Transforming Cybersecurity Response with Zscaler™ using the MITRE ATT&CK™ Framework

zscaler™

Table of contents

©2021 Zscaler, Inc. All rights reserved.

Table of contents

©2021 Zscaler, Inc. All rights reserved.

Introduction

Defending an enterprise network against modern-day attacks remains an increasingly difficult challenge that requires, among other things, advanced technologies and innovative approaches for thwarting an adversary's goals. Because new and complex attacks are continuously created, there is a need for a common framework to understand how attackers operate to achieve their objectives. This framework should not only help in understanding the attack but also to understand existing defenses and what mitigations can be put in place to thwart attacks.

What is MITRE ATT&CK?

To help address the challenges of defending against modern attacks, MITRE Corporation developed a process for modeling an adversary's post-compromise behavior at a granular level with a common taxonomy. This model is named the ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework, and it serves as a knowledge base of commonly observed adversarial behaviors to support the efforts of threat intelligence functions, with adversary emulation and defensive gap analysis.

ATT&CK was created out of the need to systematically document and catalog adversaries' behaviors based on millions of data points observed from real-life attacks and breaches. The model describes the Tactics, Techniques, and Procedures (TTPs) of adversarial behavior and breaks them into categories based on the sequence of steps involved in an attack. It is not intended to be exhaustive and is very much a living framework that is continuously updated as new TTPs are discovered.

How can an enterprise benefit from ATT&CK?

The goal of ATT&CK is to break down, classify, and document adversarial behaviors from previously observed attacks in a common language that is consistent and clear. This type of cataloging and identification provides a number of benefits, such as the following use cases:

- **Adversary emulation:** Test and verify defenses against common techniques of adversaries.
- **Red team:** Create plans and organize operations to avoid certain defensive measures that may be in place within a network.
- **Evaluate current defenses:** Assess tools, monitoring, and mitigation capabilities of existing defenses within an organization's environment.
- **Finding gaps in coverage:** Identify gaps as a way to prioritize investments for security improvements. Similar security products can also be compared against a common adversarial behavior model to determine coverage.
- **Prioritize detections:** Identify and rank alerts based on their potential threat level.

What is the difference between ATT&CK and the Lockheed Martin Cyber Kill Chain framework?

ATT&CK and **Lockheed Martin's Cyber Kill Chain** resemble each other in that both are models that define the steps attackers use to achieve their goals. ATT&CK sits at a lower level of definition to describe adversarial behavior than the Cyber Kill Chain. ATT&CK tactics are unordered and may not all occur in a single intrusion because adversaries' tactical goals change throughout an operation, whereas the Cyber Kill Chain uses ordered phases to describe high-level adversarial objectives.

©2021 Zscaler, Inc. All rights reserved.

What is the ATT&CK model?

The ATT&CK model is an ordered list of observed behaviors from known attacks. These behaviors are known as tactics and techniques.

ATT&CK is visually organized into a few different matrices: PRE-ATT&CK, Enterprise, and Mobile. Each of these matrices contains various tactics and techniques relevant to its domain.

PRE-ATT&CK is a matrix of tactics and techniques related to what attackers do before they try to exploit a particular target network or system. The Enterprise matrix contains tactics and techniques that apply to Windows, Linux, and/or MacOS systems. Mobile matrix contains tactics and techniques that apply to mobile devices.

The scope of this paper is limited to the Enterprise Matrix.

The MITRE ATT&CK Framework: Matrix for Enterprise

The MITRE ATT&CK framework Enterprise Matrix is composed of 14 tactics, including the first two PRE-ATT&CK tactics, each with associated techniques. The tactics appear in roughly sequential order, following the general stages of a comprehensive (read: worst-case) adversarial attack:

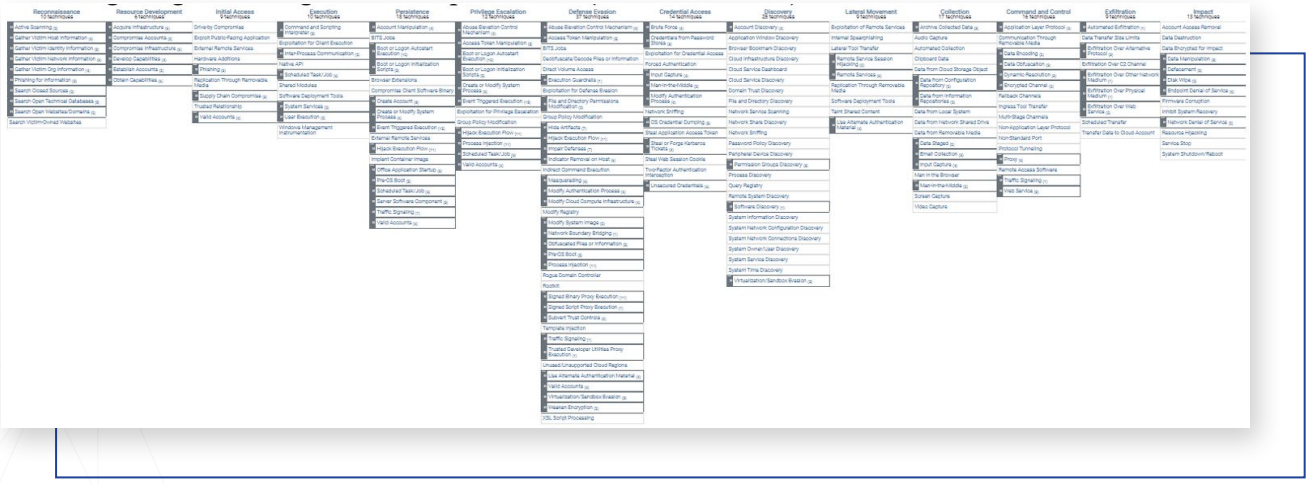

Figure 1. The MITRE ATT&CK framework Matrix for Enterprise. Source: MITRE Corporation.

Tactics

Tactics represent the "**why**" of an ATT&CK technique. It is the adversary's tactical objective for performing an action. Tactics serve as useful contextual categories for individual techniques and cover standard notations for things adversaries do during an operation, such as persist, discover information, move laterally, execute files, and exfiltrate data. There are 12 tactics that have been observed from previous attacks and are defined in the ATT&CK matrix.

©2021 Zscaler, Inc. All rights reserved.

Techniques

Techniques represent "**how**" an adversary achieves a tactical objective by performing an action. Techniques may also represent "**what**" an adversary gains by performing an action. For example, an adversary may dump credentials from an operating system to gain access to useful credentials within a network. There may be many ways, or techniques, to achieve tactical objectives, so there are multiple techniques in each tactic category.

Sub-techniques

Sub-techniques describe "**how**" an adversary achieves a tactical objective in more detail, including the specific tools used. For example, an adversary may use spear-phishing as a targeted phishing attack to make the phishing technique look more genuine.

Procedure

Procedure details the steps an adversary takes to achieve a goal, including specific tools, methods, and operating systems used.

How customers can use Zscaler with ATT&CK

The Zscaler Internet Access™ (ZIA™) solution can be directly mapped to mitigations for various ATT&CK techniques. Leveraging ZIA engines, such as the Advanced Cloud Firewall, Advanced Threat Protection, Malware Protection, CASB, DLP, Advanced Cloud Sandbox, and others, ZIA has a multitude of mechanisms available to not only detect documented ATT&CK techniques, but also defend against them.

Additionally, Zscaler integrates with endpoint detection and response (EDR) vendors to complement their solutions and provide protections against adversarial tactics and techniques that are endpoint-specific or that are insider attacks or laterally moving threats.

Zscaler's unique cloud-native multitenant architecture

ZIA is a secure internet and web gateway delivered as a service from the cloud. ZIA is a truly distributed multitenant, custom-built TCP forward-proxy architecture that is cloud-native, making it highly scalable to allow for full content inspection with SSL decryption. The Zscaler Zero Trust Exchange™, the platform on which all Zscaler services are delivered, processes more than 160 billion requests per day at peak periods and receives 175,000 unique threat updates daily.

The complete Zscaler platform is expertly positioned to disrupt the kill chain in several areas.

Its layered approach helps stop inbound threats from reputation-based blocking all the way down to advanced behavioral analysis. An integrated approach helps provide full threat context and visibility. It's important to note that customers looking for this level of inspection from other vendors would have to piece together several solutions.

For outbound protection, Zscaler can deliver complete protection from botnet callbacks and malicious outbound activity, which helps disrupt data exfiltration and malware attempting to persist within the network.

©2021 Zscaler, Inc. All rights reserved.

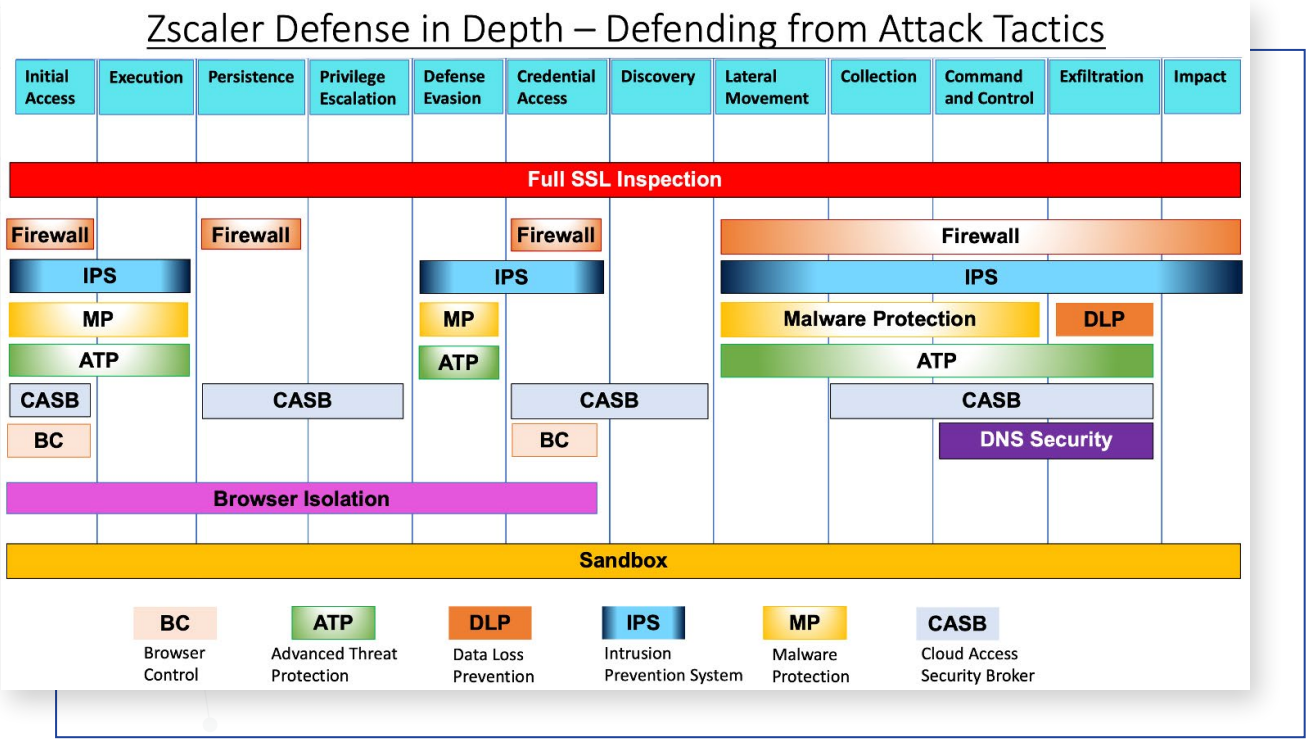

Figure 2. How ZIA security engines align with the MITRE ATT&CK framework tactics.

