IOCs: Indicators of CRAP

Xavier Ashe

VP, Security Engineering | SunTrust

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or plans of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results may differ materially. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only, and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionalities described or to include any such feature or functionality in a future release.

About Me

Hacking since the late 80's

Working in Security since the early 90's

10+ years working with Splunk in the Security space

Have worked for Gartner, IBM, Carbon Black, independent consultant, and a few Startups

Currently head of Security Engineering for an Atlanta based bank

Native language is smart ass

Twitter: @xavierashe

https://www.linkedin.com/in/xavierashe

What is an IOC?

Indicators of Compromise

Things that Indicate

…

Compromise

End of Presentation

splunk> .conf19

Where did IoCs come from?

TREND: COMPLEX INDICATORS ARE MORE LIKELY TO DETECT UNKNOWN APT-RELATED ACTIVITY

Detecting the APT is incredibly difficult and many organizations are not prepared to effectively identify that they have been compromised. In most cases, initial notification of an APT intrusion originated from a third-party, primarily law enforcement. The primary reason organizations fail to identify the APT is that most of their security devices examine inbound traffic at the perimeter. Most organizations rely solely on anti-virus solutions to provide host-based monitoring. In addition, implementing the ability to monitor internal to internal communications on a network is costly and challenging. In both instances, being able to respond quickly and to deploy APT indicators is difficult, as organizations' security arsenals are not configured to monitor using this methodology.

Host- and network-based signatures used to detect malicious activity have previously consisted of data like MD5, file size, file name, and service name, etc. Although useful, the lifespan of these type of signatures is often short because attackers can routinely modify their malware to avoid detection. Although those signatures will periodically work to identify attacker activity, MANDIANT has found greater success in adapting specific signatures into what are known as Indicators of Compromise ("IOC" or "indicators").

These indicators not only look for specific file and system information, but also use logical statements that characterize malicious activity in greater detail.

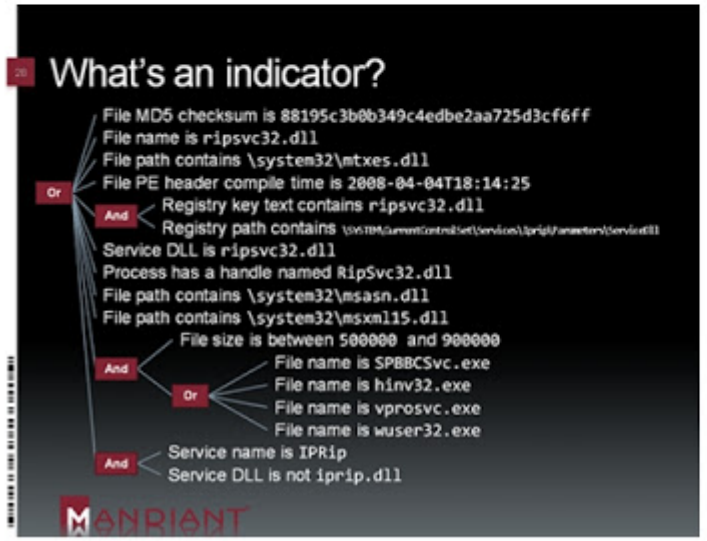
MANDIANT has determined that the majority of APT custom-developed tools typically contain code segments from other, similarly developed malware. The code segments could also be upgrades to previously identified malware. Indicators derived from this information remain fairly consistent between the various malware and their subsequent upgrades. Victims are more likely to detect APT-related activity using code segments when it is possible new APT malware might be used. In many cases, previously unidentified malware and backdoors were identified through the use of these indicators in both network traffic and host-based information.

The combination of both host- and network-based indicators continues to be the most reliable way to identify APT-related malware on a network. In two separate investigations, network-based information from a generic packed file transfer revealed suspected malicious activity. Upon further research, the file transfer was identified as malicious activity that was then immediately validated through the use of host-based indicators and forensic analysis.

Historically, compromise data has been exchanged in CSV or PDFs laden with tables of "known bad" malware information - name, size, MD5 hash values and paragraphs of imprecise descriptions supplemented by ad-hoc exchanges between targets.

MANDIANT, inspired by field pressures, operation after operation, imagined a way to exchange not only indicators of specific compromises but structures which formalize the human-intelligence of decision-making, rules, exceptions, and ongoing adaptability. Our Indicators of Compromise (IOCs) were shaped operationally detecting real-world threats. We help our clients detect the APT right now, and they're exchanging information about it using IOCs.

