Incident Response - Sofacy

Key evolutions

Across all four iterations, the Sofacy group predominantly used similar tactics and techniques in their attacks, namely:

- Their reliance on **Spear Phishing Attachment (T1193)** as the initial attack vector.
- **User Execution (T1023)** to run malicious code.
- **Process Discovery (T1057)**, **Screen Capture (T1113)** and **System Information Discovery(T1082)** to gather system-specific information.
- **Remote File Copy (T1105)** and **Standard Application Layer Protocol (T1071)** to interact with and copy files from the C2 server.

However, some key evolutions involve their efforts to avoid being detected. By the last iteration, the Sofacy group had heavily employed **Template Injection (T1221)** and **Multi-Stage Channels (T1104)** to evade static detection as no typical indicators are present until after the malicious payload is fetched and to obfuscate the C2 channel. Likewise, instead of extending the functionality of their **Zebrocy (S0251)** and **Cannon (S0351)** tools, the group opted to focus on delivering the Trojan in variant programming languages in an effort to make detection more difficult. Furthermore, their experimentation with **Standard Application Layer Protocol (T1071)** is also quite interesting with their usage of an email-based C2 communication channel which would be a difficult C2 channel to detect and act against due to encryption and legitimacy of email services.

Datasets/feeds for detection

Detection of	Data Sources
Initial Access	Det…ail gateway, File monitoring, Mail server, Network intr…m, Packet capture
Execution	Ant…on logs, DLL monitoring, Email gateway, File mor…Ls, Netflow/Enclave netflow, Network intrusion dete…Shell logs, Process command-line parameters, Process monitoring, Web logs, Windows event logs.
Defense Evasion	Binary file metadata, File monitoring, Process command-line parameters, Process monitoring.
Discovery	API monitoring, AWS CloudTrail logs, Azure activity logs, PowerShell logs, Process command-line parameters, Process monitoring, Stackdriver logs.
Collection	API monitoring, Data loss prevention, File monitoring, Process command-line parameters, Process monitoring.
Command and Control	DNS records, File monitoring, Host network interface, Netflow/Enclave netflow, Network device logs, Network protocol analysis, Packet capture, Process command-line parameters, Process monitoring, Process use of network, SSL/TLS inspection.
Exfiltration	Netflow/Enclave netflow, Packet capture, Process monitoring, Process use of network

Static correlation vs User/Entity Behaviour Analysis

The challenge with static correlation lies in the fact that due to the sheer amount of logs generated, SOCs are inundated with noise and false alerts which consequently make detection of adversaries difficult. Using both static correlation and U/EBA aims to overcome these limitations and reduce false positives, helping to eliminate alert fatigue and allowing focus on credible, high-alert risks.

Examples of static correlation rules:

Trigger an alert if:

- A malicious [illegible]e an attachment is opened. (such as a Microsoft Word docum[illegible]he internet or spawning Powershell.exe)
- PowerShell/c[illegible]n PowerShell is executed.
- Command-li[illegible]ions that could be taken to gather system and network information are executed. (e.g. tasklist, systeminfo, wmic)
- Changes are made to the registry that do not correlate to known software, patch cycles, etc.
- An unusual process performs sequential file opens and copy actions to another location on the file system for many files at once.
- Data flow is uncommon. (e.g. significantly more data sent than data received)

For U/EBA, we would be interested in defining behaviours akin to those that trigger the alerts listed above (e.g. Trigger an alert if a user executes PowerShell/change in policy to run PowerShell). Furthermore, although privileged users are the key population of interest within the environment, non-privileged users should not be discounted, as standard accounts are often escalated in privileges.

In regards to responding to these rules, a fair share of these rules are tailored to the Sofacy attacks and can be seen as somewhat effective to respond to in the sense that they mostly trigger on suspect circumstances specific to the nature of the Sofacy attacks. However, some of these rules (e.g. PowerShell and Windows Management Instrumentation) lose their effectiveness if these activities are already commonly used in an environment. In these

circumstances, these rules should be seen as something that influences a larger data model and to increase confidence of malicious activity, data and events should not be viewed in isolation but as part our that could lead to other activities.