The MITRE ATT&CK-relevant Zscaler security services include Advanced Threat Protection (ATP), Browser Control capabilities, Data Loss Prevention (DLP), File Type Control, Cloud Firewall, Intrusion Prevention System (IPS), Malware Protection, Cloud Sandbox, SSL Inspection, and URL Filtering.

ATT&CK Tactics & Zscaler-Recommended Security Engines

Below are descriptions of each of the MITRE ATT&CK Enterprise Matrix tactics and our mapping of TTPs (Tactics, Techniques, and Procedures) to Zscaler engines that can detect and mitigate the associated techniques.

Note: Zscaler engine detection of these TTPs is limited to malicious traffic, which includes payloads passing through ZIA cloud security. Since attackers have a multitude of mechanisms available to compromise an endpoint, such as when it's outside the corporate network or in the case of hardware additions, and more, we strongly recommend an endpoint detection and response (EDR) solution to complement and provide additional protections against adversarial tactics and techniques that are endpoint-specific, as well as prevention of insider attacks and lateral movement of threats. Zscaler integrates with EDR vendors through API support.

Enterprise Matrix

The Enterprise Matrix is defined with 12 tactics (initial access, execution, persistence, privilege escalation, defense evasion, credential access, discovery, lateral movement, collection, command and control, exfiltration, and impact) and more than 250 techniques and sub-techniques categorized by the tactic the adversary is trying to achieve.

©2021 Zscaler, Inc. All rights reserved.

Initial access

Initial access tactics consist of various techniques an adversary might use to gain an initial foothold in your network. Techniques include targeted spear-phishing and drive-by compromise.

Example: One technique used by **APT19** is a drive-by compromise. A drive-by compromise occurs when an adversary gains access to a system through a user visiting a website over the normal course of browsing. Attackers executed a watering-hole attack in 2014 to lure targets to a site they were known to visit—forbes.com—and performed drive-by compromises on those targets.

Spear-phishing web email is another example of a technique used by attackers. While the spear-phishing web email itself may not be blocked by ZIA engines, any phishing/malicious link or attachment embedded within the web email will be blocked, thereby blocking the attacker's ability to compromise the end-user system.

Initial Access		Zscaler Security Engine	Recommended Policy Actions			
Drive-by Compromise		● ● ●	Configure ATP	Configure Browser Control	Configure Sandbox	
Exploit Public-Facing Application		○				
External Remote Services		○				
Hardware Additions		○				
Replication Through Removable Media		○				
Phishing (3)	Spearphishing Attachment	● ● ●	Configure ATP	Configure Malware Protection	Configure IPS	Configure Sandbox
	Spearphishing Link	● ● ●	Configure ATP	Configure IPS	Configure Sandbox	
	Spearphishing via Service	● ● ●	Configure ATP	Configure Malware Protection	Configure IPS	Configure Sandbox
Supply Chain Compromise (3)	Compromise Software Dependencies and Development Tools	●	Configure Sandbox			
	Compromise Software Supply Chain	● ●	Configure Sandbox	Configure Malware Protection		
	Compromise Hardware Supply Chain	○				
Trusted Relationship		● ● ● ●	Configure Firewall	Configure ATP	Configure IPS	Configure Sandbox
Valid Accounts (4)	Default Accounts	○				
	Domain Accounts	○				
	Local Accounts	○				
	Cloud Accounts	●	Configure CASB			

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

Figure 3. Initial access tactics, ZIA security engine detection, and recommended actions.

Zscaler-recommended security engines to protect against initial access techniques:

- Apply **Advanced Threat Protection** to protect against phishing attempts, malicious active content and sites, cross-site scripting (XSS), proxy anonymizers, and peer-to-peer (P2P) file sharing.
- Configure **Malware Protection** policy to protect against malware, viruses, spyware, and users clicking on a "malicious link" or downloading a "malicious file" from the internet.
- Configure **Browser Control** policy to reduce risk of older and/or vulnerable browsers being exploited for drive-by compromise attacks.
- Implement **Sandbox** to shield against zero-day and any unknown threats.

See Appendix A for details on Zscaler security engines and recommended policy configuration.

Execution

Execution tactics consist of techniques that result in adversary-controlled code running on a local or remote system. This technique is often paired with other techniques such as network discovery or remote system discovery.

Example: APT41 leveraged PowerShell to deploy malware families in victims' environments. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code.

©2021 Zscaler, Inc. All rights reserved.

While techniques such as command-line interface, PowerShell, service execution, user execution, etc., are typically executed in an end-user system locally and is detected by endpoint security, these techniques can also be detected by ZIA engines when an attacker's suspicious code is run within a virtual environment that mimics end-user system and behavior. Since ZIA inspects the traffic inline and has the ability to quarantine the suspicious file while it's being analyzed, we can block the malicious code from reaching the user.

Note: Zscaler engine detection of these TTPs is limited to malicious traffic that includes payloads passing through ZIA cloud security.

See Appendix C for an example of ZIA engine detection when "PowerShell" is used as part of an attack.

See Appendix D for more examples from Zscaler Security Research Blogs detailing Zscaler engine detection of various attack techniques and MITRE ATT&CK Mapping.

Execution		Zscaler Security Engine	Recommended Policy Actions		
Command and Scripting Interpreter (7)	PowerShell	●	Configure Sandbox		
	AppleScript	○			
	Windows Command Shell	●	Configure Sandbox		
	Unix Shell	○			
	Visual Basic	● ●	Configure Sandbox	Configure Malware Protection	
	Python	●	Configure Malware Protection		
	JavaScript/JScript	● ●	Configure Malware Protection	Configure IPS	
Exploitation for Client Execution		●	Configure Sandbox		
Inter-Process Communication (2)	Component Object Model	●	Configure Sandbox		
	Dynamic Data Exchange	●	Configure Sandbox		
Native API		● ●	Configure Sandbox	Configure Malware Protection	
Scheduled Task/Job (5)	At (Windows)	○			
	Scheduled Task	○			
	At (Linux)	○			
	Launchd	○			
	Cron	○			
Shared Modules		●	Configure Sandbox		
Software Deployment Tools		●	Configure Sandbox		
System Services (2)	Launchctl	○			
	Service Execution	●	Configure Sandbox		
User Execution (2)	Malicious Link	● ● ●	Configure Malware Protection	Configure ATP	Configure IPS
	Malicious File	● ●	Configure Malware Protection	Configure Malware Protection	
Windows Management Instrumentation		●	Configure Sandbox		

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

Figure 4. Execution tactics, ZIA security engine detection, and recommended actions.

Zscaler-recommended security engines to protect against execution techniques:

- Apply **Malware Protection** policy to protect against malware, viruses, spyware, and users clicking on a "malicious link" or downloading a "malicious file" from the internet.
- Configure **URL Filtering** to limit enterprise risk exposure by managing user access to web content based on site categorization.
- Implement **Sandbox** to shield against zero-day and any unknown threats.

See Appendix A for details on Zscaler security engines and recommended policy configuration.

©2021 Zscaler, Inc. All rights reserved.

Persistence

Persistence tactics consist of techniques that adversaries use to keep access to systems across restarts, changed credentials, and other interruptions that could cut off their access. Techniques used for persistence include any access, action, or configuration changes that let them maintain their foothold on systems, such as replacing or hijacking legitimate code or adding startup code. More than 60 techniques are detailed under this tactic.

Example: Adversaries may use more than one remote access tool with varying command-and-control protocols or credentialed access to remote services so they can continue to maintain access even if an access mechanism is detected or mitigated. **APT3** has been known to use multiple backdoors per campaign.

Persistence		Zscaler Security Engine	Recommended Policy Actions	
Account Manipulation (4)	Additional Azure Service Principal Credentials	○		
	Exchange Email Delegate Permissions	○		
	Add Office 365 Global Administrator Role	○		
	SSH Authorized Keys	○		
BITS Jobs		●	Configure Sandbox	
Boot or Logon Autostart Execution (11)	Registry Run Keys / Startup Folder	●	Configure Sandbox	
	Authentication Package	●	Configure Sandbox	
	Time Providers	●	Configure Sandbox	
	Winlogon Helper DLL	●	Configure Sandbox	
	Security Support Provider	●	Configure Sandbox	
	Kernel Modules and Extensions	●	Configure Sandbox	
	Re-opened Applications	○		
	LSASS Driver	●	Configure Sandbox	
	Shortcut Modification	●	Configure Sandbox	
	Port Monitors	●	Configure Sandbox	
	Plist Modification	○		
Boot or Logon Initialization Scripts (5)	Logon Script (Windows)	●	Configure Sandbox	
	Logon Script (Mac)	○		
	Network Logon Script	○		
	Rc.common	○		
	Startup Items	○		
Browser Extensions		●	Configure Sandbox	
Compromise Client Software Binary		●	Configure Sandbox	
Create Account (3)	Local Account	●	Configure Sandbox	
	Domain Account	○		
	Cloud Account	○		
Create or Modify System Process (4)	Launch Agent	○		
	Systemd Service	○		
	Windows Service	●	Configure Sandbox	
	Launch Daemon	○		
Event Triggered Execution (15)	Change Default File Association	●	Configure Sandbox	
	Screensaver	●	Configure Sandbox	
	Windows Management Instrumentation Event Subscription	●	Configure Sandbox	
	.bash_profile and .bashrc	○		
	Trap	○		
	LC_LOAD_DYLIB Addition	○		
	Netsh Helper DLL	●	Configure Sandbox	
	Accessibility Features	●	Configure Sandbox	
	AppCert DLLs	●	Configure Sandbox	
	AppInit DLLs	●	Configure Sandbox	
	Application Shimming	●	Configure Sandbox	
	Image File Execution Options Injection	●	Configure Sandbox	
	PowerShell Profile	●	Configure Sandbox	
	Emond	○		
	Component Object Model Hijacking	●	Configure Sandbox	

[contd.]

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

©2021 Zscaler, Inc. All rights reserved.

Persistence		Zscaler Security Engine	Recommended Policy Actions	
External Remote Services		○		
Hijack Execution Flow (11)	Services File Permissions Weakness	●	Configure Sandbox	
	Executable Installer File Permissions Weakness	●	Configure Sandbox	
	Services Registry Permissions Weakness	●	Configure Sandbox	
	Path Interception by Unquoted Path	●	Configure Sandbox	
	Path Interception by PATH Environment Variable	●	Configure Sandbox	
	Path Interception by Search Order Hijacking	●	Configure Sandbox	
	DLL Search Order Hijacking	●	Configure Sandbox	
	DLL Side-Loading	○		
	LD_PRELOAD	○		
	Dylib Hijacking	○		
	COR_PROFILER	●	Configure Sandbox	
Implant Container Image		○		
Office Application Startup (6)	Add-ins	●	Configure Sandbox	
	Office Template Macros	●	Configure Sandbox	
	Outlook Forms	●	Configure Sandbox	
	Outlook Rules	●	Configure Sandbox	
	scheduled Task	●	Configure Sandbox	
	Office Test	●	Configure Sandbox	
Pre-OS Boot (3)	System Firmware	○		
	Component Firmware	○		
	Bootkit	●	Configure Sandbox	
Scheduled Task/Job (5)	At (Windows)	●	Configure Sandbox	
	Scheduled Task	●	Configure Sandbox	
	At (Linux)	○		
	Launchd	○		
	Cron	○		
Server Software Component (3)	SQL Stored Procedures	●	Configure Sandbox	
	Transport Agent	○		
	Web Shell	○		
Traffic Signaling (1)	Port Knocking	● ●	Configure Sandbox	Configure Firewall
Valid Accounts (4)	Default Accounts	●	Configure Sandbox	
	Domain Accounts	○		
	Local Accounts	●	Configure Sandbox	
	Cloud Accounts	○		

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

Figure 5. Persistence tactics, ZIA security engine detection, and recommended actions.

