Detecting The Adversary Post-compromise With Threat Models And Behavioral Analytics

Michael Kemmerer

MITRE

.conf2016

splunk>

Two Projects, One Goal

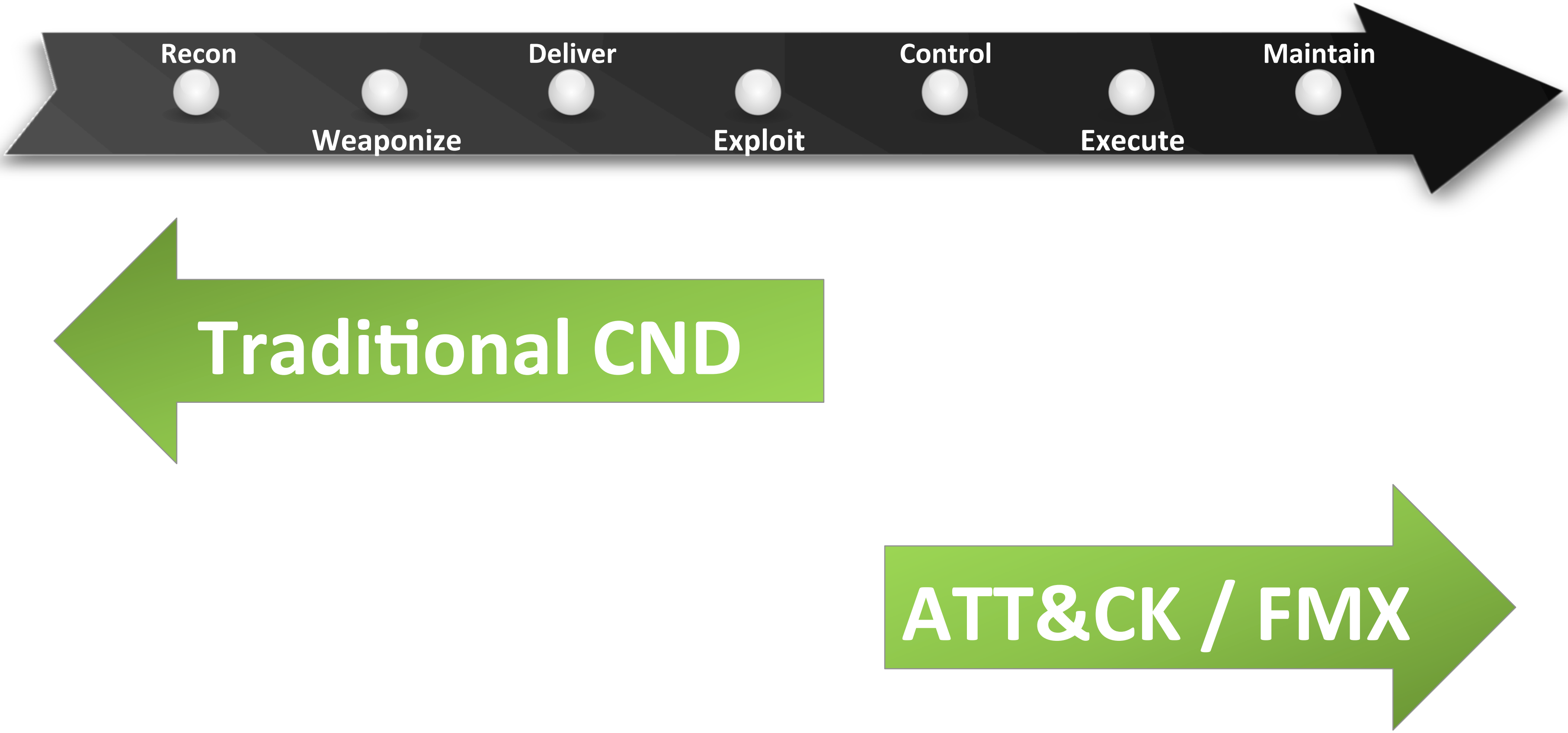
Adversarial Tactics, Techniques and Common Knowledge (ATT&CK™)

The Fort Meade eXperiment (FMX)

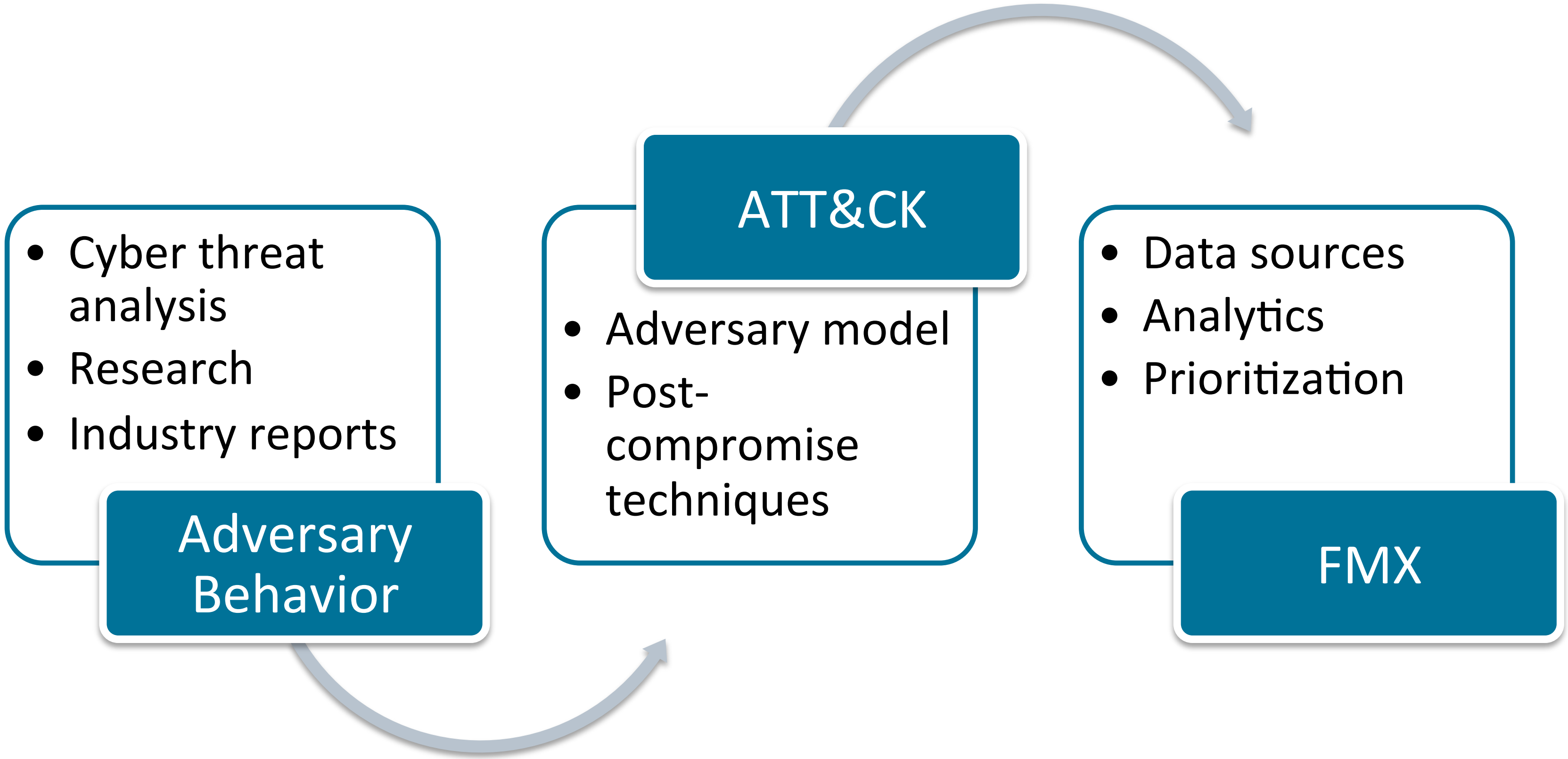
146 days - The median time an adversary is in a network before being detected

