**Week 4**

**Penetration Testing Report**

**Introduction**

This report document hereby describes the proceedings and results of a Black Box security assessment conducted against the **Week 4 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 4 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** | **Open Redirect, Exchangeable Image File Format** |
| --- | --- |

**3. Summary**

Outlined is a Black Box Application Security assessment for the **Week 4 Labs**.

**Total number of Sub-labs: 9 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. Open Redirect

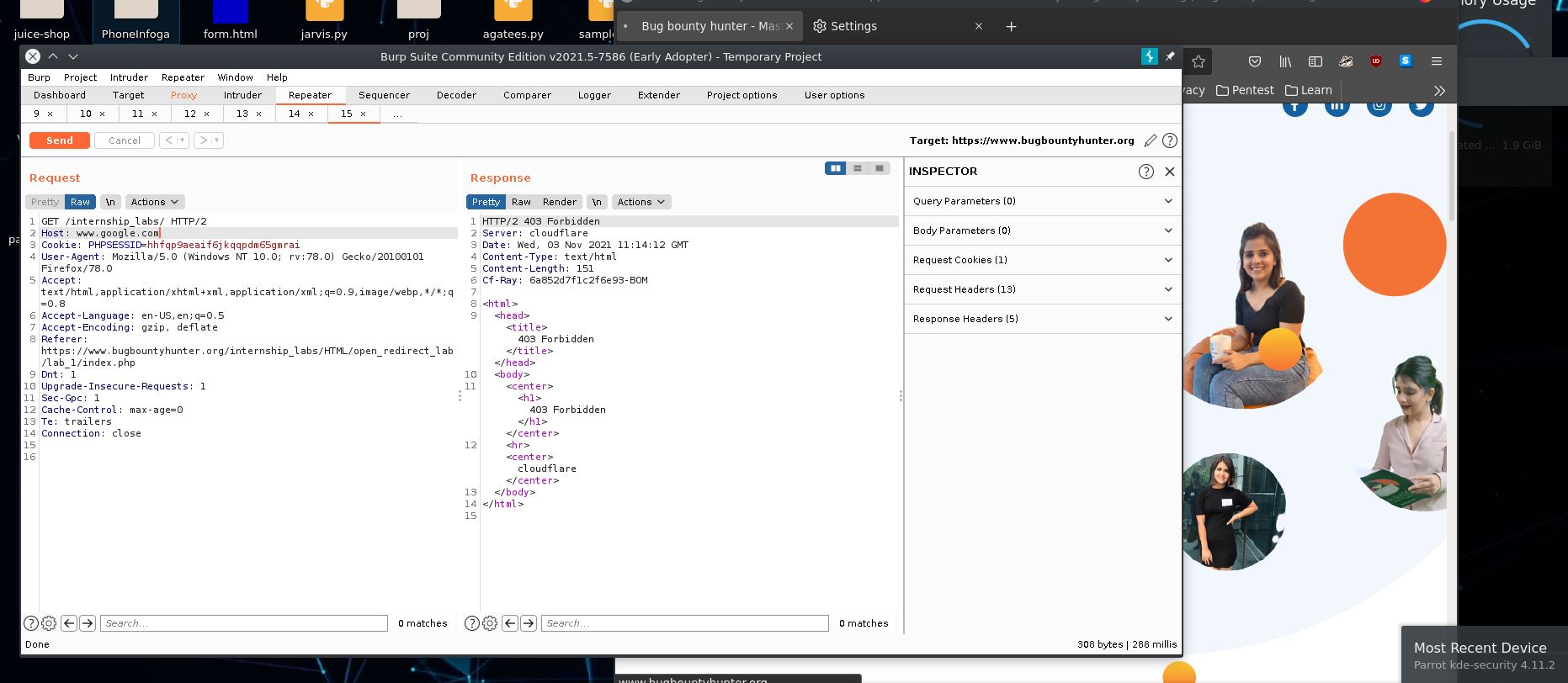
# 1.1.A Simple Host!

