Pi-hole - Multiple Vulnerabilities

Feb 18, 2021
4 minutes read

TL;DR

1 - Reflected Cross-site Scripting: A remote user is able to inject arbitrary web script or HTML due to incorrect sanitization of Origin header and achieve a Reflected Cross-Site Scripting attack against other users and steal the session cookie.
2 - Session Fixation: Session fixation vulnerability allows remote attackers to hijack web sessions via the PHPSESSID parameter.

Product Information

- **Product Name:** Pi-hole
- **Vulnerable Version:** 5.0, 5.1, 5.1.1
- **Organization Name:** Pi-hole, LLC
- **Product Web Page:** https://pi-hole.net/
- **Email:** disclosure@pi-hole.net
- **Vulnerability Disclosure Info Web Page:** https://pi-hole.net/contact/

Pi-hole is a Linux network-level advertisement and Internet tracker blocking application which acts as a DNS sinkhole and optionally a DHCP server, intended for use on a private network.
During a security auditing of the product, I have found two issues. The first is a Reflected XSS, the second is a Session Fixation. Those issues can be combined to steal a valid administrative session.

The fixes were implemented by the owners after my vulnerability notification.
It is recommended to apply the latest update Pi-hole FTL v5.7 and Web v5.4: https://pi-hole.net/2021/02/16/pi-hole-ftl-v5-7-and-web-v5-4-released/

Vulnerability Details

1 - Reflected Cross-site Scripting - Improper Neutralization of Input During Web Page Generation (Cross-site Scripting) - CWE-79

- **Summary:** A remote user is able to inject arbitrary web script or HTML due to incorrect sanitization of user-supplied data and achieve a Reflected Cross-Site Scripting attack against other users and steal the session cookie.
- **Prerequisites:** No special configuration is required to reproduce the issue.
- **CVE and CVSS Score:** CVE-2020-35592 | **5.4** (Medium)

Step-by-step instructions and PoC

An attacker is able to craft an HTTP request and send it to a user, authenticated to Pi-hole, to inject arbitrary web script or HTML into the HTTP **Origin** header, which reflect the user input.

Affected Endpoints

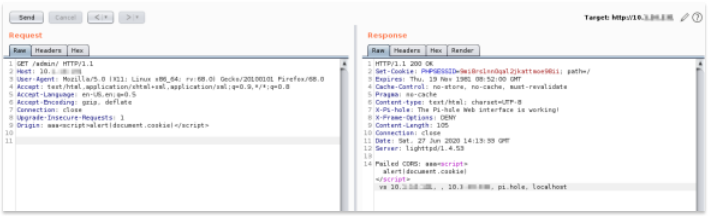
- **URL:** http://hostname/admin
- **HTTP Parameter:** Origin header

Below are the evidences with the vulnerability details and the payloads used.

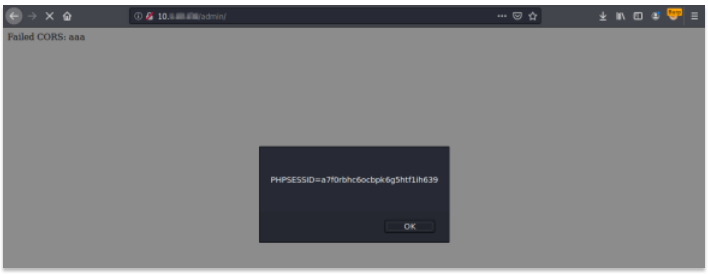
Payload used to exploit the vulnerability:

```
GET /admin/ HTTP/1.1
Host: hostname
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefo
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Origin: aaa<script>alert(document.cookie)</script>
```

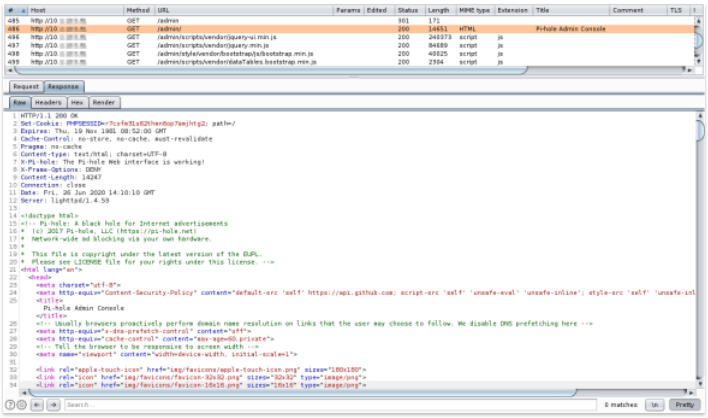
The user is invited to click on a malicious page that have JavaScript code to execute the following HTTP request on the browser of the victim:



The XSS payload gets reflected to the screen.



It is important to note that the **PHPSESSID** session cookie does not have the **HttpOnly** flag, when first assigned to the user:



Security Impact

By exploiting this issue an attacker is able to target an application user with several type of direct or indirect impacts such as, credentials stealing, integrity compromising and different type of phishing attacks. This type of reflected XSS does require user interaction.