As for the interactio elation and U/EBA, in our case, we could administer static correlation rul niques employed by the Sofacy group and leverage U/EBA to detect unknown behaviour unable to be detected by static correlation such as new advances in the Sofacy campaign. By combining both of these approaches into a hybrid analytic we are provided insight into patterns of behaviour and an additional context around known and unknown threats, in conjunction with a more accurate identification of threats.

Mitigation

In order of priority,

1. Prevent initial access into your network. The Sofacy group has a heavy reliance on **Spear Phishing Attachment (T1193)** as the initial attack vector. In the best case, initial access would be thwarted and there would be no subsequent activities.
2. Prevent the execution of malicious code. Likewise, the Sofacy group has a heavy reliance on **User Execution (T1023)** and the **Command and Scripting Interpreter (T1059)** to run malicious code, in which both play a key part in allowing the attack to progress. Preventing execution can thwart further activities.
3. Mitigate the impact of **Zebrocy (S0251).** The Trojan is a routine key tool in the Sofacy attack for the collection and exfiltration of data. As it's unfeasible to prevent its execution due to its dynamic nature, the next best course of action would be to downplay its impact either by preventing its capabilities or allowing early detection.

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command And Control	Exfiltration	Impact
Drive-by Compromise	AppleScript	.bash_profile and .bash	Access Token Manipula	Access Token Manipula	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Ap	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Dis	Application Deploymen	Automated Collection	Communication Throug	Data Compressed	Data Destruction
External Remote Servic	Command-Line Interfac	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Disc	Component Object Mo	Clipboard Data	Connection Proxy	Data Encrypted	Data Encrypted for Impact
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Cc	Credential Dumping	Domain Trust Discover	Exploitation of Remote	Data from Information	Custom Command and	Data Transfer Size Limit	Defacement
Replication Through Re	Component Object Mo	AppInit DLLs	Application Shimming	Clear Command History	Credentials from Web B	File and Directory Disco	Internal Spearphishing	Data from Local Syst	Custom Cryptographic	Exfiltration Over Alterna	Disk Content Wipe
Spearphishing Attachm	Control Panel Items	Application Shimming	Bypass User Account Cc	CMSTP	Credentials in Files	Network Service Scanni	Logon Scripts	Data from Network Sha	Data Encoding	Exfiltration Over Comm	Disk Structure Wipe
Spearphishing Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijack	Code Signing	Credentials in Registry	Network Share Discover	Pass the Hash	Data from Removable M	Data Obfuscation	Exfiltration Over Other	Endpoint Denial of Service
Spearphishing via Servic	Execution through API	BITS Jobs	Dylib Hijacking	Compile After Delivery	Exploitation for Creden	Network Sniffing	Pass the Ticket	Data Staged	Domain Fronting	Exfiltration Over Physic	Firmware Corruption
Supply Chain Compron	Execution through Mod	Bootkit	Elevated Execution with	Compiled HTML File	Forced Authentication	Password Policy Discov	Remote Desktop Protoc	Email Collection	Domain Generation Alg	Scheduled Transfer	Inhibit System Recovery
Trusted Relationship	Exploitation for Client E	Browser Extensions	Emond	Component Firmware	Hooking	Peripheral Device Disco	Remote File Copy	Input Capture	Fallback Channels		Network Denial of Service
Valid Accounts	Graphical User Interfac	Change Default File Ass	Exploitation for Privileg	Compon		Groups Disc	Remote Services	Man in the Browser	Multi-hop Proxy		Resource Hijacking
