May 02 2024

NETMANAGEIT

Intelligence Report

Threat Brief: Post-Exploitation Activity Related to CVE-2024-3400 (Updated May 1)

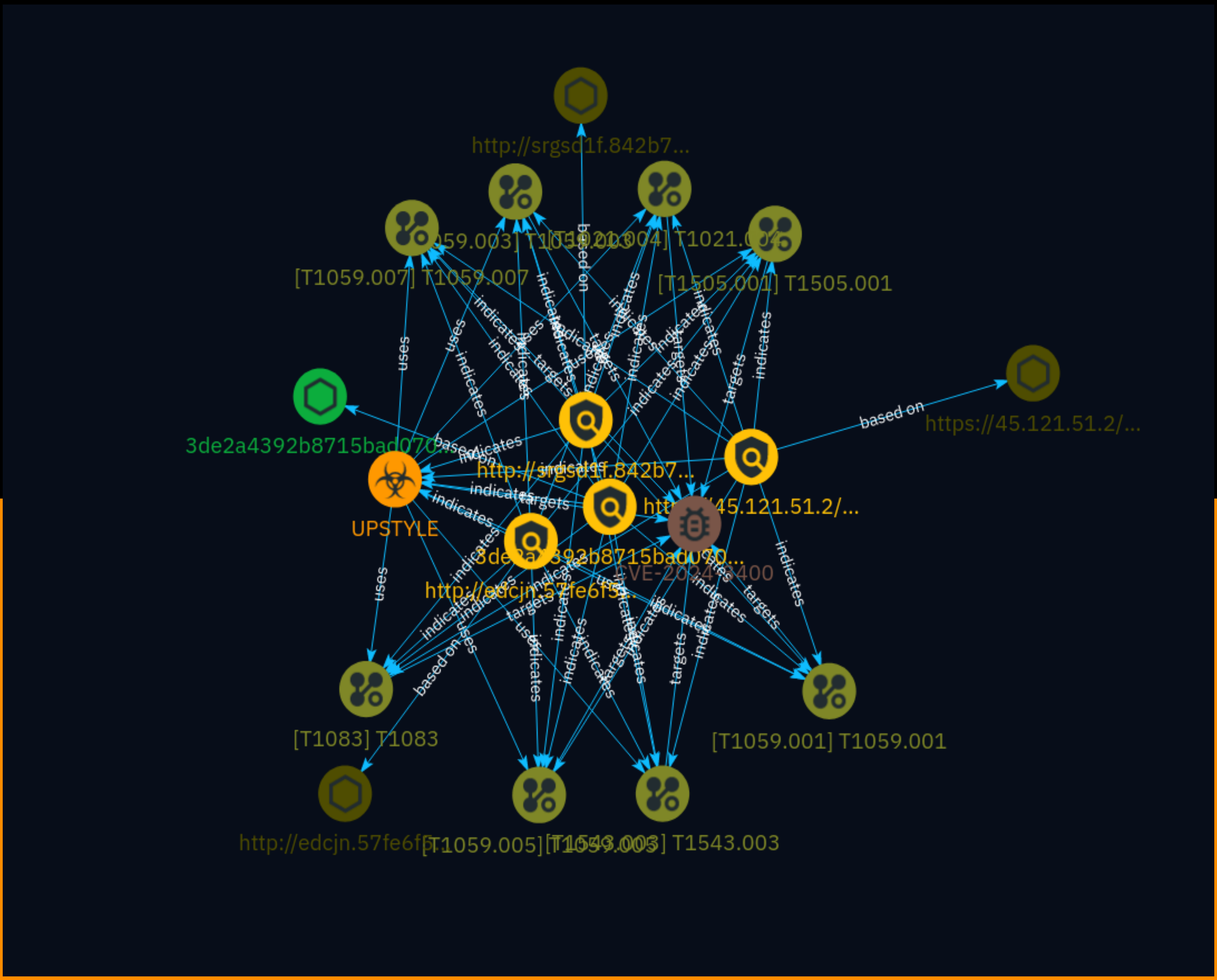

Table of contents

Overview

Entities

Observables

External References

Overview

Description

This threat brief covers a critical command injection vulnerability in Palo Alto Networks PAN-OS software (CVE-2024-3400) that enables an unauthenticated attacker to execute arbitrary code with root privileges on vulnerable firewalls. The vulnerability is being actively exploited in a campaign referred to as Operation MidnightEclipse. The report details the scope of the attack, including various levels of exploitation observed, and provides information on post-exploitation activity involving the installation of backdoors and exfiltration of configuration files. Guidance for mitigation and detection is provided, along with indicators of compromise and resources for further information.

