Network Security - IE3032

**Vulnerability Analysis Report**.

A blue and orange logo

Description automatically generated

Sri Lanka Institute of Information Technology

Lecturer: Dr. Harinda Fernando

Deadline: 2024/04/19

|  |  |
| --- | --- |
| **IT numbers** | **Name** |
| IT21826368 | Nanayakkara Y.D.T. D |
| IT21822612 | Mendis H.R.M |
| IT21831904 | Weerasinghe K.M |
| IT21828348 | Dissanayaka K.D.A.R. A |

**Contents**

[1. Actively Exploited PAN-OS vulnerability on Palo Alto firewalls 3](#_Toc164708847)

[1.1 Affected Versions 3](#_Toc164708848)

[1.2 Exploitation 3](#_Toc164708849)

[1.3 Mitigations 4](#_Toc164708850)

[2 OpenNMS XSS Flaw Let Attackers Inject JavaScript Payload. 5](#_Toc164708851)

[2.1 Affected Versions. 5](#_Toc164708852)

[2.2 Exploitations. 5](#_Toc164708853)

[2.3 Mitigations. 6](#_Toc164708854)

[3 Unpacking the Blackjack Group's Fuxnet Malware. 7](#_Toc164708855)

[3.1 Affected Items. 7](#_Toc164708856)

[3.2 Exploitation. 7](#_Toc164708857)

[3.3 Mitigations. 8](#_Toc164708858)

[4 D-Link NAS Command Injection vulnerability 9](#_Toc164708859)

[4.1 Affected items. 9](#_Toc164708860)

[4.2 Exploitation 9](#_Toc164708861)

[4.3 Mitigations. 10](#_Toc164708862)

[5 The Use of a Hard-coded Cryptographic Key vulnerability. 11](#_Toc164708863)

[5.1 Affected items. 11](#_Toc164708864)

[5.2 Exploitations. 11](#_Toc164708865)

[5.3 Mitigations. 12](#_Toc164708866)

[6 References 13](#_Toc164708867)

# Actively Exploited PAN-OS vulnerability on Palo Alto firewalls

Leading US cybersecurity company Palo Alto has discovered a serious flaw in the **PAN-OS** "GlobalProtect" function. This vulnerability targets the command-injection vulnerability (**CVE-2024-3400**) and enables **remote code execution (RCE)** with root privileges. The business published a warning revealing that a few certain PAN-OS setups can provide an unauthorized attacker the capacity to remotely run arbitrary code as root. To lessen the issue, Palo Alto has advised its clients to apply interim patches.

According to the MITRE framework, CVE-2024-3400 is a vulnerability classified as "Improper Neutralization of Special Elements used in a Command," **CVSS score of 10.0**.

## Affected Versions

* PAN-OS 11.1 - <11.2.2-h3
* PAN-OS 11.0 - <11.0.4-h1
* PAN-OS 10.2 - <10.2.9-h1

## Exploitation

The exploitation of the vulnerability is what makes Operation **MIDNIGHTECLIPSE** possible. It involves setting up a **cronjob** that runs once per minute to get commands stored on an external server ("**172.233.228[.]93/policy" or "172.233.228[.]93/patch**"), which are then run via the bash shell.

It is claimed that the attackers manually maintained the command-and-control (C2) the access control list (ACL) of the server to make sure that only the device connecting to it could access it.

Although the exact nature of the command is unknown, it is believed that the URL is a delivery mechanism for a Python-based backdoor on the firewall that is hosted on a different server ("**144.172.79[.]92" and "nhdata.s3-us-west-2.amazonaws[.]com**"). **VOLEXITY**, which discovered the in-the-wild exploitation of **CVE-2024-3400** on April 10, 2024, is tracking as **UPSTYLE**.

The purpose of the Python script is to build and execute a second Python script ("**system.pth**"), which then decodes and activates the embedded backdoor component that sends instructions to the threat actor's file named "**sslvpn\_ngx\_error.log**." The output of the process is saved in a different file called "**bootstrap.min.css**."

The fact that the files used to write the results and extract the instructions are both authentic files connected to the firewall is what makes the attack chain the most fascinating.

**/var/appweb/sslvpndocs/global-protect/portal/css/bootstrap.min.css** and **/var/log/pan/sslvpn\_ngx\_error.log**

Regarding the orders' writing to the web server error log, the threat actor spoofs network requests to a fictitious website that follows a predetermined pattern. After parsing the log file, the backdoor looks for a line that matches the same regular expression ("**img\[([a-zA-Z0-9+/=]+)\]**") in order to decode and execute the command contained inside.