| **Lab Name** | **Risk Rating** |  |
| --- | --- | --- |
| A Simple Host! | **Low** |  |
| **Tools Used** | |  |
| Burp suite | |  |
| **Vulnerability Description** | |  |
| An open redirect vulnerability exists when the destination of the redirect is provided by the client and it is not filtered or validated. An attacker can construct a URL within the application that causes a redirection to an arbitrary external domain | |  |
| **How It Was Discovered** | |  |
| Manual Analysis | |  |
| **Vulnerable URLs** | |  |
| <https://www.bugbountyhunter.org/internship_labs/> | |  |
| **Consequences of not Fixing the Issue** | |  |
|  | |  |
| **Suggested Countermeasures** | |  |
| Check the Referrer When Doing Redirects | |  |
| **References** | |  |
| <https://www.hacksplaining.com/prevention/open-redirects> | |  |

# 

# Proof of Concept

**payload:** Host: www.google.com



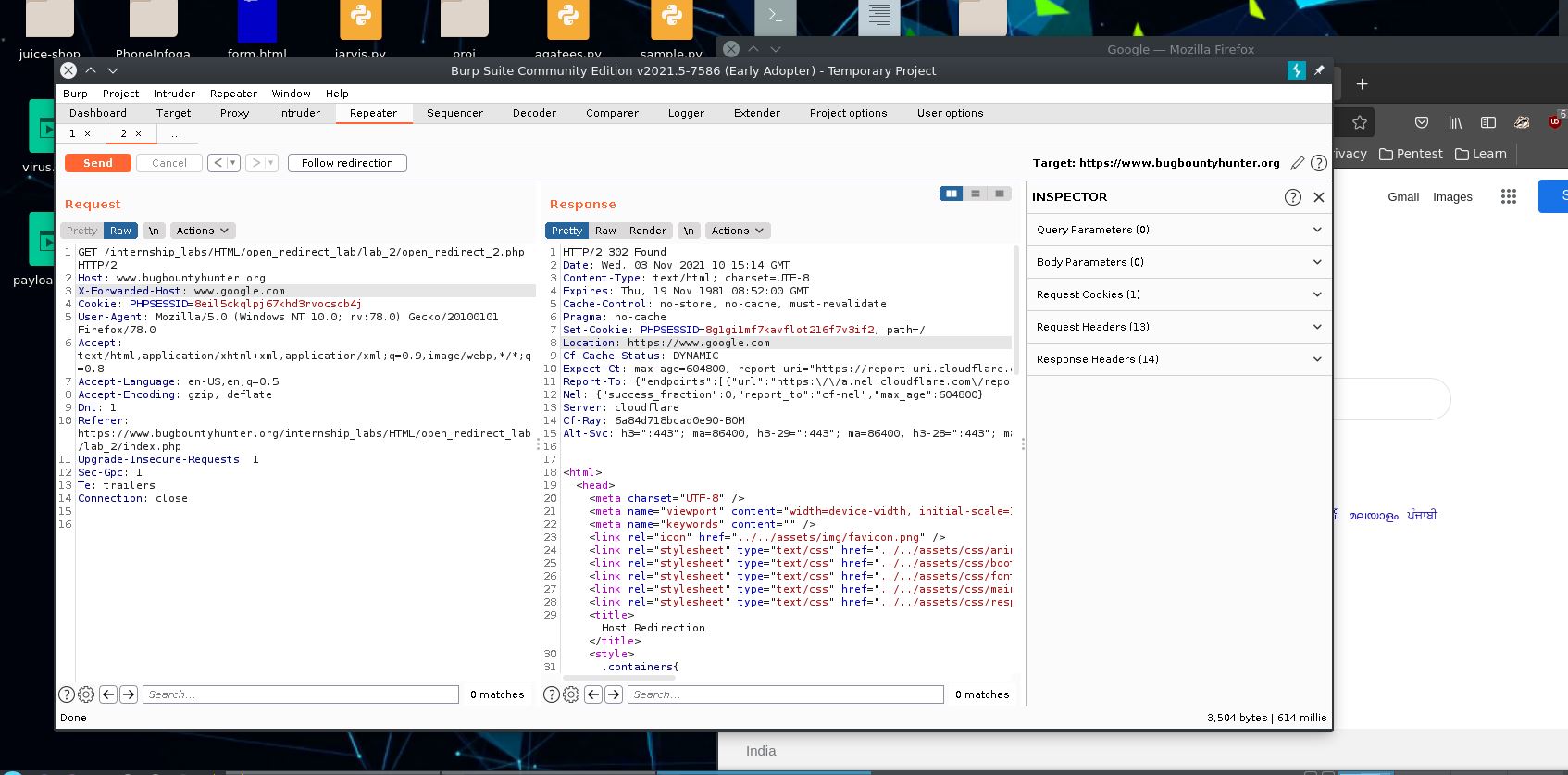
# 

# 1.2 Story Of A Beautiful Header!

| **Lab Name** | **Risk Rating** |
| --- | --- |
| Story Of A Beautiful Header! | **Low** |
| **Tools Used** | |