Confidence

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

100 / 100

Content

N/A

Indicator

Name

https://45.121.51.2/abc.txt

Description

- **Unsafe:** False - **Server:** N/A - **Domain Rank:** 0 - **DNS Valid:** False - **Parking:** False - **Spamming:** False - **Malware:** False - **Phishing:** False - **Suspicious:** True - **Adult:** False - **Category:** N/A - **Domain Age:** {'human': 'N/A', 'timestamp': None, 'iso': None} - **IPQS: Domain:** 45.121.51.2 - **IPQS: IP Address:** N/A

Pattern Type

stix

Pattern

[url:value = 'https://45.121.51.2/abc.txt']

Name

http://edcjn.57fe6f5d9d.ipv6.1433.eu.org

Description

- **Unsafe:** True - **Server:** N/A - **Domain Rank:** 949729 - **DNS Valid:** True - **Parking:** True - **Spamming:** False - **Malware:** True - **Phishing:** True - **Suspicious:** True - **Adult:** False - **Category:** Web Tracker - **Domain Age:**

{'human': '2 years ago', 'timestamp': 1654523505, 'iso': '2022-06-06T09:51:45-04:00'} - **IPQS: Domain:** edcjn.57fe6f5d9d.ipv6.1433.eu.org - **IPQS: IP Address:** 10.10.10.10

Pattern Type

stix

Pattern

[url:value = 'http://edcjn.57fe6f5d9d.ipv6.1433.eu.org']

Name

http://srgsd1f.842b727ba4.ipv6.1433.eu.org

Description

- **Unsafe:** True - **Server:** N/A - **Domain Rank:** 949729 - **DNS Valid:** True - **Parking:** False - **Spamming:** False - **Malware:** True - **Phishing:** True - **Suspicious:** True - **Adult:** False - **Category:** Web Tracker - **Domain Age:** {'human': '2 years ago', 'timestamp': 1654523505, 'iso': '2022-06-06T09:51:45-04:00'} - **IPQS: Domain:** srgsd1f.842b727ba4.ipv6.1433.eu.org - **IPQS: IP Address:** 10.10.10.10

Pattern Type

stix

Pattern

[url:value = 'http://srgsd1f.842b727ba4.ipv6.1433.eu.org']

Name

3de2a4392b8715bad070b2ae12243f166ead37830f7c6d24e778985927f9caac

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '3de2a4392b8715bad070b2ae12243f166ead37830f7c6d24e778985927f9caac']

Vulnerability

Name
CVE-2024-3400
Description
Palo Alto Networks PAN-OS GlobalProtect feature contains a command injection vulnerability that allows an unauthenticated attacker to execute commands with root privileges on the firewall.

Malware

Name
UPSTYLE

Attack-Pattern

Name
T1505.001
ID
T1505.001
Description
Adversaries may abuse SQL stored procedures to establish persistent access to systems. SQL Stored Procedures are code that can be saved and reused so that database users do not waste time rewriting frequently used SQL queries. Stored procedures can be invoked via SQL statements to the database using the procedure name or via defined events (e.g. when a SQL server application is started/restarted). Adversaries may craft malicious stored procedures that can provide a persistence mechanism in SQL database servers.(Citation: NetSPI Startup Stored Procedures)(Citation: Kaspersky MSSQL Aug 2019) To execute operating system commands through SQL syntax the adversary may have to enable additional functionality, such as xp_cmdshell for MSSQL Server.(Citation: NetSPI Startup Stored Procedures)(Citation: Kaspersky MSSQL Aug 2019)(Citation: Microsoft xp_cmdshell 2017) Microsoft SQL Server can enable common language runtime (CLR) integration. With CLR integration enabled, application developers can write stored procedures using any .NET framework language (e.g. VB .NET, C#, etc.).(Citation: Microsoft CLR Integration 2017) Adversaries may craft or modify CLR assemblies that are linked to stored procedures since these CLR assemblies can be made to execute arbitrary commands.(Citation: NetSPI SQL Server CLR)
Name
T1059.005

