SECTOR

Security Education Conference

Lies, and Damn Lies:

Getting Past the Hype of Endpoint Security Solutions

Disclaimer

The testing methodology and techniques used during this presentation are not meant to discredit any endpoint protection solution.

All results represent a point in time and results may differ based on different testing scenarios. Solutions tested at the time were current, up to date and configured by each vendor. Some products may have changed or may have been revised since testing was last performed.

This presentation serves only to give back and provide a testing framework to help you to effectivity conduct EPP testing on your own. The information in this presentation is not for financial gain. Opinions are my observations.

Thanks to the Consumer Review Fairness Act of 2016 contracts that purport to restrict our ability to publish these reviews, are void.

Who am I?

Lidia Giuliano @pink_tangent

- Information Security Professional for the past 15 years
- Curious nerd by nature and there is always a solution
- In my spare time I research and play with new technologies, build, break, rinse, repeat
- Interest in:
  - Vulnerability Management
  - Malware Evasion Techniques
  - Data Security and Defensive Tactics
  - Linux and playing with Github repos

E: tangentmelb@gmail.com

Agenda

- Background
- Endpoint Summary – Lies and Truths
- How to Prepare
- Pre-Execution Testing
- Execution Testing
- Post-Execution Testing
- Environment / Business Testing
- Summary

Plus
DEMOs

Background

Task: To resolve the issue of rampant ransomware, specifically impacting network shares

Challenges faced:

- Clicking on Phishing Campaigns
- Multiple mapping to file shares
- Endpoint User files are encrypted, resulting in encrypted file shares
- Backups and recovery services equated to 2-3 days loss attempting to bring the environment back to 100%

Goal: Dramatically reduce ransomware events (from 10 major to 1 p/yr)

Result: Creation of a framework that went beyond ransomware and using the marketing hype to perform a reusable testing methodology

Ransomware Demo

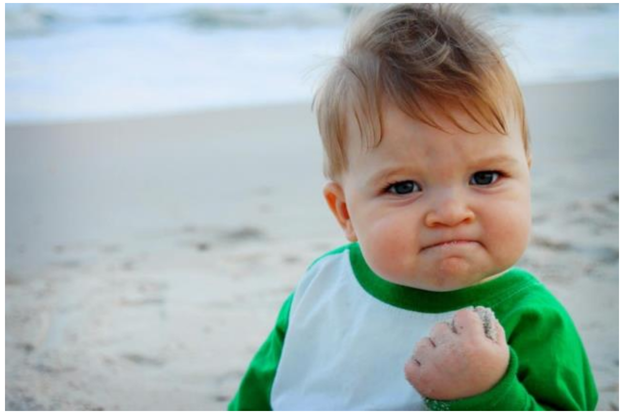
Fiction: Protect only your critical servers!

Fact: Deployment is essential!

- File Share protected with an EPP agent
- Patient 0 is not protected or is using traditional AV
- Patient 0 clicks on a malicious attachment and resulting in local files being encrypted on the endpoint
- Will the files on the share drive be spared?????

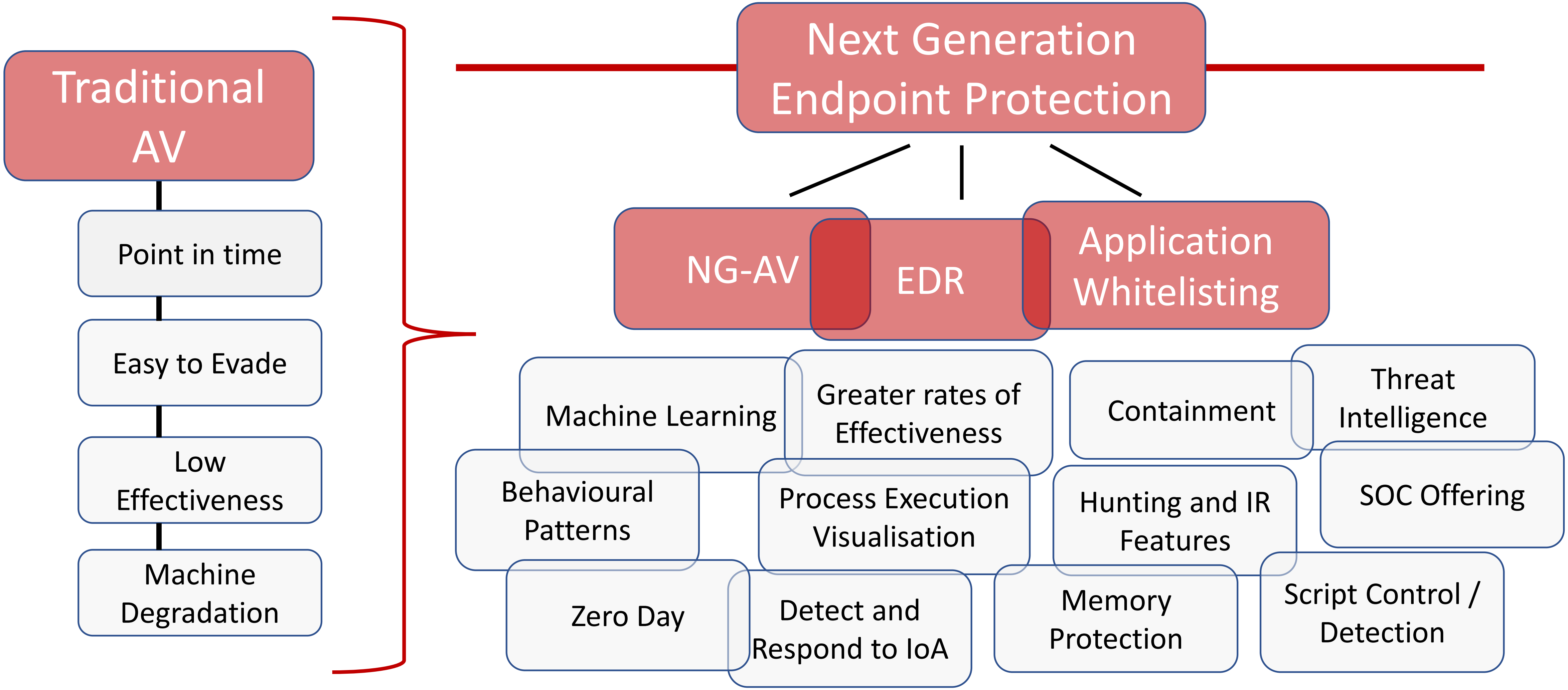
Objectives

- Provide an overview of endpoint protection products (EPP)
- Knowing where to start
- Company business requirements vs. EPP Solutions
- Planning your POC:
  - Plan
  - Preparation
  - Testing and Evaluations
- Provide you with tools to enable you to test solutions yourself
- **<u>You:</u>** Knowledge!
  - Know the questions to ask
  - Know how to do it yourself



Endpoint Protection Overview

SECTOR
Security Education Conference



Traditional AV

- Point in time
- Easy to Evade
- Low Effectiveness
- Machine Degradation

Next Generation Endpoint Protection

NG-AV

EDR

Application Whitelisting

Machine Learning

Greater rates of Effectiveness

Containment

Threat Intelligence

Behavioural Patterns

Process Execution Visualisation

Hunting and IR Features

SOC Offering

Zero Day

Detect and Respond to IoA

Memory Protection

Script Control / Detection

The Marketing Hype

SECTOR Security Education Conference

Marketing: Real Time APT Protection
Observations: No memory-based analysis

Marketing: Multi-layered Approach
Observations: Turn a layer off, hello malware

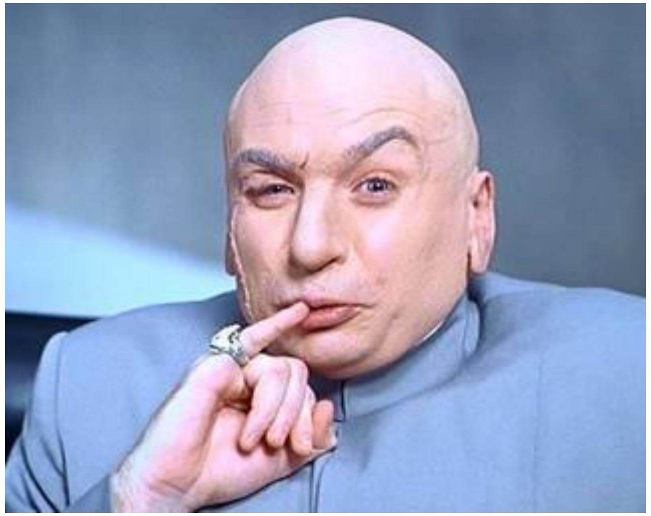
Marketing: Leader in Cloud-based Endpoint
Observations: Have a roaming user with no internet connection, product effectiveness drops

Marketing: Complete replacement of your legacy AV
Observations: Consider the impact on your compliance needs!

