



Sharing at the Oasis

Purple Teaming with MITRE ATT&CK

Matthew Gracie

Information Security Engineer

BlueCross BlueShield of Western New York

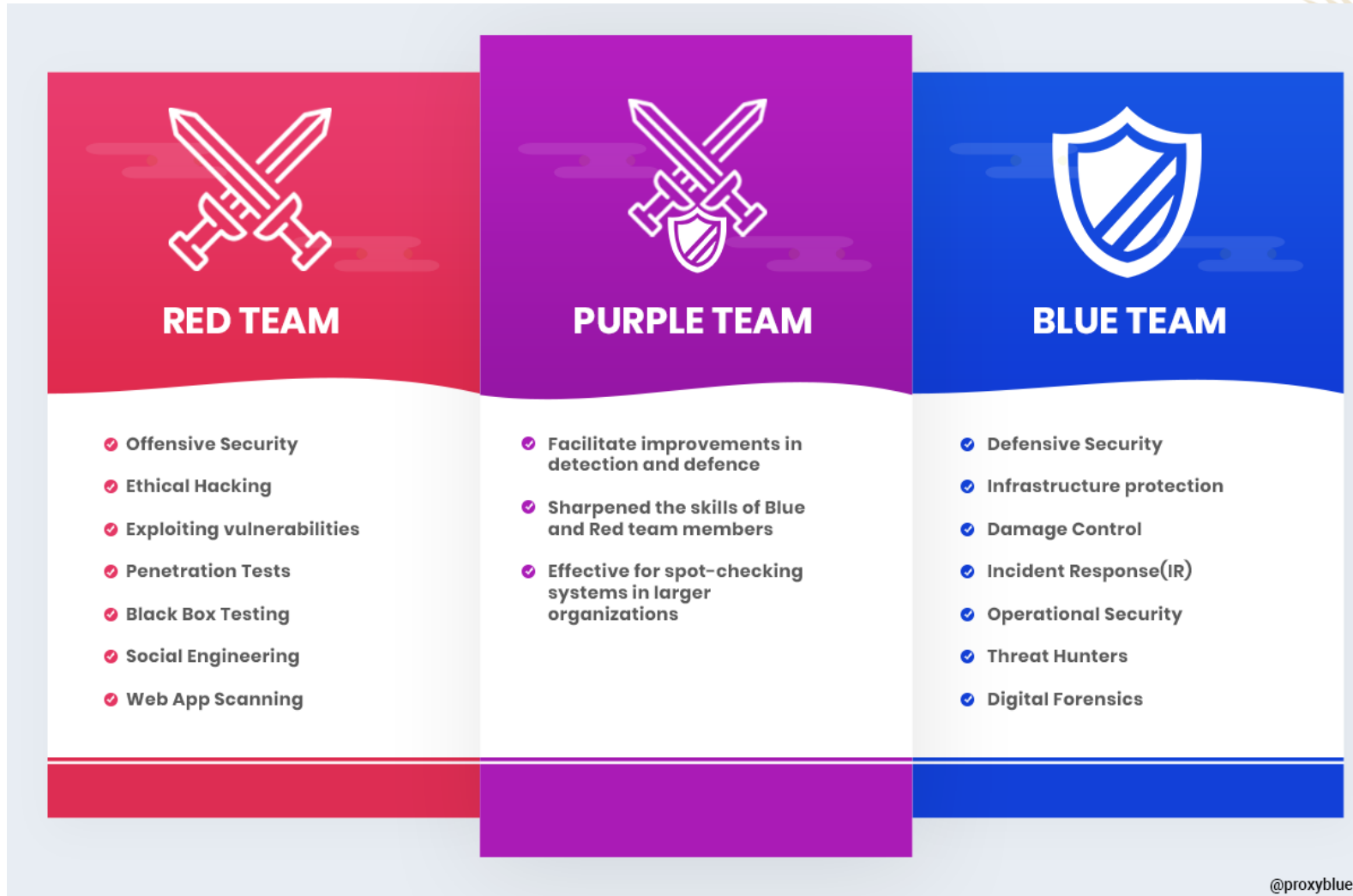
Who Am I And What Am I Talking About?