Conventional compromise datasets consist of table after table of immediately-stale data capturing few, if any, relationships. An Indicator of Compromise (IOC), however, is a Boolean decision tree that discriminates an indicator from a false-positive, theory from ground truth. What's more, when you discover an exception or extension to a well-known-IOC you can describe it concisely and proactively, authenticate its source and re-evaluate your existing data to detect new instances of old compromises. This way, as a threat group adapts to your detections, you retain an IOC's identity and maintain the value of intelligence shared with other targets over time.

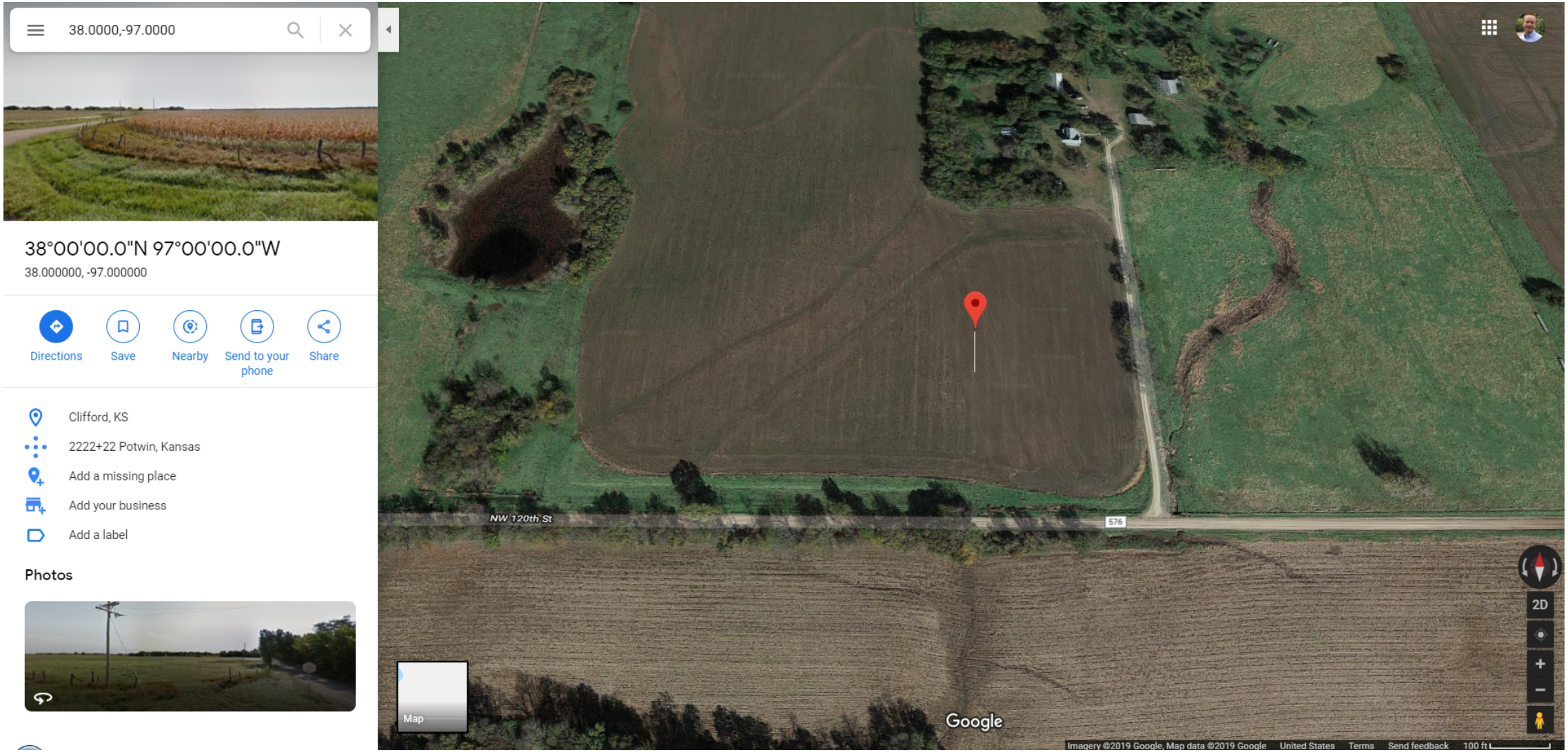


Source: https://taosecurity.blogspot.com/2018/11/the-origin-of-term-indicators-of.html

IOC Bonanza!