ID

T1059.005

Description

Adversaries may abuse Visual Basic (VB) for execution. VB is a programming language created by Microsoft with interoperability with many Windows technologies such as [Component Object Model](https://attack.mitre.org/techniques/T1559/001) and the [Native API](https://attack.mitre.org/techniques/T1106) through the Windows API. Although tagged as legacy with no planned future evolutions, VB is integrated and supported in the .NET Framework and cross-platform .NET Core.(Citation: VB .NET Mar 2020)(Citation: VB Microsoft) Derivative languages based on VB have also been created, such as Visual Basic for Applications (VBA) and VBScript. VBA is an event-driven programming language built into Microsoft Office, as well as several third-party applications.(Citation: Microsoft VBA)(Citation: Wikipedia VBA) VBA enables documents to contain macros used to automate the execution of tasks and other functionality on the host. VBScript is a default scripting language on Windows hosts and can also be used in place of [JavaScript](https://attack.mitre.org/techniques/T1059/007) on HTML Application (HTA) webpages served to Internet Explorer (though most modern browsers do not come with VBScript support).(Citation: Microsoft VBScript) Adversaries may use VB payloads to execute malicious commands. Common malicious usage includes automating execution of behaviors with VBScript or embedding VBA content into [Spearphishing Attachment](https://attack.mitre.org/techniques/T1566/001) payloads (which may also involve [Mark-of-the-Web Bypass](https://attack.mitre.org/techniques/T1553/005) to enable execution).(Citation: Default VBS macros Blocking)

Name

T1059.003

ID

T1059.003

Description

Adversaries may abuse the Windows command shell for execution. The Windows command shell ([cmd](https://attack.mitre.org/software/S0106)) is the primary command prompt on Windows systems. The Windows command prompt can be used to control almost any aspect of a system, with various permission levels required for different subsets of commands. The command prompt can be invoked remotely via [Remote Services](https://attack.mitre.org/techniques/T1021) such as [SSH](https://attack.mitre.org/techniques/T1021/004).(Citation: SSH in Windows) Batch files (ex: .bat or .cmd) also provide the shell with a list of sequential commands to run, as well as normal scripting operations such as conditionals and loops. Common uses of batch files include long or repetitive tasks, or the need to run the same set of commands on multiple systems. Adversaries may leverage [cmd](https://attack.mitre.org/software/S0106) to execute various commands and payloads. Common uses include [cmd](https://attack.mitre.org/software/S0106) to execute a single command, or abusing [cmd](https://attack.mitre.org/software/S0106) interactively with input and output forwarded over a command and control channel.

Name

T1059.001

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

T1083

ID

T1083

Description

Adversaries may enumerate files and directories or may search in specific locations of a host or network share for certain information within a file system. Adversaries may use the information from [File and Directory Discovery](https://attack.mitre.org/techniques/T1083) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Many command shell utilities can be used to obtain this information. Examples include `dir`, `tree`, `ls`, `find`, and `locate`.(Citation: Windows Commands JPCERT) Custom tools may also be used to gather file and directory information and interact with the [Native API](https://attack.mitre.org/techniques/T1106). Adversaries may also leverage a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) on network devices to gather file and directory information (e.g. `dir`, `show flash`, and/or `nvram`).(Citation: US-CERT-TA18-106A) Some files and directories may require elevated or specific user permissions to access.

Name

T1021.004

ID

T1021.004

Description

Adversaries may use [Valid Accounts](https://attack.mitre.org/techniques/T1078) to log into remote machines using Secure Shell (SSH). The adversary may then perform actions as the logged-on user. SSH is a protocol that allows authorized users to open remote shells on other computers. Many Linux and macOS versions come with SSH installed by default, although typically disabled until the user enables it. The SSH server can be configured to use standard password authentication or public-private keypairs in lieu of or in addition to a password. In this authentication scenario, the user’s public key must be in a special file on the computer running the server that lists which keypairs are allowed to login as that user.