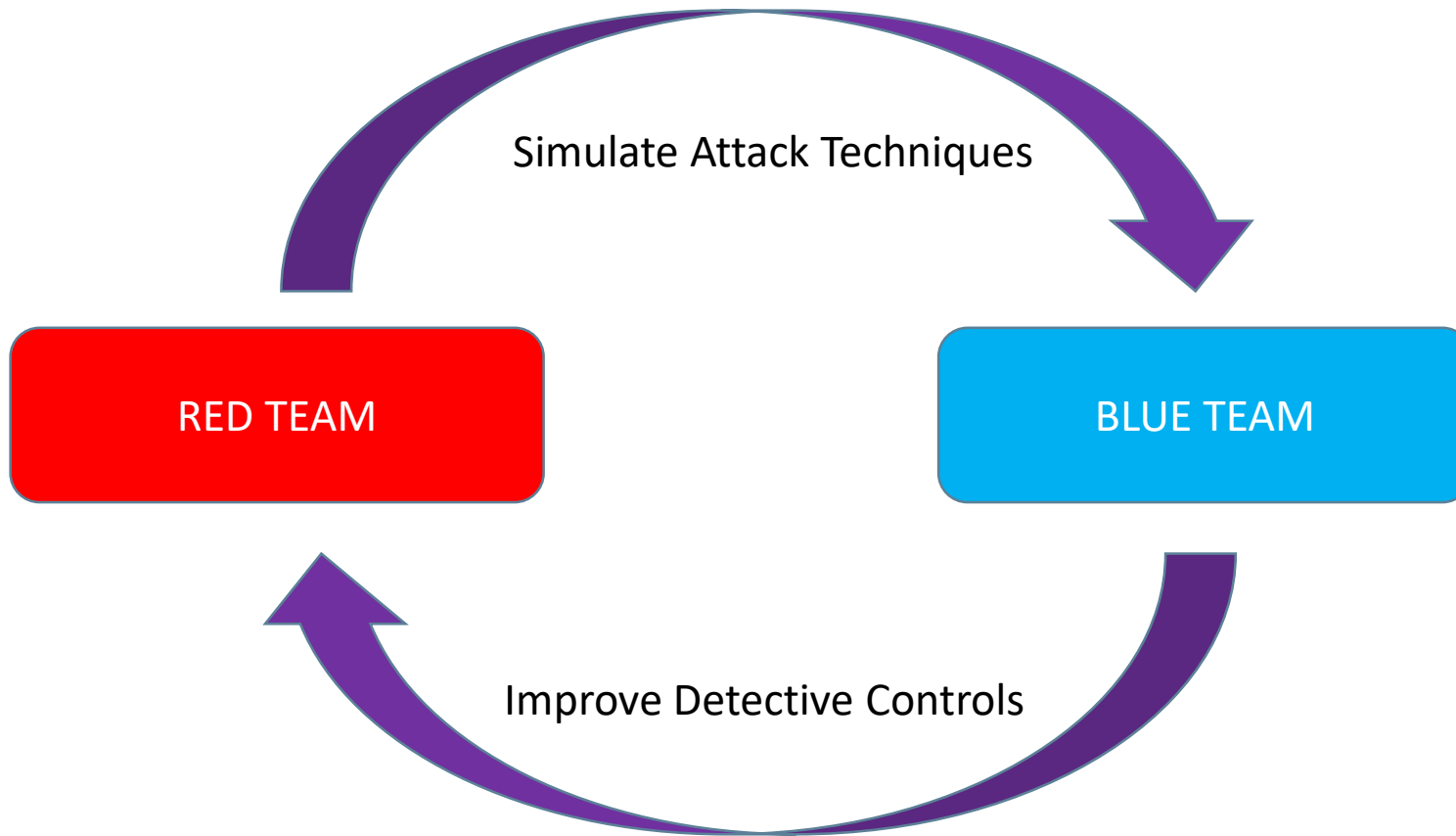
Security Complexity Is Increasing



Purple Teaming Can Help!



Sharing at the Oasis



What Should I Test?