- Hashes
- IP Addresses
- Domain Names
- Mutex Names
- URL
- File Names
- File Path
- Email Addresses
- Usernames
- Passwords
- Registry Keys
- Registry Values
- Email Subject Lines
- TLS Certificate Serial Numbers
- Service Name
- Coin Address
- MAC Addresses
- Strings
- Geolocation

A Note on Geolocation



Source: https://splinternews.com/how-an-internet-mapping-glitch-turned-a-random-kansas-f-1793856052

How can I get me some IOCs?

Steal them from the internet
- (Community Supported)

Pay for them
- (Customer Supported)

Get Lucky
- (You already paid for them)

Make them Up
- (Generate your own)

Curating & Collecting

MISP

CRIT

OTX - Open Threat Exchange

- AlienVault

ThreatConnect

PulseDive

OpenIOC

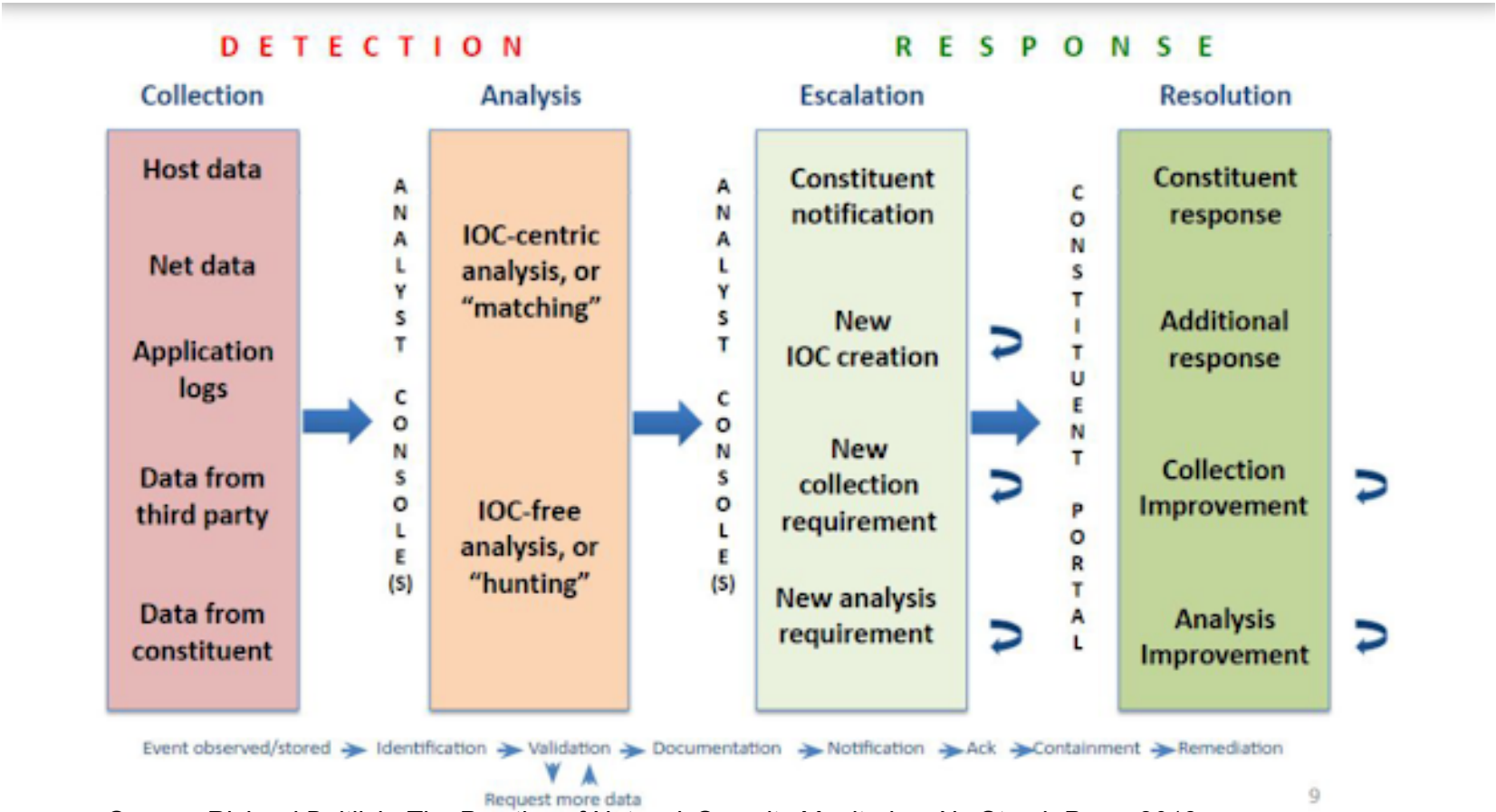
TAXII & STIX

MAEC

CybOX (defunct)

