STOP CHASING SECURITY

BREACHES WILL HAPPEN

Plan an approach; act accordingly

October 3, 2019

DEFENSES FAIL

Can you demonstrate – right now – that your enterprise isn’t breached?

How would you answer that question today? How about your staff?

Common Answers I Hear:

- Our defenses don’t tell us there is a problem, thus we have no problem.
- We modernized our defences; our EDR/EPP assures us they have us covered.
- This isn’t important – we use defense in depth strategy.
- We have spent $X and have adequate coverage to deal with any problems.
- We don’t have to worry about it – we have cyber Insurance
- We simply can’t.

SOME RECENT FAILURES I'VE SEEN

NONE of them neglected their defences

Large Retail Group

Scan of 10,000+ endpoints resulted in the discovery of hardware compromise on 400 endpoints in their retail operations. Client uses a leading EDR Solution + many layers of defence.

Ministry in Malaysia

Scan of 1000 endpoints performed to validate the effectiveness of their current defences – cryptominers, password scrapers, and more discovered in < 30 minutes.

Bank in India

Scan of 150 endpoints in a a Mumbai branch – discovered unknown ransomware on payment gateway that had not yet activated.

Oil and Gas Entity

Scan of 2000 endpoints of environment with leading EDR solution and many defensive layers. Discovered four memory injects that turned out to be malicious.

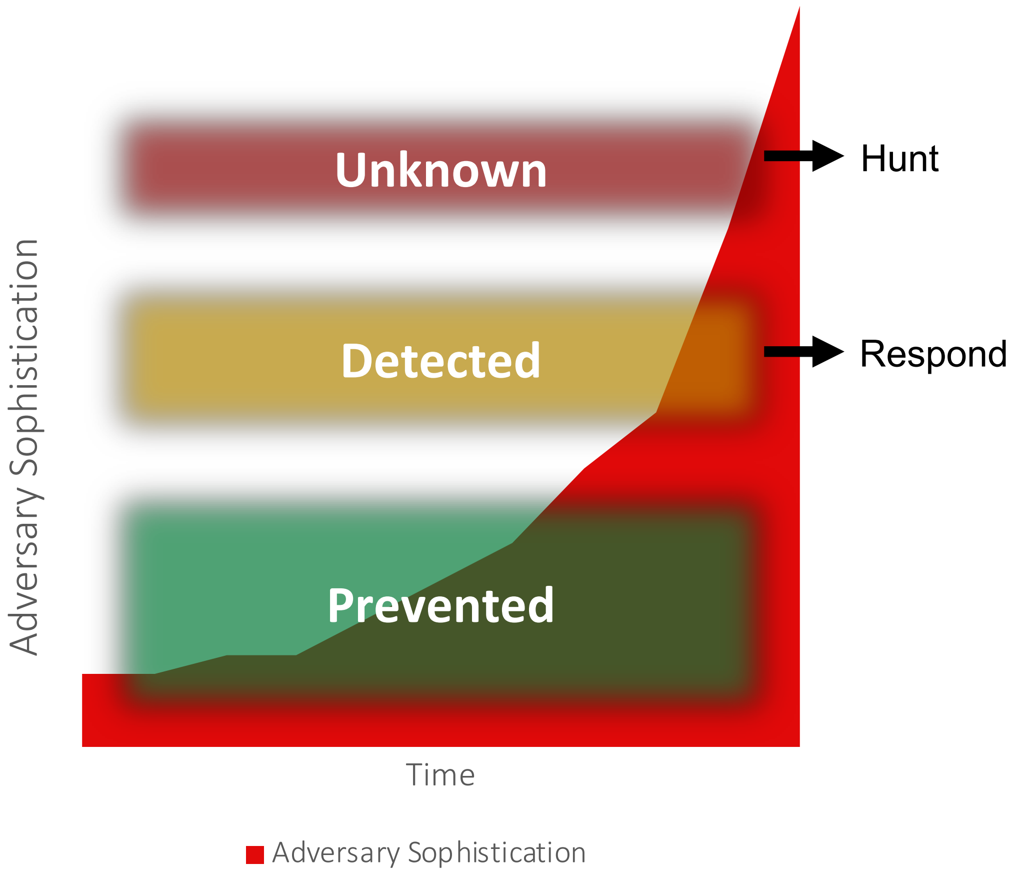
Holding Company in KSA

Scan of 500 endpoints after a ransomware attack occurred; discovered memory injects which later (2 days later) were successfully attributed to ATP33 thanks to IOCs from Florian Roth.