MITRE ATT&CK: Adversarial Tactics, Techniques, and Common Knowledge

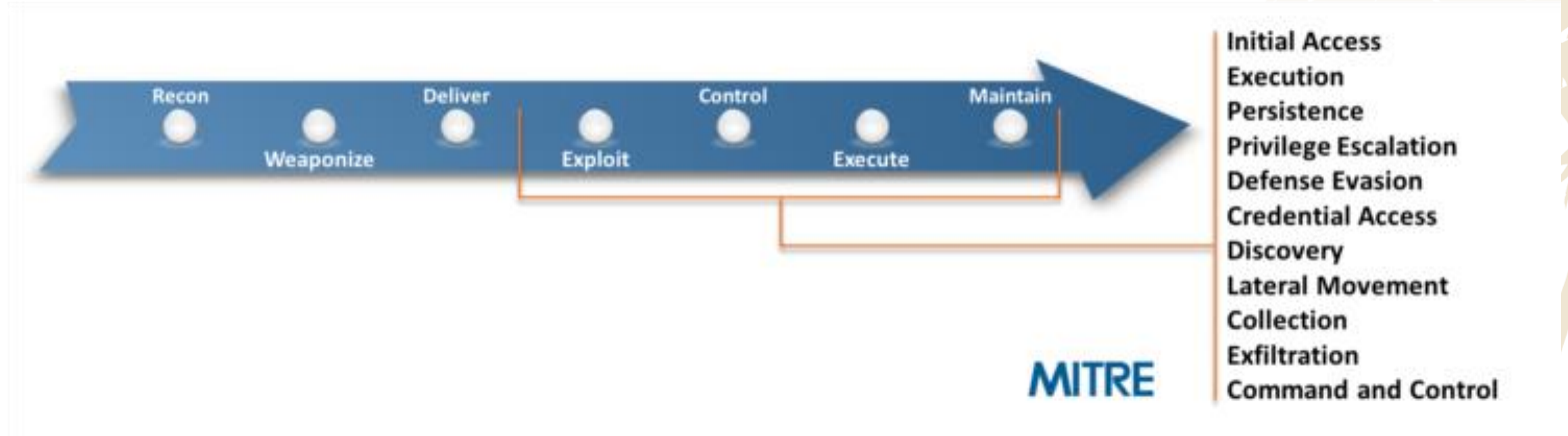
A large, light beige silhouette of a palm tree is positioned on the right side of the slide, extending from the bottom towards the top. It has several fronds and a thick trunk.

MITRE ATT&CK Matrices

- Pre-ATT&CK
- Enterprise
- Mobile
- ICS



Cyber Attack Lifecycle



Sharing at the Oasis

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Destruction
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Component Object Model and Distributed COM	Clipboard Data	Connection Proxy	Data Encrypted	Data Encrypted for Impact
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Clear Command History	Credentials from Web Browsers	File and Directory Discovery	Internal Spearphishing	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Content Wipe
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Disk Structure Wipe
Spearphishing Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	Code Signing	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Endpoint Denial of Service
Spearphishing via Service	Execution through API	BITS Jobs	Dylib Hijacking	Compile After Delivery	Exploitation for Credential Access	Network Sniffing	Pass the Ticket	Data Staged	Domain Fronting	Exfiltration Over Physical Medium	Firmware Corruption
Supply Chain Compromise	Execution through Module Load	Bootkit	Elevated Execution with Prompt	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Inhibit System Recovery
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Emond	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Input Capture	Fallback Channels		Network Denial of Service
Valid Accounts	Graphical User	Change Default File	Exploitation for Privilege	Component Object Model	Input Capture	Permission Groups	Remote Services	Man in the	Multi-hop Proxy		Resource Hijacking

TECHNIQUES

PRE-ATT&CK

Enterprise

Initial Access

Execution

Persistence

.bash_profile and
.bashrc

Accessibility Features

Account Manipulation

AppCert DLLs

AppInit DLLs

Application Shimming

Authentication Package

BITS Jobs

Bootkit

Browser Extensions

Change Default File
Association

[Home](#) > [Techniques](#) > [Enterprise](#) > Accessibility Features

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.