	InstallUtil	Component Firmware	Extra Window Memory	Connecti		covery	Replication Through Re	Screen Capture	Multi-Stage Channels		Runtime Data Manipulation
	Launchctl	Component Object Mo	File System Permissions	Control F		stry	Shared Webroot	Video Capture	Multiband Communication		Service Stop
	Local Job Scheduling	Create Account	Hooking	DCShado		tem Discove	SSH Hijacking		Multilayer Encryption		Stored Data Manipulation
	LSASS Driver	DLL Search Order Hijack	Image File Execution Op	Deobfusc		ftware Disco	Taint Shared Content		Port Knocking		System Shutdown/Reboot
	Mshta	Dylib Hijacking	Launch Daemon	Disabling		iscovery	Third-party Software		Remote Access Tools		Transmitted Data Manipulation
	PowerShell	Emond	New Service	DLL Searc		rmation Dis	Windows Admin Shares		Remote File Copy		
	Regsvcs/Regasm	External Remote Servic	Parent PID Spoofing	DLL Side-		work Config	Windows Remote Management		Standard Application Layer Protocol		
	Regsvr32	File System Permissions	Path Interception	Executio		work Connections Discovery			Standard Cryptographic Protocol		
	Rundll32	Hidden Files and Direct	Plist Modification	Exploitat		ner/User Discovery			Standard Non-Application Layer Protocol		
	Scheduled Task	Hooking	Port Monitors	Extra Wi		vice Discovery			Uncommonly Used Port		
	Scripting	Hypervisor	PowerShell Profile	File and		e Discovery			Web Service		
	Service Execution	Image File Execution Op	Process Injection	File Dele		on/Sandbox Evasion					
	Signed Binary Proxy Exe	Kernel Modules and Ext	Scheduled Task	File Syste							
	Signed Script Proxy Exec	Launch Agent	Service Registry Permis	Gatekeeper Bypass							
	Source	Launch Daemon	Setuid and Setgid	Group Policy Modification							
	Space after Filename	Launchctl	SID-History Injection	Hidden Files and Directories							
	Third-party Software	LC_LOAD_DYLIB Additic	Startup Items	Hidden Users							
	Trap	Local Job Scheduling	Sudo	Hidden Window							
	Trusted Developer Utilit	Login Item	Sudo Caching	HISTCONTROL							
	User Execution	Logon Scripts	Valid Accounts	Image File Execution Options Injection							
	Windows Management	LSASS Driver	Web Shell	Indicator Blocking							
	Windows Remote Mana	Modify Existing Service		Indicator Removal from Tools							
	XSL Script Processing	Netsh Helper DLL		Indicator Removal on Host							
		New Service		Indirect Command Execution							
		Office Application Startup		Install Root Certificate							
		Path Interception		InstallUtil							
		Plist Modification		Launchctl							
		Port Knocking		LC_MAIN Hijacking							
		Port Monitors		Masquerading							
		PowerShell Profile		Modify Registry							
		Rc.common		Mshta							
		Re-opened Applications		Network Share Connection Removal							
		Redundant Access		NTFS File Attributes							
		Registry Run Keys / Startup Folder		Obfuscated Files or Information							
		Scheduled Task		Parent PID Spoofing							
		Screensaver		Plist Modification							
		Security Support Provider		Port Knocking							
		Server Software Component		Process Doppelgänging							
		Service Registry Permissions Weakness		Process Hollowing							
		Setuid and Setgid		Process Injection							
		Shortcut Modification		Redundant Access							
		SIP and Trust Provider Hijacking		Regsvcs/Regasm							
		Startup Items		Regsvr32							
		System Firmware		Rootkit							
		Systemd Service		Rundll32							
		Time Providers		Scripting							
		Trap		Signed Binary Proxy Execution							
		Valid Accounts		Signed Script Proxy Execution							
		Web Shell		SIP and Trust Provider Hijacking							
		Windows Management Instrumentation Event S		Software Packing							
		Winlogon Helper DLL		Space after Filename							
				Template Injection							
				Timestomp							
				Trusted Developer Utilities							
				Valid Accounts							
				Virtualization/Sandbox Evasion							
				Web Service							
				XSL Script Processing							