| Burp suite | |
| **Vulnerability Description** | |
| An open redirect vulnerability exists when the destination of the redirect is provided by the client and it is not filtered or validated. An attacker can construct a URL within the application that causes a redirection to an arbitrary external domain. This behavior can be leveraged to facilitate phishing attacks against users of the application. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
| <https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_2/open_redirect_2.php> | |
| **Consequences of not Fixing the Issue** | |
| It may lead to phishing and in this attack the victim may redirect to the attacker controlled website and the attacker can able to steal victim credentials or useful information. | |
| **Suggested Countermeasures** | |
| Check the Referrer When Doing Redirects | |
| **References** | |
| <https://www.hacksplaining.com/prevention/open-redirects> | |

# Proof of Concept

**payload:** X-Forwarded-Host: www.google.com



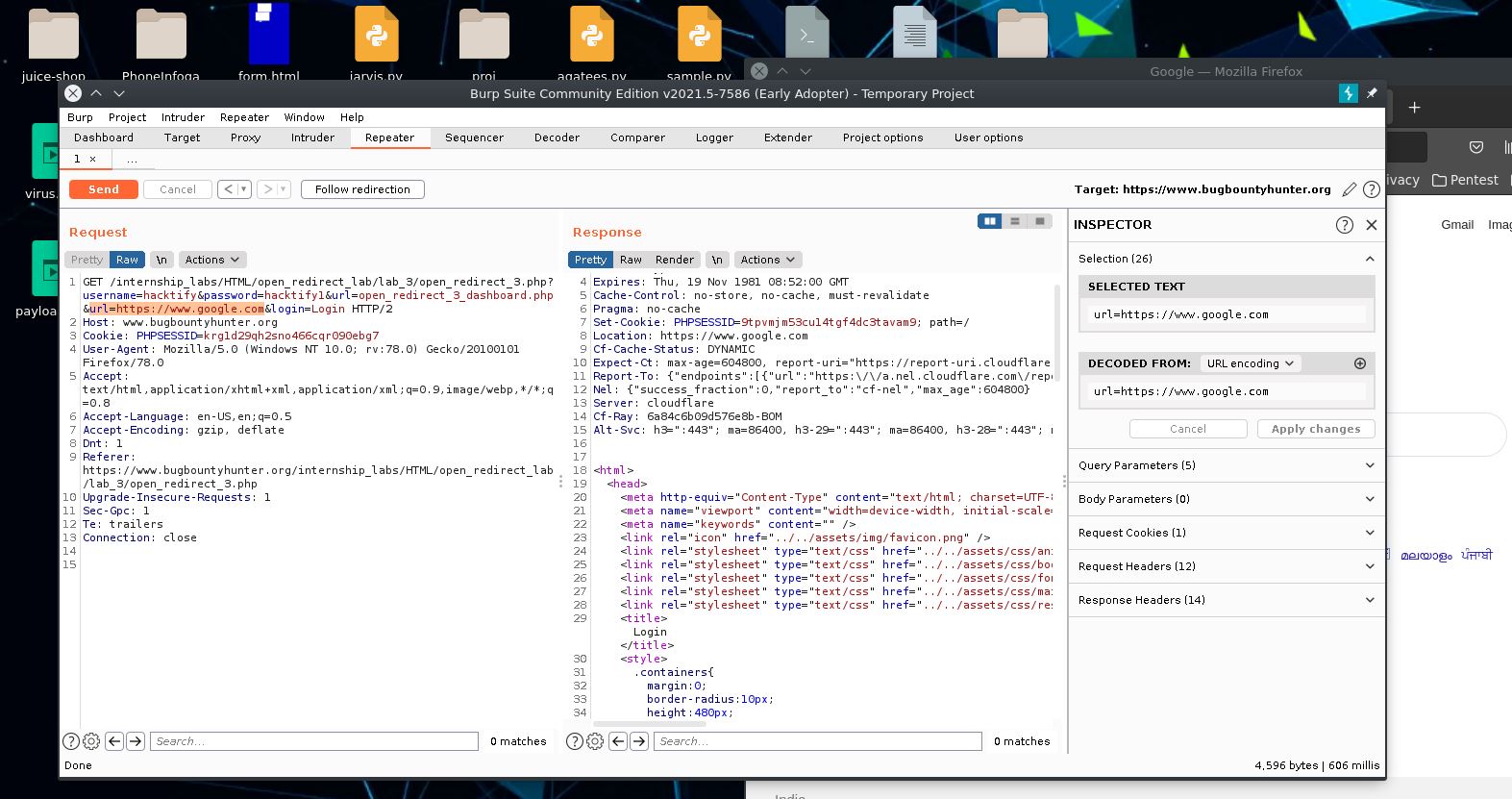
# 1.3. Sanitize Params!!

