**Week 04**

**Penetration Testing Report**

**Introduction**

This report document hereby describes the proceedings and results of a Black Box security assessment conducted against the **Week 04 Labs**. The report hereby lists the findings and corresponding best practice mitigation actions and recommendations.

**1. Objective**

The objective of the assessment was to uncover vulnerabilities in the **Week 04 Labs** and provide a final security assessment report comprising vulnerabilities, remediation strategy and recommendation guidelines to help mitigate the identified vulnerabilities and risks during the activity.

**2. Scope**

This section defines the scope and boundaries of the project.

| **Application Name** | Lab 1-Exchangeable Image File Format , Lab2-[Open Redirect](https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/index.php) |
| --- | --- |

**3. Summary**

Outlined is a Black Box Application Security assessment for the **Week 04 Labs**.

**Total number of Sub-labs: {count} Sub-labs**

| **High** | **Medium** | **Low** |
| --- | --- | --- |
| **3** | **2** | **4** |

**High - Number of Sub-labs with hard difficulty level**

**Medium - Number of Sub-labs with Medium difficulty level**

**Low - Number of Sub-labs with Easy difficulty level**

# 1. Exchangeable Image File Format

# 1.1. [Let's PII!](https://www.bugbountyhunter.org/internship_labs/HTML/exif_lab/lab_1/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Let’s PII! | **Low** |
| **Tools Used** | |
| Web Browser | |
| **Vulnerability Description** | |
| When a user uploads an image in example.com, the uploaded image’s EXIF Geolocation Data does not gets stripped. As a result, anyone can get sensitive information of example.com users like their Geolocation, their Device information like Device Name, Version, Software & Software version used etc. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
| An attacker could download public group logos and find sensitive metadata. Some phones attach metadata with the latitude/longitude of where the photo was taken which could leak important information, and it's just best practice as well to strip all metadata from images when uploaded.This vulnerability violates the privacy of a User and shares sensitive information of the user who  uploads an image on the vulnerable website. | |
| **Suggested Countermeasures** | |
| To prevent EXIF Data Exposure we can do the following:Disable geotagging on the digital device you use to take photographs,Use an image processing software or EXIF data remover tool to delete metadata | |
| **References** | |
| <https://beaglesecurity.com/blog/vulnerability/exif-data-information-leakage.html>  https://github.com/CartoDB/cartodb/issues/15241 | |