Two common accessibility programs are `C:\Windows\System32\sethc.exe`, launched when the shift key is pressed five times and `C:\Windows\System32\utilman.exe`, launched when the Windows + U key combination is pressed. The sethc.exe program is often referred to as "sticky keys", and has been used by adversaries for unauthenticated access through a remote desktop login screen. ^[1]

Depending on the version of Windows, an adversary may take advantage of these features in different ways because of code integrity enhancements. In newer versions of Windows, the replaced binary needs to be digitally signed for x64 systems, the binary must reside in `%systemdir%`, and it must be protected by Windows File or Resource Protection (WFP/WRP). ^[2] The debugger method was likely discovered as a potential workaround because it does not require the corresponding accessibility feature binary to be replaced. Examples for both methods:

For simple binary replacement on Windows XP and later as well as Windows Server 2003/R2 and later, for example, the program (e.g., `C:\Windows\System32\utilman.exe`) may be replaced with "cmd.exe" (or another program that provides backdoor access). Subsequently, pressing the appropriate key combination at the login screen while sitting at the keyboard or when connected over [Remote Desktop Protocol](#) will cause the replaced file to be executed with SYSTEM privileges. ^[3]

ID: T1015

Tactic: Persistence, Privilege Escalation

Platform: Windows

Permissions Required: Administrator

Effective Permissions: SYSTEM

Data Sources: Windows Registry, File
monitoring, Process monitoring

CAPEC ID: [CAPEC-558](#)

Contributors: Paul Speulstra, AECOM Global
Security Operations Center

Version: 1.0

Created: 31 May 2017

Last Modified: 16 July 2019

Mitigations

Mitigation	Description
Execution Prevention	Adversaries can replace accessibility features binaries with alternate binaries to execute this technique. Identify and block potentially malicious software executed through accessibility features functionality by using application whitelisting tools, like Windows Defender Application Control, AppLocker, or Software Restriction Policies where appropriate. ^{[6][7][8][9][10][11]}
Limit Access to Resource Over Network	If possible, use a Remote Desktop Gateway to manage connections and security configuration of RDP within a network. ^[5]
Operating System Configuration	To use this technique remotely, an adversary must use it in conjunction with RDP. Ensure that Network Level Authentication is enabled to force the remote desktop session to authenticate before the session is created and the login screen displayed. It is enabled by default on Windows Vista and later. ^[4]

Detection

Changes to accessibility utility binaries or binary paths that do not correlate with known software, patch cycles, etc., are suspicious. Command line invocation of tools capable of modifying the Registry for associated keys are also suspicious. Utility arguments and the binaries themselves should be monitored for changes. Monitor Registry keys within `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options`.

How Do I Prioritize?



Techniques Used In The Wild

Rank	Type
1	Credential Dumping
2	PowerShell
3	Account Discovery
4	Command Line Interface
5	Scripting

Table 4. Top five MITRE ATT&CK techniques observed in 2019

Threat Group Profiles

Techniques Used

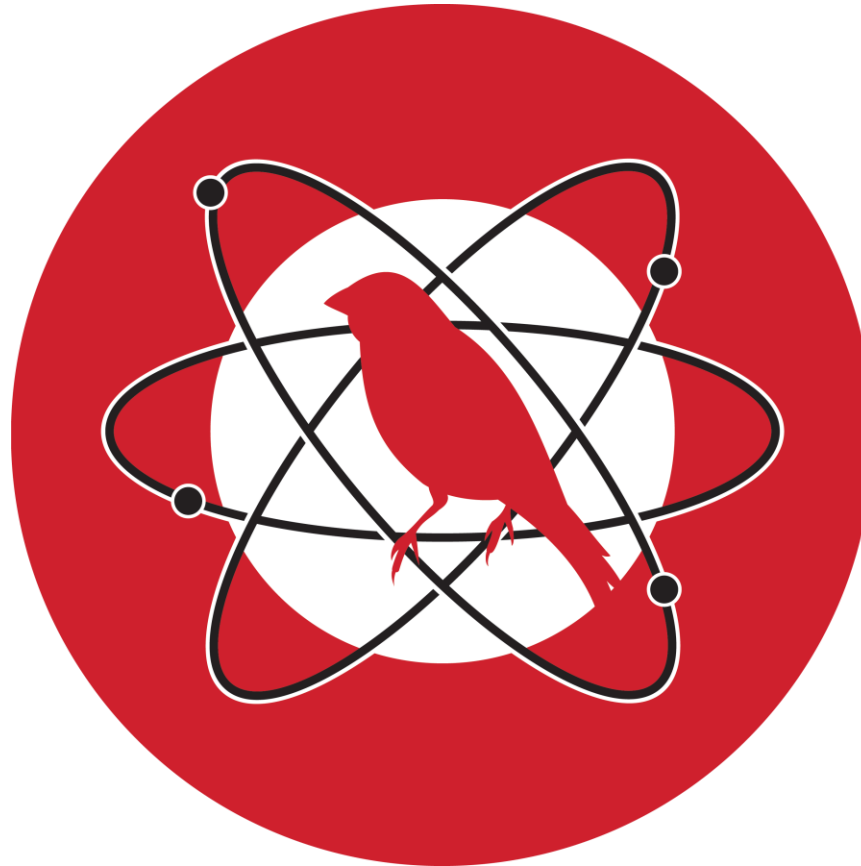
Domain	ID	Name
PRE-ATT&CK	T1328	Buy domain name
PRE-ATT&CK	T1346	Obtain/re-use payloads
Enterprise	T1134	.001 Access Token Manipulation: Token Impersonation/Theft
Enterprise	T1071	.003 Application Layer Protocol: Mail Protocols
		.001 Application Layer Protocol: Web Protocols
Enterprise	T1560	Archive Collected Data
Enterprise	T1119	Automated Collection
Enterprise	T1037	.001 Boot or Logon