MALWARE HUNTING VS THREAT HUNTING

Malware hunting is a subset of broader threat hunting

- What is threat hunting?
  - Process of proactively and iteratively searching through networks (and endpoint) to detect and isolate advanced threats that evade existing security solutions.
- So what is malware hunting?
  - Malware hunting is a set or series of workflows specifically designed to find software (malicious and unwanted) that has breached existing defences and is running undiscovered in the estate.



WHAT THE MARKET TRIES TO SELL

Repurposing Defences: Big Data Analytics, AI/ML and More

- Everyone's a threat hunter – so much noise
- Advocate log-based/event-based approach
  - Consolidate EDR data with traditional SEIM based data
  - Layer on Security Intelligence tooling/capabilities
  - Identify outliers and high quality leads missed by defensive tools
- Active Hunting: Hypothesis and then validate
  - Some of this is automated – as new attacks are discovered and IOCs documented, go back and comb through logs to see they were captured
- New Pivot: Managed EDR (Sales Tactic)
  - You're missing problems because you don't know how to use [Vendor A's] tooling, let us manage it for you.
  - When we see a problem everyone benefits.

Challenges

- Data Access
  - most organizations I know won't have the historical data required to do this.
- Data Completeness
  - there are no data quality guarantees and the source data will have gaps
  - without the current state of the compromise, we will always have an incomplete picture
- Independence
  - reliance on data from the very defensive tools/infrastructure that let malware breach to then go and find it

WHY DEFENSIVES TOOLS MAKE POOR HUNTERS

Protection Vs Hunt Solutions

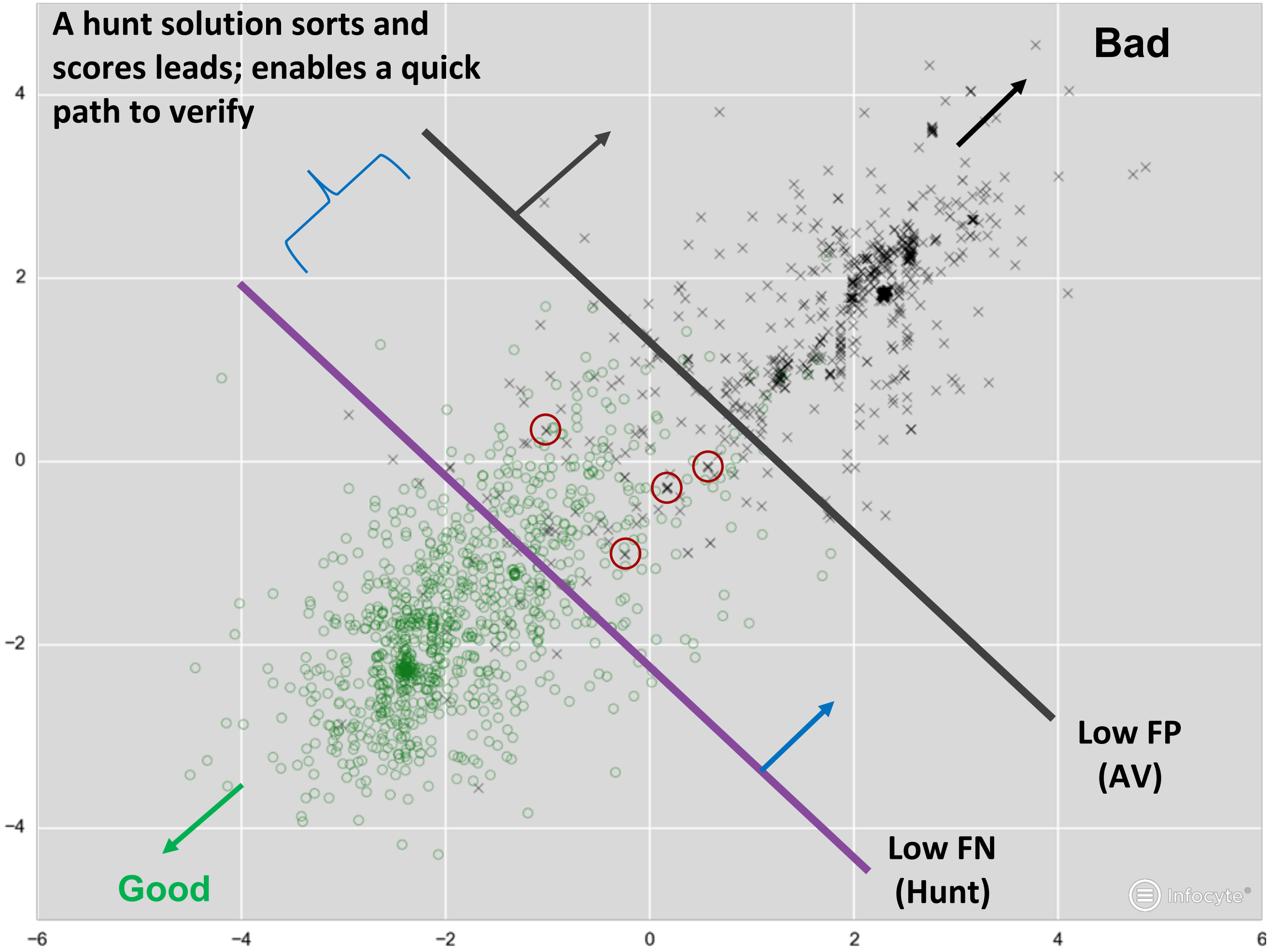
Protection solutions (AV/EDR) employ scoring models that tune for real-time detection of attacks while minimizing *False Positives (FP)*