| **Lab Name** | **Risk Rating** |
| --- | --- |
| Sanitize Params!! | **Low** |
| **Tools Used** | |
| Burp suite | |
| **Vulnerability Description** | |
| An open redirect vulnerability exists when the destination of the redirect is provided by the client and it is not filtered or validated. An attacker can construct a URL within the application that causes a redirection to an arbitrary external domain. This behavior can be leveraged to facilitate phishing attacks against users of the application.  And in this vulnerability the parameters are sanitized. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
| <https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_3/open_redirect_3_dashboard.php> | |
| **Consequences of not Fixing the Issue** | |
| It may lead to phishing .the attacker can able to steal victim credentials or useful information | |
| **Suggested Countermeasures** | |
| don’t allow same parameters in two times and check the url while open url . | |
| **References** | |
| <https://portswigger.net/kb/issues/00500100_open-redirection-reflected> | |

# 

# Proof of Concept

**payload:**url=https://www.google.com



# 

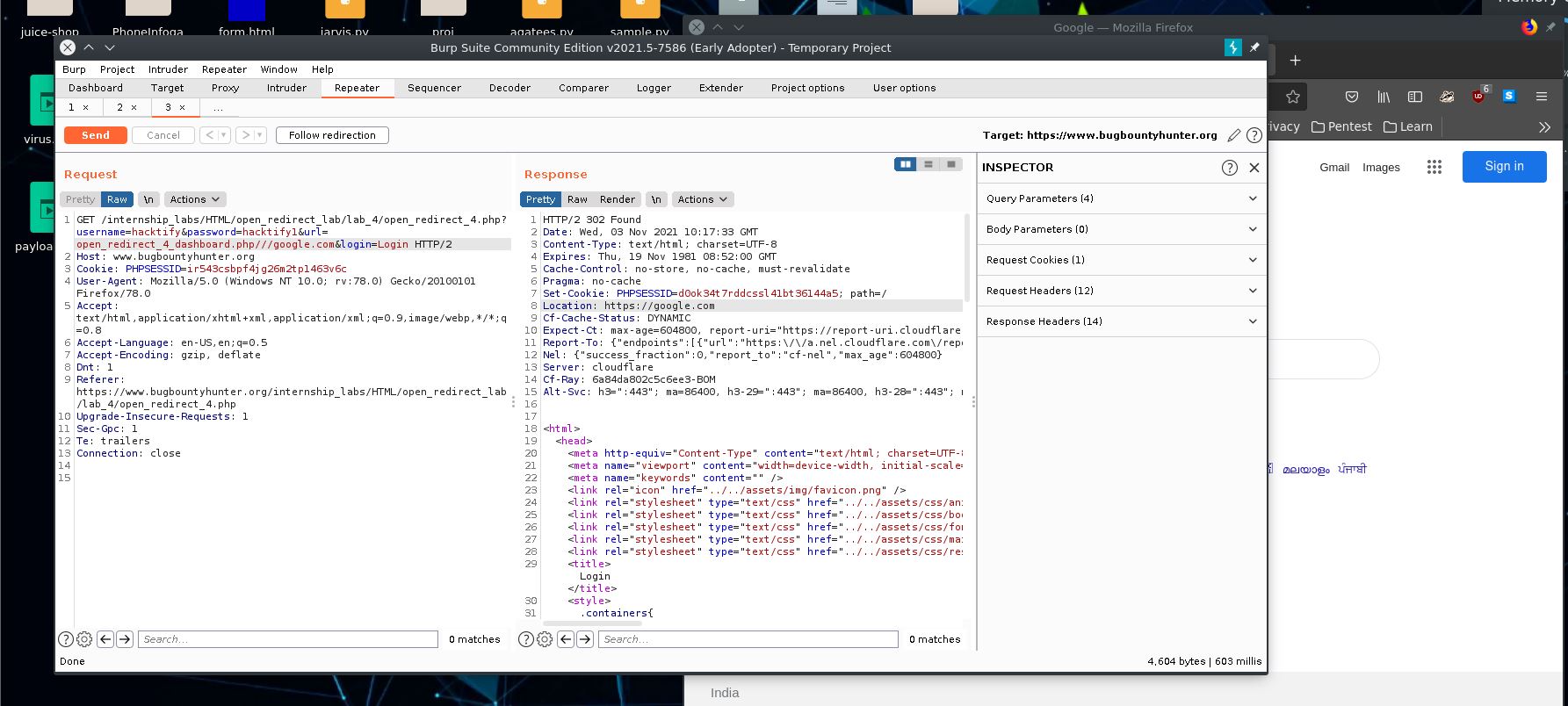
# 1.4. Patterns Are Important!

| **Lab Nme** | **Risk Rating** |
| --- | --- |
| Patterns Are Important! | **Low** |
| **Tools Used** | |
| Burp suite | |
| **Vulnerability Description** | |
| Open redirection vulnerabilities arise when an application incorporates user-controllable data into the target of a redirection in an unsafe way. An attacker can construct a URL within the application that causes a redirection to an arbitrary external domain. This behavior can be leveraged to facilitate phishing attacks against users of the application. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
| <https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_4/open_redirect_4_dashboard.php> | |