How Do I Test My Defenses?



Testing With Atomic Red Team



Atomic Tests

- [Atomic Test #1 - Attaches Command Prompt as a Debugger to a List of Target Processes](#)

Atomic Test #1 - Attaches Command Prompt as a Debugger to a List of Target Processes

Attaches cmd.exe to a list of processes. Configure your own Input arguments to a different executable or list of executables.

Supported Platforms: Windows

Inputs:

Name	Description	Type	Default Value
parent_list	Comma separated list of system binaries to which you want to attach each #{attached_process}. Default: "osk.exe"		
String	osk.exe, sethc.exe, utilman.exe, magnify.exe, narrator.exe, DisplaySwitch.exe, atbroker.exe		
attached_process	Full path to process to attach to target in #{parent_list}. Default: cmd.exe		
Path	C:\windows\system32\cmd.exe		

Attack Commands: Run with `powershell` ! Elevation Required (e.g. root or admin)

```
$input_table = "#{parent_list}".split(",")
$Name = "Debugger"
$Value = "#{attached_process}"
Foreach ($item in $input_table){
    $item = $item.trim()
    $registryPath = "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\$item"
    IF(!(Test-Path $registryPath))
    {
```

Sharing at the Oasis

Readme.md

AtomicTestHarnesses PowerShell Module

The `AtomicTestHarnesses` PowerShell module contains a suite of tools for simulating attack techniques. It is designed to be used on its own or as a dependency for `Atomic Red Team` tests. `AtomicTestHarnesses` is designed to run on PowerShell version 5 and above.

What problem does AtomicTestHarnesses aim to address?

Have you ever been asked the question and been held accountable to answer the following? "Do we detect attack technique X?" If so, you may be familiar with the initial level of discomfort involved in not knowing how to confidently answer that question. In order to tackle such a potentially broadly-scoped question, at Red Canary, one of our first questions will be, "can we see the technique in the first place independent of benign, suspicious, or malicious behaviors?" In order to "see" techniques, one would ideally have a handle on as many variants of a technique as possible and to then build test code that can exercise all those variants in a *repeatable* and *modular* fashion. Implementation of all known technique variations in an abstracted and repeatable fashion is the niche that `AtomicTestHarnesses` aims to fill. If you can observe all known technique variations, then you've laid a foundation to detect behaviors that employ a technique in a fashion that is resilient to evasion.

Installing the AtomicTestHarnesses Module



Sharing at the Oasis



Automated Testing Tools



How Do I Track My Results?