- Must score and categorize within seconds!
- Those that add Behavior and static Analysis at the endpoint have to move even faster

Hunt solutions widen the scope and seek threats that made it through protection by minimizing *False Negatives (FN)*

- For Hunters: anomalies, outliers, and suspicious activity are leads, not FPs to be tuned out

Diagram: Sample output of ML-based malware categorization
Original Source: Crowdstrike



Positive in either axis = *more malware-like*
Anything near 0 is hard to differentiate, noisy

SO, MALWARE HUNTING…

WHERE DOES ONE START?

A look into the Four Emerging Malware Hunt Methodologies

FOUR EMERGING METHODOLOGIES

Script Based Hunting

- Involves the ability to query one or more logical groupings (or all) endpoints in real time to determine whether any indicators of known IOCs are in the live environment (right now)
- Generally requires an analyst know what they are looking for with ongoing education in TTPs.
- Has complexity and traps – and it tends to be labour intensive and time consuming
- Popular in large enterprise estates – piggy backs off whatever tooling may already be in place.

Commercial Options

- Tanium
- Sentinel One

Open/Free Tools

- OS Query
- PowerShell Remoting
- WMI
- Survey in Any Scripting Language

FOUR EMERGING METHODOLOGIES

IOC Based Hunting

- Involves running a one time scan or querying a generally dormant agent on endpoints for the purpose of validating for the presence of known IOCs
- Lots of great sources and sources of IOCs, but needs a discerning eye
- Requires knowledge of how to build YARA rules
- Suitable for experienced SOC environments, less so for general enterprise environments that don’t have the specialized skills.

Commercial Tools

- Thor ATP (Nextron Systems)

Open Tools

- Loki
- Many Other Yara Rules Engines

FOUR EMERGING METHODOLOGIES

Repurposing of Forensic Solutions

- This is by far the most comprehensive method to make the determination, but it has its drawbacks and limits:
  - No bread crumb trail to follow
  - Not highly scalable (think ROI when done iteratively)
  - Enough complexity that it required highly skilled resources

Commercial Tools

- FTK
- Encase
- Google GRR
- Fidelis

Open Tools

- Tons of options out there

FOUR EMERGING METHODOLOGIES

Forensic State Analysis (FSA)

- Automated workflows collect live host forensic data:
  - Memory (volatile and non-volatile)
  - Persistence Mechanisms
  - Forensic Artifacts (Shimcache, Prefetch, Amcache)
- Data is then analyzed using a variety of post-breach analytics techniques, reputational, and multiple threat intelligence sources.
- Combining live host forensic data and analytic techniques, FSA determines the compromise state of endpoints.

Commercial Tools

- Infocyte HUNT

Open Tools

- PSHunt
- PSForensics
- And Tons More…

FORENSIC STATE ANALAYSIS

WHAT IT REALLY LOOK LIKE?

A closer look into the framework promise

FORENSIC STATE ANALYSIS PROGRAM

Five Functional Components

Scanners and Surveys

Discovery Capability or Engine

Deployment (Transport Execution and Functions)

Survey Analysis

File Analysis

FORENSIC STATE ANALYSIS PROGRAM

Scanners and Surveys

Need a method or means to rapidly query or survey an endpoint to collect data from one or more systems in a consistent way.