While techniques such as dylib hijacking, hidden files and directories, logon scripts, PowerShell profile, etc., are typically executed in an end-user system locally and are detected by endpoint security, these techniques can also be detected by ZIA engines when an attacker's suspicious code is run within a virtual machine environment that mimics end-user systems and behaviors. Since ZIA inspects the traffic inline and has the ability to quarantine the suspicious file while it's being analyzed, we can capture and analyze the malicious code before it reaches the endpoint and blocks the malicious code from reaching the user.

Note: Zscaler engine detection of these TTPs is limited to malicious traffic that includes payloads passing through ZIA cloud security.

See Appendix C for an example of ZIA engine detection when "hooking" and "registry run keys/startup folder" techniques are used as part of an attack.

Zscaler-recommended steps to protect against persistence techniques:

- Configure a **Sandbox** policy to detect and block malicious code before delivery to endpoint.
- Employ EDR endpoint security to complement protection against adversarial persistence techniques.

See Appendix A for details on Zscaler security engines and recommended policy configuration.

©2021 Zscaler, Inc. All rights reserved.

Privilege escalation

Privilege escalation tactics consist of techniques used by adversaries to gain higher-level permissions on a system or network. Adversaries often enter and explore a network with unprivileged access, but look for ways, such as system weaknesses, misconfigurations, and vulnerabilities, to elevate their privilege to a higher system or admin-level permissions to follow through on their objectives.

Example: **APT28** has used CVE-2015-1701 to access the SYSTEM token and copy it into the current process as part of privilege escalation.

Privilege Escalation		Zscaler Security Engine	Recommended Policy Actions
Abuse Elevation Control Mechanism (4)	Setuid and Setgid	○	
	Bypass User Access Control	●	Configure Sandbox
	Sudo and Sudo Caching	○	
	Elevated Execution with Prompt	○	
Access Token Manipulation (5)	Token Impersonation/Theft	●	Configure Sandbox
	Create Process with Token	●	Configure Sandbox
	Make and Impersonate Token	●	Configure Sandbox
	Parent PID Spoofing	●	Configure Sandbox
	SID-History Injection	●	Configure Sandbox
Boot or Logon Autostart Execution (11)	Registry Run Keys / Startup Folder	●	Configure Sandbox
	Authentication Package	●	Configure Sandbox
	Time Providers	●	Configure Sandbox
	Winlogon Helper DLL	●	Configure Sandbox
	Security Support Provider	●	Configure Sandbox
	Kernel Modules and Extensions	●	Configure Sandbox
	Re-opened Applications	○	
	LSASS Driver	●	Configure Sandbox
	Shortcut Modification	●	Configure Sandbox
	Port Monitors	●	Configure Sandbox
	Plist Modification	○	
Boot or Logon Initialization Scripts (5)	Logon Script (Windows)	●	Configure Sandbox
	Logon Script (Mac)	○	
	Network Logon Script	○	
	Rc.common	○	
	Startup Items	○	
Create or Modify System Process (4)	Launch Agent	○	
	Systemd Service	○	
	Windows Service	●	Configure Sandbox
	Launch Daemon	○	
Event Triggered Execution (15)	Change Default File Association	●	Configure Sandbox
	Screensaver	●	Configure Sandbox
	Windows Management Instrumentation Event Subscription	●	Configure Sandbox
	.bash_profile and .bashrc	○	
	Trap	○	
	LC_LOAD_DYLIB Addition	○	
	Netsh Helper DLL	●	Configure Sandbox
	Accessibility Features	●	Configure Sandbox
	AppCert DLLs	●	Configure Sandbox
	AppInit DLLs	●	Configure Sandbox
	Application Shimming	●	Configure Sandbox
	Image File Execution Options Injection	●	Configure Sandbox
	PowerShell Profile	●	Configure Sandbox
	Emond	○	
	Component Object Model Hijacking	●	Configure Sandbox
Exploitation for Privilege Escalation		●	Configure Sandbox
Group Policy Modification		●	Configure Sandbox

[contd.]

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

©2021 Zscaler, Inc. All rights reserved.

Privilege Escalation		Zscaler Security Engine	Recommended Policy Actions
Hijack Execution Flow (11)	Services File Permissions Weakness	●	Configure Sandbox
	Executable Installer File Permissions Weakness	●	Configure Sandbox
	Services Registry Permissions Weakness	●	Configure Sandbox
	Path Interception by Unquoted Path	●	Configure Sandbox
	Path Interception by PATH Environment Variable	●	Configure Sandbox
	Path Interception by Search Order Hijacking	●	Configure Sandbox
	DLL Search Order Hijacking	●	Configure Sandbox
	DLL Side-Loading	●	Configure Sandbox
	LD_PRELOAD	○	
	Dylib Hijacking	○	
	COR_PROFILER	○	
Process Injection (11)	Dynamic-link Library Injection	●	Configure Sandbox
	Portable Executable Injection	●	Configure Sandbox
	Thread Execution Hijacking	●	Configure Sandbox
	Asynchronous Procedure Call	●	Configure Sandbox
	Thread Local Storage	●	Configure Sandbox
	Ptrace System Calls	○	
	Proc Memory	○	
	Extra Window Memory Injection	●	Configure Sandbox
	Process Doppelgänging	●	Configure Sandbox
	Process Hollowing	●	Configure Sandbox
	VDSO Hijacking	○	
Scheduled Task/Job (5)	At (Windows)	●	Configure Sandbox
	Scheduled Task	●	Configure Sandbox
	At (Linux)	○	
	Launchd	○	
	Cron	○	
Valid Accounts (4)	Default Accounts	●	Configure Sandbox
	Domain Accounts	○	
	Local Accounts	●	Configure Sandbox
	Cloud Accounts	○	

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

Figure 6. Privilege escalation tactics, ZIA security engine detection, and recommended actions.

While techniques such as access token manipulation, hooking, path interception, etc., are typically executed in an end-user system locally and detected by endpoint security, these techniques can also be detected by ZIA engines when an attacker's suspicious code is run within a virtual machine environment that mimics end-user systems and behaviors. Since ZIA inspects the traffic inline and has the ability to quarantine the suspicious file while it's being analyzed, we can capture and analyze the malicious code before it reaches the endpoint and blocks the malicious code from reaching the user.

Note: Zscaler engine detection of these TTPs is limited to malicious traffic that includes payloads passing through ZIA cloud security.

See Appendix C for an example of ZIA engine detection when "hooking" or "process injection" techniques are used as part of an attack.

Zscaler-recommended steps to protect against privilege escalation techniques:

- Configure a **Sandbox** policy to detect and block malicious code before delivery to an endpoint.
- Employ EDR endpoint security to complement protection against adversarial privilege escalation techniques.

See Appendix A for details on Zscaler security engines and recommended policy configuration.

©2021 Zscaler, Inc. All rights reserved.

Defense evasion

Defense evasion tactics consist of techniques that adversaries use to avoid detection. Techniques include uninstalling/disabling security software or obfuscating/encrypting data and scripts or clearing security logs or history. Adversaries also leverage and abuse trusted processes to hide and masquerade their malware.

Example: Adversaries may use a connection proxy to direct network traffic between systems or act as an intermediary for network communications to a command-and-control server to avoid direct connections to their infrastructure. **APT28** used other victims as proxies to relay command traffic, for instance using a compromised Georgian military email server as a hop-point to NATO victims.

Defense Evasion		Zscaler Security Engine	Recommended Policy Actions		
Abuse Elevation Control Mechanism (4)	Setuid and Setgid	○			
	Bypass User Access Control	●	Configure Sandbox		
	Sudo and Sudo Caching	○			
	Elevated Execution with Prompt	○			
Access Token Manipulation (5)	Token Impersonation/Theft	●	Configure Sandbox		
	Create Process with Token	●	Configure Sandbox		
	Make and Impersonate Token	●	Configure Sandbox		
	Parent PID Spoofing	●	Configure Sandbox		
	SID-History Injection	●	Configure Sandbox		
BITS Jobs		●	Configure Sandbox		
Deobfuscate/Decode Files or Information		●	Configure Sandbox		
Direct Volume Access		●	Configure Sandbox		
Execution Guardrails (1)	Environmental Keying	●	Configure Sandbox		
Exploitation for Defense Evasion		●	Configure Sandbox		
File and Directory Permissions Modification (2)	Windows File and Directory Permissions Modification	●	Configure Sandbox		
	Linux and Mac File and Directory Permissions Modification	○			
Group Policy Modification		●	Configure Sandbox		
Hide Artifacts (6)	Hidden Files and Directories	●	Configure Sandbox		
	Hidden Users	○			
	Hidden Window	●	Configure Sandbox		
	NTFS File Attributes	●	Configure Sandbox		
	Hidden File System	●	Configure Sandbox		
	Run Virtual Instance	●	Configure Sandbox		
Hijack Execution Flow (11)	Services File Permissions Weakness	●	Configure Sandbox		
	Executable Installer File Permissions Weakness	●	Configure Sandbox		
	Services Registry Permissions Weakness	●	Configure Sandbox		
	Path Interception by Unquoted Path	●	Configure Sandbox		
	Path Interception by PATH Environment Variable	●	Configure Sandbox		
	Path Interception by Search Order Hijacking	●	Configure Sandbox		
	DLL Search Order Hijacking	●	Configure Sandbox		
	DLL Side-Loading	●	Configure Sandbox		
	LD_PRELOAD	○			
	Dylib Hijacking	○			
	COR_PROFILER	●	Configure Sandbox		
Impair Defenses (6)	Disable or Modify Tools	●	Configure Sandbox		
	Disable Windows Event Logging	●	Configure Sandbox		
	HISTCONTROL	○			
	Disable or Modify System Firewall	●	Configure Sandbox		
	Indicator Blocking	●	Configure Sandbox		
	Disable or Modify Cloud Firewall	●	Configure Sandbox		
Indicator Removal on Host (6)	Clear Windows Event Logs	●	Configure Sandbox		
	Clear Linux or Mac System Logs	○			
	Clear Command History	○			
	File Deletion	●	Configure Sandbox		
	Network Share Connection Removal	●	Configure Sandbox		
	Timestomp	●	Configure Sandbox		

[contd.]

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

©2021 Zscaler, Inc. All rights reserved.

