e1st / **CVE-2025-56015** Public

Code Issues Pull requests Actions Projects Security and quality

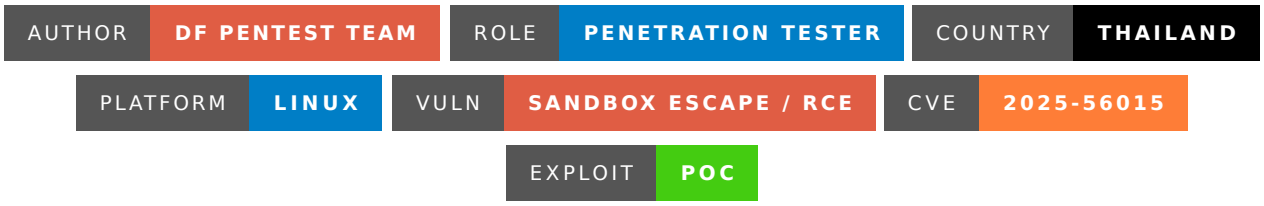
1 Branch 0 Tags Go to file Go to file Code

e1st Update vulnerable target to v1.2.13 b9bfc87 · last week

genieacs-vuln-arm64	Update vulnerable target to …	last week
.gitignore	poc and docker vuln env	last week
README.md	Update vulnerable target to …	last week
docker-compose.yml	poc and docker vuln env	last week
exploit.py	poc and docker vuln env	last week
pic.jpeg	poc and docker vuln env	last week

README

CVE-2025-56015 GenieACS RCE



Overview

Proof-of-Concept exploit for **CVE-2025-56015** a critical sandbox escape and code injection vulnerability in **GenieACS** allowing arbitrary JavaScript execution in the provisioning context. By leveraging the `declare.constructor.constructor` prototype chain, an attacker can escape the restricted JavaScript environment, access the underlying Node.js `child_process` and `net` modules, and achieve full **remote code execution (RCE)**.

Exploit type: Remote
Authentication: Requires network access to GenieACS NBI (Port 7557) and ACS (Port 7547)
Impact: Full server compromise, reverse shell execution, network pivoting

Environment Information

- **Test Environment:** GenieACS Docker Image
- **Tested Version:** GenieACS v1.2.13
- **Untested Versions:** GenieACS v1.2.14 - v1.2.16+ (or Latest version, behavior might differ or be patched)

Vulnerability Details

Vulnerability 1: UnAuth Send API

The NBI endpoint on Port 7557 allows unauthenticated access to retrieve sensitive data. Example API endpoints include:

Get Users Data:

```
$ curl 'http://myhost:7557/users/'
```

Response includes sensitive information such as password hashes and salts.

Get Files List:

```
$ curl 'http://myhost:7557/files/'
```

Get Provisions Details:

```
$ curl 'http://myhost:7557/provisions/'
```

Additionally, files can be downloaded directly from the file server (Port 7567) without authentication:

```
wget 'http://myhost:7567/SCR-20250623-qjoi.png'
```

Vulnerability 2: JS Sandbox Escape (Pre/Post Auth RCE)

A test debug endpoint inside a sandbox restricts direct access to the Node.js `process` object. However, bypassing this restriction is possible via object traversal, specifically using the constructor chain (e.g., `constructor.constructor`).

By calling this on available objects (like `declare`), attackers can escape the sandbox context and gain access to the global `process` object. This grants the ability to read environment variables and execute arbitrary commands.

Attack Vectors for RCE

1. **Post-Auth RCE:** Provision scripts can be executed directly via the API endpoint `/api/devices/{device_id}/tasks` to achieve a shell. Requests sent through the NBI alone are queued but not executed immediately.
2. **Pre-Auth RCE (Unauthenticated RCE):** Using the NBI API, an attacker can create provisions and presets without authentication. The preset serves to run provision scripts automatically when a device sends an event through the CWMP interface.