-Mandiant, M-Trends 2016

Cyber Attack Lifecycle



Threat Based Modeling



Adversary Behavior

- Cyber threat analysis
- Research
- Industry reports

ATT&CK

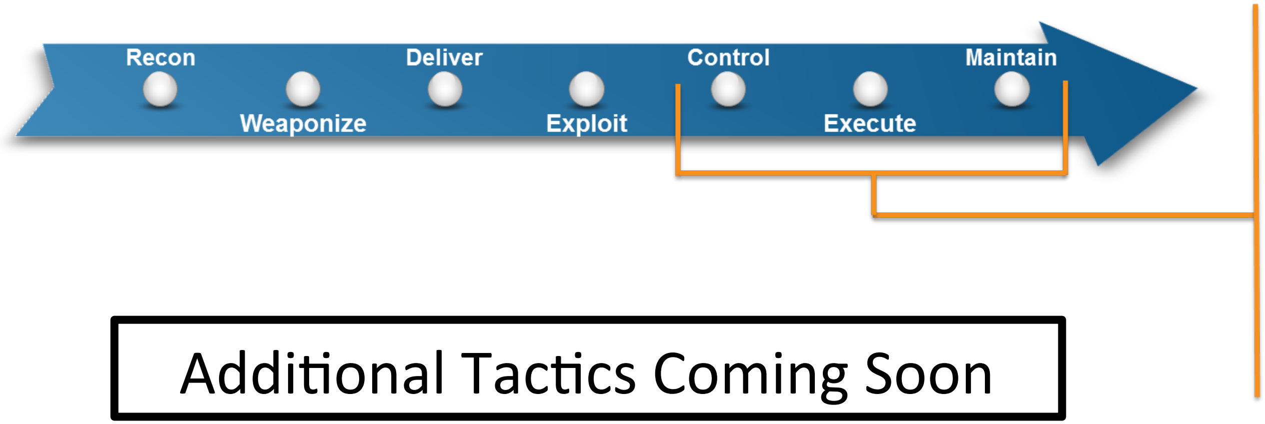
- Adversary model
- Post-compromise techniques

FMX

- Data sources
- Analytics
- Prioritization

For Public Release

ATT&CK: Deconstructing the Lifecycle



- Persistence
- Privilege Escalation
- Credential Access
- Host Enumeration
- Defense Evasion
- Lateral Movement
- Execution
- Command and Control
- Exfiltration

Additional Tactics Coming Soon

Threat data informed adversary model

Higher fidelity on right-of-exploit, post-access phases

Describes behavior sans adversary tools

For Public Release

The ATT&CK Model

- **Consists of:**
  1. Tactic phases derived from Cyber Attack Lifecycle
  2. List of techniques available to adversaries for each phase
  3. Possible methods of detection and mitigation
  4. Documented adversary use of techniques

- **Publically available adversary information is a problem**
  - Not granular enough
  - Insufficient volume

Image source: www.mrpotatohead.net

Mr. Potato Head is a registered trademark of Hasbro Inc.

For Public Release

Example of Technique Details

Persistence – New Windows Service

- **Description:** When Windows starts, it also starts programs called services. A service's configuration information, including the service's executable, is stored in the registry. Adversaries may install a new service which will be executed at startup by directly modifying the registry or by using tools.
- **Platform**: Windows
- **Permissions required**: Administrator, SYSTEM
- **Effective permissions**: SYSTEM
- **Use**: Part of initial infection vector or used during operation to locally or remotely execute persistent malware. May be used for privilege escalation.
- **Detection**: Monitor new service creation. Look for out of the ordinary service names and activity that does not correlate with known-good software, patches, etc. New services may show up as outlier processes that have not been seen before when compared against historical data.
- **Data Sources:** Windows Registry, process monitoring

Information on Threat Actors and Tools Coming Soon

ATT&CK Matrix™ Tactics and Techniques

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Host Enumeration	Lateral Movement	Execution	C2	Exfiltration
Legitimate Credentials Accessibility Features AddMonitor DLL Search Order Hijack Edit Default File Handlers New Service Path Interception Scheduled Task Service File Permission Weakness Shortcut Modification Web shell BIOS Hypervisor Rootkit Logon Scripts Master Boot Record Mod. Exist'g Service Registry Run Keys Serv. Reg. Perm. Weakness Windows Mgmt Instr. Event Subsc. Winlogon Helper DLL	Bypass UAC DLL Injection Exploitation of Vulnerability	Binary Padding DLL Side-Loading Disabling Security Tools File System Logical Offsets Process Hollowing Rootkit Indicator blocking on host Indicator removal from tools Indicator removal from host Masquerad-ing NTFS Extended Attributes Obfuscated Payload Rundll32 Scripting Software Packing Timestomp	Credential Dumping Credentials in Files Network Sniffing User Interaction Credential manipulation	Account enumeration File system enumeration Group permission enumeration Local network connection enumeration Local networking enumeration Operating system enumeration Owner/User enumeration Process enumeration Security software enumeration Service enumeration Window enumeration	Application deployment software Exploitation of Vulnerability Logon scripts Pass the hash Pass the ticket Peer connections Remote Desktop Protocol Windows management instrumentation Windows remote management Remote Services Replication through removable media Shared webroot Taint shared content Windows admin shares	Command Line File Access PowerShell Process Hollowing Registry Rundll32 Scheduled Task Service Manipulation Third Party Software	Commonly used port Comm through removable media Custom application layer protocol Custom encryption cipher Data obfuscation Fallback channels Multiband comm Multilayer encryption Peer connections Standard app layer protocol Standard non-app layer protocol Standard encryption cipher Uncommonly used port	Automated or scripted exfiltration Data compressed Data encrypted Data size limits Data staged Exfil over C2 channel Exfil over alternate channel to C2 network Exfil over other network medium Exfil over physical medium From local system From network resource From removable media Scheduled transfer