IOC Lifecycle

How to do Threat Hunting With IOCs



Source: Richard Bejtlich, The Practice of Network Security Monitoring, No Starch Press 2013

Indicator Matching (SOC)

Probably Have

IP

Domains

URLs

Email Addresses

Email Subject lines

Usernames

Rare

Registry Key and Values

TLS Cert Serials

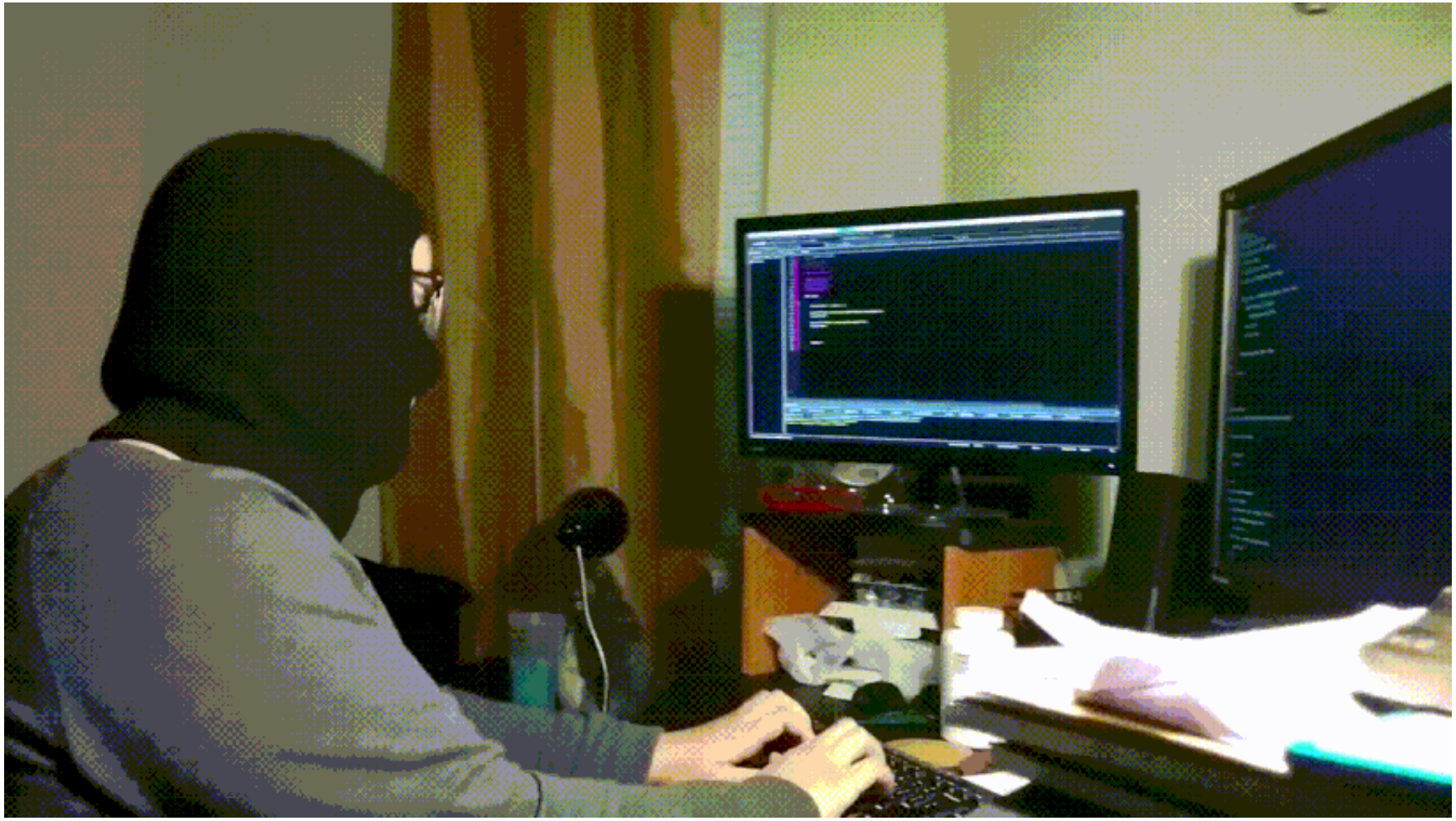
Services

Coin Address

MAC Address

File Hashes

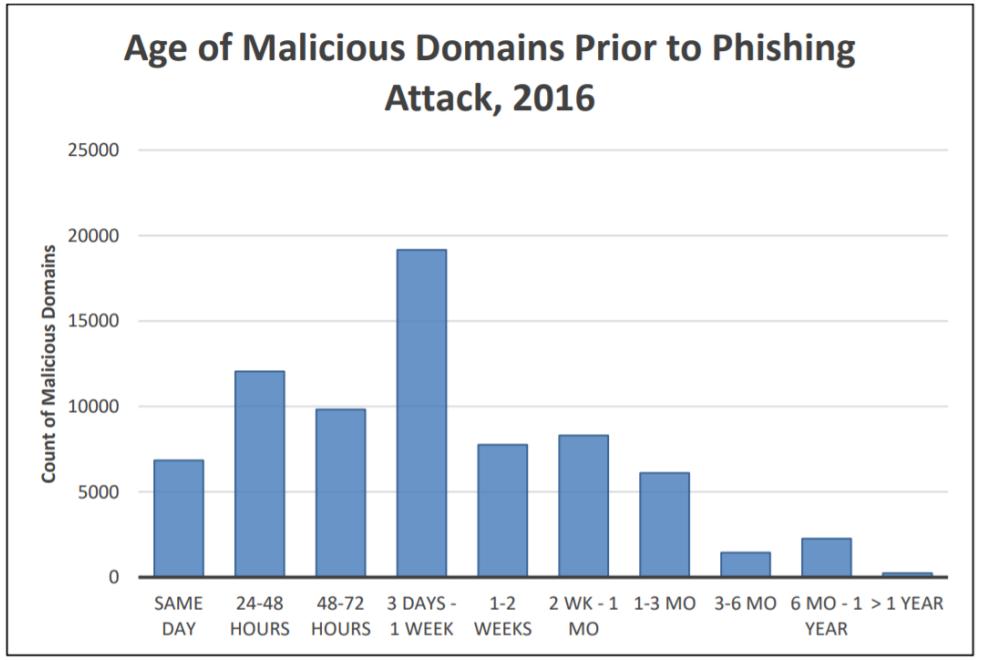
Indicator Matching (DFIR)