| **Consequences of not Fixing the Issue** | |
| it may lead to phishing .the attacker can able to steal victim credentials or useful information | |
| **Suggested Countermeasures** | |
| don’t allow double or triple slashes (// or ///) sanitize those symbols | |
| **References** | |
| <https://www.hacksplaining.com/prevention/open-redirects> | |

# 

# Proof of Concept

**payload:** url=open\_redirect\_4\_dashboard.php///google.com



# 1.5. File Upload!? Redirect IT!

| **Lab Name** | **Risk Rating** |
| --- | --- |
| File Upload!? Redirect IT! | **Low** |
| **Tools Used** | |
| Burp suite | |
| **Vulnerability Description** | |
| An open redirect vulnerability exists when the destination of the redirect is provided by the client and it is not filtered or validated. An attacker can construct a URL within the application that causes a redirection to an arbitrary external domain. This behavior can be leveraged to facilitate phishing attacks against users of the application. The ability to use an authentic application URL, targeting the correct domain and with a valid SSL certificate (if SSL is used), lends credibility to the phishing attack because many users, even if they verify these features, will not notice the subsequent redirection to a different domain. In short, if an attacker is able to redirect the user to a malicious website then it can be termed as Open Redirect. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
| <https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_5/open_redirect_5.php> | |
| **Consequences of not Fixing the Issue** | |
| What will be the consequences if the vulnerability is not patched? | |
| **Suggested Countermeasures** | |
| sanitize the url while passing. | |
| **References** | |
| <https://www.hacksplaining.com/prevention/open-redirects> | |