Defense Evasion		Zscaler Security Engine	Recommended Policy Actions		
Indirect Command Execution		●	Configure Sandbox		
Masquerading (6)	Invalid Code Signature	●	Configure Sandbox		
	Right-to-Left Override	●	Configure Sandbox		
	Rename System Utilities	●	Configure Sandbox		
	Masquerade Task or Service	●	Configure Sandbox		
	Match Legitimate Name or Location	●	Configure Sandbox		
	Space after Filename	● ● ●	Configure Sandbox	Configure IPS	Configure Malware Protection
Modify Authentication Process (3)	Domain Controller Authentication	●	Configure Sandbox		
	Password Filter DLL	●	Configure Sandbox		
	Pluggable Authentication Modules	○			
Modify Cloud Compute Infrastructure (4)	Create Snapshot	○			
	Create Cloud Instance	○			
	Delete Cloud Instance	○			
	Revert Cloud Instance	○			
Modify Registry		●	Configure Sandbox		
Obfuscated Files or Information (5)	Binary Padding	●	Configure Sandbox		
	Software Packing	●	Configure Sandbox		
	Steganography	○			
	Compile After Delivery	●	Configure Sandbox		
	Indicator Removal from Tools	●	Configure Sandbox		
Pre-OS Boot (3)	System Firmware	○			
	Component Firmware	○			
	Bootkit	●	Configure Sandbox		
Process Injection (11)	Dynamic-link Library Injection	●	Configure Sandbox		
	Portable Executable Injection	●	Configure Sandbox		
	Thread Execution Hijacking	●	Configure Sandbox		
	Asynchronous Procedure Call	●	Configure Sandbox		
	Thread Local Storage	●	Configure Sandbox		
	Ptrace System Calls	○			
	Proc Memory	○			
	Extra Window Memory Injection	●	Configure Sandbox		
	Process Doppelgänging	●	Configure Sandbox		
	Process Hollowing	●	Configure Sandbox		
	VDSO Hijacking	○			
Rogue Domain Controller		○			
Rootkit		●	Configure Sandbox		
Signed Binary Proxy Execution (10)	Rundll32	●	Configure Sandbox		
	Compiled HTML File	●	Configure Sandbox		
	Control Panel	●	Configure Sandbox		
	CMSTP	●	Configure Sandbox		
	InstallUtil	●	Configure Sandbox		
	Mshta	●	Configure Sandbox		
	Regsvcs/Regasm	●	Configure Sandbox		
	Regsvr32	●	Configure Sandbox		
	Msiexec	●	Configure Sandbox		
	Odbcconf	●	Configure Sandbox		
Signed Script Proxy Execution (1)	PubPrn	●	Configure Sandbox		

[contd]

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

Defense Evasion		Zscaler Security Engine	Recommended Policy Actions		
Subvert Trust Controls (4)	Gatekeeper Bypass	○			
	Code Signing	●	Configure Sandbox		
	SIP and Trust Provider Hijacking	●	Configure Sandbox		
	Install Root Certificate	●	Configure Sandbox		
Template Injection		● ● ●	Configure Sandbox	Configure IPS	Configure ATP
Traffic Signaling (1)	Port Knocking	●	Configure Firewall		
Trusted Developer Utilities Proxy Execution (1)	MSBuild	●	Configure Sandbox		
Unused/Unsupported Cloud Regions		○			
Use Alternate Authentication Material (4)	Pass the Hash	○			
	Pass the Ticket	○			
	Application Access Token	○			
	Web Session Cookie	○			
Valid Accounts (4)	Default Accounts	●	Configure Sandbox		
	Domain Accounts	○			
	Local Accounts	●	Configure Sandbox		
	Cloud Accounts	○			
Virtualization/Sandbox Evasion (3)	System Checks	●	Configure Sandbox		
	User Activity Based Checks	●	Configure Sandbox		
	Time Based Evasion	●	Configure Sandbox		
XSL Script Processing		○			

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

Figure 7. Defense evasion tactics, ZIA security engine detection, and recommended actions.

While techniques such as file and directory permissions modification, file deletion, install root certificate, etc., are typically executed in an end-user system locally and are detected by endpoint security, these techniques can also be detected by ZIA engines when an attacker's suspicious code is run within a virtual machine environment that mimics end-user systems and behaviors. Since ZIA inspects the traffic inline and has the ability to quarantine the suspicious file while it's being analyzed, we can capture and analyze the malicious code before it reaches the endpoint and blocks the malicious code from reaching the user.

©2021 Zscaler, Inc. All rights reserved.

Note: Zscaler engine detection of these TTPs is limited to malicious traffic that includes payloads passing through ZIA cloud security.

See Appendix C for an example of ZIA engine detection when "FileDeletion" or "Install Root Certificate" techniques are used as part of an attack.

Zscaler-recommended steps to protect against defense evasion techniques:

- Configure a **Sandbox** policy to detect and block malicious code before delivery to an endpoint.
- Employ EDR endpoint security to complement protection against adversarial defense evasion techniques.

See Appendix A for details on Zscaler security engines and recommended policy configuration.

Credential access

Credential access tactics consist of techniques for stealing credentials such as account names and passwords. Techniques used to get credentials include keylogging or credential dumping. Using legitimate credentials can give adversaries access to systems, make them harder to detect, and provide the opportunity to create more accounts to help achieve their goals.

Example: Adversaries may use brute-force techniques to attempt access to accounts when passwords are unknown or when password hashes are obtained. **APT41** performed password brute-force attacks on the local admin account.

Credential Access		Zscaler Security Engine	Recommended Policy Actions	
Brute Force (4)	Password Guessing	○		
	Password Cracking	○		
	Password Spraying	○		
	Credential Stuffing	○		
Credentials from Password Stores (3)	Keychain	○		
	Securityd Memory	○		
	Credentials from Web Browsers	●	Configure Sandbox	
Exploitation for Credential Access		●	Configure Sandbox	
Forced Authentication		●	Configure Firewall	
Input Capture (4)	Keylogging	●	Configure Sandbox	
	GUI Input Capture	●	Configure Sandbox	
	Web Portal Capture	●	Configure IPS	
	Credential API Hooking	●	Configure Sandbox	
Man-in-the-Middle (1)	LLMNR/NBT-NS Poisoning and SMB Relay	●	Configure Sandbox	
Modify Authentication Process (3)	Domain Controller Authentication	●	Configure Sandbox	
	Password Filter DLL	●	Configure Sandbox	
	Pluggable Authentication Modules	○		
Network Sniffing		●	Configure Sandbox	
OS Credential Dumping (8)	LSASS Memory	●	Configure Sandbox	
	Security Account Manager	●	Configure Sandbox	
	NTDS	●	Configure Sandbox	
	DCSync	●	Configure Sandbox	
	Proc Filesystem	○		
	/etc/passwd and /etc/shadow	○		
	Cached Domain Credentials	●	Configure Sandbox	
	LSA Secrets	●	Configure Sandbox	
Steal Application Access Token		● ●	Configure Sandbox	Configure CASB
Steal or Forge Kerberos Tickets (3)	Golden Ticket	●	Configure Sandbox	
	Silver Ticket	●	Configure Sandbox	
	Kerberoasting	●	Configure Sandbox	
Steal Web Session Cookie		●	Configure Sandbox	
Two-Factor Authentication Interception		●	Configure Sandbox	
Unsecured Credentials (6)	Credentials In Files	●	Configure Sandbox	
	Credentials in Registry	●	Configure Sandbox	
	Bash History	○		
	Private Keys	●	Configure Sandbox	
	Cloud Instance Metadata API	○		
	Group Policy Preferences	●	Configure Sandbox	

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

Figure 8. Credential access tactics, ZIA security engine detection, and recommended actions.

©2021 Zscaler, Inc. All rights reserved.

While techniques such as account manipulation, credentials from web browsers, input capture, input prompt, etc., are typically executed in an end-user system locally and are detected by endpoint security, these techniques can also be detected by ZIA engines when an attacker's suspicious code is run within a virtual machine environment that mimics end-user systems and behaviors. Since ZIA inspects the traffic inline and has the ability to quarantine the suspicious file while it's being analyzed, we can capture and analyze the malicious code before it reaches the endpoint and blocks the malicious code from reaching the user.

Note: Zscaler engine detection of these TTPs is limited to malicious traffic that includes payloads passing through ZIA cloud security.

See Appendix C for an example of ZIA engine detection when "credentials from web browsers" or "hooking" techniques are used as part of an attack.

Zscaler-recommended steps to protect against credential access techniques:

- Configure a **Sandbox** policy to detect and block malicious code before delivery to endpoint.
- Employ EDR endpoint security to complement protection against adversarial credential access techniques.

See Appendix A for details on Zscaler security engines and recommended policy configuration.

Discovery

Discovery tactics consist of techniques an adversary may use to gain knowledge about a system and internal network. These techniques help adversaries observe the environment and orient themselves before deciding how to act. They also allow adversaries to explore what they can control and what's around their entry point in order to discover how it could benefit their current objective. Native operating system tools are often used toward this post-compromise information-gathering objective.

Example: Adversaries may attempt to get a list of services running on remote hosts to identify services that are vulnerable to remote software exploitation. Methods to acquire this information include port scans and vulnerability scans using tools that are brought onto a system. **APT32** performed network scanning on the network to search for open ports, services, OS fingerprinting, and other vulnerabilities.

©2021 Zscaler, Inc. All rights reserved.

Discovery		Zscaler Security Engine	Recommended Policy Actions
Account Discovery (4)	Local Account	●	Configure Sandbox
	Domain Account	●	Configure Sandbox
	Email Account	●	Configure Sandbox
	Cloud Account	●	Configure Sandbox
Application Window Discovery		●	Configure Sandbox
Browser Bookmark Discovery		●	Configure Sandbox
Cloud Service Dashboard		○	
Cloud Service Discovery		○	
Domain Trust Discovery		●	Configure Sandbox
File and Directory Discovery		●	Configure Sandbox
Network Service Scanning		●	Configure Sandbox
Network Share Discovery		●	Configure Sandbox
Network Sniffing		●	Configure Sandbox
Password Policy Discovery		●	Configure Sandbox
Peripheral Device Discovery		●	Configure Sandbox
Permission Groups Discovery (3)	Domain Groups	●	Configure Sandbox
	Cloud Groups	●	Configure Sandbox
	Local Groups	●	Configure Sandbox
Process Discovery		●	Configure Sandbox
Query Registry		●	Configure Sandbox
Remote System Discovery		●	Configure Sandbox
Software Discovery (1)	Security Software Discovery	●	Configure Sandbox
System Information Discovery		●	Configure Sandbox
System Network Configuration Discovery		●	Configure Sandbox
System Network Connections Discovery		●	Configure Sandbox
System Owner/User Discovery		●	Configure Sandbox
System Service Discovery		●	Configure Sandbox
System Time Discovery		●	Configure Sandbox
Virtualization/Sandbox Evasion (3)	System Checks	●	Configure Sandbox
	User Activity Based Checks	●	Configure Sandbox
	Time Based Evasion	●	Configure Sandbox

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

Figure 9. Discovery tactics, ZIA security engine detection, and recommended actions.

While techniques such as account discovery, domain trust discovery, network sniffing, process discovery, etc., are typically executed in an end-user system locally and are detected by endpoint security, these techniques can also be detected by ZIA engines when an attacker's suspicious code is run within a virtual machine environment that mimics end-user systems and behavior. Since ZIA inspects the traffic inline and has the ability to quarantine the suspicious file while it's being analyzed, we can capture and analyze the malicious code before it reaches the endpoint and block the malicious code from reaching the user.

Note: Zscaler engine detection of these TTPs is limited to malicious traffic that includes payloads passing through ZIA cloud security.

Zscaler-recommended steps to protect against discovery techniques:

- Configure a **Sandbox** policy to detect and block malicious code before delivery to endpoint.
- Employ EDR endpoint security to complement protection against adversarial discovery techniques.

See Appendix A for details on Zscaler security engines and recommended policy configuration.

©2021 Zscaler, Inc. All rights reserved.

Lateral movement

Lateral movement tactics consist of techniques that adversaries use to enter and control remote systems on a network. Following through on their primary objective often requires exploring the network to find their target and subsequently gaining access to it. Reaching their objective often involves pivoting through multiple systems and accounts. Adversaries might install their own remote access tools to accomplish lateral movement or use legitimate credentials with native network and operating system tools, which may be stealthier.

Example: Adversary may exploit a software vulnerability to execute adversary-controlled code in the remote system. **APT28** exploited a Windows SMB remote code execution vulnerability to conduct lateral movement.

Lateral Movement		Zscaler Security Engine	Recommended Policy Actions			
Exploitation of Remote Services		● ● ● ●	Configure Sandbox	Configure IPS	Configure Firewall	Configure Malware Protection
Internal Spearphishing		● ● ●	Configure Sandbox	Configure IPS	Configure ATP	
Lateral Tool Transfer		●	Configure Firewall			
Remote Service Session Hijacking (2)	SSH Hijacking	○				
	RDP Hijacking	●	Configure Firewall			
Remote Services (6)	Remote Desktop Protocol	●	Configure Firewall			
	SMB/Windows Admin Shares	●	Configure Firewall			
	Distributed Component Object Model	●	Configure Sandbox			
	SSH	○				
	VNC	○				
	Windows Remote Management	○				
Replication Through Removable Media		○				
Software Deployment Tools		○				
Taint Shared Content		●	Configure Sandbox			
Use Alternate Authentication Material (4)	Pass the Hash	○				
	Pass the Ticket	○				
	Application Access Token	○				
	Web Session Cookie	○				

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

Figure 10. Lateral movement tactics, ZIA security engine detection, and recommended actions.