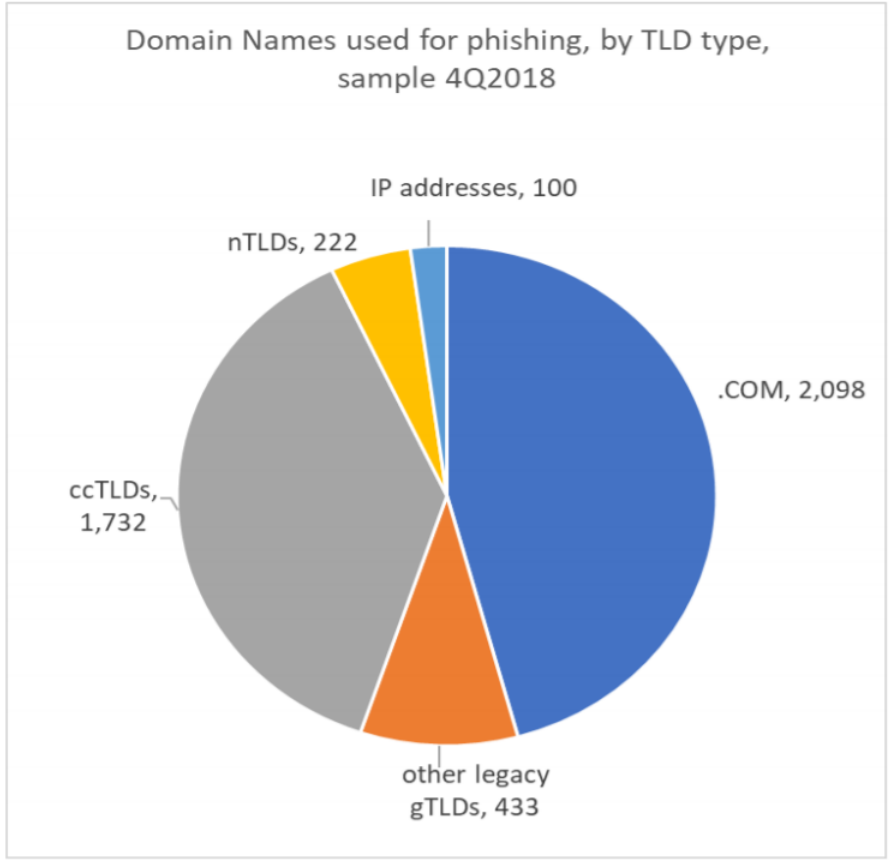


IP Addresses

Domain Names

.com is still king
New domains are evil





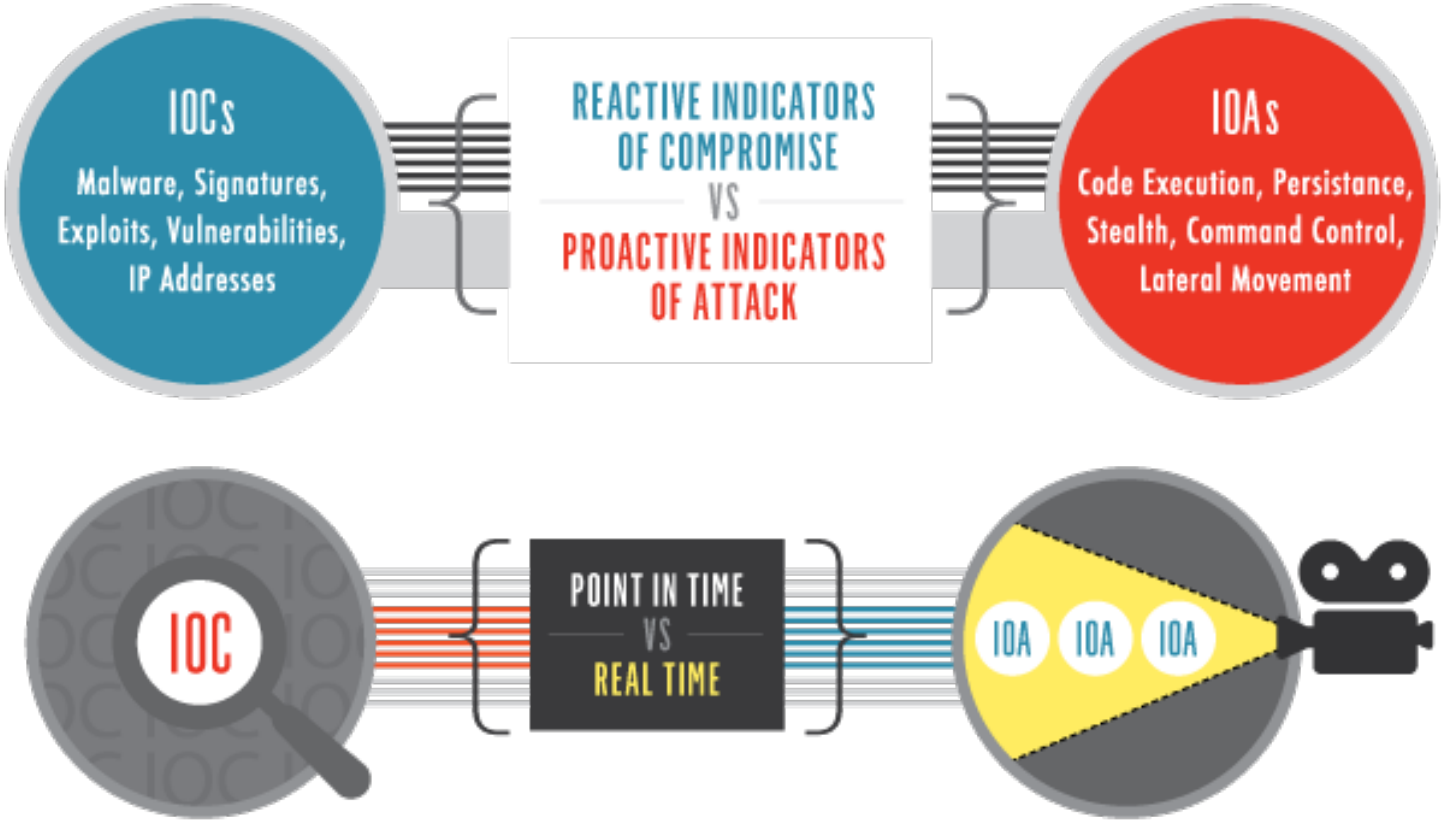
1: Phishing Activity Trends Report 4th Quarter 2018, Anti-Phishing Working Group, Inc.
2: Global Phishing Survey 2016: Trends and Domain Name Use, Anti-Phishing Working Group, Inc.