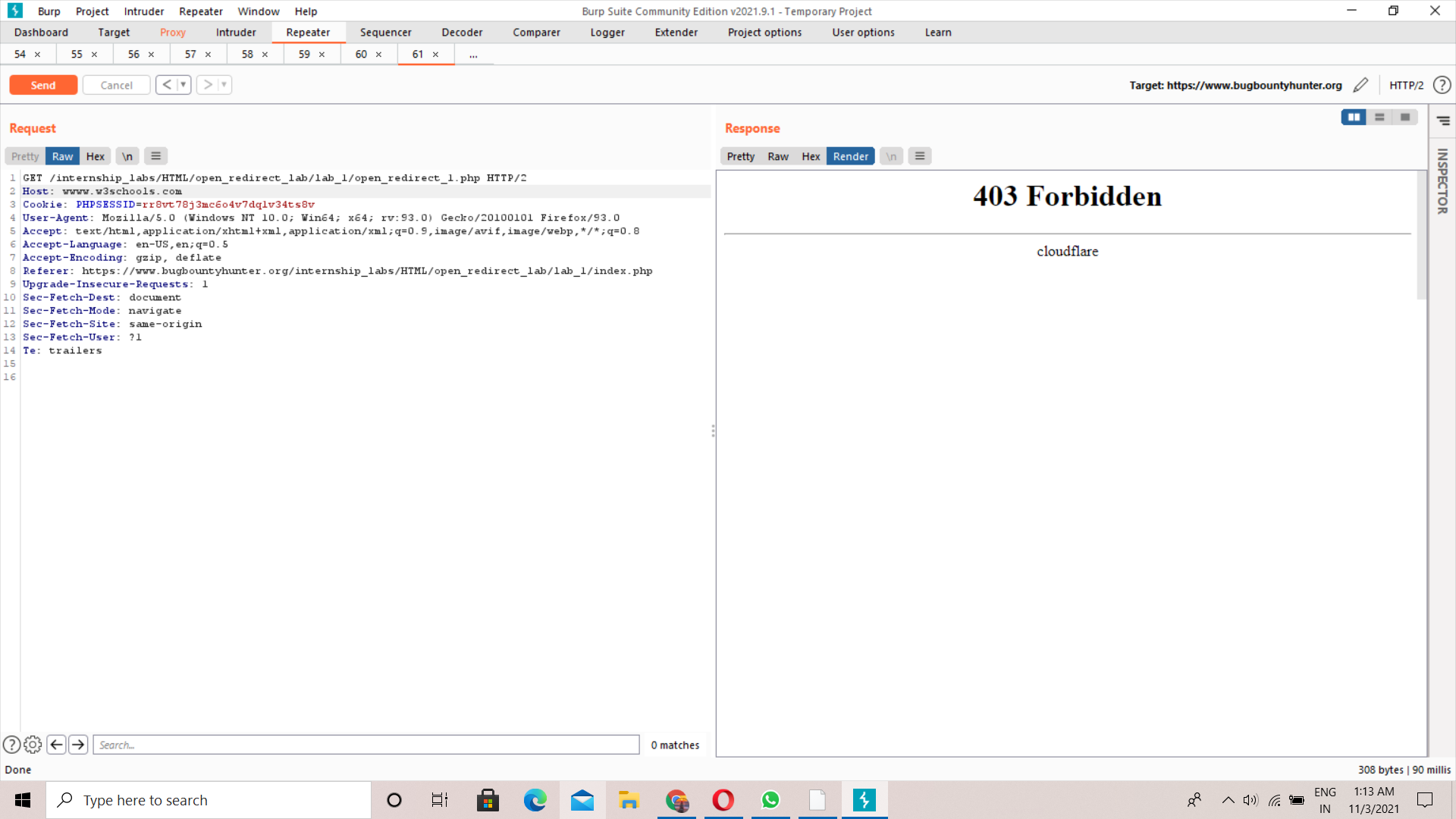
# Proof of Concept

# 2. [Open Redirect](https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/index.php)

# 2.1. [A Simple Host!](https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_1/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| A Simple Host! | **Low** |
| **Tools Used** | |
| Web Browser,Burp Suite | |
| **Vulnerability Description** | |
| Modifying and redirecting hosts file entries is a common tactic used by malware to silently send users to an unwanted website. Often, this is used to expose the user's machine to web-based attacks, as well as exposing the user to unsolicited and/or malicious content. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
| The user may be subjected to phishing attacks by being redirected to an untrusted page. The phishing attack may point to an attacker controlled web page that appears to be a trusted web site. The phishers may then steal the user's credentials and then use these credentials to access the legitimate web site. | |
| **Suggested Countermeasures** | |
|  | |
| **References** | |
| <https://cwe.mitre.org/data/definitions/601.html>  https://www.virtuesecurity.com/kb/url-redirection-attack-and-defense/  https://www.sciencedirect.com/topics/computer-science/vulnerable-host | |