Business Problem

More of the Benefits:

- Reduction of Incidents
- People Costs
- Reputation
- Keep the business running
- Protect PII data



Less of the Problem:

- Ransomware
- Insider Threat
- Malicious Outsider
- Threat Hunting
- Incident Response

Requirements:

- Functional
- Non-Functional

Measurements:

- Must’s + Should’s + Nice to have’s
- Weighting + Scoring

POC Timeline

Planning and Research

3 months

- ~80 business requirements
- ~20 non-functional
- Test scenarios
- Investigated impact on different users, roles, remote workers, platforms in operation
- Cloud vs No Cloud

Solution Testing

4-5 months

- Preparation of Test Environment
- Collect Malware Samples and Scripts
- Malware Mutation
- Varied sample data
- Pre-Execution
- Execution
- Post-Execution
- Document Findings

Business Testing

2 months

- Install Agent in Business Environment
- Monitor Mode Only
- Test Packaging
- Test Against Custom Applications
- Test Against Deployment Methods
- Test Other Dependencies
- Document Findings

Preparation of Environment

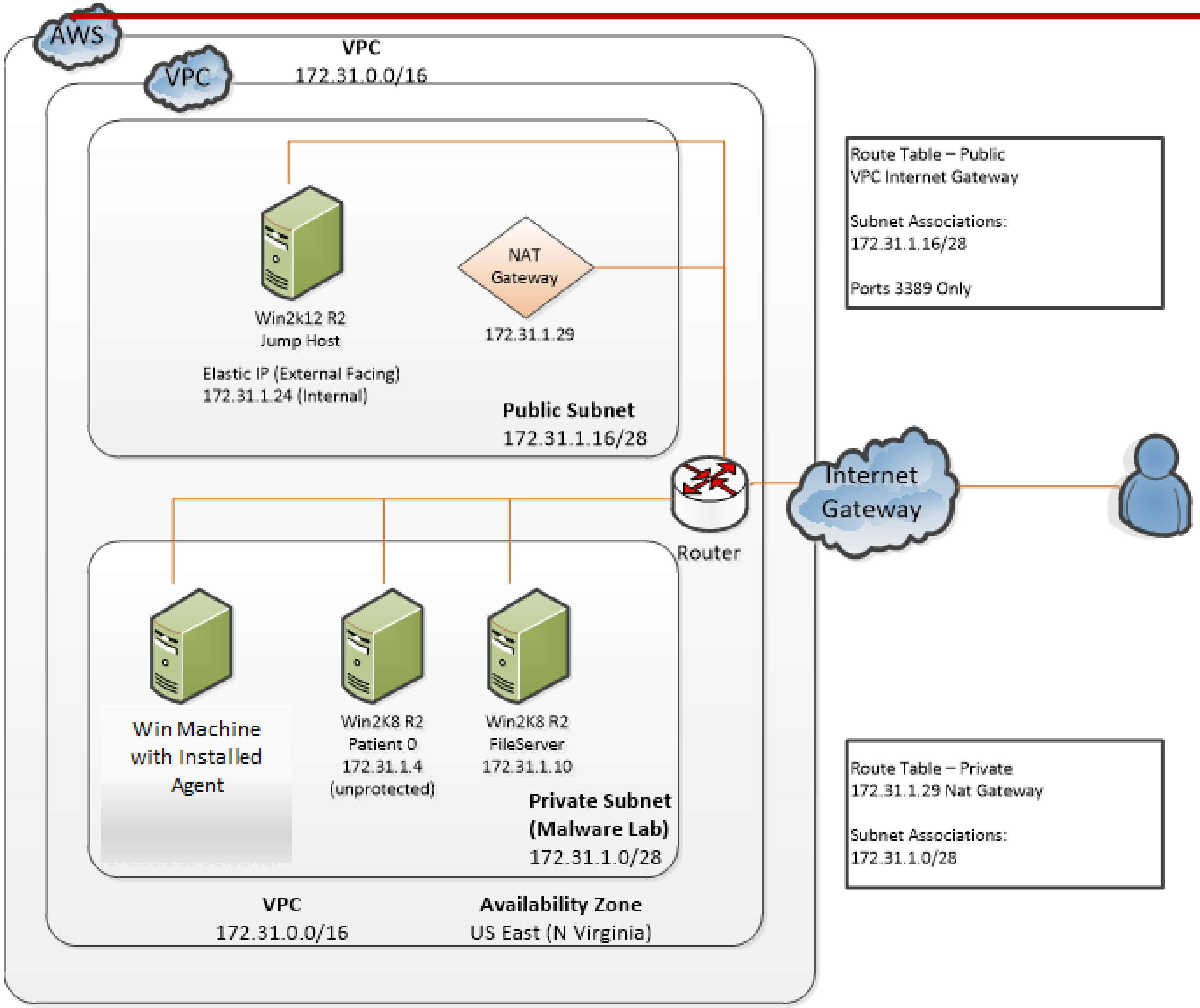
Recommend to build your own test environment consisting of:

- Victim machine
- Attacker machine
- Malware machine
- Victim machine II which is connected to Victim 1

Considerations for virtual environments:

- Not all VMs will execute malware in the same way
- VirtualBox (for example) compared to KVM, AWS or VMWare will all behave differently
- Consider vendor cloud setups, convenience yes, ability to compare solutions side by side no.

Testing Environment



Route Table – Public
VPC Internet Gateway

Subnet Associations:
172.31.1.16/28

Ports 3389 Only

Route Table – Private
172.31.1.29 Nat Gateway

Subnet Associations:
172.31.1.0/28

- All our test machines were fully patched with the EPP agent installed on them.
- Vendor worked with us to create the prevention policies either in their SaaS environment or virtual servers.
- We used their environment to validate and monitor only; no settings were changed.

Testing Recommendations

Recommendations:

- Test the different layers of the software, and then disable each layers to determine tight-coupling constraints
- Connected agents and non-connected agents
- If your organisation has different user profiles for different roles, consider testing these to check for different results
- When testing malware, as it’s the easiest scenario, choose a variety, new, old, grey and don’t just mutate them x1, do it a few times different ways (packers, hash modifiers)
- Test more than binaries try other file types such as .zip, .jar, .com, .vbs, change an ext, rename a file, .ps, false-positive directories

Where to source malware?

- In house / private collect / ask your forensic teams
- Github Repo
  - Maltrieve
  - theZoo
  - Malware-samples
- Other dedicated malware sites (subscribed / free)
  - VirusTotal
  - VirusShare
  - Malwr
  - TestMyAv
  - Malshare
  - MalwareDB
  - Malware Traffic Analysis
  - AlienVault

Don’t be afraid to handle malware
Most are password protected

** Important to have a variety of families **

Test Case 1: Pre-Execution

Static Malware Testing (file exists)

- Focus on dormant files / no running processes
- Importance of background scanning
- File changes (on-write, modify, delete)