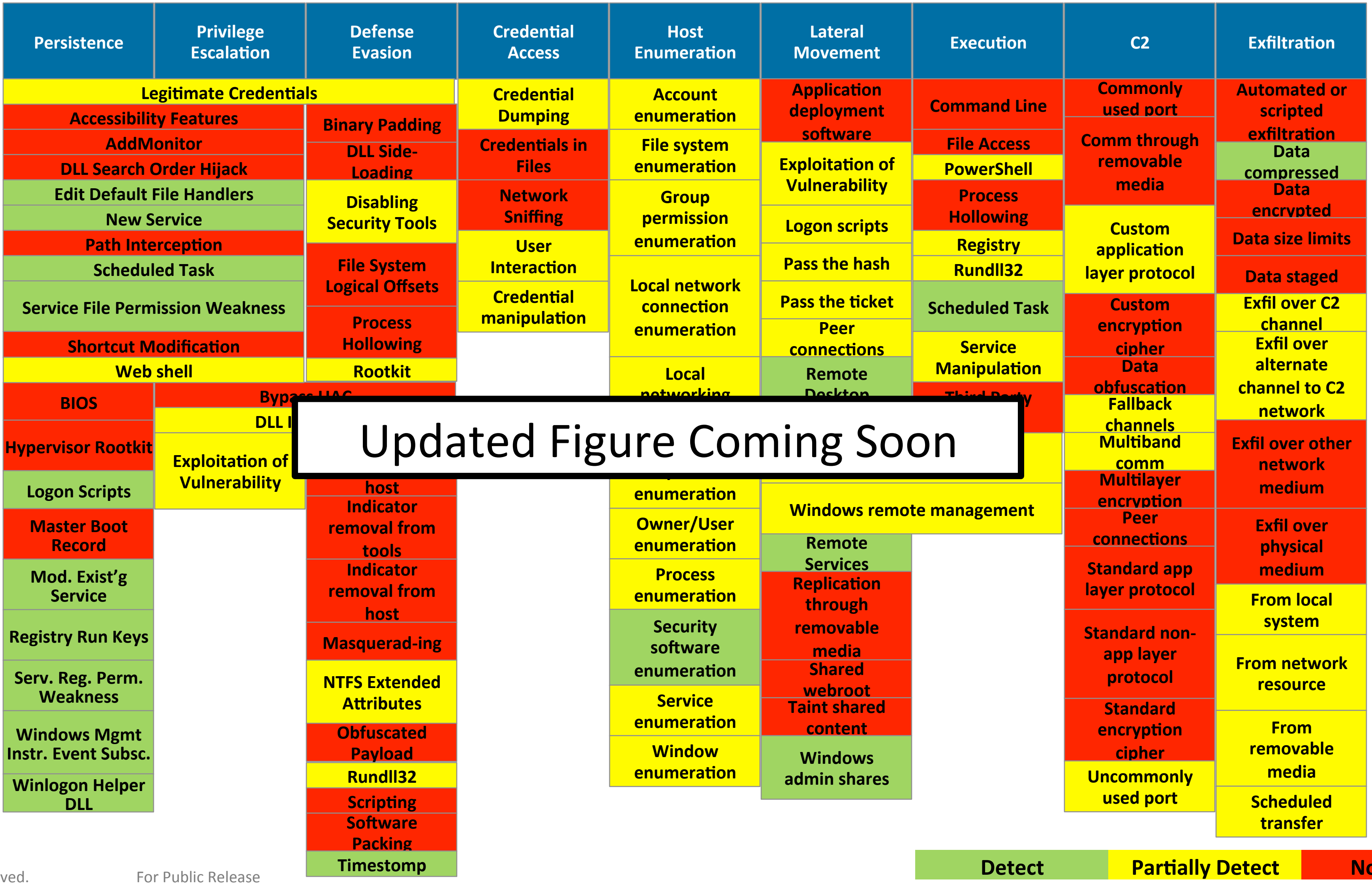
Updated Figure Coming Soon

For Public Release

Use Cases

- Gap analysis with current defenses
- Prioritize detection/mitigation of heavily used techniques
- Information sharing
- Track a specific adversary's set of techniques
- Simulations, exercises
- New technologies, research

Notional Defense Gaps



Updated Figure Coming Soon

Persistence
- 🟨 Legitimate Credentials (also Privilege Escalation, Defense Evasion)
- 🟥 Accessibility Features (also Privilege Escalation)
- 🟥 AddMonitor (also Privilege Escalation)
- 🟥 DLL Search Order Hijack (also Privilege Escalation)
- 🟩 Edit Default File Handlers (also Privilege Escalation)
- 🟩 New Service (also Privilege Escalation)
- 🟥 Path Interception (also Privilege Escalation)
- 🟩 Scheduled Task (also Privilege Escalation)
- 🟩 Service File Permission Weakness (also Privilege Escalation)
- 🟥 Shortcut Modification (also Privilege Escalation)
- 🟨 Web shell (also Privilege Escalation)
- 🟥 BIOS
- 🟥 Hypervisor Rootkit
- 🟩 Logon Scripts
- 🟥 Master Boot Record
- 🟩 Mod. Exist’g Service
- 🟩 Registry Run Keys
- 🟩 Serv. Reg. Perm. Weakness
- 🟩 Windows Mgmt Instr. Event Subsc.
- 🟩 Winlogon Helper DLL

Privilege Escalation
- 🟥 Bypass UAC (also Defense Evasion)
- 🟨 DLL I (also Defense Evasion)
- 🟨 Exploitation of Vulnerability

Defense Evasion
- 🟥 Binary Padding
- 🟥 DLL Side-Loading
- 🟨 Disabling Security Tools
- 🟥 File System Logical Offsets
- 🟥 Process Hollowing
- 🟨 Rootkit
- 🟥 host
- 🟥 Indicator removal from tools
- 🟥 Indicator removal from host
- 🟥 Masquerad-ing
- 🟨 NTFS Extended Attributes
- 🟥 Obfuscated Payload
- 🟨 Rundll32
- 🟥 Scripting
- 🟥 Software Packing
- 🟩 Timestomp

Credential Access
- 🟨 Credential Dumping
- 🟥 Credentials in Files
- 🟥 Network Sniffing
- 🟨 User Interaction
- 🟨 Credential manipulation

Host Enumeration
- 🟨 Account enumeration
- 🟨 File system enumeration
- 🟨 Group permission enumeration
- 🟨 Local network connection enumeration
- 🟨 Local networking
- 🟨 enumeration
- 🟨 Owner/User enumeration
- 🟨 Process enumeration
- 🟩 Security software enumeration
- 🟨 Service enumeration
- 🟨 Window enumeration

Lateral Movement
- 🟥 Application deployment software
- 🟨 Exploitation of Vulnerability
- 🟨 Logon scripts
- 🟨 Pass the hash
- 🟨 Pass the ticket
- 🟨 Peer connections
- 🟩 Remote Desktop
- 🟨 Windows remote management (also Execution)
- 🟩 Remote Services
- 🟥 Replication through removable media
- 🟥 Shared webroot
- 🟥 Taint shared content
- 🟩 Windows admin shares

Execution
- 🟥 Command Line
- 🟥 File Access
- 🟨 PowerShell
- 🟥 Process Hollowing
- 🟨 Registry
- 🟨 Rundll32
- 🟩 Scheduled Task
- 🟨 Service Manipulation
- 🟥 Third Party

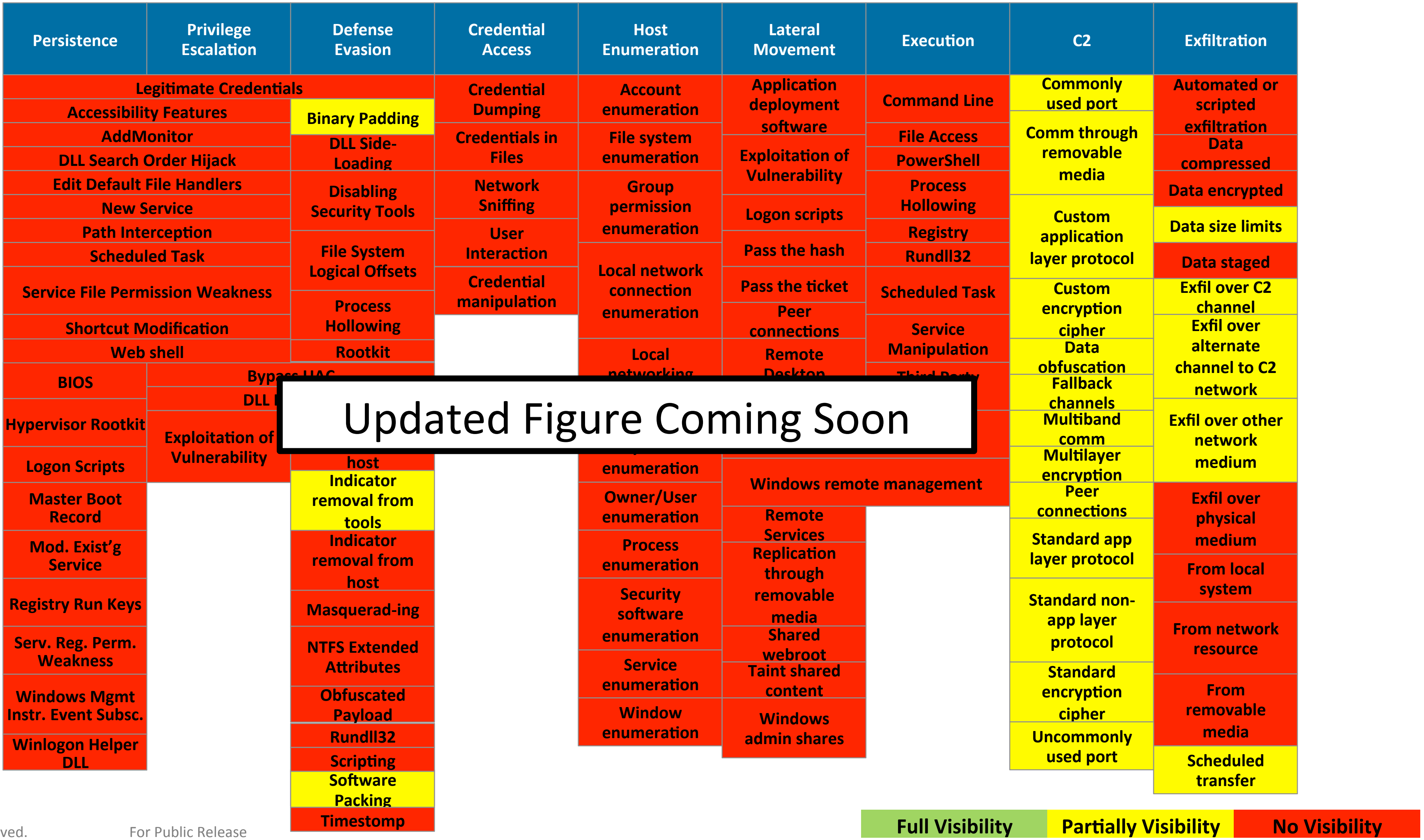
C2
- 🟥 Commonly used port
- 🟥 Comm through removable media
- 🟨 Custom application layer protocol
- 🟥 Custom encryption cipher
- 🟥 Data obfuscation
- 🟨 Fallback channels
- 🟨 Multiband comm
- 🟥 Multilayer encryption
- 🟥 Peer connections
- 🟥 Standard app layer protocol
- 🟥 Standard non-app layer protocol
- 🟥 Standard encryption cipher
- 🟨 Uncommonly used port