# 

# Proof of Concept

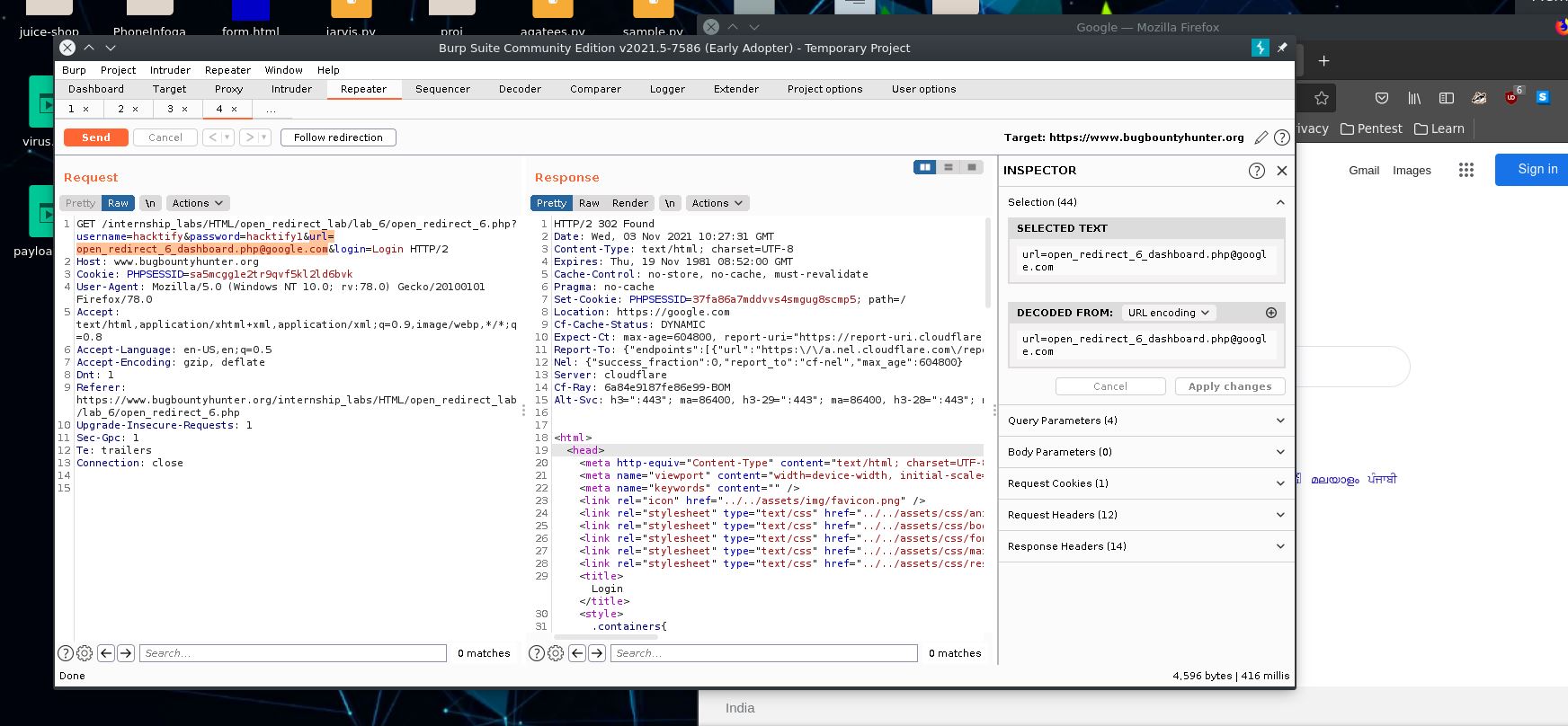
# 1.6. Same Param Twice!