Output

- Detection
- Quarantine

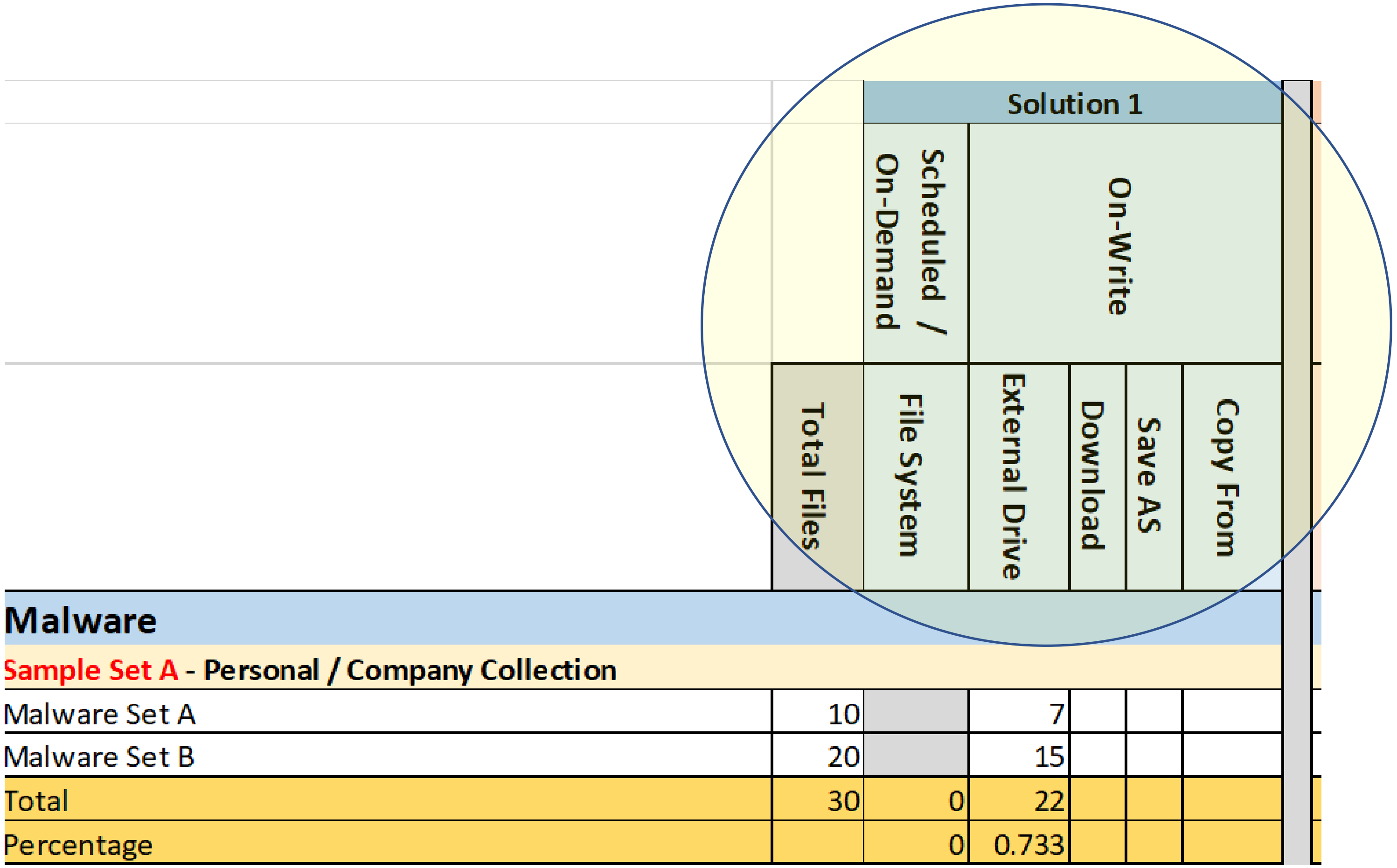
Testing Scenarios:

- Cloud / No Cloud Connectivity
- If business requirement, test different file introductions such as USB device, network copy, save as download etc.

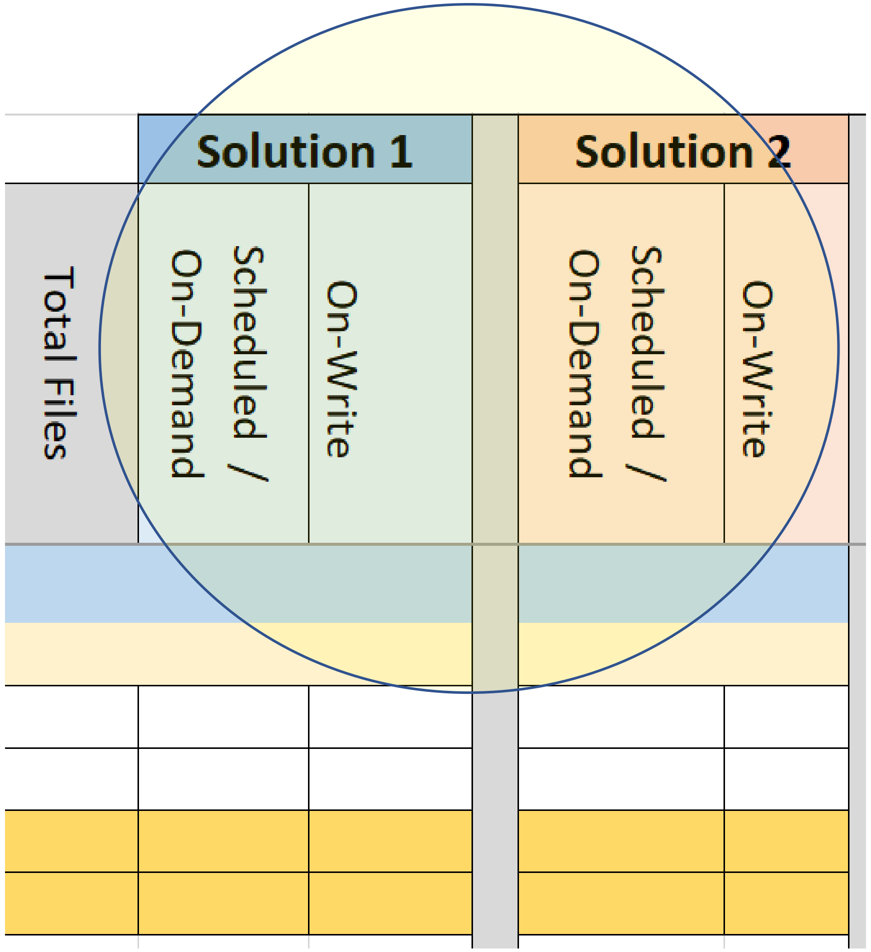
File Types:

- Don't just focus on PEs, try different file-types, .dll, .bin, .jar, .tar, .com, .ps
- Other file-types and scripts are not are available in pre-execution as yet.
- Sandboxing is not pre-execution

Pre-Execution Scoring Sample

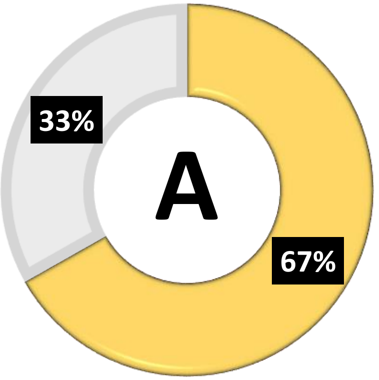


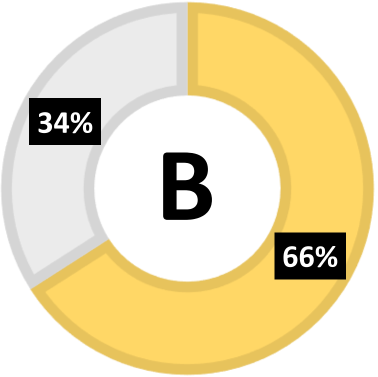
		Solution 1				
		Scheduled / On-Demand	On-Write			
	Total Files	File System	External Drive	Download	Save AS	Copy From
Malware						
Sample Set A - **Personal / Company Collection**						
Malware Set A	10		7			
Malware Set B	20		15			
Total	30	0	22			
Percentage		0	0.733			

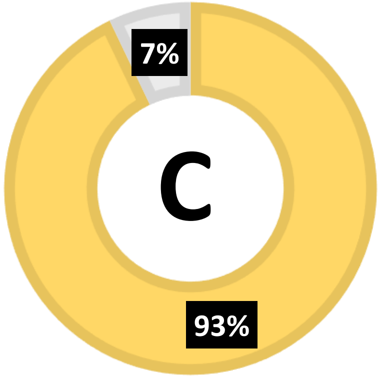


	Solution 1		Solution 2	
Total Files	Scheduled / On-Demand	On-Write	Scheduled / On-Demand	On-Write

