Nov 26 2023

NETMANAGEIT

Intelligence Report
Attack Signals Possible Return of Genesis Market, Abuses Node.js, and EV Code Signing

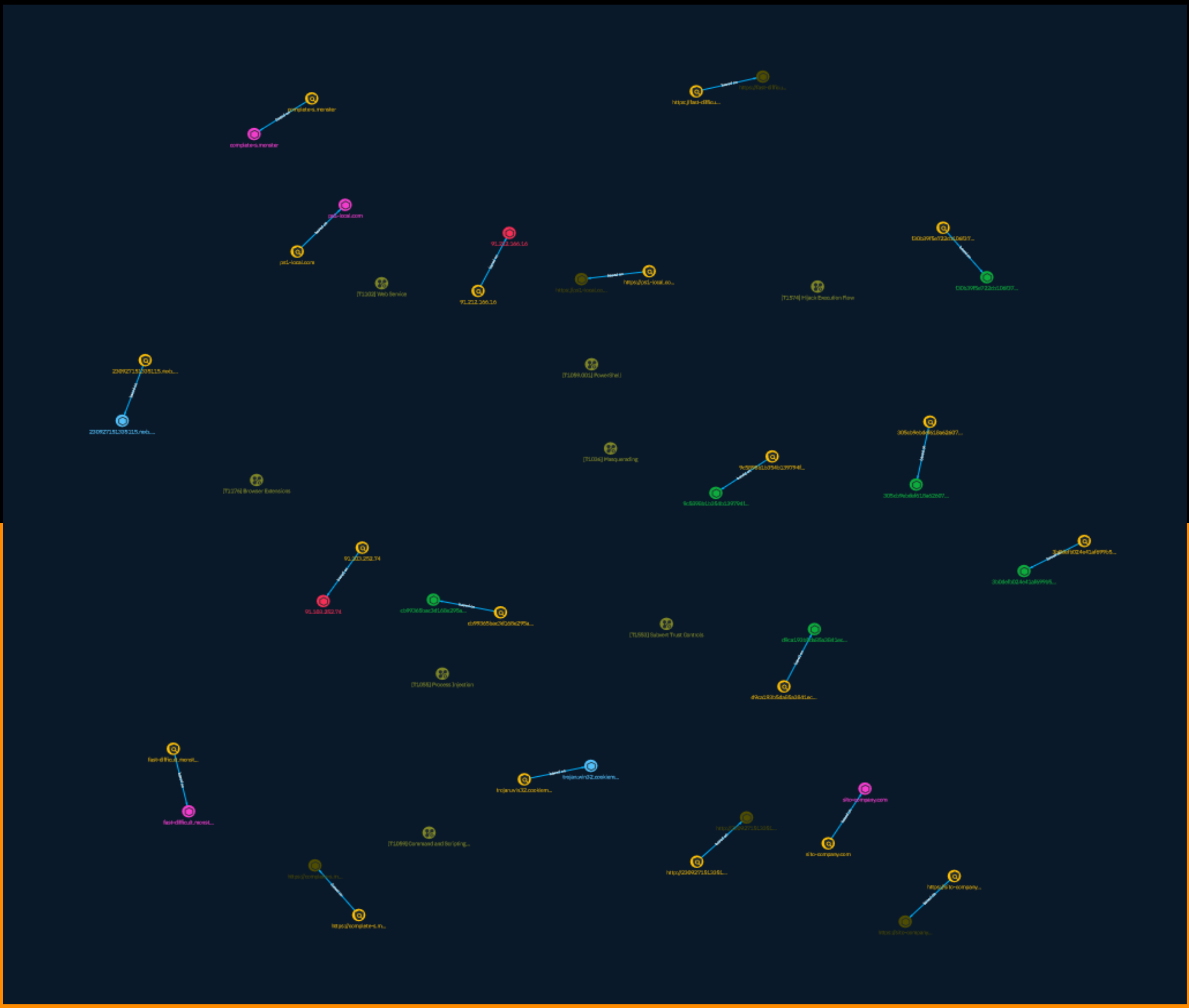

Table of contents

External References

Overview

Description

The Trend Micro Managed XDR team encountered malicious operations that used techniques similar to the ones used by Genesis Market. The threat actor behind these operations abused Node.js to act as a platform for the backdoor, Extended Validation (EV) Code Signing for defense evasion, and possibly Google Colab to host search engine-optimized download sites.