Scanners

- Similar to script hunting, this adds an ability to make a single remote query to an endpoint over an available service to collect a single piece of data
- Good for one-off instances where you need to check for presence of an IOC (registry key, maybe events, or more) across hundreds of systems fast. (Know what you're looking for.)

Surveys

- Surveys are little programs (any language) or scripts (any flavour) that will collect data/information from an endpoint and post/sent it back to a central place.
- Good for when you want to collect a lot of data for later enrichment and review (don't know what you're looking for.)

I personally tend to work in Windows heavy environments and rely on WMI or PowerShell remoting for Scanner type functionality and I tend to write my surveys in .Net (C#).

FORENSIC STATE ANALYSIS PROGRAM

Additional Notes for Surveys

Some pointers on how to go about building a survey.

Recommended Minimum Data Sets

- Active Processes
- Loaded Modules/Drivers
- Floating/Injection Modules
- Active Connections
- Persistence Mechanisms
  - Scheduled Tasks (don't forget AT jobs), Registry run keys, Windows Start Folders / Jump lists
- Key Event Logs
- Accounts
  - Local and AD (if you are storing things in a DB)
- Artifacts
  - Shimcache, Prefetch, Amcache

Pointers:

- Get-WmiObject -Class Win32_Process
  - + Get-Process –Module
  - + Get-Hashes
  - + $Process.GetOwner()
- PSForensics has better libraries you can leverage (reading from disk rather than OS is awesome way to get access to amcache)
- Registry access is challenging with powershell – super easy with .Net
  - Don't forget you'll need to load user hives – not just current user
  - Amcache has big leaves – might need a custom hive reader for that (.Net doesn't have that support out of the box)
  - Could wrap Sysinternals Autorunsc (best opensource collection of autostart locations).

FORENSIC STATE ANALYSIS PROGRAM

Discovery Engine

Discovery is concerned with how we can identify endpoints within our estate – in effect we need a port scanner that is able to not only scan for services but validating entitlements. We require is basically an enumeration engine.

Method/Way to Define Targets

- Require an enumeration capability that can do discovery by:
  - IP Range
  - CIDR (if you have large network spans)
  - Pull from Active Directory
    - Hint: add group support, not just machine discovery
  - Lists (IP/Hostnames)

Discovering and Testing Access

- Ports/Services
  - TCP 22 – SSH
  - TCP 135 – WMI/POC
  - TCP 137 – NetBIOS
  - TCP 139 – SMB over NetBIOS
  - TCP 445 – SMB
  - TCP 5985 – PS Remoting

Not up to creating a service/program? Can use powershell – Test-TCPPort, Test-TCPPorts, Get-RemoteArchitecture, Get-RemotePowershellVersion, Get-RemoteOperating System. Or – use third party tools like Dsquery, Powersploit-> Recon, PowerView

FORENSIC STATE ANALYSIS PROGRAM

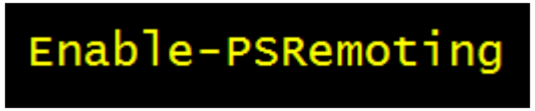
Transport Execution and Functions

With knowledge of where the endpoints are (and validation we can access them) coupled with the ability to collect data, we now need a way to execute our queries or deploy our surveys – and collect the results.

Execution Methods

- WMI (Process call create, echo/execute)
- PSRemoting (Invoke-Command)
  - Almost never enabled – but you can for your environment
- Remote Task Scheduler (Schtasks)
- Remote Service Manager (PsExec)
- SSH (linux mainly)

Protip: type this in every windows box you see:



```
Enable-PSRemoting
```

My Personal Preference

- Create a dropped that has a hard-coded download path with a custom parameter for survey:
  - https://myServer/survey + mySurveyName.exe
- Then convert survey to base64 encoding
- Echo out to a file over WMI
- Use certutil to convert back into EXE
- Execute
- Post results back to web service
- Remove Survey from endpoint

FORENSIC STATE ANALYSIS PROGRAM

Survey Enrichment

With data retrieved, now it’s time to start going through the data and enriching it with third-party intelligence or intelligence that you’ve built up internally.

Create a Local Store

- Create a local reputation store so that as you identify and decide on whether things pose a threat, you can apply that on a move forward basis.