# Proof of Concept



# 

# 

# 2.2. [Story Of A Beautiful Header!](https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_2/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Story Of A Beautiful Header! | **Low** |
| **Tools Used** | |
| web browser,burp suite | |
| **Vulnerability Description** | |
| A web application accepts a user-controlled input that specifies a link to an external site, and uses that link in a Redirect. This simplifies phishing attacks.An http parameter may contain a URL value and could cause the web application to redirect the request to the specified URL. By modifying the URL value to a malicious site, an attacker may successfully launch a phishing scam and steal user credentials. Because the server name in the modified link is identical to the original site, phishing attempts have a more trustworthy appearance. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
| The user may be redirected to an untrusted page that contains malware which may then compromise the user's machine. This will expose the user to extensive risk and the user's interaction with the web server may also be compromised if the malware conducts keylogging or other attacks that steal credentials, personally identifiable information (PII), or other important data. | |
| **Suggested Countermeasures** | |
| the application should avoid incorporating user-controllable data into redirection targets. Remove the redirection function from the application, and replace links to it with direct links to the relevant target URLs.Maintain a server-side list of all URLs that are permitted for redirection. Instead of passing the target URL as a parameter to the redirector, pass an index into this list.Don’t trust the host header. If you must use the host header as a mechanism for identifying the location of the web server, it’s highly advised to make use of a whitelist of allowed hostnames. | |
| **References** | |
| <https://portswigger.net/web-security/host-header>  <https://book.hacktricks.xyz/pentesting-web/open-redirect>  <http://software-security.sans.org/blog/2010/03/25/top-25-series-rank-23-open-redirect> | |