Confidence

This value represents the confidence in the correctness of the data contained within this report.

15 / 100

Content

N/A

Attack-Pattern

Name
Masquerading
ID
T1036
Description
Adversaries may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools. Masquerading occurs when the name or location of an object, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. This may include manipulating file metadata, tricking users into misidentifying the file type, and giving legitimate task or service names. Renaming abusable system utilities to evade security monitoring is also a form of [Masquerading](https://attack.mitre.org/techniques/T1036).(Citation: LOLBAS Main Site) Masquerading may also include the use of [Proxy](https://attack.mitre.org/techniques/T1090) or VPNs to disguise IP addresses, which can allow adversaries to blend in with normal network traffic and bypass conditional access policies or anti-abuse protections.
Name
Process Injection
ID
T1055

Description

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. There are many different ways to inject code into a process, many of which abuse legitimate functionalities. These implementations exist for every major OS but are typically platform specific. More sophisticated samples may perform multiple process injections to segment modules and further evade detection, utilizing named pipes or other inter-process communication (IPC) mechanisms as a communication channel.

Name

PowerShell

ID

T1059.001

Description

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.(Citation: TechNet PowerShell) Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the `Start-Process` cmdlet which can be used to run an executable and the `Invoke-Command` cmdlet which runs a command locally or on a remote computer (though administrator permissions are required to use PowerShell to connect to remote systems). PowerShell may also be used to download and run executables from the Internet, which can be executed from disk or in memory without touching disk. A number of PowerShell-based offensive testing tools are available, including [Empire](https://attack.mitre.org/software/S0363), [PowerSploit](https://attack.mitre.org/software/S0194), [PoshC2](https://attack.mitre.org/software/S0378), and PSAttack.(Citation: Github PSAttack) PowerShell commands/scripts can also be executed without directly invoking the `powershell.exe` binary through interfaces to PowerShell's underlying `System.Management.Automation` assembly DLL exposed through the .NET framework and

Windows Common Language Interface (CLI).(Citation: Sixdub PowerPick Jan 2016)(Citation: SilentBreak Offensive PS Dec 2015)(Citation: Microsoft PSfromCsharp APR 2014)

Name

Subvert Trust Controls

ID

T1553

Description

Adversaries may undermine security controls that will either warn users of untrusted activity or prevent execution of untrusted programs. Operating systems and security products may contain mechanisms to identify programs or websites as possessing some level of trust. Examples of such features would include a program being allowed to run because it is signed by a valid code signing certificate, a program prompting the user with a warning because it has an attribute set from being downloaded from the Internet, or getting an indication that you are about to connect to an untrusted site. Adversaries may attempt to subvert these trust mechanisms. The method adversaries use will depend on the specific mechanism they seek to subvert. Adversaries may conduct [File and Directory Permissions Modification](https://attack.mitre.org/techniques/T1222) or [Modify Registry](https://attack.mitre.org/techniques/T1112) in support of subverting these controls. (Citation: SpectorOps Subverting Trust Sept 2017) Adversaries may also create or steal code signing certificates to acquire trust on target systems.(Citation: Securelist Digital Certificates)(Citation: Symantec Digital Certificates)