| **Lab Name** | **Risk Rating** |
| --- | --- |
| Same Param Twice! | **Low** |
| **Tools Used** | |
| Tools that you have used to find the vulnerability. | |
| **Vulnerability Description** | |
| About the vulnerability and its working | |
| **How It Was Discovered** | |
| Automated Tools / Manual Analysis | |
| **Vulnerable URLs** | |
| <https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_6/open_redirect_6_dashboard.php> | |
| **Consequences of not Fixing the Issue** | |
| It may lead to phishing and account take over.the attacker can able to steal victim credentials or useful information | |
| **Suggested Countermeasures** | |
| sanitize the url. | |
| **References** | |
| <https://www.hacksplaining.com/prevention/open-redirects> | |

# 

# Proof of Concept

**payload:** url=open\_redirect\_6\_dashboard.php@google.com



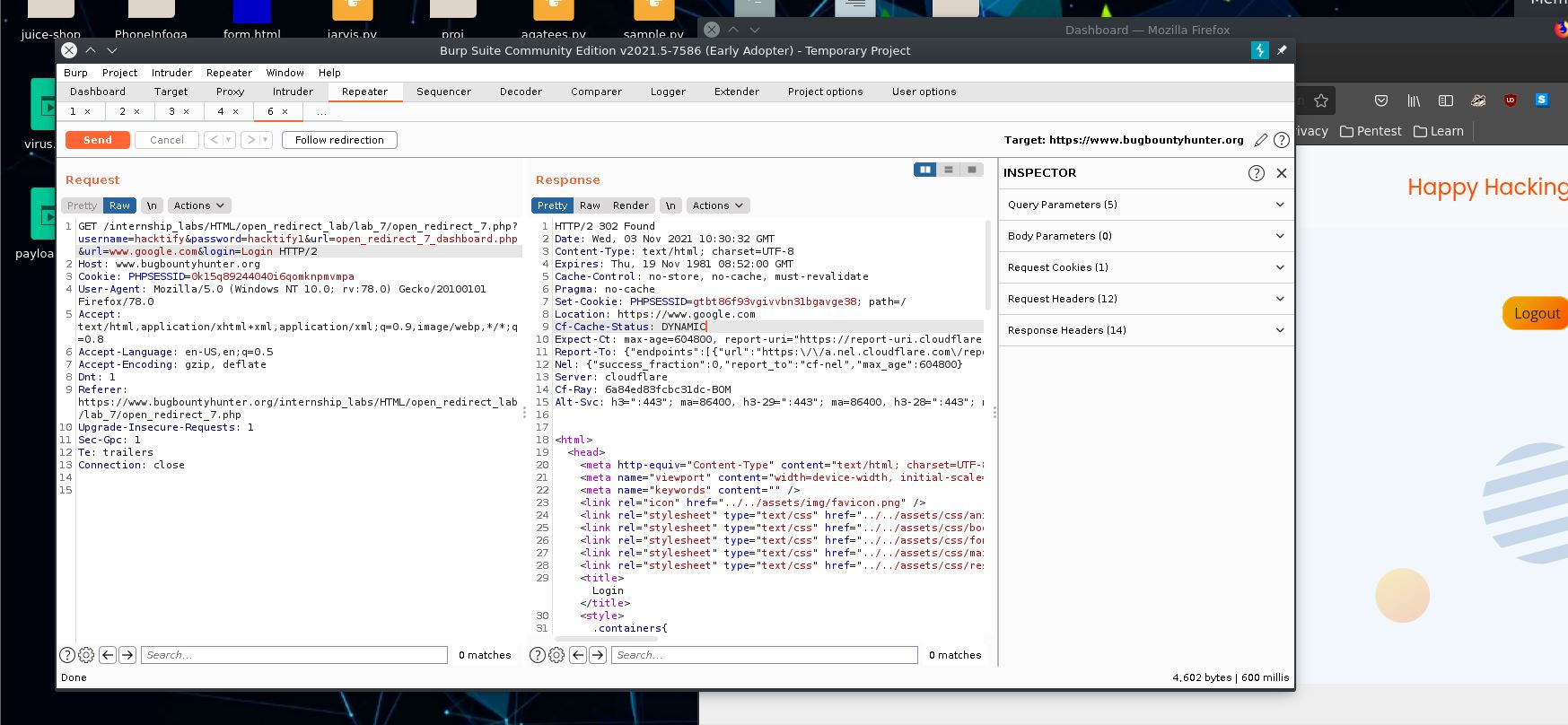
# 1.7. Domains ? Not Always!