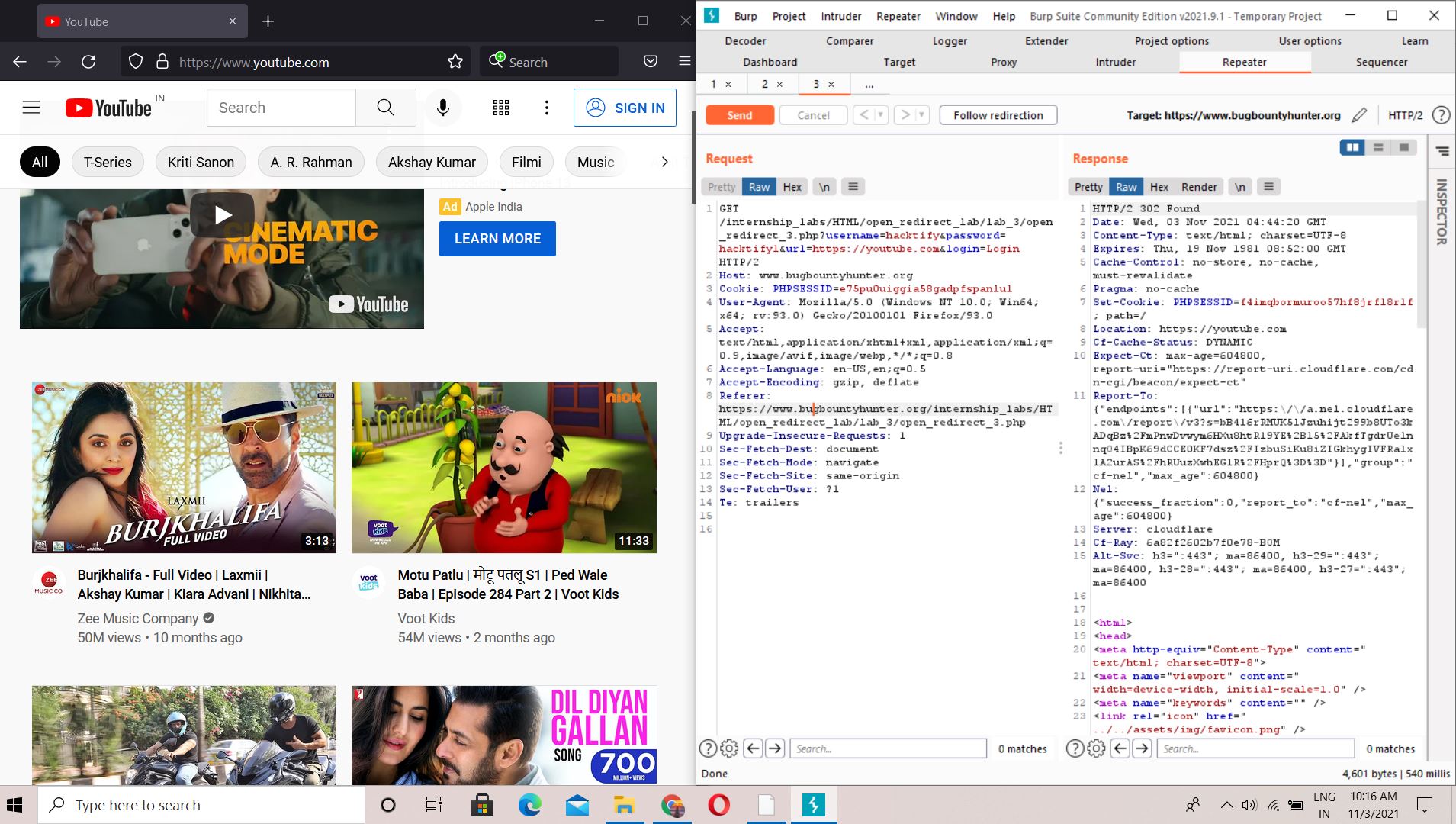
# Proof of Concept

# 

# 2.3. [Sanitize Params!!](https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_3/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Sanitize Params! | **Medium** |
| **Tools Used** | |
| Web Browser ,Burp Suite | |
| **Vulnerability Description** | |
| In this attack, the hacker exploited the webpage by changing the value of url parameter.Unvalidated redirects and forwards are possible when a web application accepts untrusted input that could cause the web application to redirect the request to a URL contained within untrusted input. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
| By modifying untrusted URL input to a malicious site, an attacker may successfully launch a phishing scam and steal user credentials. | |
| **Suggested Countermeasures** | |
| Any attack like this – where user input is echoed back to the Web page – *requires* that data be sanitized before output. Because we are defending against Web page injection, you need to escape HTML special characters including tag brackets () and the entity ampersand (&) so they are not rendered by the browser. | |
| **References** | |
| https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated\_Redirects\_and\_Forwards\_Cheat\_Sheet.html | |

# Proof of Concept



# 2.4. [Patterns Are Important!](https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_4/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Patterns Are Important | **Medium** |
| **Tools Used** | |
| Web Browser,Burp Suite | |
| **Vulnerability Description** | |
| In this the lab the value of url parameter is changed in order to redirect the victim Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to [phishing](https://spanning.com/blog/phishing-everything-you-need-to-know/) or malware sites, or use forwards to access unauthorized pages.” | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
|  | |
| **Suggested Countermeasures** | |
| increase Resistance to Attack: Utilize strong two-way authentication for all communication between client and server. This option could have significant performance implications.Increase Resilience to Attack: Minimize the amount of logic and filtering present on the client; place it on the server instead. Use white lists on server to filter and validate client input.Verify URL patterns using regular expressions to check if they belong to valid URLs. However, malicious URLs can pass that check. Therefore, add another check where URLs are vetted against a whitelist: a list of valid URLs. | |
| **References** | |
| <https://book.hacktricks.xyz/pentesting-web/open-redirect>  <https://i.blackhat.com/USA-19/Thursday/us-19-Birch-HostSplit-Exploitable-Antipatterns-In-Unicode-Normalization.pdf>  https://owasp.org/www-community/vulnerabilities/Information\_exposure\_through\_query\_strings\_in\_url | |

# Proof of Concept

**2.5.** [**File Upload!? Redirect IT!**](https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_5/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| File Upload!? Redirect IT! | **Low** |
| **Tools Used** | |
| Web Browser,Burp Suite | |
| **Vulnerability Description** | |
| File upload is one of the most common functionalities application has to offer. This functionality, however, is implemented in many different forms based on the application’s use case. While some applications only allow uploading a profile picture and support only image-related extensions, on the other hand, some applications support other extensions based on their business case. Storing and retrieving these files on the server-side is again a huge task and required to be handled with caution.Due to the involved complexity and level of caution that is required to implement a file upload functionality, this becomes one of the interesting attack vectors and can open doors to multiple critical security vulnerabilities such as Remote Code Execution. | |
| **How It Was Discovered** | |
|  | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
| If an application allows uploading files such as HTML,svg, text and the files are executed on the “Application Endpoint” itself | |
| **Suggested Countermeasures** | |
| A file upload vulnerability can be prevented by following the below mitigation techniques:Allow only certain file extension,Set maximum file size and name length,Allow only authorized users,Make sure the fetched file from the web is an expected on,Keep your website update | |
| **References** | |
| <https://book.hacktricks.xyz/pentesting-web/open-redirect>  https://www.neuralegion.com/blog/open-redirect-vulnerabilities/ | |