Name

Browser Extensions

ID

T1176

Description

Adversaries may abuse Internet browser extensions to establish persistent access to victim systems. Browser extensions or plugins are small programs that can add functionality and customize aspects of Internet browsers. They can be installed directly or through a browser's app store and generally have access and permissions to everything that the browser can access.(Citation: Wikipedia Browser Extension)(Citation: Chrome Extensions Definition) Malicious extensions can be installed into a browser through malicious app store downloads masquerading as legitimate extensions, through social engineering, or by an adversary that has already compromised a system. Security can be limited on browser app stores so it may not be difficult for malicious extensions to defeat automated scanners.(Citation: Malicious Chrome Extension Numbers) Depending on the browser, adversaries may also manipulate an extension's update url to install updates from an adversary controlled server or manipulate the mobile configuration file to silently install additional extensions. Previous to macOS 11, adversaries could silently install browser extensions via the command line using the `profiles` tool to install malicious `.mobileconfig` files. In macOS 11+, the use of the `profiles` tool can no longer install configuration profiles, however `.mobileconfig` files can be planted and installed with user interaction.(Citation: xorrior chrome extensions macOS) Once the extension is installed, it can browse to websites in the background, steal all information that a user enters into a browser (including credentials), and be used as an installer for a RAT for persistence. (Citation: Chrome Extension Crypto Miner)(Citation: ICEBRG Chrome Extensions)(Citation: Banker Google Chrome Extension Steals Creds)(Citation: Catch All Chrome Extension) There have also been instances of botnets using a persistent backdoor through malicious Chrome extensions.(Citation: Stantinko Botnet) There have also been similar examples of extensions being used for command & control.(Citation: Chrome Extension C2 Malware)

Name

Hijack Execution Flow

ID

T1574

Description

Adversaries may execute their own malicious payloads by hijacking the way operating systems run programs. Hijacking execution flow can be for the purposes of persistence, since this hijacked execution may reoccur over time. Adversaries may also use these mechanisms to elevate privileges or evade defenses, such as application control or other restrictions on execution. There are many ways an adversary may hijack the flow of execution, including by manipulating how the operating system locates programs to be executed. How the operating system locates libraries to be used by a program can also be intercepted. Locations where the operating system looks for programs/resources, such as file directories and in the case of Windows the Registry, could also be poisoned to include malicious payloads.

Name
Command and Scripting Interpreter

ID
T1059

Description
Adversaries may abuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages provide ways of interacting with computer systems and are a common feature across many different platforms. Most systems come with some built-in command-line interface and scripting capabilities, for example, macOS and Linux distributions include some flavor of [Unix Shell](https://attack.mitre.org/techniques/T1059/004) while Windows installations include the [Windows Command Shell](https://attack.mitre.org/techniques/T1059/003) and [PowerShell](https://attack.mitre.org/techniques/T1059/001). There are also cross-platform interpreters such as [Python](https://attack.mitre.org/techniques/T1059/006), as well as those commonly associated with client applications such as [JavaScript](https://attack.mitre.org/techniques/T1059/007) and [Visual Basic](https://attack.mitre.org/techniques/T1059/005). Adversaries may abuse these technologies in various ways as a means of executing arbitrary commands. Commands and scripts can be embedded in [Initial Access](https://attack.mitre.org/tactics/TA0001) payloads delivered to victims as lure documents or as secondary payloads downloaded from an existing C2. Adversaries may also execute commands through interactive terminals/shells, as well as utilize various [Remote Services](https://attack.mitre.org/techniques/T1021) in order to achieve remote Execution.(Citation: Powershell Remote Commands)(Citation: Cisco IOS Software Integrity Assurance - Command History)(Citation: Remote Shell Execution in Python)

Name
Web Service

ID
T1102

Description
Adversaries may use an existing, legitimate external Web service as a means for relaying data to/from a compromised system. Popular websites and social media acting as a mechanism for C2 may give a significant amount of cover due to the likelihood that hosts within a network are already communicating with them prior to a compromise. Using common services, such as those offered by Google or Twitter, makes it easier for adversaries to hide in expected noise. Web service providers commonly use SSL/TLS encryption, giving adversaries an added level of protection. Use of Web services may also protect back-end C2 infrastructure from discovery through malware binary analysis while also enabling operational resiliency (since this infrastructure may be dynamically changed).