Exfiltration
- 🟥 Automated or scripted exfiltration
- 🟩 Data compressed
- 🟥 Data encrypted
- 🟥 Data size limits
- 🟥 Data staged
- 🟨 Exfil over C2 channel
- 🟨 Exfil over alternate channel to C2 network
- 🟥 Exfil over other network medium
- 🟥 Exfil over physical medium
- 🟨 From local system
- 🟨 From network resource
- 🟨 From removable media
- 🟨 Scheduled transfer

🟩 Detect 🟨 Partially Detect 🟥 No Detect

For Public Release

Adversary Visibility at the Perimeter



Persistence
- 🟥 Legitimate Credentials
- 🟥 Accessibility Features
- 🟥 AddMonitor
- 🟥 DLL Search Order Hijack
- 🟥 Edit Default File Handlers
- 🟥 New Service
- 🟥 Path Interception
- 🟥 Scheduled Task
- 🟥 Service File Permission Weakness
- 🟥 Shortcut Modification
- 🟥 Web shell
- 🟥 BIOS
- 🟥 Hypervisor Rootkit
- 🟥 Logon Scripts
- 🟥 Master Boot Record
- 🟥 Mod. Exist’g Service
- 🟥 Registry Run Keys
- 🟥 Serv. Reg. Perm. Weakness
- 🟥 Windows Mgmt Instr. Event Subsc.
- 🟥 Winlogon Helper DLL

Privilege Escalation
- 🟥 Bypass UAC
- 🟥 DLL I
- 🟥 Exploitation of Vulnerability

Defense Evasion
- 🟨 Binary Padding
- 🟥 DLL Side-Loading
- 🟥 Disabling Security Tools
- 🟥 File System Logical Offsets
- 🟥 Process Hollowing
- 🟥 Rootkit
- 🟥 host
- 🟨 Indicator removal from tools
- 🟥 Indicator removal from host
- 🟥 Masquerad-ing
- 🟥 NTFS Extended Attributes
- 🟥 Obfuscated Payload
- 🟥 Rundll32
- 🟥 Scripting
- 🟨 Software Packing
- 🟥 Timestomp

Credential Access
- 🟥 Credential Dumping
- 🟥 Credentials in Files
- 🟥 Network Sniffing
- 🟥 User Interaction
- 🟥 Credential manipulation

Host Enumeration
- 🟥 Account enumeration
- 🟥 File system enumeration
- 🟥 Group permission enumeration
- 🟥 Local network connection enumeration
- 🟥 Local networking
- 🟥 enumeration
- 🟥 Owner/User enumeration
- 🟥 Process enumeration
- 🟥 Security software enumeration
- 🟥 Service enumeration
- 🟥 Window enumeration

Lateral Movement
- 🟥 Application deployment software
- 🟥 Exploitation of Vulnerability
- 🟥 Logon scripts
- 🟥 Pass the hash
- 🟥 Pass the ticket
- 🟥 Peer connections
- 🟥 Remote Desktop
- 🟥 Windows remote management
- 🟥 Remote Services
- 🟥 Replication through removable media
- 🟥 Shared webroot
- 🟥 Taint shared content
- 🟥 Windows admin shares

Execution
- 🟥 Command Line
- 🟥 File Access
- 🟥 PowerShell
- 🟥 Process Hollowing
- 🟥 Registry
- 🟥 Rundll32
- 🟥 Scheduled Task
- 🟥 Service Manipulation
- 🟥 Third Party

C2
- 🟨 Commonly used port
- 🟨 Comm through removable media
- 🟨 Custom application layer protocol
- 🟨 Custom encryption cipher
- 🟨 Data obfuscation
- 🟨 Fallback channels
- 🟨 Multiband comm
- 🟨 Multilayer encryption
- 🟨 Peer connections
- 🟨 Standard app layer protocol
- 🟨 Standard non-app layer protocol
- 🟨 Standard encryption cipher
- 🟨 Uncommonly used port

Exfiltration
- 🟥 Automated or scripted exfiltration
- 🟥 Data compressed
- 🟥 Data encrypted
- 🟨 Data size limits
- 🟥 Data staged
- 🟨 Exfil over C2 channel
- 🟨 Exfil over alternate channel to C2 network
- 🟨 Exfil over other network medium
- 🟥 Exfil over physical medium
- 🟥 From local system
- 🟥 From network resource
- 🟥 From removable media
- 🟨 Scheduled transfer

Updated Figure Coming Soon

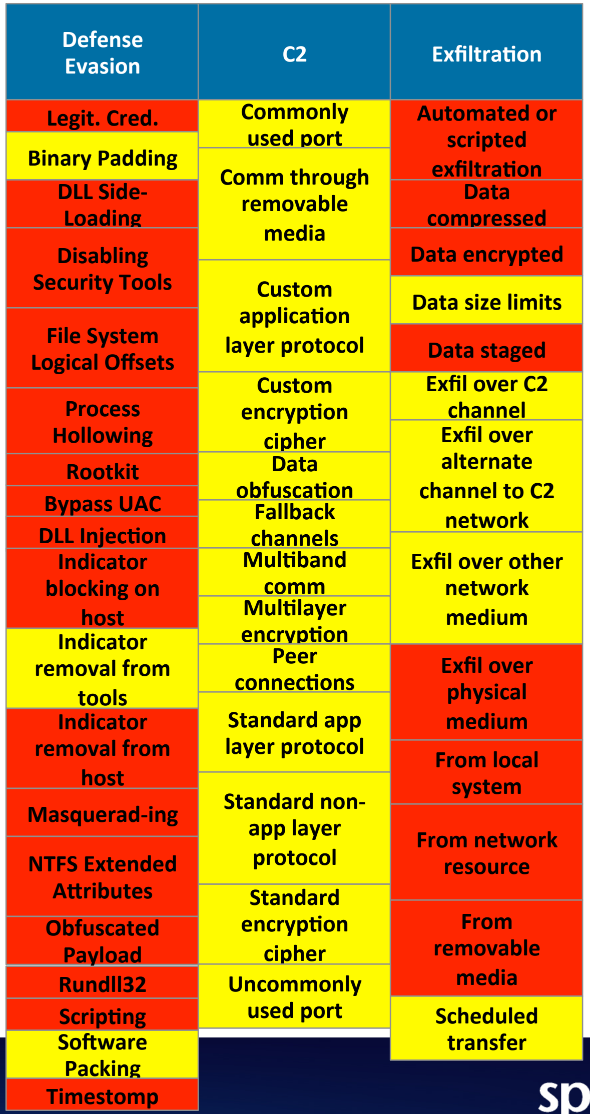
🟩 Full Visibility	🟨 Partially Visibility	🟥 No Visibility