# Proof of Concept

# 

# 2.6.[Same Param Twice!](https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_6/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Same Param Twice! | **Hard** |
| **Tools Used** | |
| Web Browser,Burp Suite | |
| **Vulnerability Description** | |
| Open redirection occurs when a vulnerable web page is redirected to an untrusted and malicious page that may compromise the user. Open redirection attacks usually come with a phishing attack because the modified vulnerable link is identical to the original site, which increases the likelihood of success for the phishing attack. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
| Unvalidated redirects and forwards are possible when a web application accepts untrusted input that could cause the web application to redirect the request to a URL contained within untrusted input. By modifying untrusted URL input to a malicious site, an attacker may successfully launch a phishing scam and steal user credentials. | |
| **Suggested Countermeasures** | |
| Simply avoid using redirects and forwards.If used, do not allow the URL as user input for the destination.Where possible, have the user provide short name, ID or token which is mapped server-side to a full target URL. | |
| **References** | |
|  | |

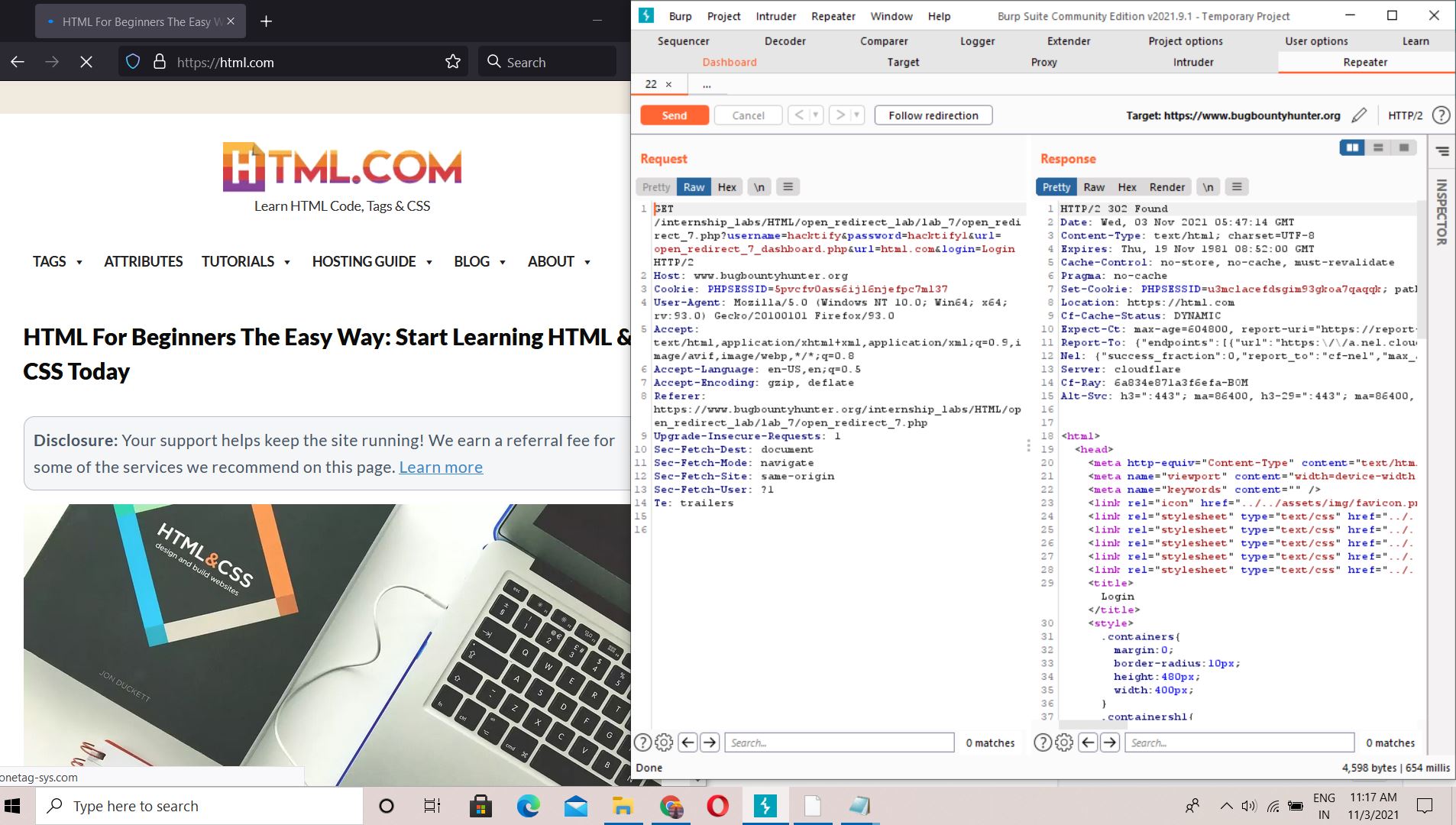
# Proof of Concept

# 

# 2.7. [Domains ? Not Always!](https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_7/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Domains ? Not Always! | **Hard** |
| **Tools Used** | |
| Web Browser,Burp Suite | |
| **Vulnerability Description** | |
| domain name in a URL is typically the only indicator for a user to recognize a legitimate website from a non-legitimate one, an attacker can abuse this trust to exploit an open redirect vulnerability on the vulnerable website, and redirect the user to a malicious page to execute further attacks, as explained in the following sections. | |
| **How It Was Discovered** | |
|  | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
| impacts can be many, and vary from theft of information and credentials, to the redirection to malicious websites containing attacker-controlled content, which in some cases even cause XSS attacks. So even though an open redirection might sound harmless at first, the impacts of it can be severe should it be exploitable. | |
| **Suggested Countermeasures** | |