Indicator

Name
http://230927151335115.mxb.ewk48.shop/f/fvgs30927001.msi
Pattern Type
stix
Pattern
[url:value = 'http://230927151335115.mxb.ewk48.shop/f/fvgs30927001.msi']
Name
91.103.252.74
Description
ISP: Shelter LLC **OS:** None ------------------------- Hostnames: ------------------------- Domains: ------------------------- Services: **22:** ``` SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.9 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAABgQC9BapdAuHeUmo8f7nlbPuWEN8lEDbhXYuHewVqvgW2 2cUf h9nhDv27NSkwK5LF2cE+A5aykH/ hzAiqCYNOyuQB5iBnM5nVCTgJf6atCd5CN0T79Z6Ae1tIQcjk ibwY+Pcgot3WnccoUdcCNfecZKKd3lTyIao9YN3n1IlRk6mz1Prjg9CKDTBLQVqF5EyGS1ia7DHM 57VVynZqN9fQyT70g+rTt4yhgiRzKAhBlfCwk0iDNsp7gEEB5RP/PoqjoTUcvNbw3GUDOZKR76VC Q4pRm1DmbULSUTN2Q2rLL38g2jtt3vYsgMbvGrBtVlyvP6QakpdV7SZXzzax1ADJ4GEOtkVB2vic C9KB8d6sCgCfH5hLOMev1IngPp5SagHot5bRf76VGcyZucTl+UyLcwkNza3tRYdfAf25jbzdivoB Xcr2E87y5HNHELowd0dz8BYz5cmkhj4tJUcw7BndZzh96EjmpFQyKOA99zXuNjvqDH5SQkNrfdTH

RQDsQWT+lK0= Fingerprint: 90:cb:45:97:93:c6:6e:0f:18:d8:d9:b8:63:6b:fa:55 Kex Algorithms: curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group14-sha256 Server Host Key Algorithms: rsa-sha2-512 rsa-sha2-256 ssh-rsa ecdsa-sha2-nistp256 ssh-ed25519 Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr aes128-gcm@openssh.com aes256-gcm@openssh.com MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ``` ------------------

Pattern Type
stix
Pattern
[ipv4-addr:value = '91.103.252.74']
Name
https://ps1-local.com/obfs3ip2.bs64
Pattern Type
stix
Pattern
[url:value = 'https://ps1-local.com/obfs3ip2.bs64']
Name
ps1-local.com
Pattern Type

stix
Pattern
[domain-name:value = 'ps1-local.com']
Name
305cb9ebdef618a626075f71fce3f4a64091e7a875a5ddff983aaeeea0f1fd41
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = '305cb9ebdef618a626075f71fce3f4a64091e7a875a5ddff983aaeeea0f1fd41']
Name
https://complete-s.monster/upd.php
Pattern Type
stix
Pattern
[url:value = 'https://complete-s.monster/upd.php']
Name
230927151335115.mxb.ewk48.shop

Pattern Type

stix

Pattern

[hostname:value = '230927151335115.mxb.ewk48.shop']

Name

3b0defb024e41af699b5dfc424a9ff276409f447edd24af024b34941f5ab62a9

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '3b0defb024e41af699b5dfc424a9ff276409f447edd24af024b34941f5ab62a9']

Name

fast-difficult.monster

Pattern Type

stix

Pattern

[domain-name:value = 'fast-difficult.monster']

Name

https://fast-difficult.monster/api7.php?name=microsoft_barcode_control_16.0_download

Pattern Type

stix

Pattern

[url:value = 'https://fast-difficult.monster/api7.php?name=microsoft_barcode_control_16.0_download']