For Public Release

Adversary Visibility At The Perimeter

- **Adversary has the most latitude for variation at the network level**
- **Firewall, IDS/IPS, netflow, proxy, mail gateway, WCF, SSL MitM, protocol decoders, anomaly detection etc…**
- **All partial solutions**
  - Don't add up to a complete one
- **Often require specific prior knowledge**
  - IPs, domains, malware changed easily
    - Sector, organization specific infrastructure
    - Frequently modify tools
    - Use legitimate channels
- **Better coverage with host sensing**

Updated Figures Coming Soon



Defense Evasion	C2	Exfiltration
Legit. Cred. Binary Padding DLL Side-Loading Disabling Security Tools File System Logical Offsets Process Hollowing Rootkit Bypass UAC DLL Injection Indicator blocking on host Indicator removal from tools Indicator removal from host Masquerad-ing NTFS Extended Attributes Obfuscated Payload Rundll32 Scripting Software Packing Timestomp	Commonly used port Comm through removable media Custom application layer protocol Custom encryption cipher Data obfuscation Fallback channels Multiband comm Multilayer encryption Peer connections Standard app layer protocol Standard non-app layer protocol Standard encryption cipher Uncommonly used port	Automated or scripted exfiltration Data compressed Data encrypted Data size limits Data staged Exfil over C2 channel Exfil over alternate channel to C2 network Exfil over other network medium Exfil over physical medium From local system From network resource From removable media Scheduled transfer

Tactic Breakdown

Persistence	20	Lateral Movement	14
Privilege Escalation	14	Execution	11
Credential Access	5	Command and Control	13
Host Enumeration	11	Exfiltration	13
Defense Evasion	19		

Updated Figures Coming Soon

For Public Release

Publicly Known Adversary Use

Persistence	20	8	Lateral Movement	14	8
Privilege Escalation	14	9	Execution	11	7
Credential Access	5	5	Command and Control	13	12
Host Enumeration	11	10	Exfiltration	13	10
Defense Evasion	19	14			

Updated Figures Coming Soon

For Public Release

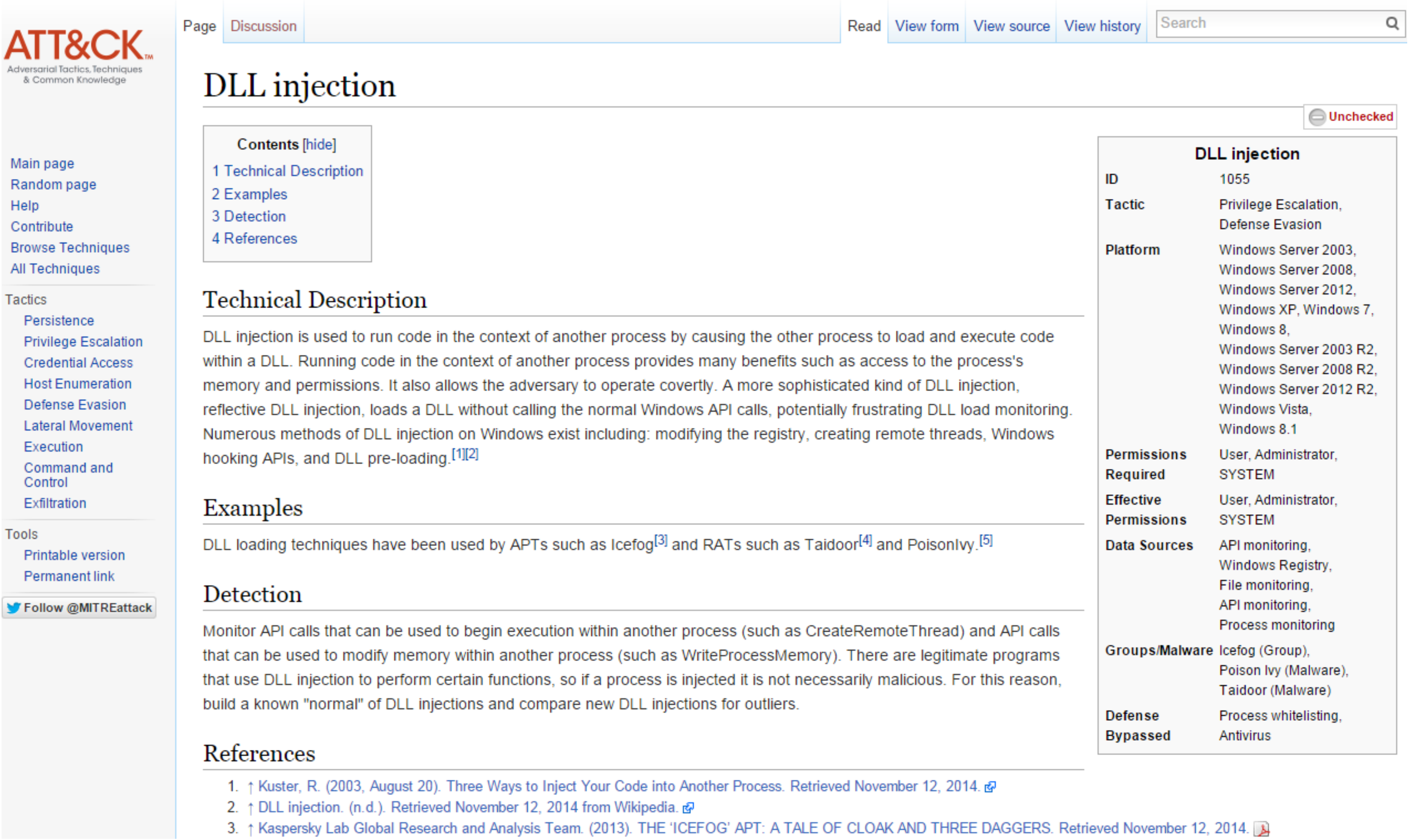
Publically Reported Technique Use

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Host Enumeration	Lateral Movement	Execution	C2	Exfiltration
Legitimate Credentials		Binary Padding	Credential Dumping	Account enumeration	Application deployment software	Command Line	Commonly used port	Automated or scripted exfiltration
Accessibility Features		DLL Side-Loading	Credentials in Files	File system enumeration	Exploitation of Vulnerability	File Access	Comm through removable media	Data compressed
AddMonitor		Disabling Security Tools	Network Sniffing	Group permission enumeration	Logon scripts	PowerShell	Custom application layer protocol	Data encrypted
DLL Search Order Hijack		File System Logical Offsets	User Interaction	Local network connection enumeration	Pass the hash	Process Hollowing	Custom encryption cipher	Data size limits
Edit Default File Handlers		Process Hollowing	Credential manipulation	Local networking	Pass the ticket	Registry	Data obfuscation	Data staged
New Service		Rootkit		enumeration	Peer connections	Rundll32	Fallback channels	Exfil over C2 channel
Path Interception		host		Owner/User enumeration	Remote Desktop	Scheduled Task	Multiband comm	Exfil over alternate channel to C2 network
Scheduled Task		Indicator removal from tools		Process enumeration	Windows remote management	Service Manipulation	Multilayer encryption	Exfil over other network medium
Service File Permission Weakness		Indicator removal from host		Security software enumeration	Remote Services	Third Party Software	Peer connections	Exfil over physical medium
Shortcut Modification		Masquerad-ing		Service enumeration	Replication through removable media	ement on	Standard app layer protocol	From local system
Web shell		NTFS Extended Attributes		Window enumeration	Shared webroot		Standard non-app layer protocol	From network resource
BIOS	Bypass UAC	Obfuscated Payload			Taint shared content		Standard encryption cipher	From removable media
Hypervisor Rootkit	Exploitat Vulnerability	Rundll32			Windows admin shares		Uncommonly used port	Scheduled transfer
Logon Scripts		Scripting						
Master Boot Record		Software Packing						
Mod. Exist'g Service		Timestomp						
Registry Run Keys								
Serv. Reg. Perm. Weakness								
Windows Mgmt Instr. Event Subsc.								
Winlogon Helper DLL								