Zscaler-recommended steps to protect against lateral movement techniques:

- Configure and apply a **Firewall Control** policy to control which services are available to specific users. Use firewall filtering to configure policies that define which types of traffic are allowed from which sources and to which destinations.
- Configure a **Sandbox** policy to detect and block zero-day and unknown threats.

See Appendix A for details on Zscaler security engines and recommended policy configuration.

Collection

Collection tactics consist of techniques adversaries may use to gather information and the sources information is collected from that are relevant to following through on the adversary's objectives. Frequently, the goal after collecting data is to steal (exfiltrate) the data. Common target sources include various drive types, browsers, audio, video, and email. Common collection methods include capturing screenshots and keyboard input.

Example: Adversary may store the collected data in a central location or directory prior to exfiltration. Data may be kept in separate files or combined into one file through techniques such as compression or encryption. **APT3** has been known to stage files for exfiltration in a single location.

©2021 Zscaler, Inc. All rights reserved.

Collection		Zscaler Security Engine	Recommended Policy Actions	
Archive Collected Data (3)	Archive via Utility	● ●	Configure Sandbox	Configure Malware Protection
	Archive via Library	●	Configure Malware Protection	
	Archive via Custom Method	○		
Audio Capture		●	Configure Sandbox	
Automated Collection		●	Configure Sandbox	
Clipboard Data		●	Configure Sandbox	
Data from Cloud Storage Object		○		
Data from Information Repositories (2)	Confluence	○		
	Sharepoint	○		
Data from Local System		●	Configure Sandbox	
Data from Network Shared Drive		●	Configure Sandbox	
Data from Removable Media		●	Configure Sandbox	
Data Staged (2)	Local Data Staging	○		
	Remote Data Staging	○		
Email Collection (3)	Local Email Collection	○		
	Remote Email Collection	○		
	Email Forwarding Rule	○		
Input Capture (4)	Keylogging	●	Configure Sandbox	
	GUI Input Capture	●	Configure Sandbox	
	Web Portal Capture	●	Configure IPS	
	Credential API Hooking	●	Configure Sandbox	
Man in the Browser		●	Configure Sandbox	
Man-in-the-Middle (1)	LLMNR/NBT-NS Poisoning and SMB Relay	●	Configure Sandbox	
Screen Capture		●	Configure Sandbox	
Video Capture		●	Configure Sandbox	

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

Figure 11. Collection tactics, ZIA security engine detection, and recommended actions.

While techniques such as audio capture, clipboard data, data from the local system, man-in-the-browser, screen capture, etc., are typically executed in an end-user system locally and are detected by endpoint security, these techniques can also be detected by ZIA engines when an attacker's suspicious code is run within a virtual machine environment that mimics end-user systems and behaviors. Since ZIA inspects the traffic inline and has the ability to quarantine the suspicious file while it's being analyzed, we can capture and analyze the malicious code before it reaches the endpoint and blocks the malicious code from reaching the user.

Note: Zscaler engine detection of these TTPs is limited to malicious traffic that includes payloads passing through ZIA cloud security.

Zscaler-recommended steps to protect against collection techniques:

- Configure a **Sandbox** policy to detect and block malicious code before delivery to endpoint.
- Employ EDR endpoint security to complement protection against adversarial collection techniques.

See Appendix A for details on Zscaler security engines and recommended policy configuration.

Command and control

Command-and-control tactics consist of techniques that adversaries may use to communicate with systems under their control within a victim network. Adversaries commonly attempt to mimic normal, expected traffic to avoid detection. There are many ways an adversary can establish command and control with various degrees of stealth, depending on the victim's network structure and defenses.

Example: Adversaries may communicate over a commonly used port such as TCP:80/443 (HTTP/HTTPS) or TCP/UDP:53 (DNS) to bypass firewalls or network detection systems and to blend with normal network activity to avoid suspicion. **APT33** has used port 443 for command and control.

©2021 Zscaler, Inc. All rights reserved.

Command and Control		Zscaler Security Engine	Recommended Policy Actions		
Application Layer Protocol (4)	Web Protocols		Configure IPS	Configure ATP	
	File Transfer Protocols		Configure IPS	Configure Firewall	
	Mail Protocols		Configure Firewall		
	DNS		Configure Firewall		
Communication Through Removable Media					
Data Encoding (2)	Standard Encoding		Configure Malware Protection		
	Non-Standard Encoding				
Data Obfuscation (3)	Junk Data		Configure IPS		
	Steganography				
	Protocol Impersonation				
Dynamic Resolution (3)	Domain Generation Algorithms		Configure Sandbox	Configure IPS	Configure ATP
	Fast Flux DNS				
	DNS Calculation		Configure Firewall		
Encrypted Channel (2)	Symmetric Cryptography				
	Asymmetric Cryptography		Configure IPS	Configure Firewall	
Fallback Channels			Configure Sandbox	Configure IPS	Configure ATP
Ingress Tool Transfer			Configure Malware Protection	Configure IPS	
Multi-Stage Channels			Configure Sandbox	Configure IPS	Configure ATP
Non-Application Layer Protocol			Configure Firewall		
Non-Standard Port			Configure Firewall		
Protocol Tunneling			Configure Firewall		
Proxy (4)	Internal Proxy		Configure Firewall		
	External Proxy		Configure Firewall		
	Multi-hop Proxy		Configure Sandbox	Configure IPS	Configure ATP
	Domain Fronting		Configure Sandbox	Configure IPS	Configure ATP
Remote Access Software			Configure Firewall		
Traffic Signaling (1)	Port Knocking		Configure Firewall		
Web Service (3)	Dead Drop Resolver		Configure IPS		
	Bidirectional Communication		Configure IPS		
	One-Way Communication		Configure IPS		

Sandbox IPS URL Reputation Adv CFW+IPS Inline-AV DLP CASB Not Supported

Figure 12. Command-and-control tactics, ZIA security engine detection, and recommended actions.

See Appendix C for an example of ZIA engine detection when "uncommonly used port" is used as part of an attack.

Zscaler-recommended steps to protect against command-and-control techniques:

- Configure and apply a **Firewall Control** policy to control which services are available to specific users.
- Set an **IPS Control** policy using signature-based detection to control and protect traffic from intrusion over all ports and protocols.
- Configure an **Advanced Threat Protection** policy to block threats delivered via HTTP, HTTPS, or FTP web traffic.
- Configure a **Sandbox** policy to detect and block zero-day and unknown threats.
- Employ EDR endpoint security to complement protection against adversarial command-and-control techniques.

See Appendix A for details on Zscaler security engines and recommended policy configuration.

©2021 Zscaler, Inc. All rights reserved.

Exfiltration

Exfiltration tactics consist of techniques that adversaries may use to steal data from your network. Once they've collected data, adversaries often package it to avoid detection while removing it. This can include compression and encryption. Techniques for getting data out of a target network typically include transferring it over their command-and-control channel or an alternate channel and may also include putting size limits on the transmission.

Example: Adversaries may encrypt the collected data in order to hide the information that is being exfiltrated from detection or to make the exfiltration less conspicuous upon inspection by a defender. **Emotet** has been observed encrypting the data it collects before sending it to the command-and-control server.

Exfiltration		Zscaler Security Engine	Recommended Policy Actions			
Automated Exfiltration		● ● ● ●	Configure Sandbox	Configure IPS	Configure ATP	Configure DLP
Data Transfer Size Limits		● ●	Configure Firewall	Configure DLP		
Exfiltration Over Alternative Protocol (3)	Exfiltration Over Symmetric Encrypted Non-C2 Protocol	○				
	Exfiltration Over Asymmetric Encrypted Non-C2 Protocol	●	Configure IPS			
	Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol	● ●	Configure IPS	Configure Firewall		
Exfiltration Over C2 Channel		● ● ● ●	Configure Sandbox	Configure IPS	Configure ATP	Configure DLP
Exfiltration Over Other Network Medium (1)	Exfiltration Over Bluetooth	○				
Exfiltration Over Physical Medium (1)	Exfiltration over USB	○				
Exfiltration Over Web Service (2)	Exfiltration to Code Repository	●	Configure IPS			
	Exfiltration to Cloud Storage	●	Configure IPS			
Scheduled Transfer		● ● ●	Configure IPS	Configure ATP	Configure DLP	
Transfer Data to Cloud Account		● ●	Configure IPS	Configure Firewall		

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

Figure 13. Exfiltration tactics, ZIA security engine detection, and recommended actions.

Zscaler-recommended steps to protect against exfiltration techniques:

- Configure and apply a **Firewall Control** policy to control which services are available to specific users.
- Configure a **Data Loss Prevention** policy.
- Set an **IPS Control** policy using signature-based detection to control and protect traffic from intrusion over all ports and protocols.
- Configure an **Advanced Threat Protection** policy to block threats delivered via HTTP, HTTPS, or FTP web traffic.
- Configure a **Sandbox** policy to detect and block zero-day and unknown threats.
- Employ EDR endpoint security to complement protection against adversarial exfiltration techniques.

See Appendix A for details on Zscaler security engines and recommended policy configuration.

Impact

Impact tactics consist of techniques that adversaries use to disrupt availability or compromise integrity by manipulating business and operational processes. Techniques used for impact may include destroying or tampering with data. In some cases, business processes can look fine, but may have been altered to benefit the adversaries' goals. These techniques might be used by adversaries to follow through on their end-goal or to provide cover for a confidentiality breach.

©2021 Zscaler, Inc. All rights reserved.

Example: Adversaries may leverage the resources of co-opted systems in order to solve resource-intensive problems that may impact system and/or hosted service availability. One common purpose for resource hijacking is to validate transactions of cryptocurrency networks and earn virtual currency. **APT41** deployed a Monero cryptocurrency mining tool in a victim's environment.

Impact		Zscaler Security Engine	Recommended Policy Actions
Account Access Removal		●	Configure Sandbox
Data Destruction		●	Configure Sandbox
Data Encrypted for Impact		●	Configure Sandbox
Data Manipulation (3)	Stored Data Manipulation	○	
	Transmitted Data Manipulation	○	
	Runtime Data Manipulation	○	
Defacement (2)	Internal Defacement	○	
	External Defacement	●	Configure IPS
Disk Wipe (2)	Disk Content Wipe	●	Configure Sandbox
	Disk Structure Wipe	●	Configure Sandbox
Endpoint Denial of Service (4)	OS Exhaustion Flood	○	
	Service Exhaustion Flood	○	
	Application Exhaustion Flood	○	
	Application or System Exploitation	○	
Firmware Corruption		○	
Inhibit System Recovery		●	Configure Sandbox
Network Denial of Service (2)	Direct Network Flood	●	Configure Firewall
	Reflection Amplification	○	
Resource Hijacking		●	Configure Sandbox
Service Stop		●	Configure Sandbox
System Shutdown/Reboot		●	Configure Sandbox

● Sandbox ● IPS ● URL Reputation ● Adv CFW+IPS ● Inline-AV ● DLP ● CASB ○ Not Supported

Figure 14. Impact tactics, ZIA security engine detection, and recommended actions.

While techniques such as account access removal, data destruction, disk content wipe, service stop, etc., are typically executed in an end-user system locally and are detected by endpoint security, these techniques can also be detected by ZIA engines when an attacker's suspicious code is run within a virtual machine environment that mimics end-user systems and behaviors. Since ZIA inspects the traffic inline and has the ability to quarantine the suspicious file while it's being analyzed, we can capture and analyze the malicious code before it reaches the endpoint and blocks the malicious code from reaching the user.