| the easiest and most effective way to prevent vulnerable open redirects would be to not let the user control where your page redirects him to. If you have to redirect the user based on URLs, you should always use an ID which is internally resolved to the respective URL. If you want the user to be able to issue redirects you should use a redirection page that requires the user to click on the link instead of just redirecting them. | |
| **References** | |
| <https://book.hacktricks.xyz/pentesting-web/open-redirect>  https://cwe.mitre.org/data/definitions/601.html | |

# Proof of Concept



# 

# 

# 

# 

# 2.8. [Style Digit Symbols <3](https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_8/index.php)

| **Reference** | **Risk Rating** |
| --- | --- |
| Style Digit Symbols <3 | **Hard** |
| **Tools Used** | |
| Web Browser,Burp Suite | |
| **Vulnerability Description** | |
| Web servers may be configured to send redirects to client requests. In some cases, specially crafted queries may be used to expose internal IP addresses/ Host name. Typically, this is done by sending a  blank host header which can result in the server sending a redirect using its own IP address/ internal host name as the host name. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
|  | |
| **Consequences of not Fixing the Issue** | |
| Disclosure of Internal IP address/ Host name can be used by an attacker to exploit the server, its hosting network, etc. This helps an attacker to chain multiple issues and launch specific attacks against the internal environment of the application. | |
| **Suggested Countermeasures** | |
| There is not usually any good reason to disclose the internal IP addresses used within an organization's infrastructure. It is recommended to mask or obfuscate the private addresses. Cases where the address is being returned in HTTP header, there header should be configured to mask the private address. Depending on the type of server, there should be a way to prevent the Internal IP address/ Host name from ever being sent in 302 Redirects on the Server. However, if this is not possible, a new rule can be created & used to block any Internal IP addresses/ Host name from being exposed in a 302 redirect. | |
| **References** | |
| <https://www.site24x7.com/find-ip-address-of-web-site.html>  https://support.kemptechnologies.com/hc/en-us/articles/203522429-Mitigating-Against-Internal-IP-Address-Domain-Name-Disclosure-In-Real-Server-Redirects | |

# Proof of Concept

### 