"The script will subsequently establish a new thread that runs a process called restore,".

## Mitigations

* Threat Prevention subscription can block attacks for this vulnerability using Threat ID 95187
* This issue is fixed in hotfix releases of PAN-OS 10.2.9-h1, PAN-OS 11.0.4-h1, PAN-OS 11.1.2-h3, and in all later PAN-OS versions.
* Monitoring the network for unusual activities like spikes in traffic or unauthorized access attempts. Investigating unexpected activity involves digging into these anomalies to understand their cause and impact, using tools like log files and network data.
* **Disabling the telemetry feature, which means stopping the automatic collection and sending data from devices to a central location for analysis or monitoring. Usually, this data includes information like device performance, usage, or other metrics.** **There are several reasons to disable telemetry, including following rules, safeguarding privacy, and consuming less resources and network traffic. It is crucial to acknowledge that turning off telemetry could lead to the deletion of crucial data regarding the safety and operation of the apparatus. Therefore, before turning it off, enterprises should carefully consider the advantages and disadvantages.**

# OpenNMS XSS Flaw Let Attackers Inject JavaScript Payload.

A Widely used enterprise-grade *network monitoring solution* named “**OpenNMS**” has been identified with a critical vulnerability, allowing attackers to inject malicious javascript payloads though a ***Cross-Site Scritpting(XSS)*** flaw.

Identified as the ***CVE-2023-0846*** vulnerability is causing considerable alarm due to its capacity to jeopardize the security of networks under OpenNMS monitoring. This XSS vulnerability originates from inadequate sanitization of user input within the OpenNMS web application.

Attackers exploit this flaw by **sending crafted data**, allowing them to execute JavaScript in the victim's session, which also exploits another vulnerability in the application to **execute arbitrary code** on the OpenNMS server once an admin views the dashboard, potentially leading to session hijacking and data theft.

According to the MITRE framework, **CVE-2023-0846** is a vulnerability classified as “Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')” with the score of **CVSS score of 6.1.**

## Affected Versions.

 OpenNMS Horizon Versions < 31.0.4

 OpenNMS Meridian Versions < 2023.1.0

## Exploitations.

**Unauthenticated XSS (CVE-2023-0846)**

The OpenNMS server. While SNMP polling involves the manager actively retrieving data from devices, traps allow devices to proactively send information. OpenNMS has a dedicated SNMP listener to process these traps, converting them into events if they are deemed relevant. However, by manipulating SNMP traps, attackers can inject XSS payloads into the OpenNMS admin dashboard.

**Spoofing SNMP Traps**

Attackers can manipulate SNMP traps to inject XSS payloads into OpenNMS's admin dashboard. By crafting deceptive trap messages, attackers can trick the SNMP listener into processing these traps as legitimate events, bypassing security measures.

This technique can be used for various malicious purposes, like False Alarms, Evasion of Detection, Exploitations.

**Authenticated Command Injection**

The combination of an authenticated command injection flaw with the XSS vulnerability in OpenNMS poses a significant security risk. With the command injection flaw, attackers who have authenticated access to the system can execute arbitrary commands, gaining the ability to manipulate system resources and potentially compromise its integrity.

## Mitigations.

* The vulnerabilities were patched in all versions subsequent to **OpenNMS version 31.0.4**, effectively addressing XSS attacks and command injection exploits.
* Implementing rigorous input validation and sanitization mechanisms involves thoroughly checking and cleaning any data that enters your system, especially from external or untrusted sources like SNMP (Simple Network Management Protocol) traps.
* Input validation ensures that the data conforms to expected formats, lengths, and ranges, helping to prevent various types of attacks such as injection attacks (e.g., SQL injection) or buffer overflows.
* Sanitization is the process of eliminating or encoding potentially dangerous characters or sequences from the input data to prevent attacks such as command injection and cross-site scripting (XSS).
* Enforce strong access controls and authentication mechanisms to restrict access to sensitive functionalities within **OpenNMS**, minimizing the likelihood of unauthorized access and exploitation.

# Unpacking the Blackjack Group's Fuxnet Malware.

**Blackjack,** a hacking collective that is thought to be connected to Ukraine's security services, has declared that it has attacked multiple important Russian institutions. The hackers allegedly caused major damage and exfiltrated critical information by targeting data centers, utilities, ISPs, and the Russian military.