Note: Zscaler engine detection of these TTPs is limited to malicious traffic that includes payloads passing through ZIA cloud security.

See Appendix C for an example of ZIA engine detection when "data destruction" is used as part of an attack.

Zscaler-recommended steps to protect against impact techniques:

- Configure a **Sandbox** policy to detect and block malicious code before delivery to endpoint.
- Employ EDR endpoint security to complement protection against adversarial impact techniques.

See Appendix A for details on Zscaler security engines and recommended policy configuration.

©2021 Zscaler, Inc. All rights reserved.

Conclusion

Zscaler's unique architecture secures customers with comprehensive coverage and extensibility to enable a defense-in-depth approach (an approach that can be supplemented even further with additional partner integrations).

The Zscaler Zero Trust Exchange security engines—specifically, Advanced Threat Protection (ATP), ZIA's Browser Control capabilities, Data Loss Prevention (DLP), File Type Control, Cloud Firewall, Intrusion Prevention System (IPS), Malware Protection, Cloud Sandbox, SSL Inspection, and URL Filtering—identify and deter the attack and post-attack tactics (and supporting techniques and sub-techniques) of the generalized kill chain diagrammed by the MITRE ATT&CK framework enterprise matrix.

The MITRE ATT&CK framework allows customers to assess their security posture with Zscaler, determine security coverage, identify gaps, and repel attacks. Those customers also benefit from additional Zscaler and partner security engines to protect against an ever-evolving threat landscape.

Appendix A – Zscaler Security Engines & Recommended Policy

Advanced Threat Protection (ATP)

Hackers routinely embed malicious scripts and applications not only on their own websites but on legitimate websites that they have hacked. Zscaler ATP can identify a variety of these objects and scripts and prevent them from downloading to the end user's browser. Configuring an ATP policy protects your traffic from fraud, unauthorized communication, and other malicious objects and scripts. When you configure the Advanced Threat Protection policy, you can set a suspicious content protection (PageRisk) value. The Zscaler service calculates the PageRisk Index score of a web page in real-time. This score is then evaluated against the value that you set. Zscaler also has a **Recommended Advanced Threat Protection Policy**.

Browser Control

With ZIA, customers can configure a Browser Control policy to warn users from going out to the internet when they are using outdated or vulnerable browsers, plugins, and applications. The service examines browser versions and patches (including beta browsers), internet applications (for example, Adobe Flash, Sun Java, Apple QuickTime), and media download applications (for example, Windows Media Player). You can also reduce the security risk of your organization by blocking the use of browsers or specific browser versions that are older or that have known vulnerabilities. The ZIA Admin Portal displays the last 12 versions for most browsers. Zscaler also has a **Recommended Browser Control Policy**.

Data Loss Prevention (DLP)

Corporate data can be leaked in different ways, such as through webmail, cloud storage, social media, and a variety of other applications. Zscaler DLP can protect your organization from data loss. If your organization has a third-party DLP solution, Zscaler can forward information about transactions that trigger DLP policies to your third-party solution. Zscaler uses secure Internet Content Adaptation Protocol (ICAP) to do this. However, the Zscaler service does not take ICAP responses from your DLP solution. Zscaler only monitors or blocks content according to the policy you configure, then forwards information about transactions so that your organization can take any necessary remediation steps.

©2021 Zscaler, Inc. All rights reserved.

Cloud Firewall

The Zscaler Cloud Firewall service provides integrated cloud-based next-generation firewall capabilities that allow granular control over your organization's outbound TCP, UDP, and ICMP traffic. By default, Cloud Firewall allows all non-HTTP/HTTPS traffic from your network to the internet. With firewall filtering, you can configure policies that define which types of traffic are allowed from specific sources and to specific destinations. Cloud Firewall also includes a dashboard, giving your organization visibility into your networks. Zscaler also has a **Recommended Firewall Control Policy**.

Intrusion Prevention System (IPS)

With IPS, you can use signature-based detection to control and protect your traffic from intrusion over all ports and protocols. The Zscaler service uses custom signatures built and updated by Zscaler's security research team as well as signatures from industry-leading vendors. Using these signatures, the Zscaler service is able to monitor your traffic in real time. As soon as the IPS has examined the contents of your traffic and found a pattern match, it can enforce your policies inline. Zscaler also has a **Recommended IPS Control Policy**.

Malware Protection

Malware Protection service uses an industry-leading AV vendor for signature-based detection and protection so it can provide comprehensive web security. In addition to virus and spyware protection, the service uses malware feeds from its trusted partners, such as Microsoft and Adobe, as well as its own technologies to detect and block malware. The malware policy applies globally to all of an organization's locations. Zscaler also has a **Recommended Malware Protection Policy**.

Cloud Sandbox

Zscaler Cloud Sandbox provides an additional layer of security against zero-day, unknown threats and Advanced Persistent Threats (APTs) through sandbox analysis, an integrated file behavioral analysis. Cloud Sandbox runs and analyzes files in a virtual environment to detect malicious behavior and propagates a hash of malicious files to all ZIA Public Service Edges (formerly Zscaler Enforcement Nodes or ZENs) throughout the cloud, effectively maintaining a real-time blacklist so it can prevent users anywhere in the world from downloading malicious files. Please refer to the following help text article link for recommended policy configuration: **Recommended Sandbox Policy**.

SSL Inspection

As more and more websites use HTTPS, including social media such as Facebook and Twitter, the ability to control and inspect traffic to and from these sites has become an important piece of the security posture of an organization. The Zscaler service can inspect HTTPS traffic from your organization. The service can scan data transactions and apply policies to it. It functions as a full SSL proxy, or SSL man-in-the-middle (MITM) proxy. The Zscaler service provides two options to protect your organization's HTTPS traffic: SSL inspection or, if SSL inspection is not feasible for your organization, you can configure a global block of specific HTTPS content.

When you enable SSL inspection, the Zscaler service establishes a separate SSL tunnel with the user's browser and with the destination server. Please refer to **About SSL Inspection** for more information.

©2021 Zscaler, Inc. All rights reserved.

URL Filtering

Through URL filtering, you can limit your exposure to liability by managing access to web content based on a site's categorization. To allow granular control of filtering, the Zscaler service organizes URLs into a hierarchy of categories. Zscaler also has a **Recommended URL & Cloud App Control Policy**.

Appendix B – Real-World Attacks and ATT&CK Techniques Mapping

While attackers generally use multiple techniques for a successful attack, there are some attacks that have used several techniques to infiltrate, stay active, and wreak havoc in the customer network. Here is an example of a known adversary group (APT33) and a ransomware (WannaCry) attack that have used several tactics/techniques and how they map to ATT&CK. Additional examples can be found in Appendix D.

APT33 (Advanced Persistent Threat)

APT33 is a suspected Iranian threat group that has carried out operations since at least 2013. The group has targeted organizations across multiple industries in the United States, Saudi Arabia, and South Korea, with a particular interest in the aviation and energy sectors.

Associated groups: HIDDEN COBRA, Guardians of Peace, ZINC, NICKEL ACADEMY

APT33 has used many tactics including spear-phishing and malware. Here are the tactics and techniques used by this advanced persistent threat.

Initial access

- Phishing: Spear-phishing link

  Sent spear-phishing emails containing links to .hta files.

- Phishing: Spear-phishing attachment

  Sent spear-phishing emails with archive attachments.

- Valid accounts

  Used valid accounts for initial access and privilege escalation.

- Valid accounts: Cloud accounts

  Used compromised Office 365 accounts in tandem with Ruler in an attempt to gain control of endpoints.

Execution

- Command and scripting interpreter: PowerShell

  Utilized PowerShell to download files from the command-and-control server and run various scripts.

- Command and scripting interpreter: Visual Basic

  Used VBScript to initiate the delivery of payloads.

- Exploitation for client execution

  Attempted to exploit a known vulnerability in WinRAR (CVE-2018-20250) and attempted to gain remote code execution via a security bypass vulnerability (CVE-2017-11774).

©2021 Zscaler, Inc. All rights reserved.

- Scheduled task/job: Scheduled task

  Created a scheduled task to execute a .vbe file multiple times a day.

- User execution: Malicious link

  Lured users to click links to malicious HTML applications delivered via spear-phishing emails.

- User execution: Malicious file

  Used malicious email attachments to lure victims into executing malware.

Persistence

- Boot or logon autostart execution: Registry run keys/startup folder

  Deployed a tool known as DarkComet to the startup folder of a victim and used registry run keys to gain persistence.

- Event-triggered execution: Windows Management Instrumentation event Subscription

  Attempted to use WMI event subscriptions to establish persistence on compromised hosts.

Privilege escalation

- Exploitation for privilege escalation

  Used a publicly available exploit for CVE-2017-0213 to escalate privileges on a local system.

Defense evasion

- Obfuscated files or information

  Used Base64 to encode payloads.

Credential access

- Brute-force: Password spraying

  Used password spraying to gain access to target systems.

- Credentials from password stores

  Used a variety of publicly available tools such as LaZagne to gather credentials.

- Credentials from password stores: Credentials from web browsers

  Used a variety of publicly available tools such as LaZagne to gather credentials.

- Network sniffing

  Used SniffPass to collect credentials by sniffing network traffic.

- OS credential dumping: LSASS memory

  Used a variety of publicly available tools such as LaZagne, Mimikatz, and ProcDump to dump credentials.

- OS credential dumping: LSA secrets, cached domain credentials, credentials In files

  Used a variety of publicly available tools such as LaZagne to gather credentials.

©2021 Zscaler, Inc. All rights reserved.

- Unsecured credentials: Group policy preferences

  Used a variety of publicly available tools such as Gpppassword to gather credentials.

Collection

- Archive collected data: Archive via utility

  Used WinRAR to compress data prior to exfil.

Command and control

- Application layer protocol: Web protocols

  Used HTTP for command and control.

- Data encoding: Standard encoding

  Used Base64 to encode command-and-control traffic.

- Encrypted channel: Symmetric cryptography

  Used AES for encryption of command-and-control traffic.

- Ingress tool transfer

  Downloaded additional files and programs from its command-and-control server.

- Non-standard port

  Used HTTP over TCP ports 808 and 880 for command and control.

Exfiltration

- Exfiltration over alternative protocol: Exfiltration over unencrypted/obfuscated Non-C2 Protocol

  Used FTP to exfiltrate files (separately from the command-and-control channel).

Source: **https://attack.mitre.org/groups/G0064/**

©2021 Zscaler, Inc. All rights reserved.

WannaCry ransomware

WannaCry is ransomware that was first seen in a global attack during May 2017, which affected more than 150 countries. It contains worm-like features to spread itself across a computer network using the SMBv1 exploit EternalBlue.

Other names: WanaCrypt, WanaCrypt0r, WCry

WannaCry has used many tactics and techniques, including the following:

Execution

- Windows Management Instrumentation

  Utilizes wmic to delete shadow copies.

Persistence

- Create or modify system process: Windows service

  Creates the service "mssecsvc2.0" with the display name "Microsoft Security Center (2.0) Service."

Defense evasion

- File and directory permissions modification: Windows file and directory permissions modification

  Uses attrib +h and icacls . /grant Everyone:F /T /C /Q to make some of its files hidden and grant all users full access controls.

- Hide artifacts: Hidden files and directories

  Uses attrib +h to make some of its files hidden.

Discovery

- File and directory discovery

  Searches for a variety of user files by file extension before encrypting them using RSA and AES, including Office, PDF, image, audio, video, source code, archive/compression format, and key and certificate files.

- Peripheral device discovery

  Contains a thread that will attempt to scan for new attached drives every few seconds. If one is identified, it will encrypt the files on the attached device.

