
Isolation        : "Present"
NX               : "Present"
SEH              : "Present"
CFG              : "Present"
RFG              : "NotPresent"
SafeSEH          : "NotPresent"
GS               : "Present"
Authenticode     : "NotPresent"
.NET             : "NotPresent"
```

```
Z:\AV-analysis>wincheckseс.exe Norton\NortonSecurity.exe
Dynamic Base     : "Present"
ASLR             : "Present"
High Entropy VA  : "Present"
Force Integrity  : "Present"
Isolation        : "Present"
NX               : "Present"
SEH              : "Present"
CFG              : "NotPresent"
RFG              : "NotPresent"
SafeSEH          : "NotApplicable"
GS               : "Present"
Authenticode     : "Present"
.NET             : "NotPresent"
```

Closing Statement

To prevent successful breaches, defenders need to detect and respond to attackers before they accomplish their goal. Therefore, defenders have two goals:

1. Lowering the time to detect and respond to an attacker
2. Making it take longer for an attacker to accomplish their goal

Prevention is ideal, but it's impossible to prevent 100% of incidents. Therefore, focus on minimizing, detecting, and accelerating response to incidents.

Questions?

Slides are online at https://bit.ly/bypassingav

Follow up questions? jeff@roguevalleyinfosec.com