Threat Intel

- Threat Intel Provider
  - Start with Virus Total (free)

Other Sources

- NIST RDS Hashsets
  - https://www.nist.gov/itl/ssd/software-quality-group/nsrl-download/current-rds-hash-sets
- NIST Vulnerability Listings
  - https://nvd.nist.gov/vuln/full-listing
- AT&T’s Open Threat Exchange
- Regional MISP Projects
  - In UAE we have 971Sec
- Commercial Options
  - A lot of vendors have reasonably priced file reputation or threat intel feeds – some not so reasonable. Shop it around if this is an option.

FORENSIC STATE ANALYSIS PROGRAM

File Analysis

Sometimes you need to have a closer look at files and get a better understanding about what they do and whether they pose a risk – files can include memory injects if you unmapped them in your survey.

Submit to Internal or External Services

- There are a lot of tools that can help in this space, some can be automated and some manual – in the spirit of scalability favour automation when you can.
- You'll generally do file retrieval with a query (back to that single purpose engine).
- If have no budget, better to get results than be blind – Virus Total is your friend.

Word of Caution

If you submit to external sources, remember that the bad guys monitor those too. This is one of the areas worth investing in. You can generally submit unmapped memory, but it's not a good idea to submit retrieved samples.

Multi-Av Scanner:

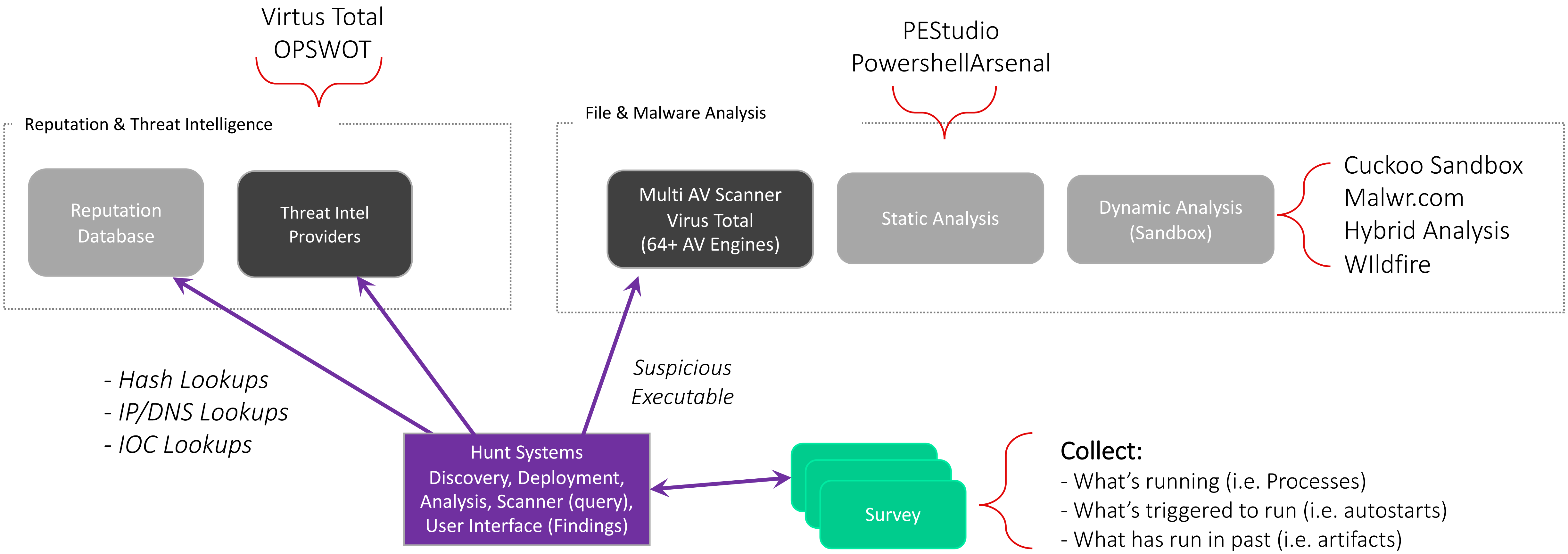
- Opswot

Sandboxing:

- Hybrid Analysis (Crowdstrike), Palo Alto Wildfire, Joe Sandbox, and more

FORENSIC STATE ANALYSIS PROGRAM

Bringing it all together



Virtus Total
OPSWOT

Reputation & Threat Intelligence

Reputation Database

Threat Intel Providers

PEStudio
PowershellArsenal

File & Malware Analysis

Multi AV Scanner
Virus Total
(64+ AV Engines)

Static Analysis

Dynamic Analysis
(Sandbox)

Cuckoo Sandbox
Malwr.com
Hybrid Analysis
WIldfire

- Hash Lookups
- IP/DNS Lookups
- IOC Lookups

Suspicious Executable

Hunt Systems
Discovery, Deployment,
Analysis, Scanner (query),
User Interface (Findings)

Survey

Collect:
- What’s running (i.e. Processes)
- What’s triggered to run (i.e. autostarts)
- What has run in past (i.e. artifacts)

BENEFITS OF MALWARE HUNTING

This is a lot of work, what are use cases?