   Steps to Exploit:

   - **Step 1:** Create a malicious provision via `PUT /provisions/{provision_name}`.
   - **Step 2:** Create a preset mapping to the provision via `PUT /presets/{preset_name}` (linking it to an event like "Periodic").
   - **Step 3:** Simulate a device by sending a Device Inform SoapXPC message to the CWMP interface to trigger the event and activate the preset. Subsequent inform requests (like event "2 PERIODIC") will execute the payload.

Features of This PoC

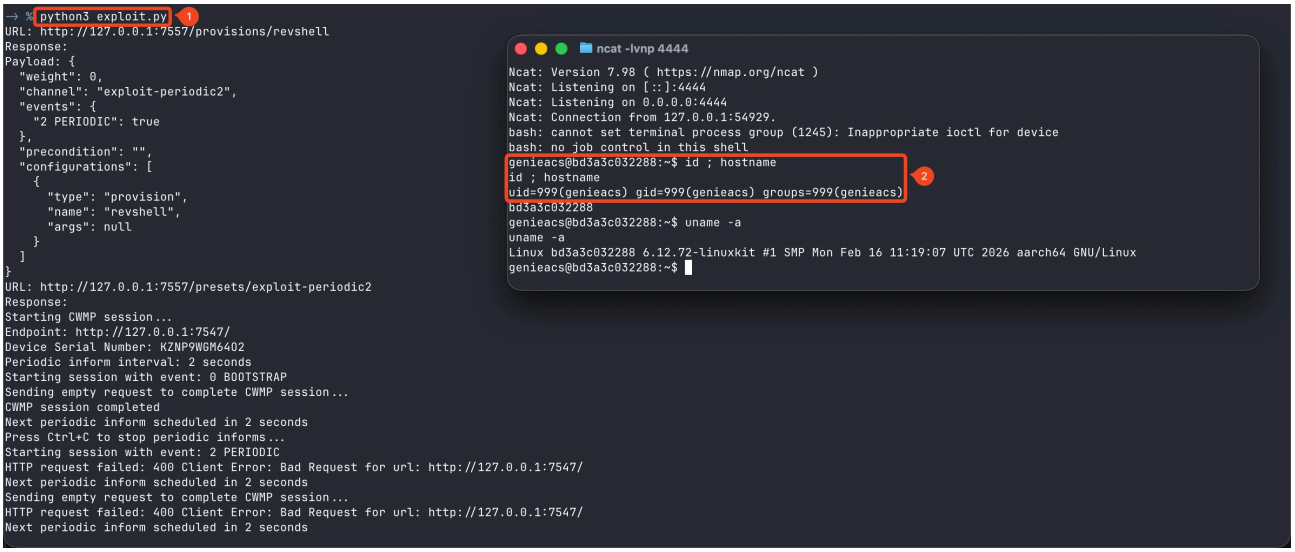
- Automatic malicious Provision creation via NBI API
- Automatic Preset configuration mapping
- CPE device simulation and automated CWMP wrapper
- Reliable Node.js Sandbox Escape using `declare.constructor.constructor`
- Reverse shell payload execution targetting Linux hosts

Usage

Prior to running the exploit, update the `ACS_URL`, `NBI_URL`, `LHOST`, and `LPORT` configurations directly within `exploit.py` to match your environment and attacker machine settings.

```
# Start listener on your attacker machine
nc -lvnp 4444

# Run the exploit
python3 exploit.py
```



Requirements

- Python 3.6+
- `requests` library (`pip install requests`)

Legal & Ethical Notice

This code is provided for educational and authorized security testing purposes only. Unauthorized use against systems you do not own or have explicit permission to test is illegal and unethical.

References

- GenieACS Documentation
- CVE-2025-56015 Advisory

Credits

Vulnerabilities discovered and exploit developed by:

- Thanasin Luangpipat
- Natchanon Jaengsuwan
- Navapon Premkasem
- Suebpong Sittichotpong

From **Datafarm Co., Ltd.**

Releases

No releases published

Packages

No packages published

Contributors 1

e1st E1st

Languages

Python 67.0% **Dockerfile** 17.2% **Go Template** 15.0% **Shell** 0.8%