- Remote system discovery

  Scans its local network segment for remote systems to try to exploit and copy itself onto.

- System network configuration discovery

  Attempts to determine the local network segment it is a part of.

©2021 Zscaler, Inc. All rights reserved.

Lateral movement

- Exploitation of remote services

  Uses an exploit in SMBv1 to spread itself to other remote systems on a network.

- Lateral tool transfer

  Attempts to copy itself to remote computers after gaining access via an SMB exploit.

- Remote service session hijacking: RDP hijacking

  Enumerates current remote desktop sessions and tries to execute the malware on each session.

Command and control

- Encrypted channel: Asymmetric cryptography

  Uses Tor for command-and-control traffic and routes a custom cryptographic protocol over the Tor circuit.

- Proxy: Multi-hop proxy

  Uses Tor for command-and-control traffic.

Impact

- Data encrypted for impact

  Encrypts user files and demands that a ransom is paid in Bitcoin to decrypt those files.

- Inhibit system recovery

  Uses vssadmin, wbadmin, bcdedit, and wmic to delete and disable operating system recovery features.

- Service stop

  Attempts to kill processes associated with Exchange, Microsoft SQL Server, and MySQL to make it possible to encrypt their data stores.

Source: **https://attack.mitre.org/software/S0366/**

©2021 Zscaler, Inc. All rights reserved.

Appendix C – Zscaler ZIA Security Engine's Real-World Detection of MITRE ATT&CK TTPs

The following examples show details of ZIA engine detection of various ATT&CK techniques being used as part of an attack in the real world.

1: ZIA engine detection of hooking, credential from web browser, process injection, and registry run keys/startup folder techniques being used in an attack.

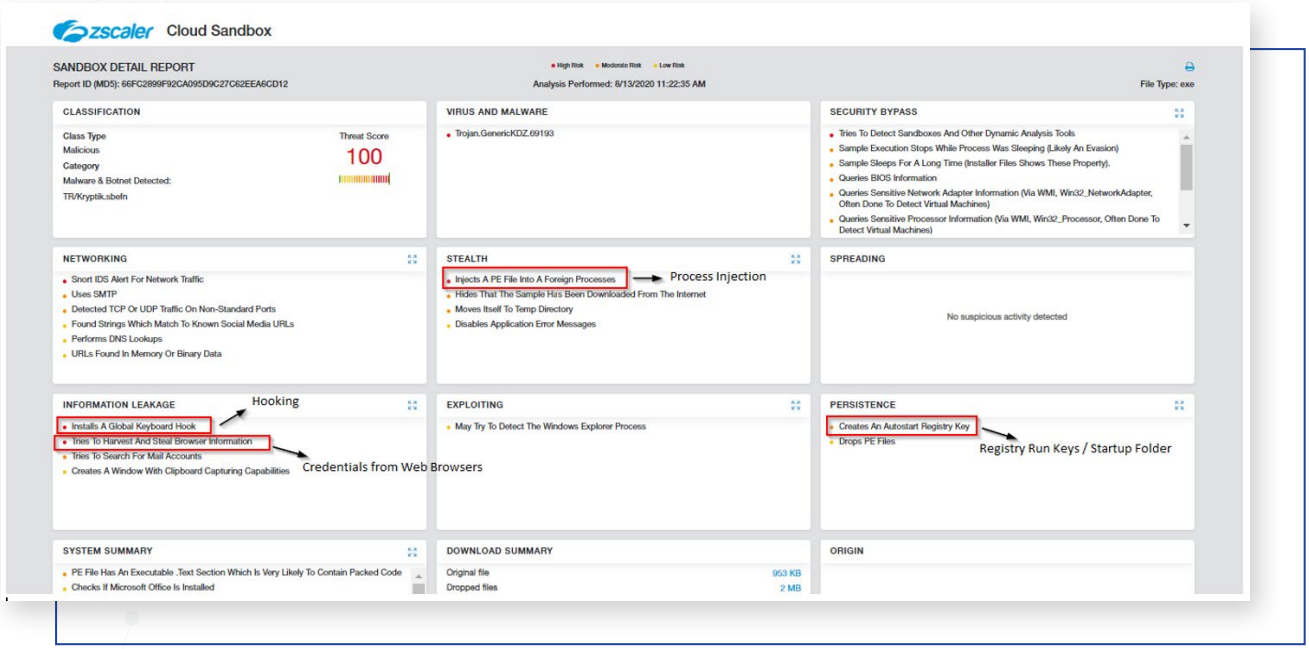

Figure 15. ZIA engine detection of hooking, credential from web browser, process injection, and registry run keys/startup folder techniques being used in an attack.

Hooking

Adversaries may log user keystrokes to intercept credentials. Keyloggers may be used for this purpose to "hook" the keyboard to steal sensitive information such as usernames and passwords as the user types them.

Activity logged in Advanced Cloud Sandbox: Installs a global keyboard hook

In this example, the Advanced Cloud Sandbox had seen the following two keyboard hooks created by the malicious process:

0 keyboard low level C:\5F34D52310080000_5F34D53000000001.exe

0 keyboard low level C:\Users\user\AppData\Roaming\nVTlJn\nVTlJn.exe

©2021 Zscaler, Inc. All rights reserved.

Credentials from web browsers

Adversaries may acquire credentials from web browsers by reading files specific to the target browser. Web browsers commonly save credentials such as website usernames and passwords so that they do not need to be entered manually in the future. Web browsers typically store the credentials in an encrypted format within a credential store; however, methods exist to extract plaintext credentials from web browsers. Malicious programs may look for browser information such as cookies, history, and passwords to steal confidential information.

Activity logged in Advanced Cloud Sandbox: Tries to harvest and steal browser information

In this example, the Advanced Cloud Sandbox had seen the malicious process trying to access the following files for stealing confidential information:

C:\Users\user\AppData\Roaming\Mozilla\Firefox\Profiles\vx19egtj.default\cookies.sqlite

C:\Users\user\AppData\Local\Google\Chrome\User Data\Default\Cookies

C:\Users\user\AppData\Roaming\Mozilla\Firefox\profiles.ini

C:\Users\user\AppData\Local\Google\Chrome\User Data\Default\Login Data

Process injection

Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process.

Activity logged in Advanced Cloud Sandbox: Injects a PE file into a foreign process

In this example, Advanced Cloud Sandbox had seen the malicious process trying to inject a portable executable (PE) file in the following processes:

C:\5F34D52310080000_5F34D53000000001.exe base: 400000 value starts with: 4D5A

C:\Users\user\AppData\Roaming\nVTlJn\nVTlJn.exe base: 400000 value starts with: 4D5A

Registry run keys/Startup folder

Adversaries may achieve persistence by adding a program to a startup folder or referencing it with a registry run key. Adding an entry to the "run keys" in the registry or startup folder will cause the program referenced to be executed when a user logs in. These programs will be executed under the context of the user and will have the account's associated permissions level.

Placing a program within a startup folder will also cause that program to execute when a user logs in. There is a startup folder location for individual user accounts as well as a system-wide startup folder that will be checked regardless of which user account logs in.

While not restricted to malicious usage, autostarting registry key additions will ensure that the user will restart the application whenever the PC is restarted.

©2021 Zscaler, Inc. All rights reserved.

Activity logged in Advanced Cloud Sandbox: Creates an autostart registry key

In this example, Advanced Cloud Sandbox had seen the following registry entries created by the malicious process for autostart:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run nVTlJn

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run nVTlJn

2: ZIA engine detection of data destruction technique being used in an attack.

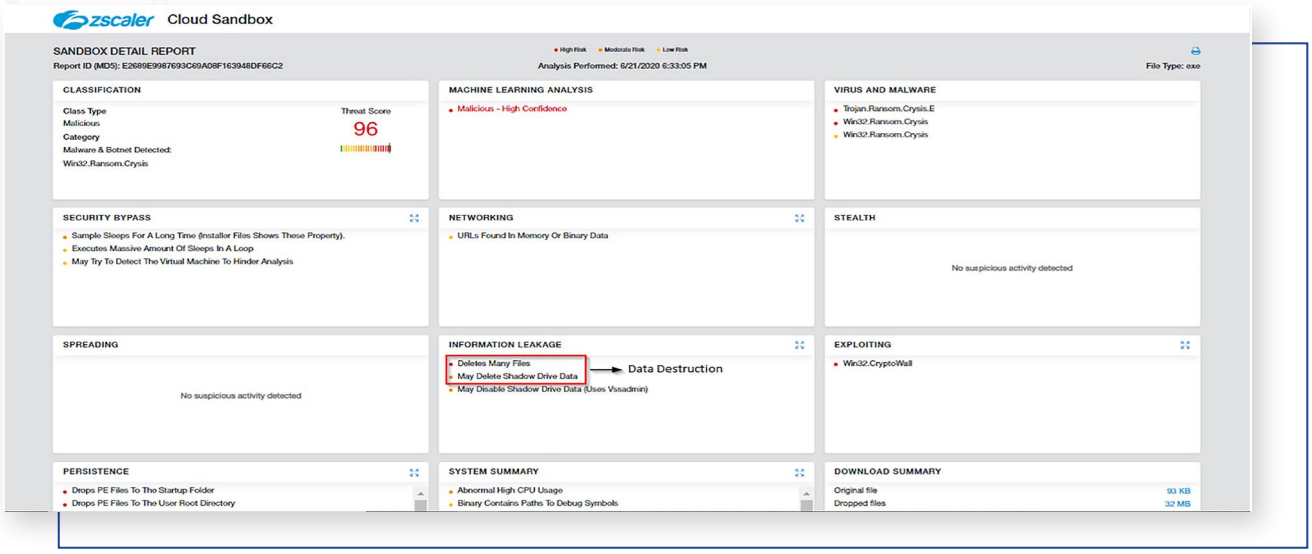

Figure 16. ZIA engine detection of data destruction technique being used in an attack.

Data destruction

Adversaries may destroy data and files on specific systems or in large numbers on a network to interrupt availability to systems, services, and network resources. Data destruction is likely to render stored data irrecoverable by forensic techniques through overwriting files or data on local and remote drives. Adversaries may attempt to overwrite files and directories with randomly generated data to make it irrecoverable.

Activity logged in Advanced Cloud Sandbox: Deletes many files

Activity logged in Advanced Cloud Sandbox: May delete shadow drive data

Checks if the process runs “vssadmin.exe” with “delete” in the command line or if the strings “vssadmin” and “delete” are contained in the binary or the process memory.

In this example, Advanced Cloud Sandbox had seen the malicious process executing the following command to delete the shadow drive data:

C:\Windows\System32\vssadmin.exe vssadmin delete shadows/all/quiet

The malicious sample has attempted to delete numerous files from the user’s system. Number of file deletions of 500 exceeded the threshold of 400.

©2021 Zscaler, Inc. All rights reserved.

3: ZIA engine detection of PowerShell and install root certificate techniques being used in an attack.

