proximus NXT
cybersecurity

Proximus Luxembourg CyberSecurity Incident Response Team

Vulnerability Notification Form

Created on 21/07/26 | Version 4.0 | Sensitivity: Public | TLP: CLEAR

The Proximus Luxembourg CSIRT (**PXS-LU-CSIRT**) is the response entity for the computer incidents related to the Autonomous System Number (ASN) **AS56665**.

To notify a vulnerability in either software or hardware products, please complete this form as detailed as possible with enough information for allowing PXS-LU-CSIRT and the Vendor to analyse it, understand it and reproduce it, and send it to csirt (at) proximus (dot) lu preferably PGP/GPG encrypted (PGP KeyID 6E2EA9F8).

PXS-LU-CSIRT hours of operation are restricted to regular business hours: 09h00-17h00 CET from Monday to Friday except during Luxembourg's public holidays. Outside of these hours and in case of emergency, the email telecomsd (at) proximus (dot) lu address, mainly dedicated to operational problems, can be contacted.

All reported information will be treated confidentially according to our policies (please refer to our rfc2350 available at https://www.proximusnxt.lu/en/proximus-luxembourg-csirt).

Bug Bounty Policy - We value those who take the time and effort to report security vulnerabilities. However, we do not offer monetary rewards for vulnerability disclosures.

Multiple Related Vulnerabilities - If several vulnerabilities are related or form an exploitation chain (e.g., a weak cipher combined with reversible password storage), they should be reported in a single form. Use Section 3 to document each vulnerability individually and Section 4 to describe how they relate to each other as a chain.

SECTION 1 - About the Reporter

Company / Organisation

Reporter Full Name

Phone Number

Email Address

PGP Key ID / Fingerprint

Provide your PGP key ID so that the PXS-LU-CSIRT can encrypt responses back to you.

PGP Public Key URL (or attach to submission)

Remain Anonymous

☐ Yes ☐ No (Default)

SECTION 2 - Impacted Product(s)

List all affected vendors, products, and versions. Add rows as needed.

Vendor	Product	Version(s) Tested	Fixed Version (if known)	Vulnerable (Yes / No)

SECTION 3 - Vulnerability Details (repeat for each vulnerability)

IMPORTANT - If reporting multiple related vulnerabilities, duplicate this entire section for each individual vulnerability. Number them (Vulnerability #1, #2, etc.) for cross-reference in the Vulnerability Chain section (Section 4).

3.1 - Vulnerability Identification

Vulnerability Reference Number

Internal numbering for this submission, e.g., Vulnerability #1, #2, etc.

Vulnerability Type

Select one or more applicable types. If "Other", specify in the field below.

- ☐ Remote Code Execution (RCE)
- ☐ Cross-Site Scripting (XSS)
- ☐ Buffer Overflow
- ☐ Privilege Escalation
- ☐ Denial of Service (DoS)
- ☐ Insecure Deserialization
- ☐ XML External Entity (XXE)
- ☐ Cryptographic Weakness
- ☐ Other (specify below)
- ☐ SQL Injection
- ☐ Cross-Site Request Forgery (CSRF)
- ☐ Authentication Bypass
- ☐ Information Disclosure
- ☐ Path / Directory Traversal
- ☐ Server-Side Request Forgery (SSRF)
- ☐ Command Injection
- ☐ Insecure Data Storage
- ☐ Unknown / Unclassified

If "Other", please specify:

CWE Classification

Common Weakness Enumeration identifier. Reference: https://cwe.mitre.org/

CWE ID	CWE Description
e.g., CWE-327	*e.g., Use of a Broken or Risky Cryptographic Algorithm*

3.2 — CVSS v3.1 Base Score

Use the Online calculator: https://www.first.org/cvss/calculator/3.1

Computed CVSS v3.1 Vector String

Vector String	Base Score	Severity
https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:A/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N	*5.7*	*Medium*

None: 0.0	Low: 0.1–3.9	Medium: 4.0–6.9	High: 7.0–8.9	Critical: 9.0–10.0

3.3 - Technical Description

Vulnerability Summary (3–5 sentences)

Describe what component is affected, how it can be exploited, and what the impact is.

Affected Component / Module / Endpoint

Specify the exact component, API endpoint, function, module, or service containing the vulnerability.

Discovery Method

Describe how the vulnerability was discovered (manual testing, automated scanning, code review, fuzzing, etc.).

Impact Assessment

Describe the potential impact on confidentiality, integrity, and availability. Include any potential for data breach, system compromise, lateral movement, or service disruption.

Suggested Remediation / Mitigation (if known)

If you have identified a possible fix or workaround, describe it here.