File Hashes

Behaviors of Compromise

- Increase network usage
- Unusual privileged account activity
- Geographical Irregularities
- HTML Response Sizes
- Applications using the wrong port
- DNS Request Anomalies
- FBI calls and tells you you've been pwned

Indicators of Attacks



Source: https://www.crowdstrike.com/blog/indicators-attack-vs-indicators-compromise/

Techniques, Tactics, and Procedures (TTPs)

Enterprise Tactics

ATT&CK™

Enterprise Tactics: 12

ID	Name	Description
TA0001	Initial Access	The initial a
TA0002	Execution	The executi conjunctior network.
TA0003	Persistence	Persistence to maintain restart or al
TA0004	Privilege Escalation	Privilege es require a hi access and access can objective m
TA0005	Defense Evasion	Defense ev variations c set of attrib
TA0006	Credential Access	Credential a environmer

Enterprise Techniques

Enterprise Techniques: 244

ID	Name	Description
T1156	.bash_profile and .bashrc	`~/.bash_profile` and `~/.bashrc` are executed in a user's context when a new shell opens or when a user logs in so that their environment is set correctly. `~/.bash_profile` is executed for login shells and `~/.bashrc` is executed for interactive non-login shells. This means that when a user logs in (via username and password) to the console (either locally or remotely via something like SSH), `~/.bash_profile` is executed before the initial command prompt is returned to the user. After that, every time a new shell is opened, `~/.bashrc` is executed. This allows users more fine grained control over when they want certain commands executed.
T1134	Access Token Manipulation	Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it belongs to someone other than the user that started the process. When this occurs, the process also takes on the security context associated with the new token. For example, Microsoft promotes the use of access tokens as a security best practice. Administrators should log in as a standard user but run their tools with administrator privileges using the built-in access token manipulation command `runas`.
T1015	Accessibility Features	Windows contains accessibility features that may be launched with a key combination before a user has logged in (for example, when the user is on the Windows logon screen). An adversary can modify the way these programs are launched to get a command prompt or backdoor without logging in to the system.
T1087	Account Discovery	Adversaries may attempt to get a listing of local system or domain accounts.
T1098	Account Manipulation	Account manipulation may aid adversaries in maintaining access to credentials and certain permission levels within an environment. Manipulation could consist of modifying permissions, modifying credentials, adding or changing permission groups, modifying account settings, or modifying how

Example

i	Time	Event
>	9/22/19 9:19:44.000 PM	09/22/2019 09:19:44 PM LogName=Security SourceName=Microsoft Windows security auditing. EventCode=4657 EventType=0 Type=Information ComputerName=[redacted] TaskCategory=Registry OpCode=Info RecordNumber=106610154740 Keywords=Audit Success Message=A registry value was modified. Subject: Security ID: [redacted] Account Name: [redacted] Account Domain: [redacted] Logon ID: [redacted] Object: Object Name: \REGISTRY\USER[redacted]\Software\Microsoft\Group Policy\Client\RunOnce Object Value Name: {F679194F-F34C-4C59-87C1-6BF157B6618F} Handle ID: 0x1af0 Operation Type: New registry value created Process Information: Process ID: 0x3b0 Process Name: C:\Windows\System32\svchost.exe Change Information:

ID: T1060

Tactic: Persistence

Platform: Windows

System Requirements: HKEY_LOCAL_MACHINE keys require administrator access to create and modify