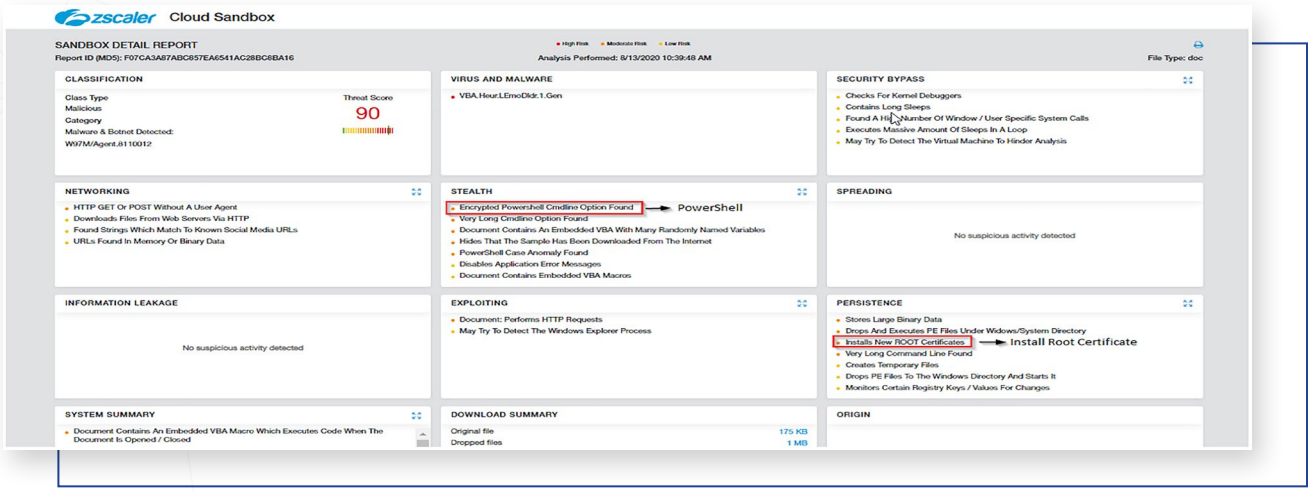

Figure 17. ZIA engine detection of PowerShell and Install root certificate techniques being used in an attack.

PowerShell

Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code.

Activity logged in Advanced Cloud Sandbox: Encrypted PowerShell command-line option found

Malicious samples will attempt to hide their attack within encrypted shellcode. In this example, Advanced Cloud Sandbox has detected the encrypted PowerShell command in the malicious sample:

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe powersheLL -e JABaAEMAUQBEAFgAeABwA-GIAPQAnAFoAWABHAE0ASwBjAHgAbAAnADsAWwBO-AGUAdAAuAFMAZQByAHYAaQBjAGUAUABvAGkAbgB0AE0AYQBuAGEAZwBlAHIAXQA6ADoAIgBTAGUAQwBVAGAAUgBgAGkAVAB5AHAAcgBPAFQAYABPAEMATwBMACI

Install root certificate

Installation of a root certificate on a compromised system would give an adversary a way to degrade the security of that system. Adversaries have used this technique to avoid security warnings that prompt users when compromised systems connect over HTTPS to adversary-controlled web servers that spoof legitimate websites in order to collect login credentials.

Activity logged in Advanced Cloud Sandbox: Installs new root certificates

In this example, Advanced Cloud Sandbox has detected the malicious sample installing the following new root certificates:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SystemCertificates\ROOT\Certificates\3F728A35DE52B2C8994A4FB101A03B95E87B06C8 Blob

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SystemCertificates\ROOT\Certificates\3F728A35DE52B2C8994A4FB101A03B95E87B06C8 Blob

©2021 Zscaler, Inc. All rights reserved.

4: ZIA engine detection of uncommonly used port and data destruction techniques being used in an attack.

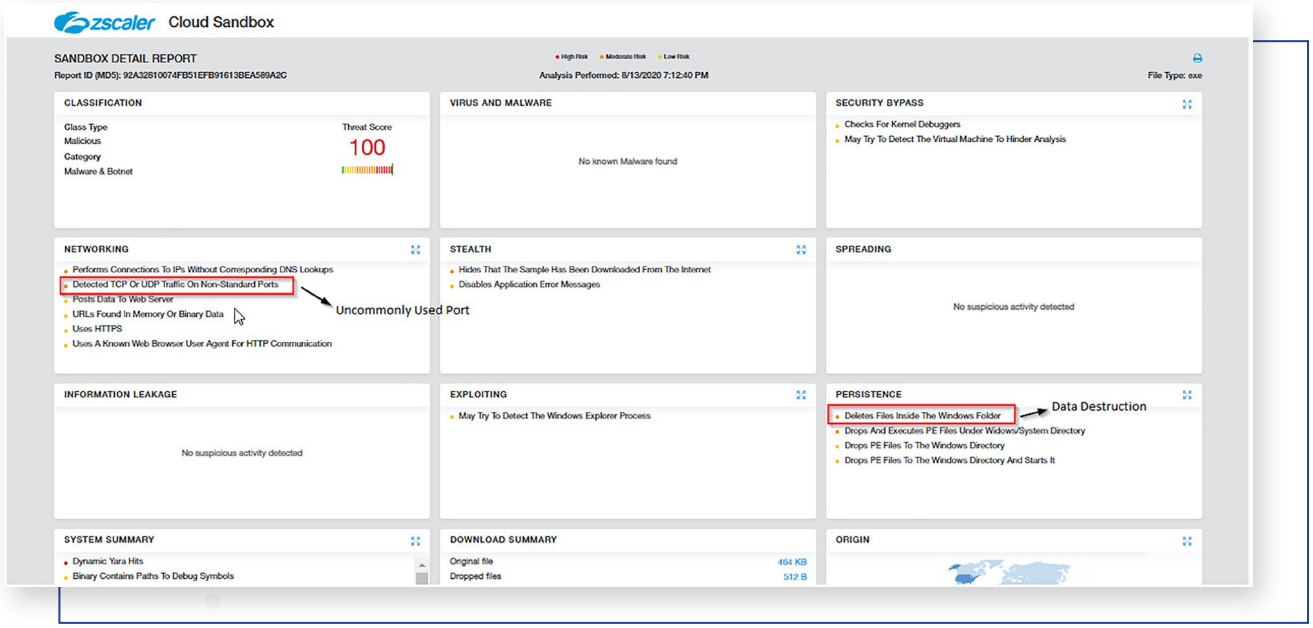

Figure 18. ZIA engine detection of uncommonly used port and data destruction being used in an attack.

Uncommonly used ports

Adversaries may use non-standard ports to evade security tools and exfiltrate information. The ZIA engine checks for TCP and UDP connections that are not on a list of standard protocol ports on a non-private IP address.

Activity logged in Advanced Cloud Sandbox: Detected TCP or UDP traffic on non-standard ports

In this example, Advanced Cloud Sandbox has detected the malicious sample using non-standard ports:

192.168.1.166:49683 -> **159.203.232.29:8080**
192.168.1.166:49685 -> **87.106.231.60:8080**

Data destruction

Adversaries may destroy data and files on specific systems or in large numbers on a network to interrupt availability to systems, services, and network resources. Data destruction is likely to render stored data irrecoverable by forensic techniques through overwriting files or data on local and remote drives. Adversaries may attempt to overwrite files and directories with randomly generated data to make it irrecoverable.

Activity logged in Advanced Cloud Sandbox: Deletes files inside the Windows folder

In this example, the malicious sample has attempted to delete the following Windows files to impede removal of the threat:

C:\Windows\SysWOW64\KBDOLDIT\ole2disp.exe:Zone.Identifier

©2021 Zscaler, Inc. All rights reserved.

Appendix D – Zscaler Security Research Materials Detailing Attack Techniques and MITRE ATT&CK Mapping Examples

The Zscaler ThreatLabZ security research team analyzes the hundreds of trillions of time-series data points collected by the Zscaler Zero Trust Exchange platform each and every day. That uniquely positions ThreatLabZ to assess the latest cyberattack methods. Below, several recent summaries of attacks are documented by the ThreatLabZ team, with each attack mapped to the MITRE ATT&CK framework.

LinkedIn Job Seeker Phishing Campaign Spreads Agent Tesla

https://www.zscaler.com/blogs/research/linkedin-job-seeker-phishing-campaign-spreads-agent-tesla

In August 2020, Zscaler ThreatLabZ researchers observed network activity to a malicious site that used LinkedIn, a popular professional networking and job search site, as the lure for a social engineering scheme designed to steal a user's credentials and spread malicious binaries. The bad actors also used a legitimate site hosting company, Yola, to host the malicious content in an attempt to further look legitimate. The .NET-based binaries hosted on this site are related to the Agent Tesla malware and another previously unseen in-the-wild malware family. Its major functionality is information-stealing and exfiltrating data through SMTP.

Attackers used many tactics to carry out this attack such as phishing, user execution, boot or logon autostart execution, indirect command execution, obfuscated files or information, steal web session cookie, exfiltration over command-and-control channel, and others.

This blog provides a detailed description of the tools, techniques, and procedures of this threat actor and the malicious binaries hosted on this site, as well as the credential-phishing methods used and how ZIA engines detected and blocked the threat.

PurpleWave—A New Infostealer from Russia

https://www.zscaler.com/blogs/research/purplewave-new-infostealer-russia

Infostealer is one of the most profitable tools for cybercriminals, as information gathered from systems infected with this malware could be sold in the cybercrime underground or used for credential stuffing attacks. The Zscaler ThreatLabZ team came across a new Infostealer called PurpleWave, which is written in C++ and silently installs itself onto a user's system. It connects to a command-and-control server to send system information and installs new malware onto the infected system.

The capabilities of the PurpleWave stealer include:

- Stealing passwords, cookies, cards, autofill data, and browser history from Chromium and Mozilla
- Collecting files from the specified path
- Capturing the screen
- Stealing system information
- Stealing Telegram session files
- Stealing Steam application data
- Stealing Electrum wallet data
- Loading and executing additional module/malware

©2021 Zscaler, Inc. All rights reserved.

Attackers used many tactics to carry out this attack such as file and directory discovery, automated exfiltration, exfiltration over command-and-control channel, credentials from web browsers, screen capture, and others, and built highly customized malware to sell it in the underground "as-a-service."

This blog provides a detailed description of the tools, techniques, and procedures used by this attacker to carry out the attack and ZIA engines detected and blocked this threat.

Malware Leveraging XML-RPC Vulnerability to Exploit WordPress Sites

https://www.zscaler.com/blogs/security-research/malware-leveraging-xml-rpc-vulnerability-exploit-wordpress-sites

One of the most common attack vectors employed by bad actors is to launch an XML-RPC attack. XML-RPC on WordPress, which is enabled by default, is actually an API that provides third-party applications and services with the ability to interact with WordPress sites, rather than through a browser. Attackers use this channel to establish a remote connection to a WordPress site and make modifications without being directly logged in to a WordPress customer's system. However, if a WordPress site didn't disable XML-RPC, there is no limit to the number of login attempts that can be made by an attacker, meaning it is just a matter of time before a cybercriminal can gain access.

Recently, ThreatLabZ researchers came across a scheme to attack WordPress sites where a malicious program gets a list of WordPress sites from a command-and-control server which are then attacked leveraging the XML-RPC pingback method to fingerprint the existing vulnerabilities on the listed WordPress sites.

Attackers used many tactics to carry out this attack such as credential access, brute-force, modify authentication process, sandbox evasion, process injection, and others.

This blog provides a detailed description of the tools, techniques, and procedures used by this attacker to carry out the attack and how ZIA engines detected and blocked the threat.

About Zscaler
Zscaler (NASDAQ: ZS) accelerates digital transformation so that customers can be more agile, efficient, resilient, and secure. The Zscaler Zero Trust Exchange protects thousands of customers from cyberattacks and data loss by securely connecting users, devices, and applications in any location. Distributed across more than 150 data centers globally, the SASE-based Zero Trust Exchange is the world's largest inline cloud security platform.. Learn more at **zscaler.com** or follow us on Twitter **@zscaler**.

Zscaler, Inc.
120 Holger Way
San Jose, CA 95134
+1 408.533.0288
www.zscaler.com

©**2021 Zscaler, Inc. All rights reserved.** Zscaler™, Zero Trust Exchange™, Zscaler Internet Access™, ZIA™, Zscaler Private Access™, and ZPA™ are either (i) registered trademarks or service marks or (ii) trademarks or service marks of Zscaler, Inc. in the United States and/or other countries. Any other trademarks are the properties of their respective owners. V.071321