Updated Figure Coming Soon

For Public Release

Public Website – Attack.Mitre.Org



DLL injection

DLL injection	
ID	1055
Tactic	Privilege Escalation, Defense Evasion
Platform	Windows Server 2003, Windows Server 2008, Windows Server 2012, Windows XP, Windows 7, Windows 8, Windows Server 2003 R2, Windows Server 2008 R2, Windows Server 2012 R2, Windows Vista, Windows 8.1
Permissions Required	User, Administrator, SYSTEM
Effective Permissions	User, Administrator, SYSTEM
Data Sources	API monitoring, Windows Registry, File monitoring, API monitoring, Process monitoring
Groups/Malware	Icefog (Group), Poison Ivy (Malware), Taidoor (Malware)
Defense Bypassed	Process whitelisting, Antivirus

Technical Description

DLL injection is used to run code in the context of another process by causing the other process to load and execute code within a DLL. Running code in the context of another process provides many benefits such as access to the process's memory and permissions. It also allows the adversary to operate covertly. A more sophisticated kind of DLL injection, reflective DLL injection, loads a DLL without calling the normal Windows API calls, potentially frustrating DLL load monitoring. Numerous methods of DLL injection on Windows exist including: modifying the registry, creating remote threads, Windows hooking APIs, and DLL pre-loading.[1][2]

Examples

DLL loading techniques have been used by APTs such as Icefog[3] and RATs such as Taidoor[4] and PoisonIvy.[5]

Detection

Monitor API calls that can be used to begin execution within another process (such as CreateRemoteThread) and API calls that can be used to modify memory within another process (such as WriteProcessMemory). There are legitimate programs that use DLL injection to perform certain functions, so if a process is injected it is not necessarily malicious. For this reason, build a known "normal" of DLL injections and compare new DLL injections for outliers.

References

1. ↑ Kuster, R. (2003, August 20). Three Ways to Inject Your Code into Another Process. Retrieved November 12, 2014.
2. ↑ DLL injection. (n.d.). Retrieved November 12, 2014 from Wikipedia.
3. ↑ Kaspersky Lab Global Research and Analysis Team. (2013). THE 'ICEFOG' APT: A TALE OF CLOAK AND THREE DAGGERS. Retrieved November 12, 2014.

For Public Release

Defender’s Problem: Adversaries Blend In

- Attackers post-exploit look very similar to normal users
- Traditional efforts aren’t effective at finding an active intrusion
  - Internal tools look for exploit, compliance, or C2 channel
  - Indicator sharing only covers what’s known and is fragile

For Public Release

End-Point Sensing

Addressing the ATT&CK TTPs requires host-level sensing beyond typical antivirus and host-based intrusion sensors

Many more opportunities to catch adversaries operating inside networks than at the perimeter

Better awareness of compromise severity and scope

– Verizon: 85% of IP thefts lacked specific knowledge of what was taken

Sensor Options

- COTS
  - Bit9, Countertack, Mandiant, CrowdStrike, Cylance, EMC, others

- Built-in and OS Integrated
  - Event Tracing for Windows, Sysmon, Autoruns, Event Logs

The Fort Meade eXperiment (FMX)

MITRE's Fort Meade site

About 250 unclassified computers

Primarily user desktops running Windows 7

For Public Release

FMX: Analytics Based On Threats

- Develop analytics based on observed adversary TTPs
- Utilize native Windows logging/tools
- Decouple sensors from analytic platform
- Data model improves exportability and flexibility of analytics
- Create a methodology that doesn't overwhelm analysts

Not This!

Sensors: FY15

- Host-based Sensors
  - Microsoft Sysmon
  - ETWmon
  - Salt (Autoruns)
  - Hostflows