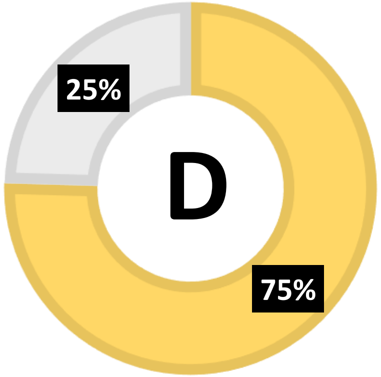
Pre-Execution - Original

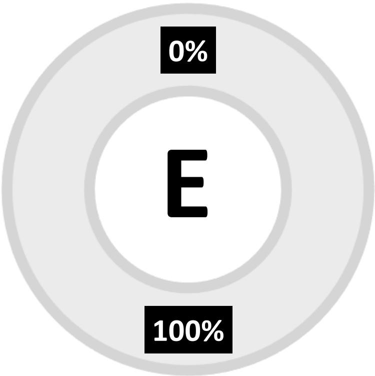






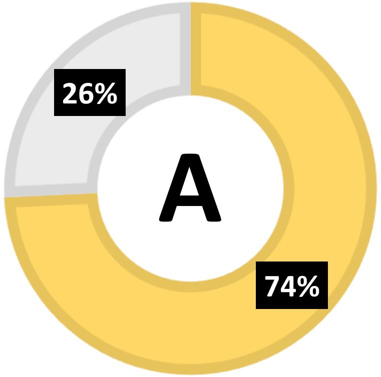
~20,000 samples used

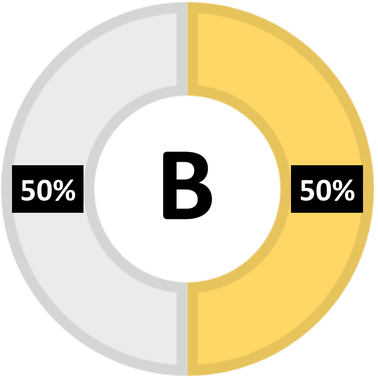


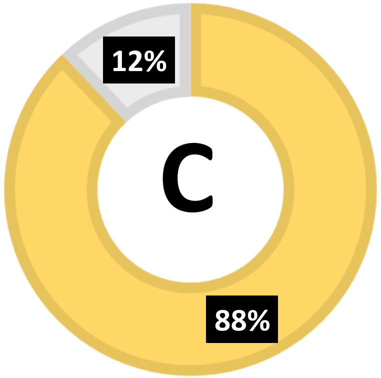


🟨 Quarantined
⬜ Not Quarantined

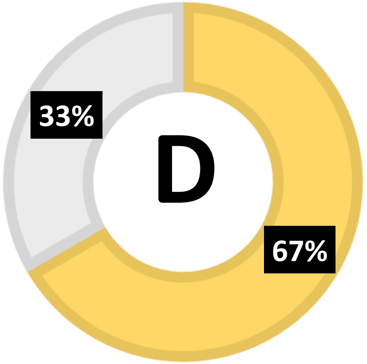
Pre-Execution - Mutated

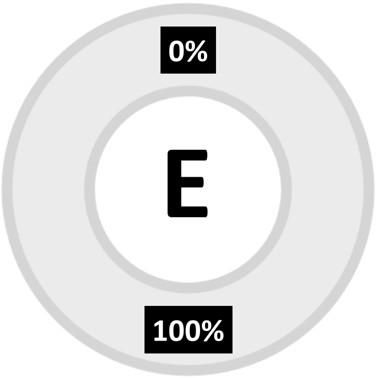






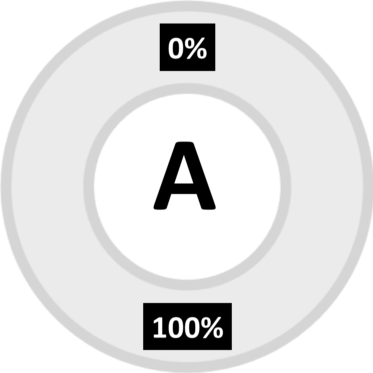
~20,000 samples used

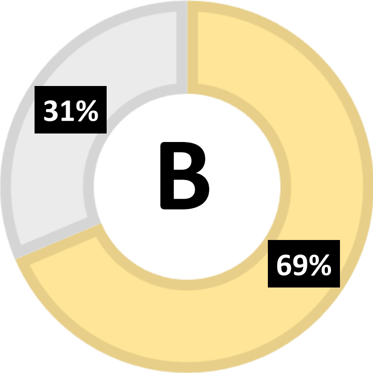


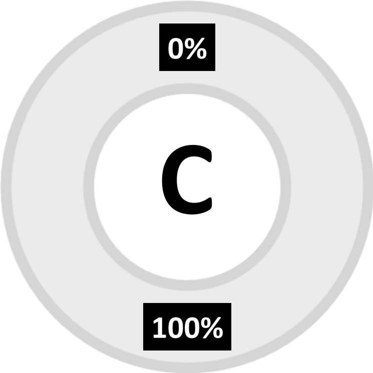


🟨 Quarantined
⬜ Not Quarantined

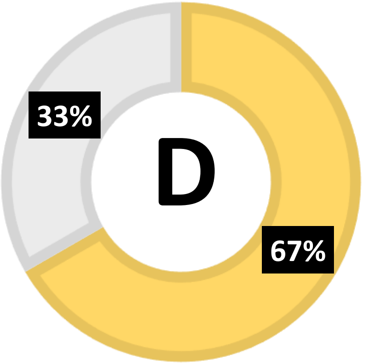
Pre-Execution – Other Types

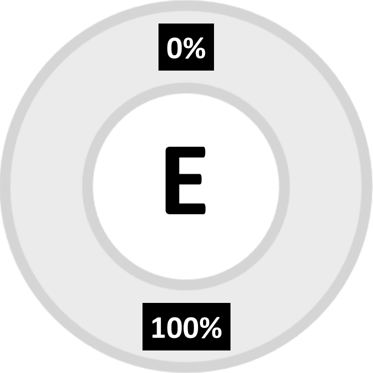






~50 samples used





🟨 Quarantined
⬜ Not Quarantined

Test Case 2a: Execution

Dynamic Malware Testing

- *Turn off pre-execution module*
- Execute malware

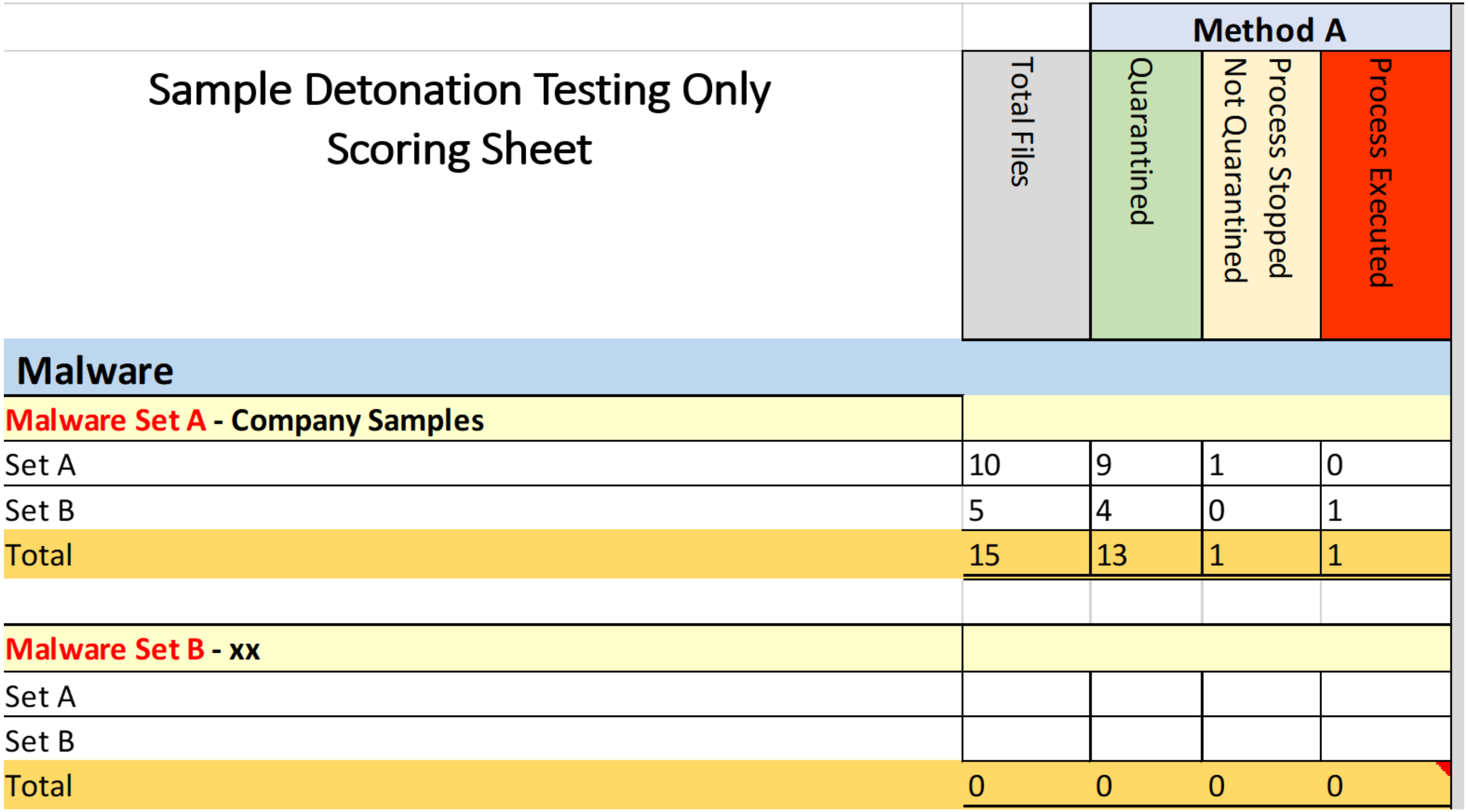
Output

- Detection
- Quarantine

Testing Scenarios:
- Cloud / No Cloud Connectivity
- Test Malware by bulk if wanted to test effectancy (such as the for loop) of the agent

- Don’t use focus on one mode of execution such as a for loop executing binaries
- Hide your malware behind scripts, rename your powershell scripts, rename files, use whats naïve to that operating system

Execution Scoring Sample