Name

cb99365bac3d168e295aa0764a1c67e1a7e582731880ad0522e9b6b3616275df

Description

InnoSetupInstaller SHA256 of 3364dd410527f6fc2c2615aa906454116462bf96

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'cb99365bac3d168e295aa0764a1c67e1a7e582731880ad0522e9b6b3616275df']

Name

trojan.win32.cookiemonster.jcb

Pattern Type

stix

Pattern
[hostname:value = 'trojan.win32.cookiemonster.jcb']

Name
9c5898b1b354b139794f10594e84e94e991971a54d179b2e9f746319ffac56aa

Description
compromised_site_redirector_fromcharcode SHA256 of 6817df1da376e8f6e68fd1ad06d78f02406b6e19

Pattern Type
stix

Pattern
[file:hashes.'SHA-256' = '9c5898b1b354b139794f10594e84e94e991971a54d179b2e9f746319ffac56aa']

Name
sito-company.com

Pattern Type
stix

Pattern
[domain-name:value = 'sito-company.com']

Name

complete-s.monster

Pattern Type

stix

Pattern

[domain-name:value = 'complete-s.monster']

Name

https://sito-company.com/launcher/auth

Pattern Type

stix

Pattern

[url:value = 'https://sito-company.com/launcher/auth']

Name

91.212.166.16

Description

ISP: Proton66 OOO **OS:** None ------------------------- Hostnames: - 0g.tel ------------------------- Domains: - 0g.tel ------------------------- Services: **443:** ``` HTTP/1.1 200 OK Server: nginx Date: Mon, 13 Nov 2023 16:49:58 GMT Content-Type: application/zip Connection: close Content-Disposition: attachment; filename=lc4w.zip Content-Length: 42374 ``` HEARTBLEED: 2023/11/13 16:50:14 91.212.166.16:443 - SAFE ------------------

Pattern Type

stix

Pattern

[ipv4-addr:value = '91.212.166.16']

Name

f30b39f5e722cb106f37d1738fff7ad20fa8e312d82e246d4a6e2175685b963b

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'f30b39f5e722cb106f37d1738fff7ad20fa8e312d82e246d4a6e2175685b963b']

Name

d9ca193b5da85a3841ec749b67168c906e21bbaac40f0a0bff40839efb3a74c1

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'd9ca193b5da85a3841ec749b67168c906e21bbaac40f0a0bff40839efb3a74c1']

Domain-Name

Value
complete-s.monster
sito-company.com
fast-difficult.monster
ps1-local.com

StixFile

Value
3b0defb024e41af699b5dfc424a9ff276409f447edd24af024b34941f5ab62a9
9c5898b1b354b139794f10594e84e94e991971a54d179b2e9f746319ffac56aa
d9ca193b5da85a3841ec749b67168c906e21bbaac40f0a0bff40839efb3a74c1
305cb9ebdef618a626075f71fce3f4a64091e7a875a5ddff983aaeeea0f1fd41
cb99365bac3d168e295aa0764a1c67e1a7e582731880ad0522e9b6b3616275df
f30b39f5e722cb106f37d1738fff7ad20fa8e312d82e246d4a6e2175685b963b

Hostname

Value
trojan.win32.cookiemonster.jcb
230927151335115.mxb.ewk48.shop

IPv4-Addr

Value
91.212.166.16
91.103.252.74

Url

Value
https://ps1-local.com/obfs3ip2.bs64
http://230927151335115.mxb.ewk48.shop/f/fvgs30927001.msi
https://complete-s.monster/upd.php
https://fast-difficult.monster/api7.php?name=microsoft_barcode_control_16.0_download
https://sito-company.com/launcher/auth

External References

- https://otx.alienvault.com/pulse/65609160cddfd2987cac2ef3
- https://www.trendmicro.com/en_us/research/23/k/attack-signals-possible-return-of-genesis-market.html
- https://www.trendmicro.com/content/dam/trendmicro/global/en/research/23/k/attack-signals-possible-return-of-genesis-market/iocs-attack-signals-possible-return-of-genesis-market.txt