Permissions Required: User, Administrator

Data Sources: Windows Registry, File monitoring

CAPEC ID: CAPEC-270

Contributors: Oddvar Moe, @oddvarmoe

Version: 1.0

splunk> .conf19

Use ATT&CK to Prioritize

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
10 items	25 items	41 items	21 items	49 items	16 items	19 items	15 items	13 items	9 items	20 items
Drive-by Compromise	CMSTP	Accessibility Features	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	Application Deployment Software	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	Command-Line Interface	AppCert DLLs	Accessibility Features	Binary Padding	Brute Force	Application Window Discovery	Distributed Component Object Model	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Control Panel Items	AppInit DLLs	AppCert DLLs	BITS Jobs	Credential Dumping	Browser Bookmark Discovery	Exploitation of Remote Services	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Dynamic Data Exchange	Application Shimming	AppInit DLLs	Bypass User Account Control	Credentials in Files	File and Directory Discovery	Logon Scripts	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Execution through API	Authentication Package	Application Shimming	CMSTP	Credentials in Registry	Network Service Scanning	Pass the Hash	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through Module Load	BITS Jobs	Bypass User Account Control	Code Signing	Exploitation for Credential Access	Network Share Discovery	Pass the Ticket	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Exploitation for Client Execution	Bootkit	DLL Search Order Hijacking	Component Firmware	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Peripheral Device Discovery	Remote File Copy	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Permission Groups Discovery	Remote Services	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	LSASS Driver	Component Firmware	File System Permissions Weakness	DCShadow	Kerberoasting	Process Discovery	Replication Through Removable Media	Input Capture		Multi-hop Proxy
	Mshta	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	LLMNR/NBT-NS Poisoning	Query Registry	Shared Webroot	Man in the Browser		Multi-Stage Channels
	PowerShell	Create Account	Image File Execution Options Injection	Disabling Security Tools	Network Sniffing	Remote System Discovery	Taint Shared Content	Screen Capture		Multiband Communication
	Regsvcs/Regasm	DLL Search Order Hijacking	New Service	DLL Search Order Hijacking	Password Filter DLL	Security Software Discovery	Third-party Software	Video Capture		Multilayer Encryption
	Regsvr32	External Remote Services	Path Interception	DLL Side-Loading	Private Keys	System Information Discovery	Windows Admin Shares			Remote Access Tools
	Rundll32	File System Permissions Weakness	Port Monitors	Exploitation for Defense Evasion	Replication Through Removable Media	System Network Configuration Discovery	Windows Remote Management			Remote File Copy
	Scheduled Task	Hidden Files and Directories	Process Injection	Extra Window Memory Injection	Two-Factor Authentication Interception	System Network Connections Discovery				Standard Application Layer Protocol
	Scripting	Hooking	Scheduled Task	File Deletion		System Owner/User Discovery				Standard Cryptographic Protocol
	Service Execution	Hypervisor	Service Registry Permissions Weakness	File System Logical Offsets		System Service Discovery				Standard Non-Application Layer Protocol
	Signed Binary Proxy Execution	Image File Execution Options Injection	SID-History Injection	Hidden Files and Directories		System Time Discovery				Uncommonly Used Port
	Signed Script Proxy Execution	Logon Scripts	Valid Accounts	Image File Execution Options Injection						Web Service
	Third-party Software	LSASS Driver	Web Shell	Indicator Blocking						
	Trusted Developer Utilities	Modify Existing Service		Indicator Removal from Tools						
	User Execution	Netsh Helper DLL		Indicator Removal on Host						
	Windows Management Instrumentation	New Service		Indirect Command Execution						
	Windows Remote Management	Office Application Startup		Install Root Certificate						
		Path Interception		InstallUtil						
				Masquerading						

IoCs or TTPs

Why Not Both?

MultiScanner

build passing

Introduction

MultiScanner is a file analysis framework that assists the user in evaluating a set of files by automatically running a suite of tools for the user and aggregating the output. Tools can be custom built Python scripts, web APIs, software running on another machine, etc. Tools are incorporated by creating modules that run in the MultiScanner framework.

Modules are designed to be quickly written and easily incorporated into the framework. Currently written and maintained modules are related to malware analytics, but the framework is not limited to that scope. For a list of modules you can look in modules/. Descriptions and config options can be found on the Analysis Modules page.

MultiScanner also supports a distributed workflow for sample storage, analysis, and report viewing. This functionality includes a web interface, a REST API, a distributed file system (GlusterFS), distributed report storage / searching (Elasticsearch), and distributed task management (Celery / RabbitMQ). Please see Architecture for more details.

Source: https://github.com/mitre/multiscanner

Summary

Detect known bad
Artifact-driven
Fewer false positives
More atomic
Higher quantity

Detect suspicious events
Behavior-driven
More false positives
Broader
Lower quantity
Longer lifetime

Thank You!

Follow me on Twitter @xavierashe!

splunk> .conf19