Sample Detonation Testing Only Scoring Sheet		Method A		
	Total Files	Quarantined	Process Stopped Not Quarantined	Process Executed
Malware				
Malware Set A - Company Samples				
Set A	10	9	1	0
Set B	5	4	0	1
Total	15	13	1	1
Malware Set B - xx				
Set A				
Set B				
Total	0	0	0	0

Execution Demo

Execution of Malware Demo

Scenario:

- Pre-execution engine disabled
- 100 pieces of malware executed sequentially using a loop within a **PowerShell** script

Test Case 2b: Execution

Dynamic Malware Testing

- *All modules enabled*
- Execute malware

Output

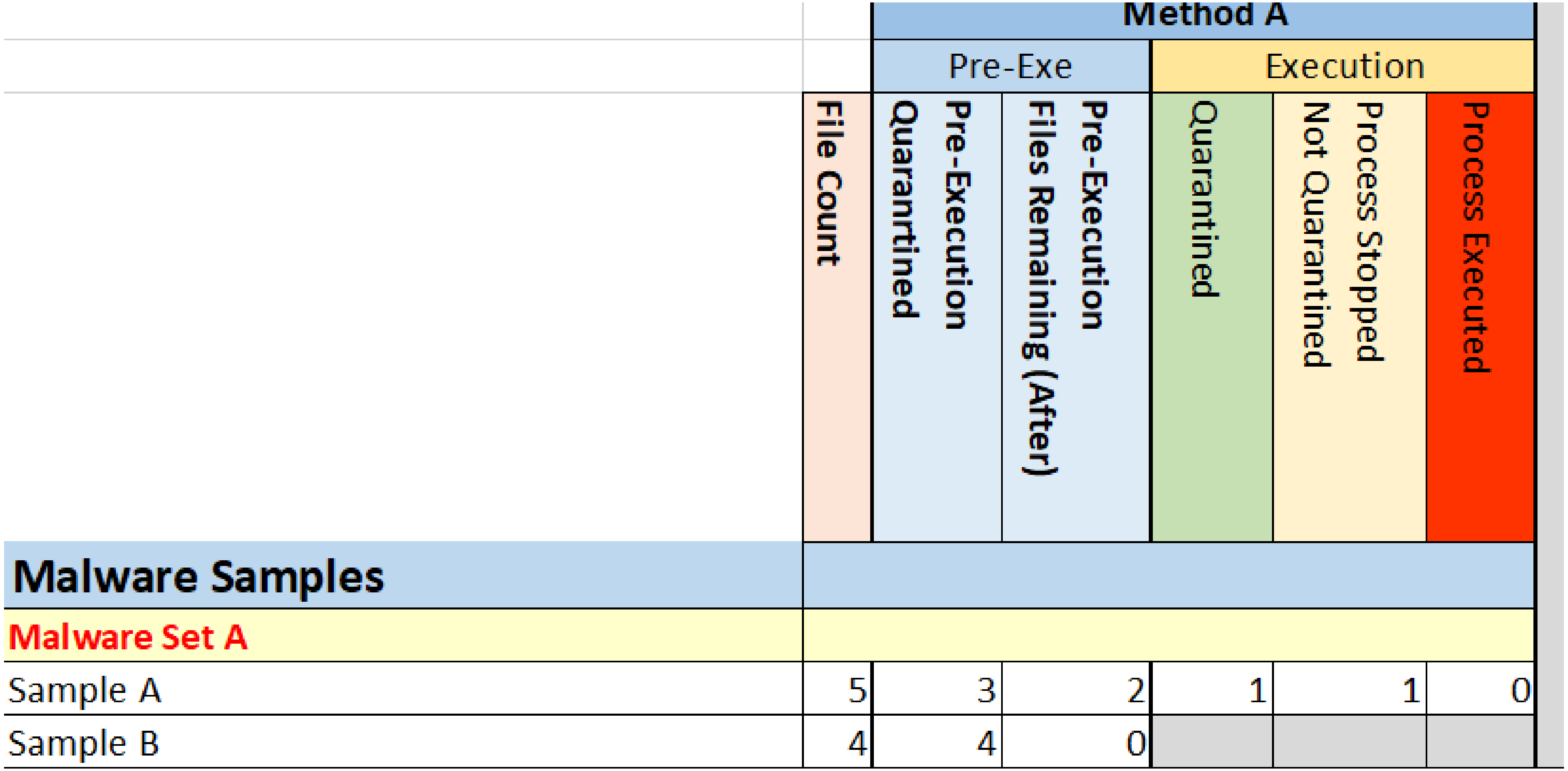
- Detection
- Quarantine

Testing Scenarios:

- Cloud / No Cloud Connectivity
- Test Malware by bulk if wanted to test effectancy (such as the for loop) of the agent

- Don't use focus on one mode of execution such as a for loop executing binaries
- Hide your malware behind scripts, rename your powershell scripts, rename files, use whats naïve to that operating system

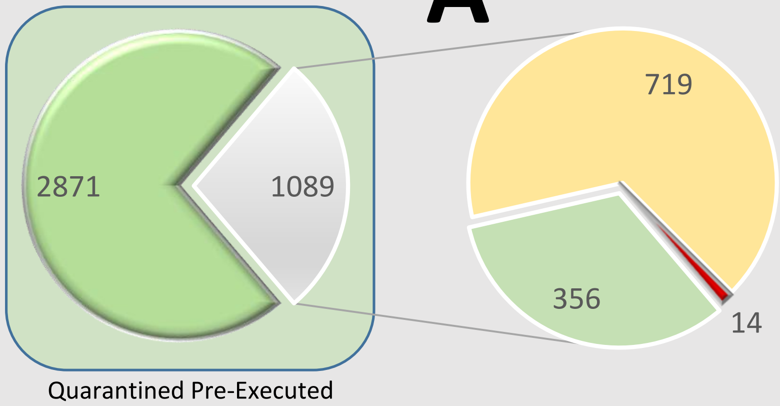
Execution Scoring Sample



		Method A				
		Pre-Exe		Execution		
	File Count	Pre-Execution Quaranrtined	Pre-Execution Files Remaining (After)	Quarantined	Process Stopped Not Quarantined	Process Executed
Malware Samples						
Malware Set A						
Sample A	5	3	2	1	1	0
Sample B	4	4	0			