| **Lab Name** | **Risk Rating** |
| --- | --- |
| Domains ? Not Always! | **Low** |
| **Tools Used** | |
| Burp suite | |
| **Vulnerability Description** | |
| An open redirect vulnerability exists when the destination of the redirect is provided by the client and it is not filtered or validated. An attacker can construct a URL within the application that causes a redirection to an arbitrary external domain. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
| <https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_7/open_redirect_7_dashboard.php> | |
| **Consequences of not Fixing the Issue** | |
| the attacker can able to steal victim credentials or useful information | |
| **Suggested Countermeasures** | |
| check the parameter in url tab. | |
| **References** | |
| <https://www.hacksplaining.com/prevention/open-redirects> | |

# 

# Proof of Concept

**payload:** url=www.google.com



# 1.8. Style Digit Symbols <3

| **Lab Name** | **Risk Rating** |
| --- | --- |
| Style Digit Symbols <3 | **Low** |
| **Tools Used** | |
| Burp suite | |
| **Vulnerability Description** | |
| An open redirect vulnerability exists when the destination of the redirect is provided by the client and it is not filtered or validated. An attacker can construct a URL within the application that causes a redirection to an arbitrary external domain. This behavior can be leveraged to facilitate phishing attacks against users of the application. The ability to use an authentic application URL, targeting the correct domain and with a valid SSL certificate (if SSL is used), lends credibility to the phishing attack because many users, even if they verify these features, will not notice the subsequent redirection to a different domain. In short, if an attacker is able to redirect the user to a malicious website then it can be termed as Open Redirect. | |
| **How It Was Discovered** | |
| Manual Analysis | |
| **Vulnerable URLs** | |
| <https://www.bugbountyhunter.org/internship_labs/HTML/open_redirect_lab/lab_8/open_redirect_8_dashboard.php> | |
| **Consequences of not Fixing the Issue** | |
| it may lead to phishing and account takeover . | |
| **Suggested Countermeasures** | |
| don’t allow those ip in the url parameter. | |
| **References** | |
| <https://www.hacksplaining.com/prevention/open-redirects> | |

# 

# Proof of Concept

**payload:** url=172.217.167.132



# 2. Exchangeable Image File Format

# 2.1 Let's PII!

| **Lab Name** | **Risk Rating** |
| --- | --- |
| Exchangeable Image File Format | **Medium** |
| **Tools Used** | |
| <http://exif.regex.info/exif.cgi> | |
| **Vulnerability Description** | |
| EXIF Data stores sensitive information like Geo-location, Date, Name of the camera, Modified date, Time, Sensing Method, File Source, Type of compression etc. in the photos you click. Now this data resides in every photo you take using cameras. Whenever you upload a picture on a website and if the website does not strip these sensitive data then this could lead to sensitive data exposure like the Geo-location, Date of the photo, Time of the photo, Camera used etc. | |
| **How It Was Discovered** | |
| Automated Tools | |
| **Vulnerable URLs** | |
| <https://www.bugbountyhunter.org/internship_labs/HTML/exif_lab/lab_1/exif.php> | |
| **Consequences of not Fixing the Issue** | |
| It may lead to extract location info and some sensitive data like focal length etc... | |
| **Suggested Countermeasures** | |
| strip those sensitive data like latitude and longitude | |
| **References** | |
| <https://photographylife.com/what-is-exif-data> | |

# 

# Proof of Concept