SECTION 4 - Vulnerability Chain (if applicable)

Complete this section only if the reported vulnerabilities are related or can be chained together to achieve a greater impact than any single vulnerability alone. For example: Vulnerability #1 (weak cipher - CWE-327) combined with Vulnerability #2 (reversible password encoding - CWE-257) enables full credential recovery.

Chain Description

Explain how the vulnerabilities relate to each other and how they can be combined. Reference each vulnerability by its number (Vulnerability #1, #2, etc.) as defined in Section 3.

Combined Impact

Describe the overall impact when the vulnerabilities are exploited together. How does the chained exploitation differ from exploiting each vulnerability individually?

Combined CVSS v3.1 Score (if different from individual scores)

If the chained exploitation results in a higher severity than individual vulnerabilities, provide the combined CVSS vector string and score. Use the Online calculator: https://www.first.org/cvss/calculator/3.1

Vector String	Base Score	Severity
https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:A/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N	*5.7*	*Medium*

None: 0.0	Low: 0.1–3.9	Medium: 4.0–6.9	High: 7.0–8.9	Critical: 9.0–10.0

Exploitation Flow / Attack Path

Describe the step-by-step attack path using the chained vulnerabilities. e.g., Step 1: Exploit Vuln #1 to obtain encrypted credentials → Step 2: Exploit Vuln #2 to reverse the encryption → Step 3: Use recovered credentials for access.

SECTION 5 – Proof-of-Concept (PoC) Code & Evidence

⚠ HANDLING INSTRUCTIONS - This section contains or references exploit code. All PoC materials must be handled as TLP:AMBER and transmitted only through encrypted channels (PGP/GPG). The PXS-LU-CSIRT will never redistribute PoC code without explicit authorisation from the reporter. PoC code will be used solely for vulnerability verification and vendor coordination purposes.

Reproduction Steps

Provide detailed step-by-step instructions to reproduce the vulnerability. Include the environment setup, prerequisites, tools required, and exact sequence of actions.

PoC Type

Select all that apply:

- ☐ Script / Code (attached separately)
- ☐ HTTP requests / cURL commands
- ☐ Video recording (attached separately)
- ☐ Log files / Output (attached separately)
- ☐ Command-line instructions (included below)
- ☐ Screenshots (attached separately)
- ☐ Network capture / PCAP (attached separately)
- ☐ Other (specify below)

PoC Code / Commands (inline)

If the PoC is short enough to include here, paste the code or commands below. For longer scripts, attach them as separate files and reference the filenames here. Use a monospaced format if possible.

PoC Language / Runtime

e.g., Python 3.11, Bash, PowerShell, JavaScript/Node.js, C, cURL, Burp Suite, etc.

Dependencies / Libraries Required

List any packages, tools, or libraries required to run the PoC (e.g., requests, pycryptodome, nmap).

List of Attached PoC Files

List all attached files related to the PoC. For each file, provide the filename, a brief description, and file hash (SHA-256) for integrity verification.
Example:
- *exploit_weak_cipher.py - Python script to decrypt stored passwords - SHA-256: abc123...*
- *screenshot_admin_access.png - Screenshot showing admin panel access - SHA-256: def456...*

Expected Outcome

Describe what a successful exploitation looks like. What output, access, or state change should the PXS-LU-CSIRT observe when the PoC is executed successfully?

Test Environment

Describe the environment used for testing (OS, browser, software version, network config, etc.). This helps the PXS-LU-CSIRT reproduce the vulnerability accurately.

SECTION 6 - Credit, Attribution & Preferences

Credit Line for CVE Entry / Advisory

How should the discoverer(s) be credited in the public CVE entry and advisory? e.g., "John Doe, ACME Security" or "Anonymous"

Embargo / Disclosure Preference

☐ Standard 90-day coordinated disclosure
☐ Extended embargo (>90d - specify reason below)
☐ Immediate disclosure (critical, actively exploited)
☐ No preference - defer to PXS-LU-CSIRT policy

Reason for extended embargo (if applicable):

Additional References / Links

Any relevant references, blog posts, existing advisories, patches, or external links.

Submission Instructions

Please send this completed form along with any attached PoC files, screenshots, and supporting evidence to:

csirt (at) proximus (dot) lu

PGP/GPG encryption is strongly recommended (PGP KeyID: 6E2EA9F8).

SECTION 7 - PXS-LU-CSIRT Coordinated Vulnerability Disclosure Process

When PXS-LU-CSIRT receives a vulnerability report from a third-party reporter (security researcher, customer, partner) and the reporter is unable or unwilling to contact the vendor directly, PXS-LU-CSIRT may act as coordinator. The coordinator role involves facilitating information sharing between reporter and vendor, synchronizing timelines, and - when multiple vendors are affected - orchestrating a joint disclosure.