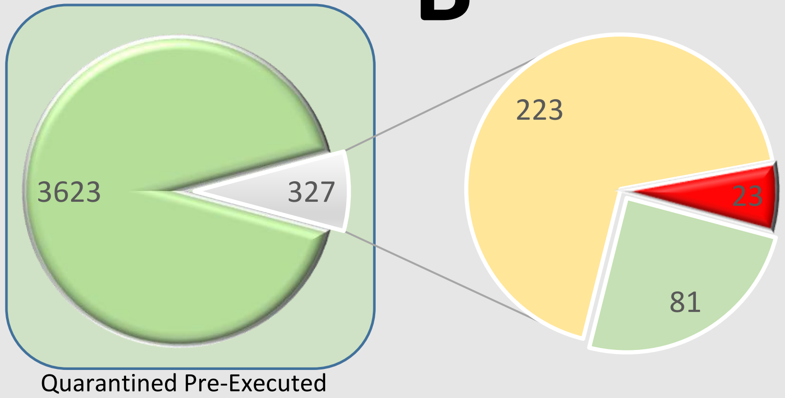
Execution – Known Malware

A



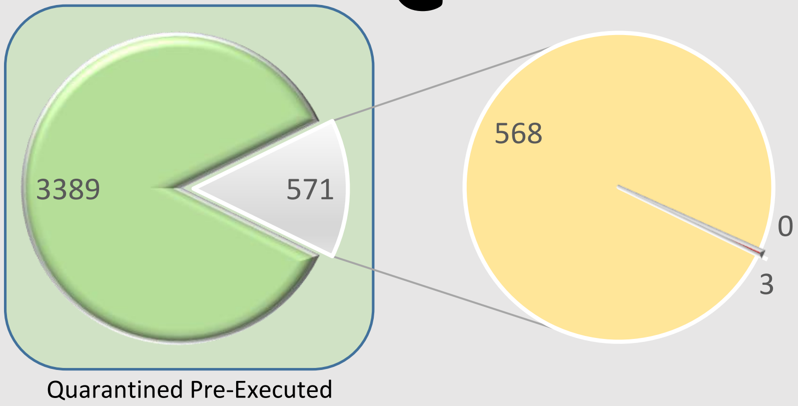
Of the malware which infected this EPP, re-tested 3 times over a period of 3 weeks. There appeared to be no machine learning or behavioral changes.

B



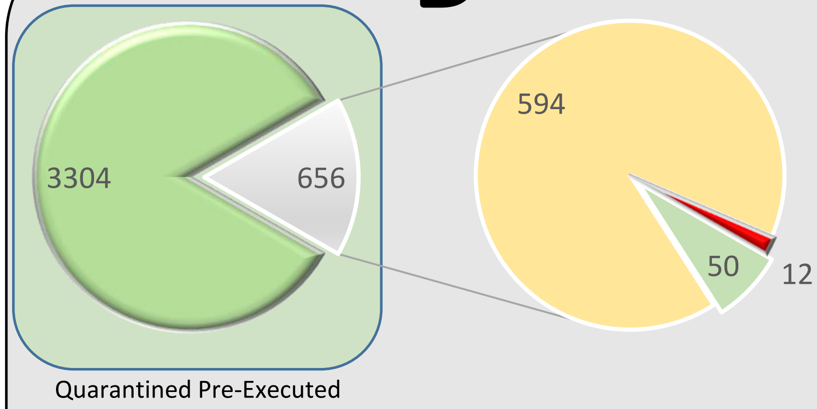
Of the Malware executed this caused the machine to be shutdown 3 times.
No learning or behavioral changes after re-testing.

C



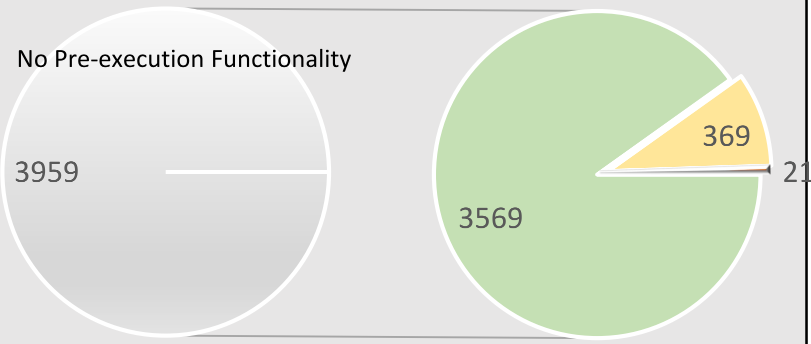
Only a few PUP software executed

D



Only a dozen PUPs executed

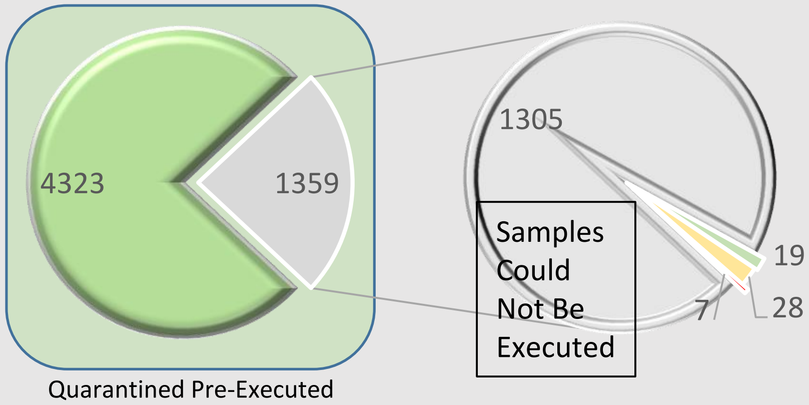
E



Of the malware executed during the first pass, none were executed during re-testing

Execution – Unknown Malware

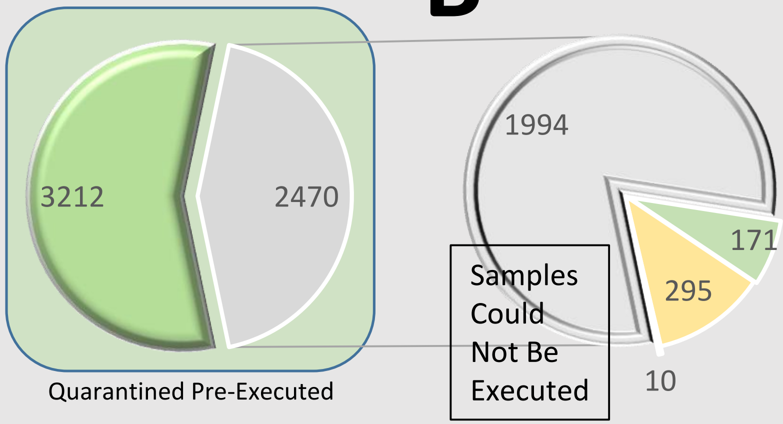
A



Re-testing showed no difference in results.

Not all files tested due to the system being white screened or shutdown.

B

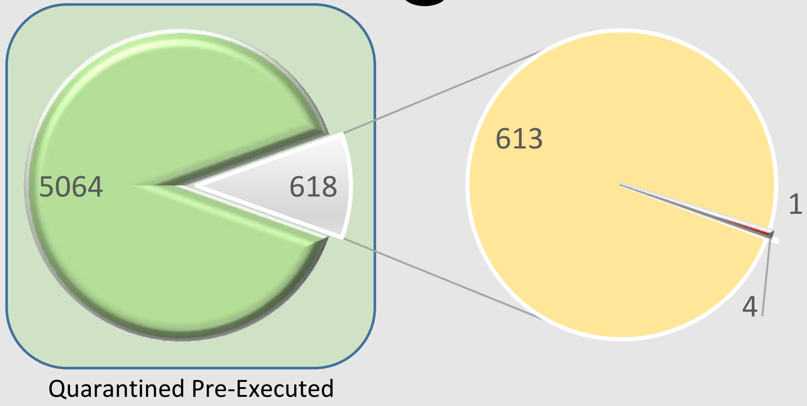


Very slow executing mutated samples.

Remaining malware caused the machine to shutdown 4-5 times.