Blackjack revealed the specifics of a purported assault against “**Moscollector**”, a Moscow-based business in charge of subterranean infrastructure, which includes Russia's Network Operation Center (NOC), which oversees and regulates gas, water, fire alarms, and a host of other services, including a sizable network of remote sensors and Internet of Things controllers.

They asserted to have turned off 87,000 sensors in total, including those connected to gas pipelines, metro systems, and airports. They claimed to have utilized "**Fuxnet**," a malware they referred to as "***Stuxnet on steroids***," to accomplish this, allowing them to physically destroy sensor equipment.

## Affected Items.

* Sensors with the RS485/Meter-Buses.

## Exploitation.

Most likely, the infection was spread remotely. *Once on a device*, it would begin wiping out critical files and folders, stopping remote access services to stop remote restoration, and wiping out routing table data to stop connecting to other devices. After that, **Fuxnet** would rebuild the device's flash memory and erase the file system.

The malware tries to physically damage the **NAND memory chip** once it has corrupted the file system and prevented access to the device. It then rewrites the **UBI volume** to prevent rebooting.

Furthermore, by overloading the serial bus and the sensors with random data, the malware tries to interfere with the sensors that are attached to the gateway.

## Mitigations.

* Implement firewalls and IDS to monitor network traffic. Firewalls function as a barrier between the outside world and your internal network by using pre-established security rules. infiltration detection systems (IDS) augment firewalls by continuously monitoring network traffic for anomalous activity or signs of infiltration.
* Employ robust antivirus and endpoint protection software across all devices and networks.
* Divide networks into more manageable, remote areas to stop the spread of viruses such as FluxNet. The process of breaking up a larger network into more manageable, isolated zones, each with its own set of access rules and security measures. By doing this, you can lessen the possible breach's extent and stop attackers from moving laterally.

# D-Link NAS Command Injection vulnerability

The D-Link NAS command injection vulnerability specifically affects D-Link network attached storage devices. A total of 92,000 devices were impacted by this vulnerability. The **D-Link DNS-320L, DNS-325, DNS-327L, and DNS-340L** models, as well as the **nas\_sharing.cgi** script, are the specific components that are impacted by this vulnerability. The suppliers have verified that these models are no longer supported and have reached their end-of-life.

The vulnerability's severity ranges from high to critical since it enables the execution of arbitrary commands on the vulnerable devices. The **NIST national vulnerability database** rates the severity of the **CVE-2024-3273** vulnerability using a **base score of 9.8**.

Attackers can take advantage of this vulnerability by sending a malicious constructed GET request with a customized payload to the vulnerable CGI endpoint.

## Affected items.

* DNS-320L Version 1.11, Version 1.03.0904.2013, Version 1.01.0702.2013
* DNS-325 Version 1.01
* DNS-327L Version 1.09, Version 1.00.0409.2013
* DNS-340L Version 1.08

## Exploitation

nas\_sharing.cgi script is the main reason for the existence of this vulnerability. The exploitation involves **backdoor and command injection** to receive remote code execution.

The script **nas\_sharing.cgi** has a hardcoded account that, by exposing the login and password, could potentially be taken advantage of to serve as a backdoor.

Because the parameter request contains an empty **password (passwd=)** and a **username (user=messagebus)**, the exploitation is easy. Attackers could be able to get illegal entry to this without the necessary authentication.

The System parameter allows for command injection since it contains a **base64-encoded** value that, once decoded, functions as a command that can be executed.

By using a malicious http get request using the user parameters and the decoded system parameter values targeting the **nas\_sharing.cgi** endpoint the vulnerability can be exploited.

**GET**

**/cgi/bin/nas\_sharing.cgi?user=messagebus&passwd=&cmd=15&system=<BASE64\_ENCODED COMMAND\_TO\_BE\_EXECUTED>**

If this vulnerability is successfully exploited, the attackers could be able to run arbitrary commands on the device, which could impact in,

* denial-of-service attacks
* modified system settings
* illegal possession of confidential data**.**

## Mitigations.

* Check for any available patches released by the supplier - Follow up with any security patches or updates that the supplier may issue. Suppliers could continue to supply updates for serious vulnerabilities as these even after the impacted products are no longer in production.
* Device retirement or replacement - Mostly impacted by this vulnerability are devices that companies have marked as end-of-life. Removal or replacing of these devices is the best way to address this issue. Since they are discontinued from getting security patches or upgrades, end-of-life devices are left open to both known and unknown security threats. Organizations may lower the risk of exploitation and vulnerability by retiring or replacing these devices.
* Frequent password change - Set a policy in which the password required for accessing the online configuration interface of the device is changed on a regular basis.
* Enabling Wi-Fi encryption with a unique password. - Ensure that the network to which the NAS device is connected has Wi-Fi encryption enabled, and set a strong, special password to get access to the network. Through doing it, unauthorized interference with the network and the NAS device can be avoided.