Name

T1543.003

ID

T1543.003

Description

Adversaries may create or modify Windows services to repeatedly execute malicious payloads as part of persistence. When Windows boots up, it starts programs or applications called services that perform background system functions.(Citation: TechNet Services) Windows service configuration information, including the file path to the service's executable or recovery programs/commands, is stored in the Windows Registry. Adversaries may install a new service or modify an existing service to execute at startup in order to persist on a system. Service configurations can be set or modified using system utilities (such as sc.exe), by directly modifying the Registry, or by interacting directly with the Windows API. Adversaries may also use services to install and execute malicious drivers. For example, after dropping a driver file (ex: `.sys`) to disk, the payload can be loaded and registered via [Native API](https://attack.mitre.org/techniques/T1106) functions such as `CreateServiceW()` (or manually via functions such as `ZwLoadDriver()` and `ZwSetValueKey()`), by creating the required service Registry values (i.e. [Modify Registry](https://attack.mitre.org/techniques/T1112)), or by using command-line utilities such as `PnPUtil.exe`.(Citation: Symantec W.32 Stuxnet Dossier)(Citation: Crowdstrike DriveSlayer February 2022)(Citation: Unit42 AcidBox June 2020) Adversaries may leverage these drivers as [Rootkit](https://attack.mitre.org/techniques/T1014)s to hide the presence of malicious activity on a system. Adversaries may also load a signed yet vulnerable driver onto a compromised machine (known as "Bring Your Own Vulnerable Driver" (BYOVD)) as part of [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068).(Citation: ESET InvisiMole June 2020)(Citation: Unit42 AcidBox June 2020) Services may be created with administrator privileges but are executed under SYSTEM privileges, so an adversary may also use a service to escalate privileges. Adversaries may also directly start services through [Service Execution](https://attack.mitre.org/techniques/T1569/002). To make detection analysis more challenging, malicious services may also incorporate [Masquerade Task or Service](https://attack.mitre.org/techniques/T1036/004) (ex: using a service and/or payload name related to a legitimate OS or benign software component). Adversaries may also create ‘hidden’ services (i.e., [Hide Artifacts](https://attack.mitre.org/techniques/T1564)), for example by using the `sc sdset` command to set service permissions via the Service Descriptor Definition Language (SDDL). This may hide a Windows service from the view of standard service enumeration methods such as `Get-Service`, `sc query`, and `services.exe`.(Citation: SANS 1)(Citation: SANS 2)

Name

T1059.007

ID

T1059.007

Description

Adversaries may abuse various implementations of JavaScript for execution. JavaScript (JS) is a platform-independent scripting language (compiled just-in-time at runtime) commonly associated with scripts in webpages, though JS can be executed in runtime environments outside the browser.(Citation: NodeJS) JScript is the Microsoft implementation of the same scripting standard. JScript is interpreted via the Windows Script engine and thus integrated with many components of Windows such as the [Component Object Model](https://attack.mitre.org/techniques/T1559/001) and Internet Explorer HTML Application (HTA) pages.(Citation: JScrip May 2018)(Citation: Microsoft JScript 2007)(Citation: Microsoft Windows Scripts) JavaScript for Automation (JXA) is a macOS scripting language based on JavaScript, included as part of Apple’s Open Scripting Architecture (OSA), that was introduced in OSX 10.10. Apple’s OSA provides scripting capabilities to control applications, interface with the operating system, and bridge access into the rest of Apple’s internal APIs. As of OSX 10.10, OSA only supports two languages, JXA and [AppleScript](https://attack.mitre.org/techniques/T1059/002). Scripts can be executed via the command line utility `osascript`, they can be compiled into applications or script files via `osacompile`, and they can be compiled and executed in memory of other programs by leveraging the OSAKit Framework.(Citation: Apple About Mac Scripting 2016)(Citation: SpecterOps JXA 2020)(Citation: SentinelOne macOS Red Team)(Citation: Red Canary Silver Sparrow Feb2021)(Citation: MDSec macOS JXA and VSCode) Adversaries may abuse various implementations of JavaScript to execute various behaviors. Common uses include hosting malicious scripts on websites as part of a [Drive-by Compromise](https://attack.mitre.org/techniques/T1189) or downloading and executing these script files as secondary payloads. Since these payloads are text-based, it is also very common for adversaries to obfuscate their content as part of [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027).

Url

Value
https://45.121.51.2/abc.txt
http://edcjn.57fe6f5d9d.ipv6.1433.eu.org
http://srgsd1f.842b727ba4.ipv6.1433.eu.org

StixFile

Value
3de2a4392b8715bad070b2ae12243f166ead37830f7c6d24e778985927f9caac

External References

- https://otx.alienvault.com/pulse/662928da3cb477552cd932ab
- https://unit42.paloaltonetworks.com/cve-2024-3400/