Sharing at the Oasis



Sharing at the Oasis



Edit T1015 - Attaches Command Prompt As Debugger To Process - utilman Test Case

Status:
NotPerformed

Attack
Start

Attack Stop

Source IPs

Red Team Details

Name
T1015 - Attaches Command Prompt As Debugger To Prc

Description
This allows adversaries to execute the attached process

Technique
Accessibility Features

Phase
Persistence

Operator Guidance
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\#{target_executable}" /v "Debugger" /t REG_SZ /d "C:\windows\system32\cmd.exe" /f

References
<https://github.com/redcanaryco/atomic-red>

Attacker
Tools

Target
Assets

Blue Team Details

Outcome
☒ TBD ☐ Blocked ☐ Detected
☐ NotDetected

Outcome Notes
outcomeNotes

Tags

Rules
Sigma

Detection
Time





Expected
Detection
Layers

Detection

Changes to accessibility utility binaries or binary paths that do not correlate with known software, patch cycles, etc., are suspicious. Command line invocation of tools capable of modifying the Registry for associated keys are also suspicious. Utility arguments and the binaries themselves should be monitored for changes. Monitor Registry keys within

Sharing at the Oasis

Edit T1015 - Attaches Command Prompt As Debugger To Process - utilman Test Case

<div>Status: Completed</div> <div></div> <div>Attack Start</div> <div>02/21/2020 13:31:10 status changed to InProgress</div> <div>Attack Stop</div> <div>02/21/2020 13:34:16 status changed to Completed</div>	<div>Red Team Details</div> <div>Name T1015 - Attaches Command Prompt As Debugger To Proc</div> <div>Description This allows adversaries to execute the attached process</div> <div>Technique Accessibility Features</div> <div>Phase Persistence</div> <div>Operator Guidance reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\#{target_executable}" /v "Debugger" /t REG_SZ /d "C:\windows\system32\cmd.exe" /f</div> <div>References https://github.com/redcanaryco/atomic-red</div> <div>Attacker Tools</div> <div>Target Assets</div>	<div>Blue Team Details</div> <div>Outcome <input type="checkbox"/> TBD <input checked="" type="checkbox"/> Blocked <input type="checkbox"/> Detected <input type="checkbox"/> NotDetected</div> <div>Detecting Blue Tool(s):</div> <div>QRadar</div> <div>Cb Response</div> <div>Cb Protection</div> <div>Was an alert triggered? <input checked="" type="checkbox"/> Yes <input type="checkbox"/> TBD <input type="checkbox"/> No</div> <div>Outcome Notes Exploitation detected and blocked by <u>EDR</u>, but did not raise account score in <u>UEBA</u>.</div> <div>Tags FOLLOWUP</div> <div>Rules Sigma</div>	<div>Detection Time</div> <div>02/21/2020 13:35:36 outcome changed to Blocked</div> <div>Expected Detection Layers</div> <div>SIEM EDR (Managed) EDR (Blocking) Behavior Analytics</div>
--	---	--	--