Updated Sensor List Coming Soon

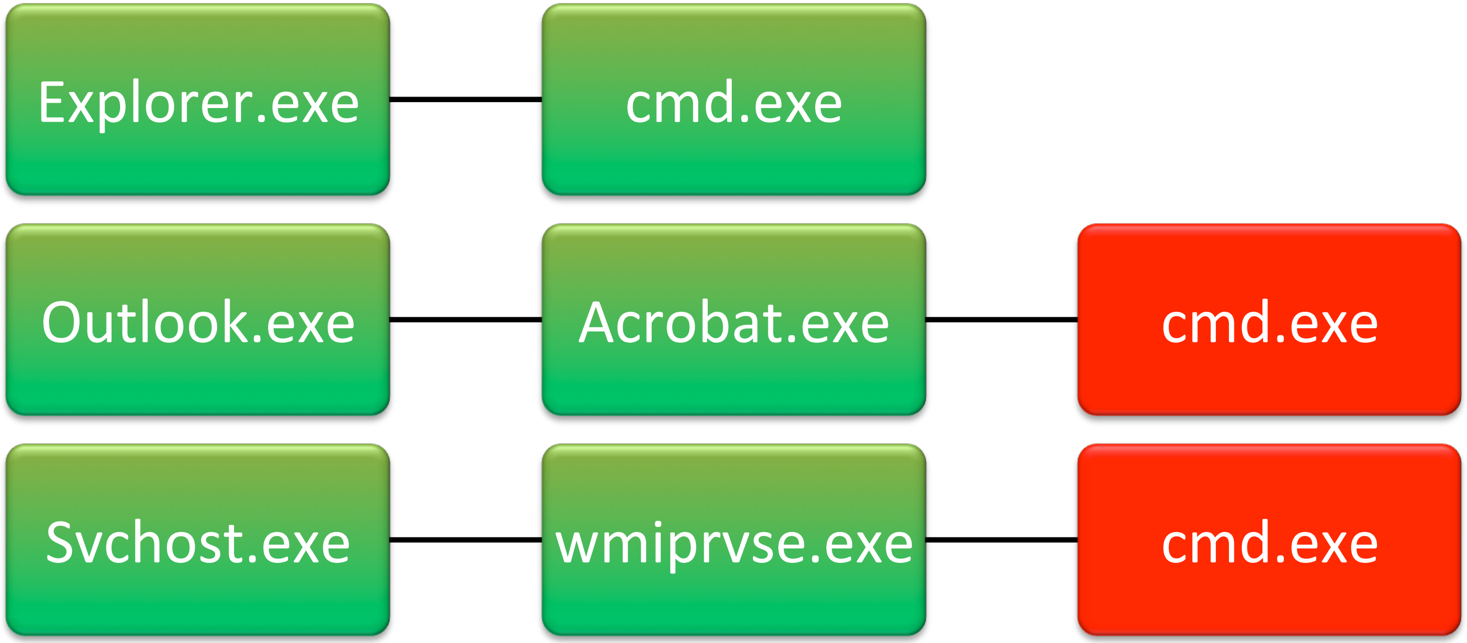
- Network Sensors
  - PCAP
  - Netflows

For Public Release

Microsoft Sysmon

Provides details on processes

Process chains provide context around system activity



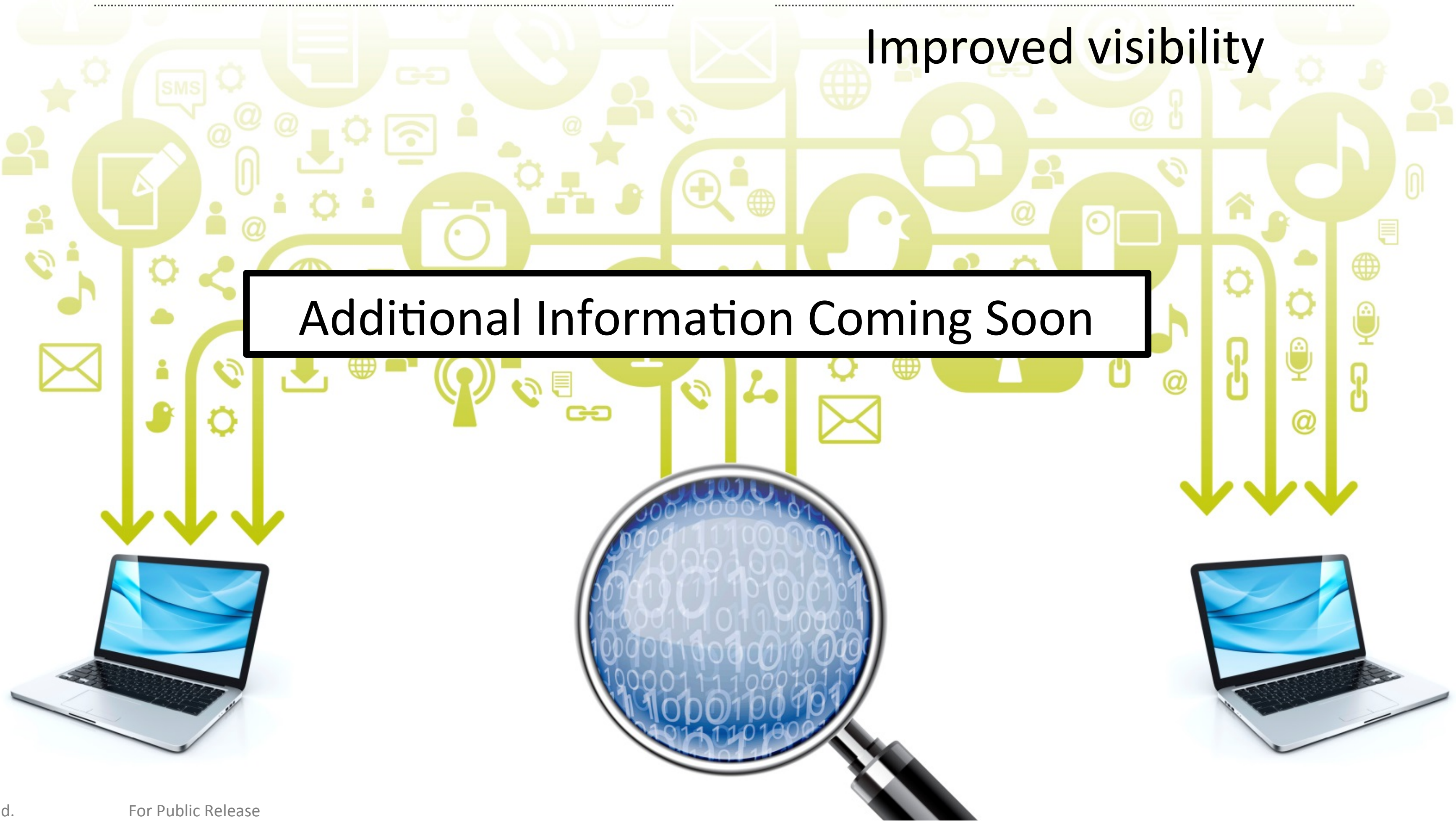
For Public Release

Hostflows

Metadata on network connections

Pivot point between host and network data

Improved visibility

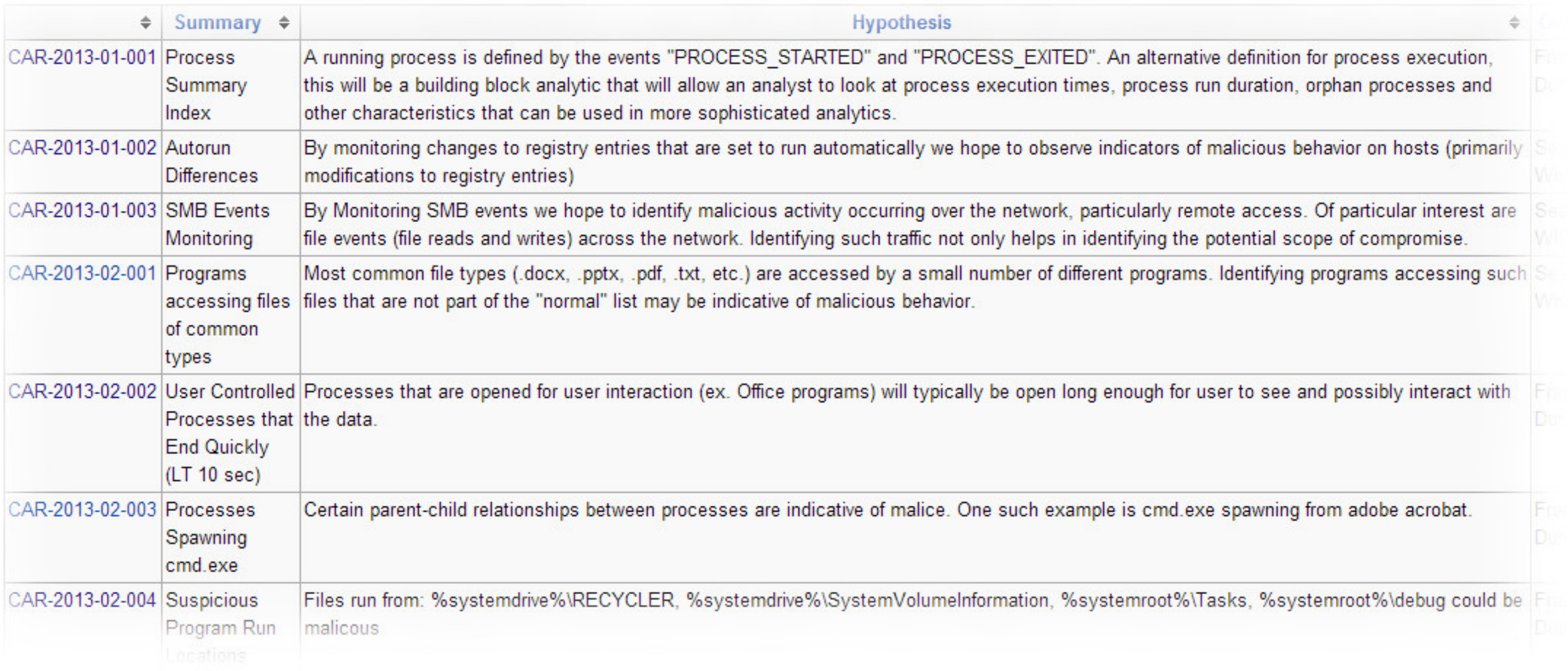


Additional Information Coming Soon

For Public Release

Analytic Lessons Captured

Tested, shareable analytics that are effective at finding attacker behavior are the output of FMX