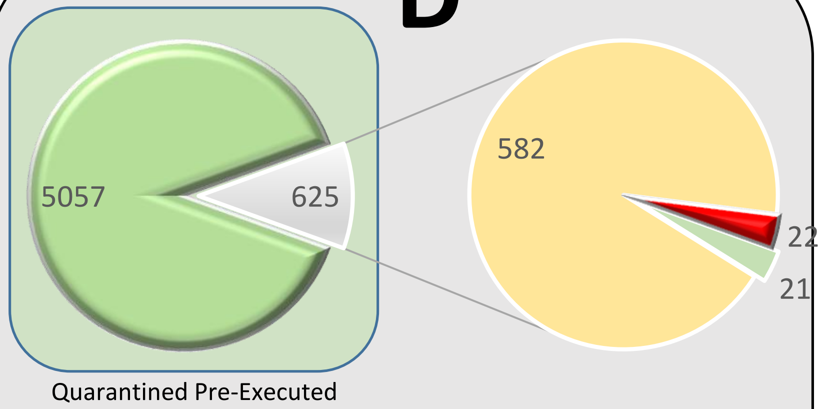
No change when re-testing.

C



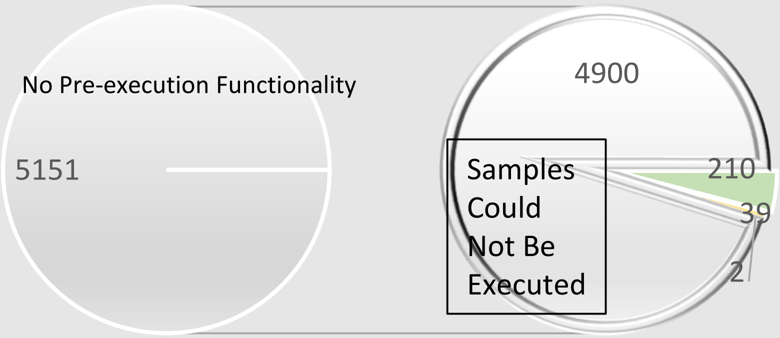
Agent quarantined or stopped almost all mutated samples.

D



Slow execution of mutated samples. Agent protected against most of the malware with machine shutdown 1-2 times.

E



Of the mutated sets, this agent machine learning ability was strong. However, the last sample set caused multiple machine shutdown.

All Capabilities Execution Demo

All Capabilities Enabled Demo

Scenario:

- 100 pieces of malware executed sequentially using a via **the command line**
- 100 pieces of malware were mutated two times using two different methods to change their hash values
- Machine is “double-ransomwared”

Execution Takeaways

Pros and Cons for the Loop

+ Test efficiency of the agent
+ Good stress test
+ Performance Test
- Hard to know which piece executed
- Cross Contamination
- Lots of rebuilding
- You can’t deep dive

Other Takeaways

- Mutate files using different methods
- Test the different components to determine tight coupling
- Sandboxing had lots of difficulty
- Re-tested mutated files weeks later demonstrated no difference in results. ML?

Test Case 3: Post-Execution

Dynamic Malware Testing

- *Pre-Execution Modules Disabled*
- *Depending on use case, turn off preventive controls for testing*
- Execute malware
- Was it stopped?
- Was it allowed to be installed?
- What data was seen?
- What additional information?

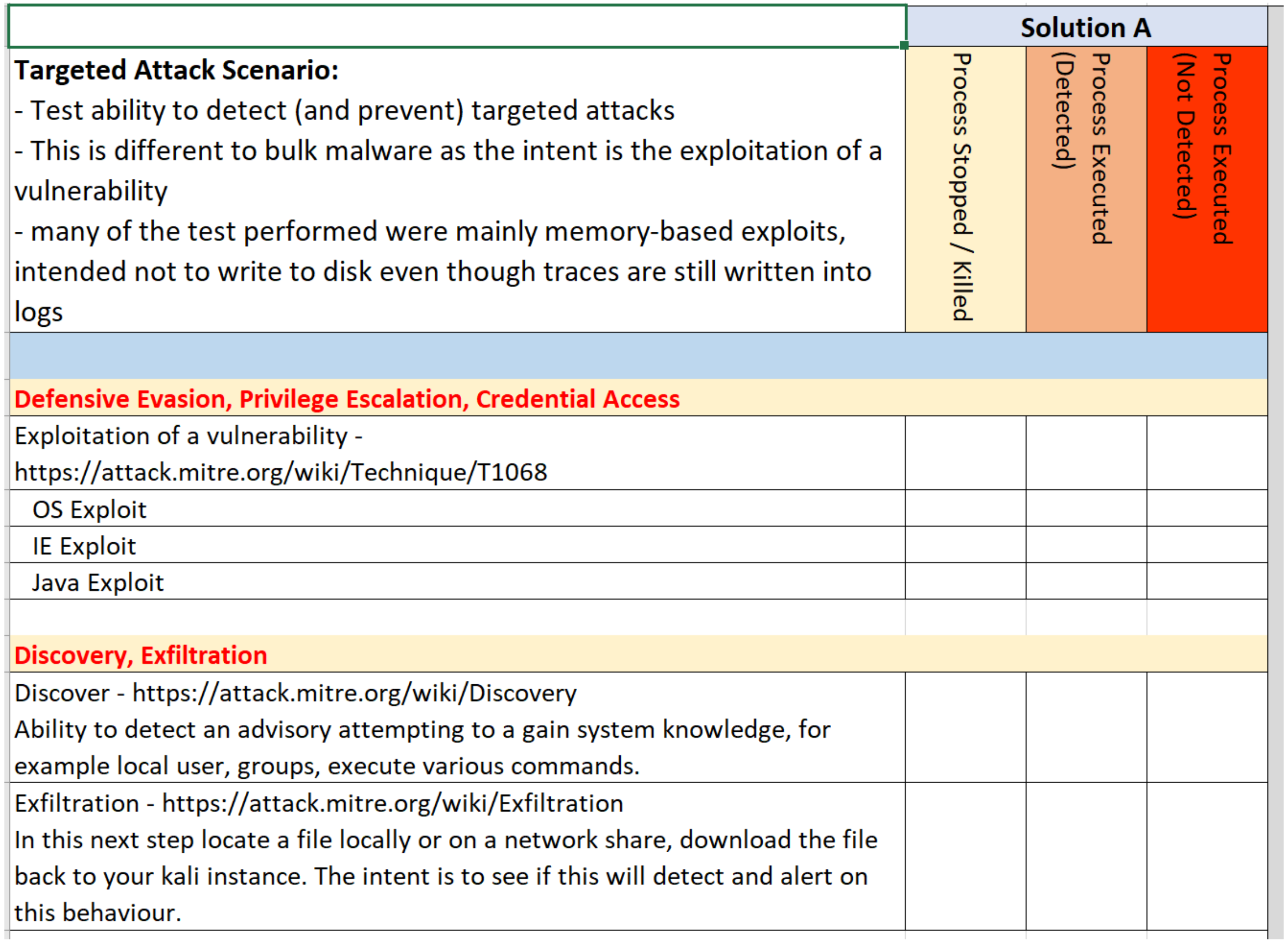
Output

- Quarantined or Process did not execute *(optional)*
- Detection
- Process Information
- IoA and IoC

Testing Scenarios:

- Cloud / No Cloud Connectivity
- Individual Pieces of Malwares
- Keep it varied such as browser exploits, embedded macros, phishing links, weaponised attachments etc

Exec & Pre-Exec Sample Scoring



	Solution A		
Targeted Attack Scenario: - Test ability to detect (and prevent) targeted attacks - This is different to bulk malware as the intent is the exploitation of a vulnerability - many of the test performed were mainly memory-based exploits, intended not to write to disk even though traces are still written into logs	Process Stopped / Killed	Process Executed (Detected)	Process Executed (Not Detected)
Defensive Evasion, Privilege Escalation, Credential Access			
Exploitation of a vulnerability - https://attack.mitre.org/wiki/Technique/T1068			
OS Exploit			
IE Exploit			
Java Exploit			
Discovery, Exfiltration			
Discover - https://attack.mitre.org/wiki/Discovery Ability to detect an advisory attempting to a gain system knowledge, for example local user, groups, execute various commands.			
Exfiltration - https://attack.mitre.org/wiki/Exfiltration In this next step locate a file locally or on a network share, download the file back to your kali instance. The intent is to see if this will detect and alert on this behaviour.			