4 On-Demand Use Cases

- Heavy investments made in defences and just want to validate that nothing has in-fact slipped through.
- Suspect something is wrong, but not sure and want to validate with independent tooling
- Know something is wrong (maybe endpoint is beaconing) but existing endpoint defences can't pinpoint what is causing it.
- Hit by malware (ransomeware is common) and need to validate no secondary malware was seeded.

Business Impact

- Ongoing scans is important – introduces a new control – ability to define and manage dwell time.
- Early discovery reduces business impact
  - Aberdeen group research sponsored by Secureworks using raw data from Verizon Data Breach report and some Monaco modeling showed:
    - 30 Days of Dwell Time results in a business impact reduction of up to 24%
    - 7 Days of Dwell Time results in a business impact reduction of up to 77%
    - 1 day of Dwell Time results in a business impact reduction of up to 97%

> “Stop Chasing Security
> Accept Malware will Breach
> Independently Validate State
> Keep it Iterative and Ongoing”

SPEAKER BACKGROUND

What is my background?

- Canadian with 17+ year working internationally in developing, frontier, and emerging markets – currently based in the United Arab Emirates
- Working with Virgin Mobile UAE as the Head of Cyber Security
- Consult and Advise to several technology startups – two uncloaked ones:
  - CSAT – cyber maturity assessment tool
  - Infocyte – Post breach Detection Tool using FSA Methodology
- Consult and advise governments
  - Solve Really Tough Problems – look for the patterns – from contested spectrum to cyber security
- Always searching for the next thing
  - Early adopters in security, if picking right, reap the greatest rewards
- Guide enterprises on technology alignment using a functional capabilities frame

FINDING STUFF (TIME PERMITTING)

Active Processes, Modules, Drivers

Some malware (even advanced types) attempt to hide in plain sight or with the noise of many programs running on your systems. With the data we have collected we can reduce this noise and then start finding things that don't belong.

- Initial Technique
  - Hash everything (done at collection in the survey) and compare to a signature and threat intel database like Virus Total (query the hash, see what you can find). This will clear all known good and bad, reducing the noise.
- Advanced Techniques
  - Stack remaining data and perform anomaly and outlier analysis
    - If a scan of 400 endpoints has been done and I have something that remains that's only on one or two, let's start with those
  - Perform static/dynamic analysis on any suspicious or out of place processes
    - This will often give rapid clearing of futher noise, leaving only a handful left that need proper/real attention

FINDING STUFF (TIME PERMITTING)

Persistance Mechanisms

Malware normally has to maintain persistence – and there are a lot of common (and less common) ways that authors do this. You won’t find them all, but over time you’ll build a large repository of mechanisms and it will keep improving your ability to perform the discovery. New techniques/mechanisms can be added to your surveys.

- Scheduled Tasks, AT Jobs, etc
  - The survey should collect all AT Job and scheduled task executables (easy through API’s) and hash the executables
- Registry Persistence (most common, still in 2019)
  - The survey should be configured to hash all identified executables in the registry hive (or at least known persistence paths) so that they can be queried against reputation and threat intel available to remove/reduce noise
- Boor Process Redirection
  - Evaluate the MBR file (first 512 bytes of disk) for redirection to an alternate boot loader

FINDING STUFF (TIME PERMITTING)

Process Memory Injection

DLL Injection / Process Hallowing:

1. Allocate chunk of unprotected Read/Write/Execute (RWX) memory inside another legitimate process.
2. Load in a malicious DLL.
3. Redirect an execution thread.
4. Profit.

•*Adv. Technique:*

- Walk Process Memory looking for PE Headers in <u>large</u> chunks of unprotected memory (Use @mattifestation's PSReflect)
  - False Positives will come from:
    1. Just-in-Time (JIT) compilers – i.e. .NET and Java Apps
    - 2. Security Software