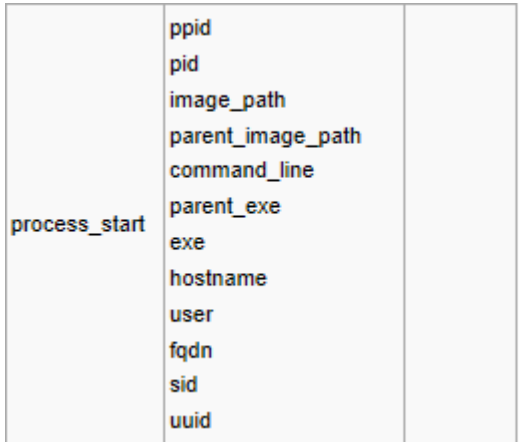
	Summary	Hypothesis
CAR-2013-01-001	Process Summary Index	A running process is defined by the events "PROCESS_STARTED" and "PROCESS_EXITED". An alternative definition for process execution, this will be a building block analytic that will allow an analyst to look at process execution times, process run duration, orphan processes and other characteristics that can be used in more sophisticated analytics.
CAR-2013-01-002	Autorun Differences	By monitoring changes to registry entries that are set to run automatically we hope to observe indicators of malicious behavior on hosts (primarily modifications to registry entries)
CAR-2013-01-003	SMB Events Monitoring	By Monitoring SMB events we hope to identify malicious activity occurring over the network, particularly remote access. Of particular interest are file events (file reads and writes) across the network. Identifying such traffic not only helps in identifying the potential scope of compromise.
CAR-2013-02-001	Programs accessing files of common types	Most common file types (.docx, .pptx, .pdf, .txt, etc.) are accessed by a small number of different programs. Identifying programs accessing such files that are not part of the "normal" list may be indicative of malicious behavior.
CAR-2013-02-002	User Controlled Processes that End Quickly (LT 10 sec)	Processes that are opened for user interaction (ex. Office programs) will typically be open long enough for user to see and possibly interact with the data.
CAR-2013-02-003	Processes Spawning cmd.exe	Certain parent-child relationships between processes are indicative of malice. One such example is cmd.exe spawning from adobe acrobat.
CAR-2013-02-004	Suspicious Program Run Locations	Files run from: %systemdrive%\RECYCLER, %systemdrive%\SystemVolumeInformation, %systemroot%\Tasks, %systemroot%\debug could be malicous

Data Model

index=old_sensor type=**PROC_EVENT_CREATE** hostname=A4123456.mitre.org imagepath=“c:\location\foo.exe”

index=sysmon Message=**“Process Create”** ComputerName=A4123456.mitre.org Image=“c:\location\foo.exe”

eventtype=**process_start** host_name=A4123456 image_path=“c:\location\foo.exe”



process_start	ppid pid image_path parent_image_path command_line parent_exe exe hostname user fqdn sid uuid	

Data Model Implementation Example Coming Soon

Current eventtypes:
file_access, process_start, process_stop, flow, logon

CAR Instantiation With Data Model

CAR-2014-07-001: Search Path Interception

Hypothesis:

As described by ATT&CK, one method of escalation is intercepting the search path for services, so that legitimate services point to the binary inserted at an intercepted location. This can be done when there are spaces in the path and it is unquoted.

Instantiation:

```
eventtype=process_start parent_image_path="*\\system32\\services.exe" command_line!="\"*" command_line="* *"
| rex field=image_path   ".*\\\(?<img_exe>.*)"
| rex field=img_exe "(?<img_base>.*)\..*"
| where NOT like(lower(command_line ), lower("%"+img_exe+"%")) AND like(lower(command_line), lower("%"+img_base+"%"))
| table _time hostname ppid pid parent_image_path image_path command_line img_exe
```

More Information About CAR Including Example Analytics Coming Soon

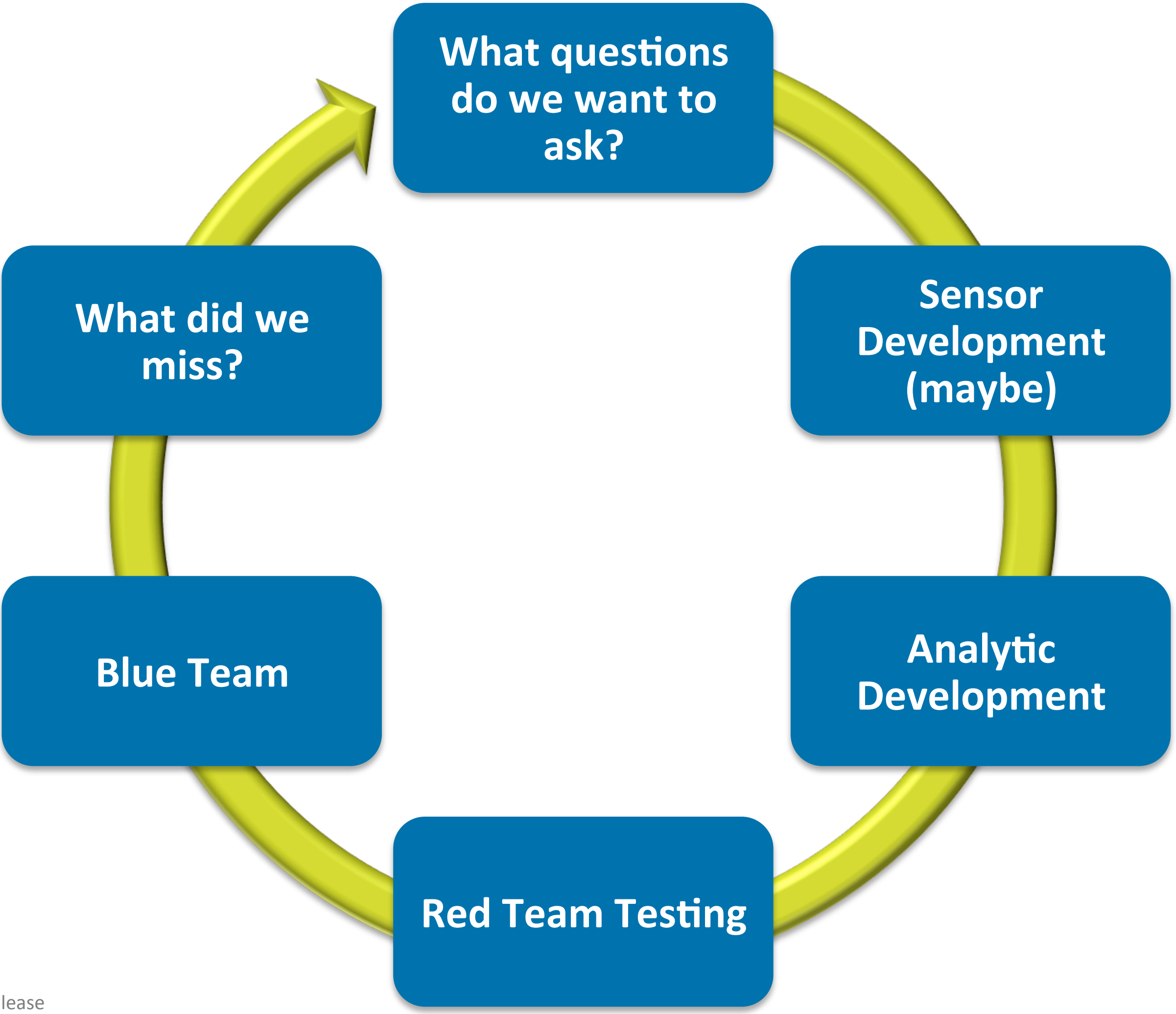
Evaluation With Cyber Games

- **Red/Blue Team operations within FMX environment**
  - Emulated adversary
  - Asynchronous
  - Designed to push analytic boundaries

Source: Tron, Walt Disney Pictures

For Public Release

FMX Analytic Development Cycle



What questions do we want to ask?

Sensor Development (maybe)

Analytic Development

Red Team Testing

Blue Team

What did we miss?

For Public Release

FMX Analytic Development Cycle

Analytic Development Example

Coming Soon

For Public Release

Questions?

ATT&CK

attack@mitre.org

Public website:

attack.mitre.org

FMX

fmx@mitre.org

THANK YOU

.conf2016

splunk>