# The Use of a Hard-coded Cryptographic Key vulnerability.

**Juniper Networks, Juniper Cloud Native Routers (JCNR) and containerized routing Protocol Deamon (cRPD)** products were vulnerable to the attacks known as perform **Person-in-the-Middle (PitM)** attacks which results in complete compromise of the containers. Due to the use of **Hard-Coded Cryptographic Key** Vulnerability.

A Person in the Middle, the attacker can surreptitiously intercept SSH communication since the container has hardcoded SSH host keys. Because the cryptographic key is hardcoded, changing it requires a software update, leaving a gap for attackers to exploit without detection.

The vulnerability's severity state high because this results in complete compromise of the container. The **NIST national vulnerability database** rates the severity of the **CVE-2024-30407** vulnerability using a **base score of 8.1**.

## Affected items.

* Containerized Routing Protocol Deamon products (cRPD) < **23.4R1 version.**
* Juniper Cloud Native Routers (JCNR) < **23.4 version.**

## Exploitations.

By reverse engineering, decomplication, or other techniques, the attacker obtains access to the application's code or binaries. Then they can quickly determine the hard-coded cryptographic key once they have access.

The following code examples in **C language** show the attempt to verify a password using a hard-coded cryptographic key.

**int VerifyAdmin(char \*password) {**

**if (strcmp(password,"68af404b513073584c4b6f22b6c63e6b")) {**

**printf("Incorrect Password!\n");**

**return(0);**

**}**

**printf("Entering Diagnostic Mode...\n");**

**return(1);**

**}**

Then the cryptographic key is within a hard-coded string value that is compared to the password and will be able to read the key and compromise the system.

## Mitigations.

* Upgrade to the latest software versions.
* Utilize a key management system provided by cloud services or third-party vendors.
* Instead of Hard coded keys, generate cryptographic keys dynamically during the runtime.
* Can use a Hardware Security module (HSM) which can leverage the hardware security even more. HSMs are specialized hardware devices designed to securely generate, store, and manage cryptographic keys. By offloading cryptographic operations to an HSM, the security can be enhanced by protecting keys from the tapering and unauthorized access. Also, they offer additional security features such as physical tamper resistance and key backup capabilities.

# References

42, U. (2024, 04 12). *cve-2024-3400*. Retrieved from palo alto networks: https://unit42.paloaltonetworks.com/cve-2024-3400/

*CVE-2024-3400*. (2024, 04 12). Retrieved from Palo Alto Networks: https://security.paloaltonetworks.com/CVE-2024-3400

cvedetails.com. (n.d.). *cvedetails.com*. Retrieved from https://www.cvedetails.com/cve/CVE-2023-0846/

Divya. (2024, March 6). *gbhackers.com*. Retrieved from https://gbhackers.com/opennms-xss-attackers-javascript/

Enumeration, C. W. (n.d.). *cwe.mitre.org*. Retrieved from https://cwe.mitre.org/data/definitions/321.html

Eswar. (2024, 4 4). *cybersecuritynews.com*. Retrieved from https://cybersecuritynews.com/d-link-nas-command-injection-impact/

Gyongyoși, L. (2024, 4 10). *heimdalsecurity.com*. Retrieved from https://heimdalsecurity.com/blog/d-link-nas-devices-vulnerable/

Joseph, A. (2024, 04 15). *palo-alto-networks-new-firewall-vulnerability*. Retrieved from thecyberexpress: https://thecyberexpress.com/palo-alto-networks-new-firewall-vulnerability/

Juniper Networks, I. (n.d.). *nvd.nist.gov*. Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2024-30407

Kovacas, E. (n.d.). *securityweek.com*. Retrieved from https://www.securityweek.com/destructive-ics-malware-fuxnet-used-by-ukraine-against-russian-infrastructure/

netsecfish. (2024, 3 26). *github*. Retrieved from https://github.com/netsecfish/dlink?tab=readme-ov-file#fix-recommendation

Schiller, S. (n.d.). *sonarsource.com*. Retrieved from https://www.sonarsource.com/blog/opennms-vulnerabilities-securing-code-against-attackers-unexpected-ways/

Team82. (n.d.). *Claroty.com*. Retrieved from https://claroty.com/team82/research/unpacking-the-blackjack-groups-fuxnet-malware