Post-Execution Demo

Scenario:

- Using Kali (attacker), exploit MS11-003 used against victim running an unpatched version of IE
- Victim gets link and clicks
- Attacker takes advantage of vulnerable IE and obtain a reserved shell on the victim's machine
- Attacker start recon

Post-Execution Takeaways

- Most challenging to test
- Many solutions based their detection on a static rule set
- Many of the EDR solutions are weaker for commodity malware detection
- If prevention rules did not trigger, validate the solution was able to detect the malware
- Threat feeds vs threat intelligence
- IoA vs IoC

<u>Tools</u>

- Red Canary Scripts
  https://github.com/redcanaryco/atomic-red-team
- Kali Payloads
- Links to Phishing Campaigns
- Create your own scripts
- Mitre ATT&CK Adversarial Tactics, Techniques and Common Knowledge

  https://attack.mitre.org/wiki/Main_Page

Test Case 4: Environment

Finally, run the solution in your environment for a good few weeks. Consider the following:

- Don’t test malware in this scenario
- Agent sitting in silent mode on a few devices
- Testing malware is important but how compatible is it with your actual work environment
- How many FPs are being generated? Is it the way your company pushes packages, or admins write update scripts?

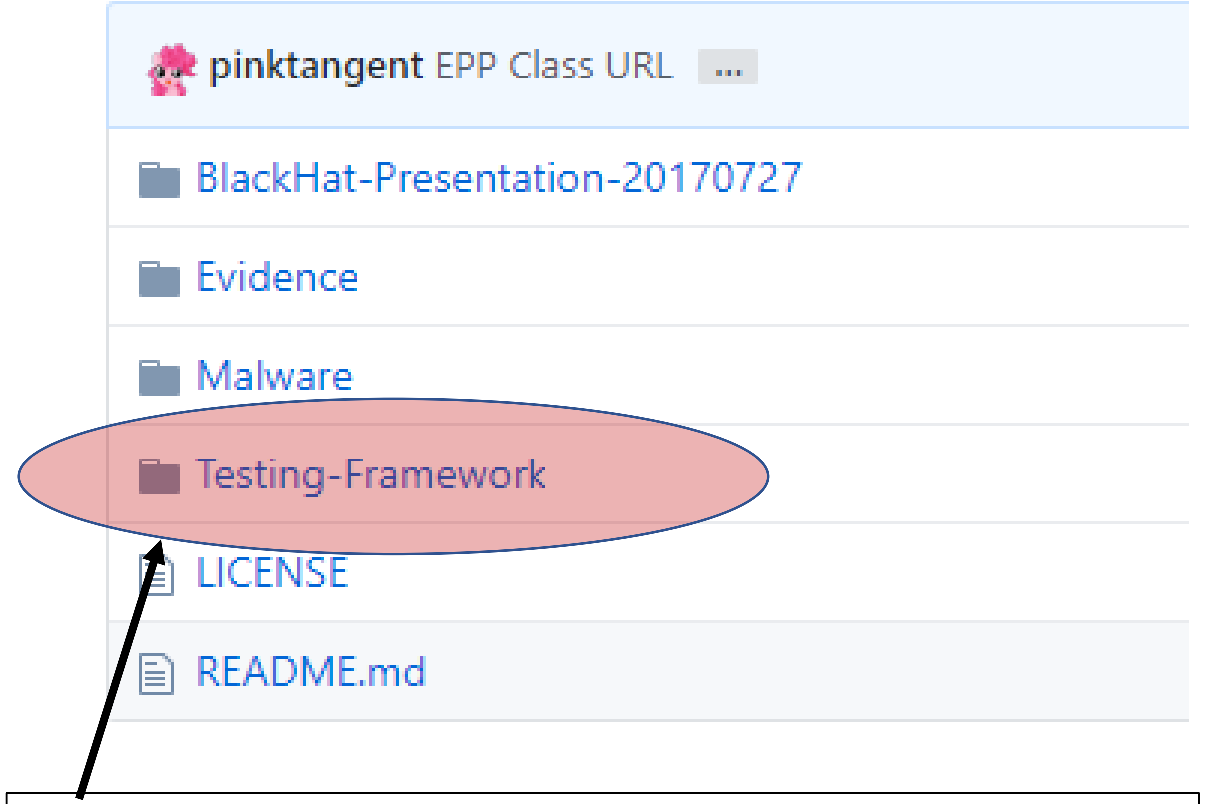
- MFA
- AD Integration
- SIEM Integration
- API’s
- Large Scale Deployment
- Log Storage and Retention $$
- Backward Compatibility
- Agent Reboot

*** If vendors push you to a tight testing timeline, then reconsider ***

Shameless Plug

SECTOR
Security Education Conference

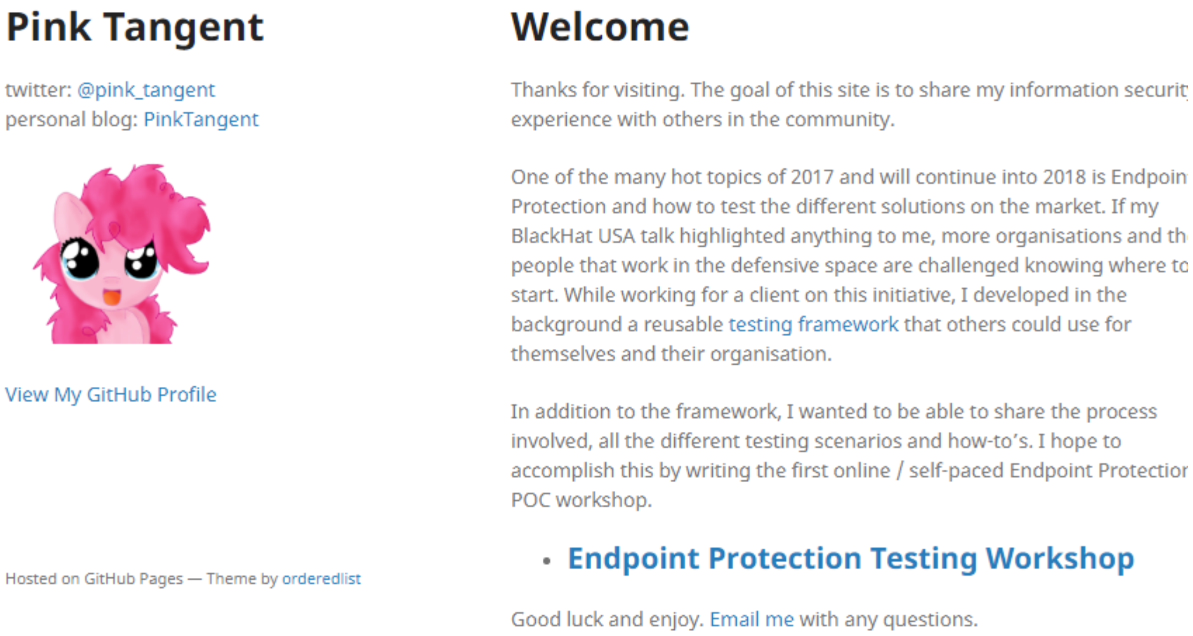
https://github.com/pinktangent/Endpoint-Testing



pinktangent EPP Class URL ...

- BlackHat-Presentation-20170727
- Evidence
- Malware
- Testing-Framework
- LICENSE
- README.md

Continuously evolving and updating content



Pink Tangent

twitter: @pink_tangent
personal blog: PinkTangent

View My GitHub Profile

Hosted on GitHub Pages — Theme by orderedlist

Welcome

Thanks for visiting. The goal of this site is to share my information securit experience with others in the community.

One of the many hot topics of 2017 and will continue into 2018 is Endpoin Protection and how to test the different solutions on the market. If my BlackHat USA talk highlighted anything to me, more organisations and th people that work in the defensive space are challenged knowing where to start. While working for a client on this initiative, I developed in the background a reusable testing framework that others could use for themselves and their organisation.

In addition to the framework, I wanted to be able to share the process involved, all the different testing scenarios and how-to's. I hope to accomplish this by writing the first online / self-paced Endpoint Protectior POC workshop.

- **Endpoint Protection Testing Workshop**

Good luck and enjoy. Email me with any questions.

https://pinktangent.github.io/ - In Progress

DIY Workshop – To get started

Summary

- There is not one solution which is better than the other
- It comes down to business need and requirements
- Problem you are trying to solve
- Not all have machine learning, many are augmenting with static rules and signatures
- Don’t be scared to denotate malware
- You don’t need to be a PenTester to do this
- Test but verify
- There is not silver bullet

Questions