# Sri Lanka Institute of Information Technology


## Specialized in Cyber Security

Year 2, Semester 2

# IE2062 – Web Security

Bug Bounty – Report 09

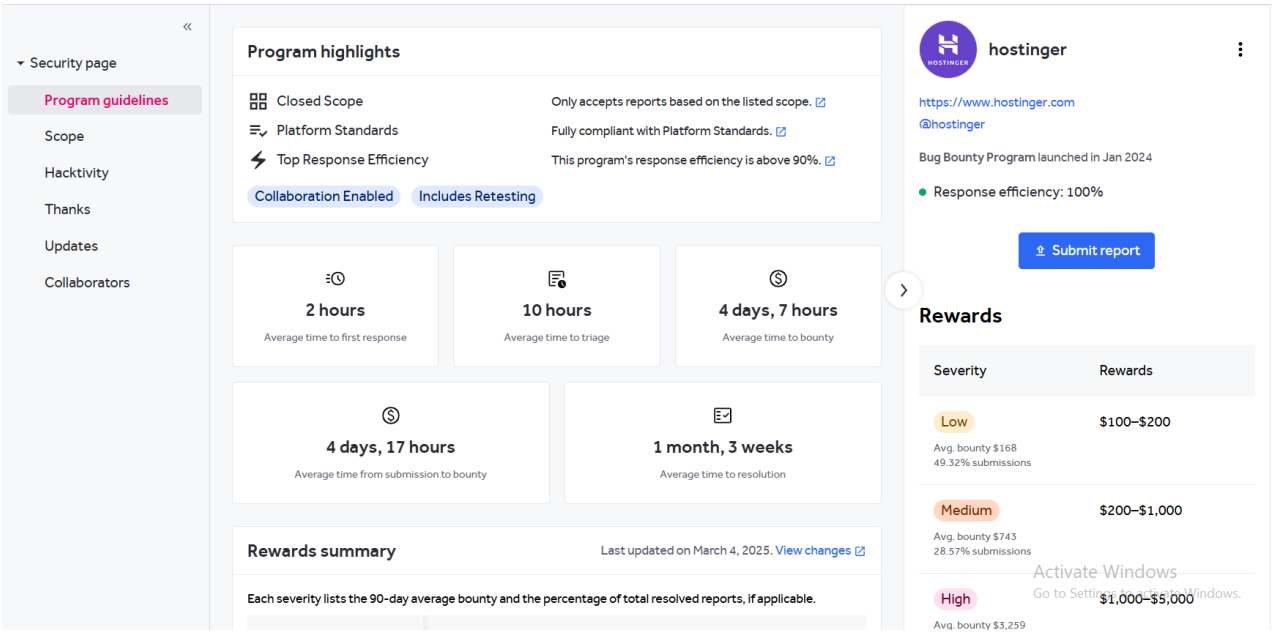
| Student ID No. | Name |
|---|---|
| IT23136106 | D.M.M. Pasindu Supushmika |

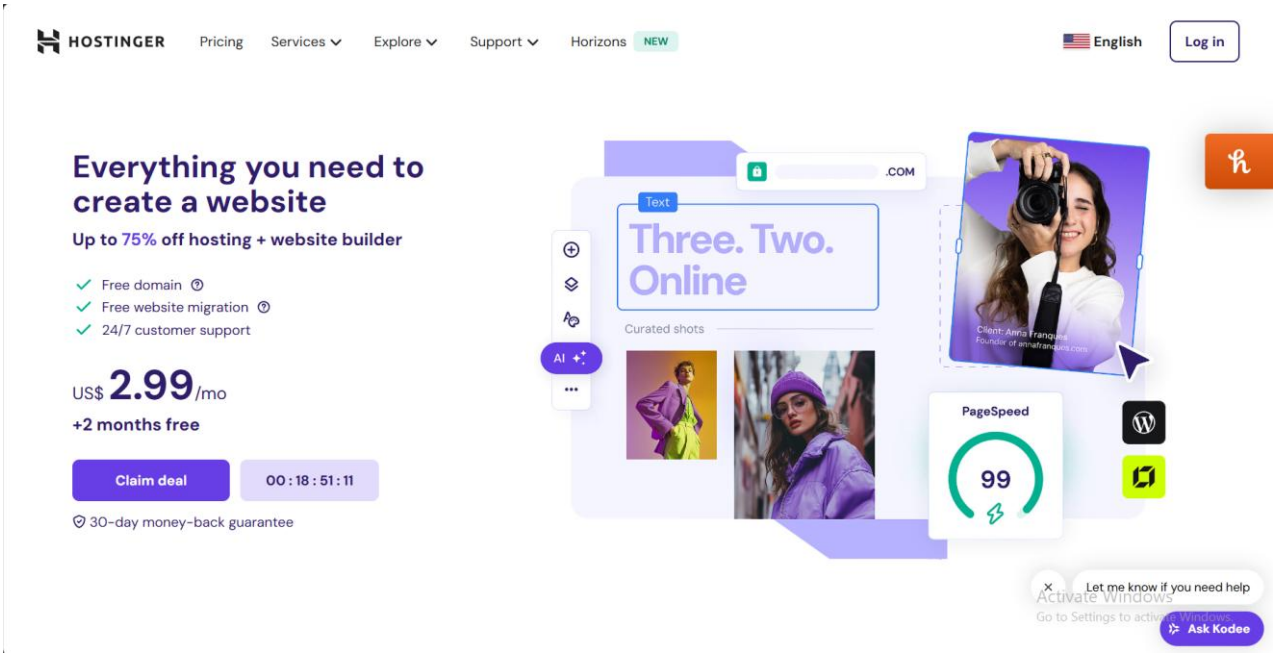
# Table of Contents

# 1. Website Overview

[Hostinger - Bring Your Idea Online With a Website]