2 - Session Fixation - CWE-384

- **Summary:** Session fixation vulnerability allows remote attackers to hijack web sessions via the PHPSESSID parameter.
- **Prerequisites:** No special configuration is required to reproduce the issue.
- **CVE and CVSS Score:** CVE-2020-35591 | **5.4** (Medium)

Step-by-step instructions and PoC

The application does not generate new session cookie after the user is logged in. A malicious user is able to create a new session cookie value and inject it to a victim. After the victim logs in, the injected cookie becomes valid, giving the attacker access to the user's account through the active session.

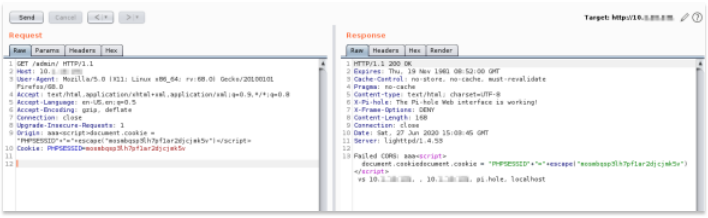
Affected Endpoints

- **URL:** http://hostname/admin/index.php?login
- **HTTP Parameter:** PHPSESSID cookie

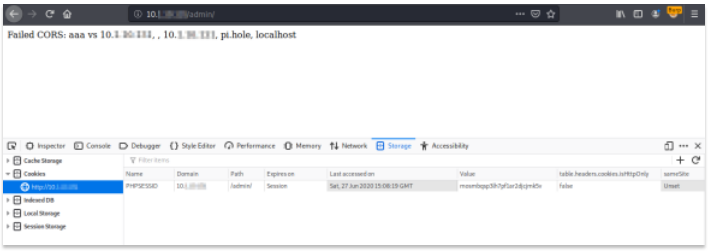
Below are the evidences with the vulnerability details and the payload used.

By leveraging the previously found XSS vulnerability, the malicious user can create a request to inject an arbitrary PHPSESSID value to the victim browser:

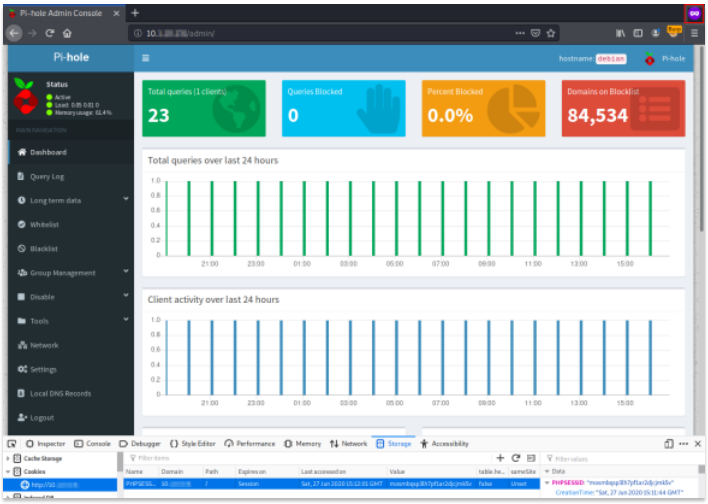
```
GET /admin/ HTTP/1.1
Host: hostname
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefo
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Origin: aaa<script>document.cookie = "PHPSESSID"+"="+escape("mosmbqsp3lh7p
Cookie: PHPSESSID=mosmbqsp3lh7pf1ar2djcjmk5v
```



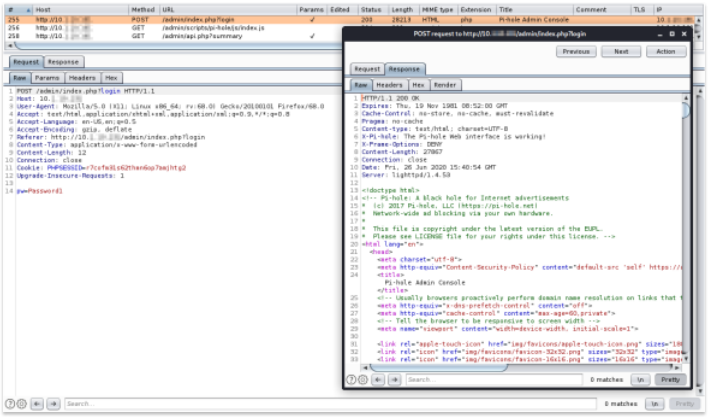
The following is the response of the payload and the new cookie added to the browser cookie storage:



Then, when the user performs a login, the new PHPSESSID becomes valid. The attacker can add it to its browser to autenticate to the web application:



Indeed, after a user login, there is no a new generation of PHPSESSID:



Security Impact

By exploiting this issue, an attacker is able to obtain a valid user session of the application. This type of attack does require user interaction.

Timeline

- **15/12/2020**: First disclosure via e-mail to disclosure@pi-hole.net
- **22/12/2020**: Acknowledge e-mail from the product owner!
- **31/12/2020**: Trying to collaborate with product owner by suggesting some remediation code. I'm really glad to collaborate with them!
- **16/02/2020**: Released the fixed versions and added a news with advisory and credits: https://pi-hole.net/2021/02/16/pi-hole-ftl-v5-7-and-web-v5-4-released/
- **26/02/2021**: NVD scored CVE-2020-35591 and CVE-2020-35592 as **5.4** (Medium).

NeDi 1.9C - Multiple Vulnerabilities

WordPress Plugin WP File Manager - Reflected XSS