- Step 1- Receive and triage the report: Assign a tracking ID. Validate the vulnerability independently before any further action. Send acknowledgement to the reporter within 24–48 hours. Confirm the tracking ID, the PXS-LU-CSIRT CVD process, and the expected timeline. Classify the communication TLP:AMBER.
- Step 2 - Identify affected vendor(s): Determine whether one or multiple vendors are impacted. For multi-vendor cases, coordinate timeline synchronization
- Step 3 - Notify each vendor as PXS-LU-CSIRT. Do not reveal the reporter's identity without explicit consent.
- Step 4 - Request a CVE Identifier: in parallel with vendor notification (or shortly after), request a CVE identifier through the appropriate channel.
- Step 5 - Synchronize timelines: The 90-day disclosure clock starts from the date the full technical details are shared with the vendor. Coordinate extensions in writing with all parties.
- Step 6 - Keep the reporter informed and get in touch with the manufacturer: Provide regular status updates (at minimum on Day 14, Day 45, Day 80). If the vendor is unresponsive, inform the reporter and proceed with the escalation path: first the Follow-Up to Manufacturer (No Response) then - if still unresponsive after 14 additional days - escalation to CIRCL.
- Step 7 - Coordinated public disclosure: Synchronize publication with all parties. Ensure the reporter receives appropriate credit in the advisory and CVE entry.

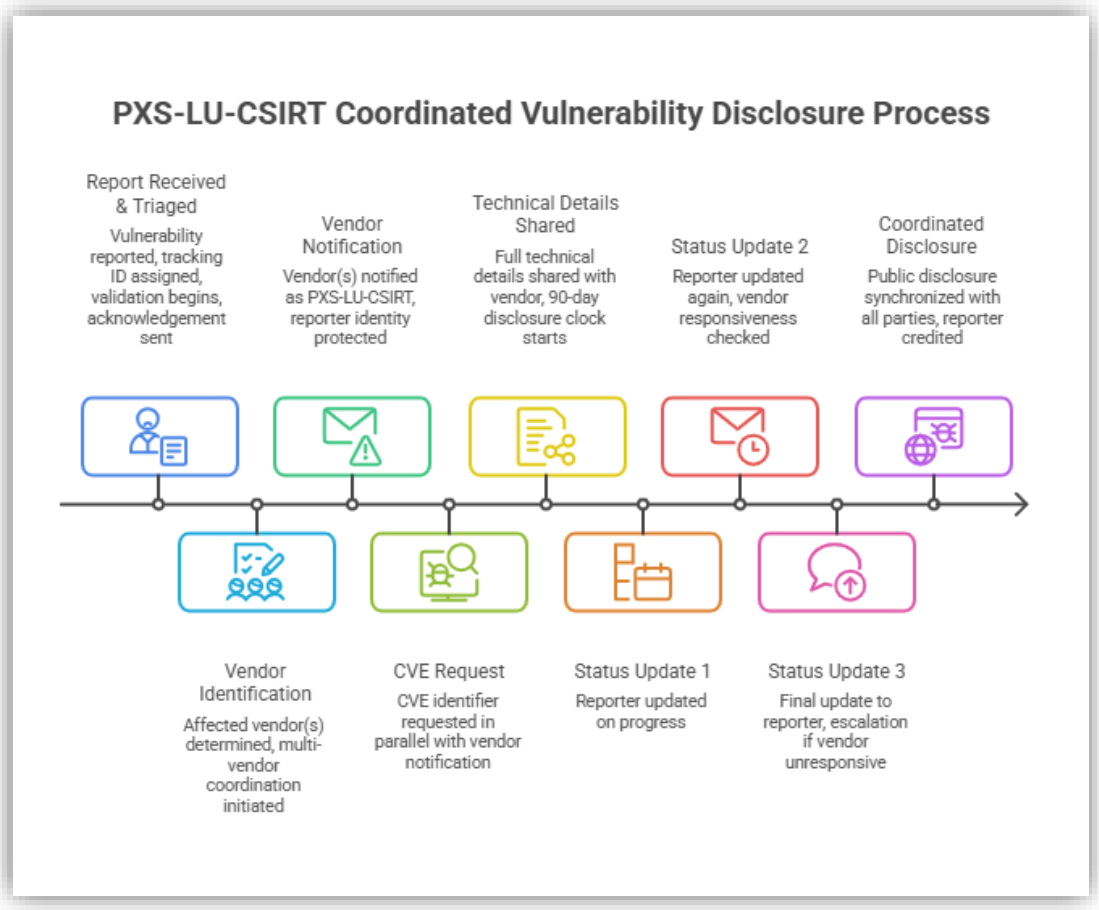

Figure 1 - PXS-LU-CSIRT Coordinated Vulnerability Disclosure Process