HackerOne Link: [hostinger | Bug Bounty Program Policy | HackerOne]



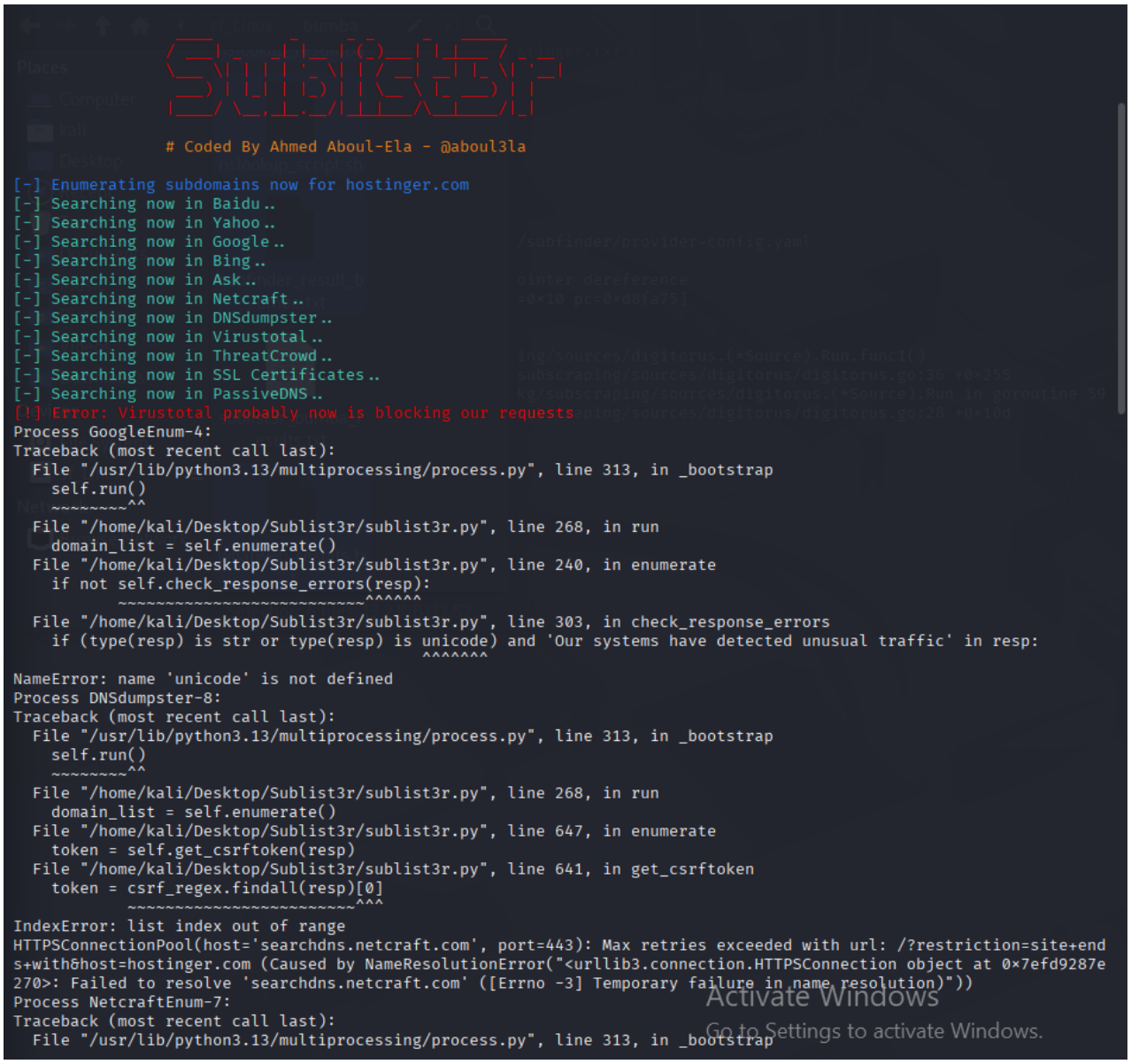



# **Step 01:** Gather Information.

## **a.** Sub-domain Discovery

### i. Sublist3r: [sublist3r_hostinger_results.txt]

**Tool** **:** Sublist3r
**Code** **:** python3 sublist3r.py -d hostinger.com -o sublist3r_hostinger_results.txt
**Explanation:**
*python3 sublist3r.py* - Run the script using python
*-d hostinger.com* - Target domain
*-o sublist3r_hostinger_results.txt* – Output file where the result is saved