Cancel

Save

Next

Sharing at the Oasis



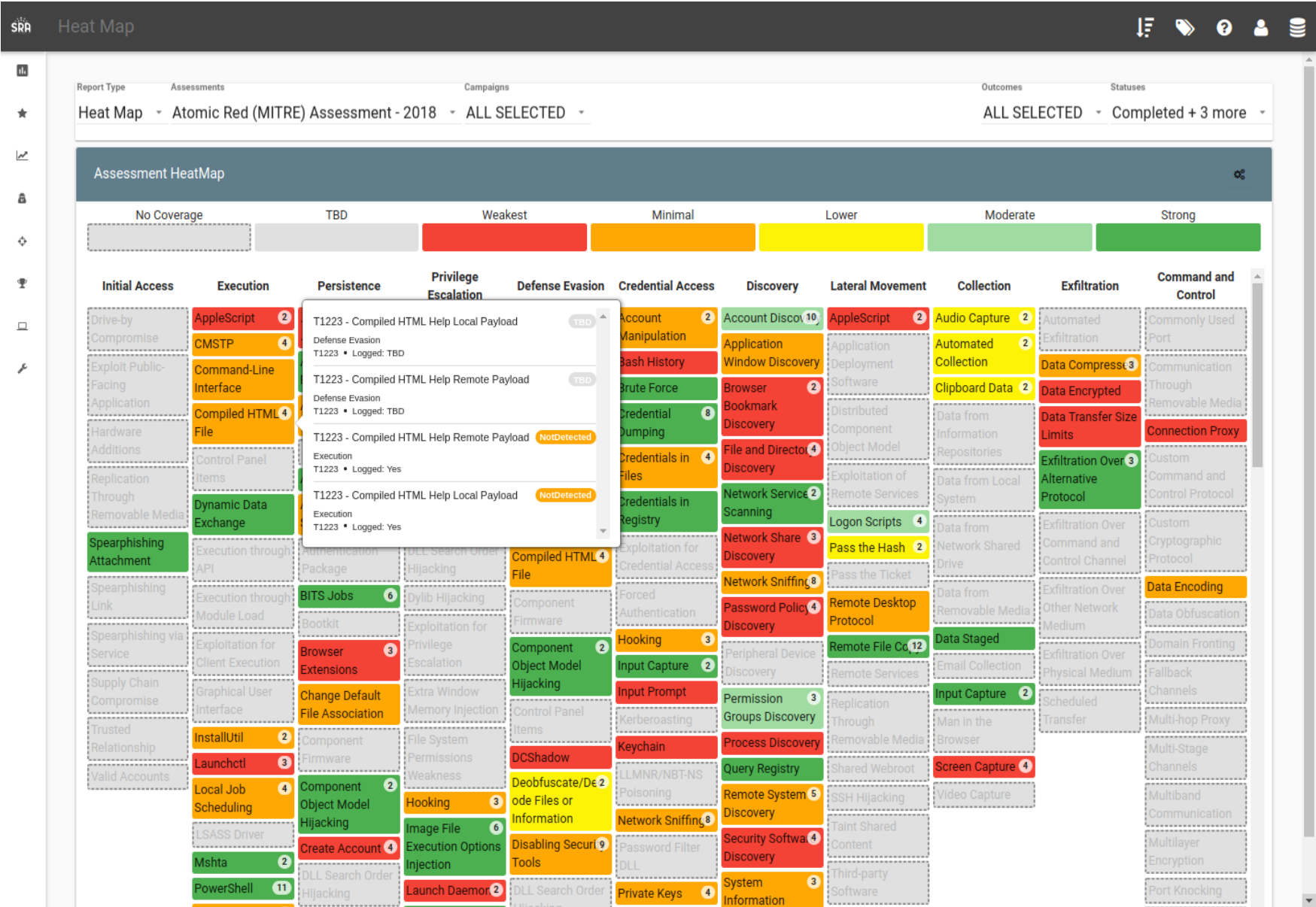
SRA DEMO_ATOMIC_RED_CE / Atomic Red (MITRE) Assessment - 2018

Assessment Dashboard

Name	Type	Progress	Outcome	Action	
Exfiltration	Campaign	100%	38% 13% 50%	LOAD	VIEW REPORT
Collection	Campaign	100%	42% 58%	LOAD	VIEW REPORT
Defense Evasion	Campaign	22% 78%	28% 11% 78%	LOAD	VIEW REPORT
Discovery	Campaign	69% 31%	29% 40% 31%	LOAD	VIEW REPORT
Privilege Escalation	Campaign	100%	9% 66% 25%	LOAD	VIEW REPORT
Lateral Movement	Campaign	100%	26% 47% 26%	LOAD	VIEW REPORT
Persistence	Campaign	30% 70%	11% 19% 70%	LOAD	VIEW REPORT
Execution	Campaign	100%	11% 51% 38%	LOAD	VIEW REPORT
Command & Control	Campaign	40% 60%	10% 30% 60%	LOAD	VIEW REPORT
Initial Access	Campaign	100%	100%	LOAD	VIEW REPORT
Credential Access	Campaign	100%	11% 32% 57%	LOAD	VIEW REPORT



Sharing at the Oasis



Sharing at the Oasis



How Do I Get Started?



Running Your First Exercise

- Provision target machine(s) identical to production standards.
- Decide ahead of time what TTPs to test and how.
- Get the blue team together.
- Designate someone to record results in VECTR.

Running Your First Exercise

- Work through the Atomic Red Team tests for the techniques you're testing.
- Record the results in VECTR.
- Anything that isn't blocked or detected as expected, tag for later followup.
- Repeat regularly.



Lessons We Have Learned

- Tool knowledge needs to be shared.
- Some vendor products don't meet their claims.
- Certain TTPs are only detected in Atomic form.
- Detection of a technique is rarely binary.
- Retest with new defenses and new attacks.

Conclusions



For More Information



@InfosecGoon



gracie.matthew@bcbswny.com



<https://github.com/InfosecGoon/>



<https://attack.mitre.org>

<https://atomicredteam.io>

<https://vectr.io>