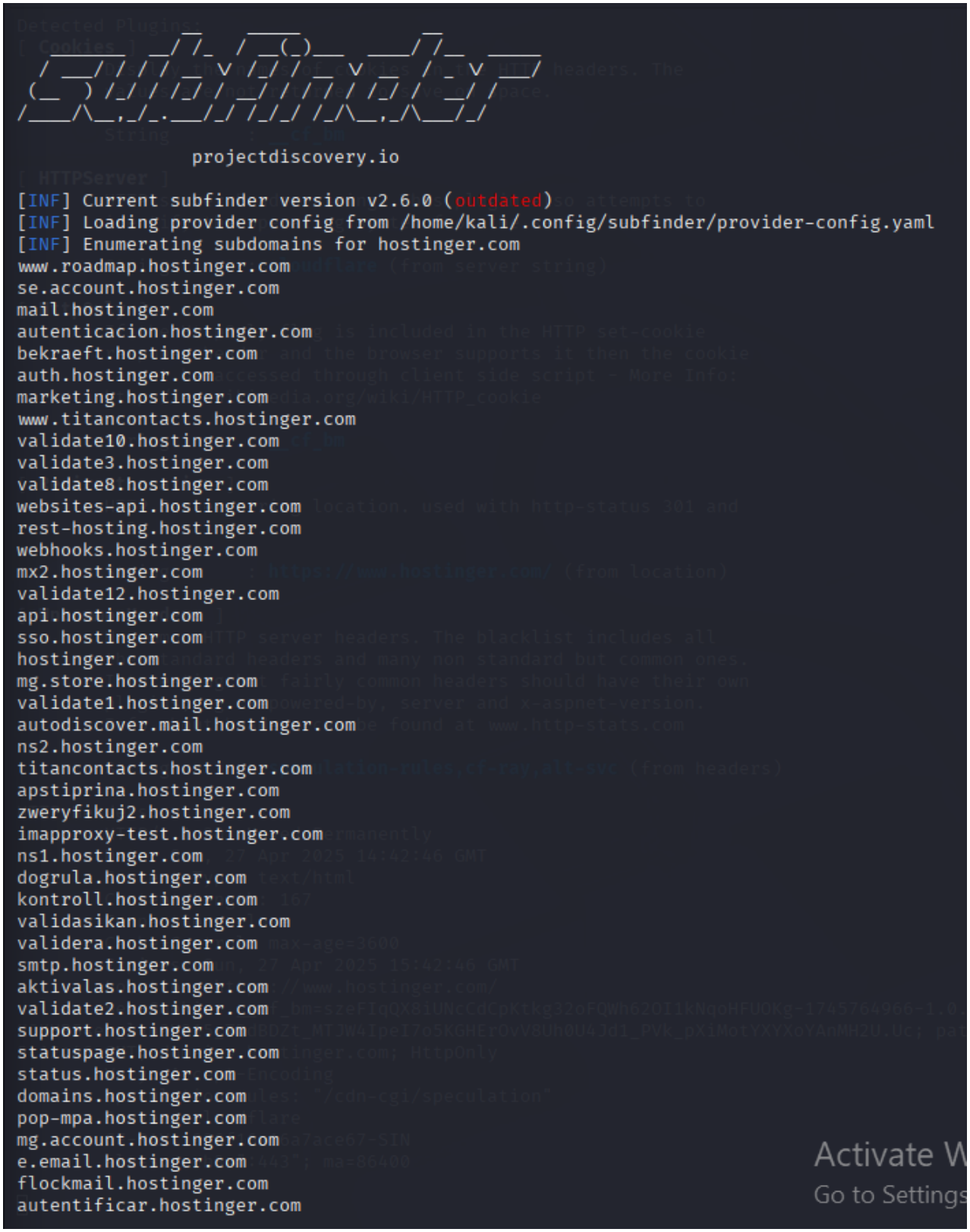


```
                 ____        _     _ _     _   _____
                / ___| _   _| |__ | (_)___| |_|___ / _ __
                \___ \| | | | '_ \| | / __| __| |_ \| '__|
                 ___) | |_| | |_) | | \__ \ |_ ___) | |
                |____/ \__,_|_.__/|_|_|___/\__|____/|_|

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for hostinger.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
Process GoogleEnum-4:
Traceback (most recent call last):
  File "/usr/lib/python3.13/multiprocessing/process.py", line 313, in _bootstrap
    self.run()
    ~~~~~~~~^^
  File "/home/kali/Desktop/Sublist3r/sublist3r.py", line 268, in run
    domain_list = self.enumerate()
  File "/home/kali/Desktop/Sublist3r/sublist3r.py", line 240, in enumerate
    if not self.check_response_errors(resp):
           ~~~~~~~~~~~~~~~~~~~~~~~~~~^^^^^^
  File "/home/kali/Desktop/Sublist3r/sublist3r.py", line 303, in check_response_errors
    if (type(resp) is str or type(resp) is unicode) and 'Our systems have detected unusual traffic' in resp:
                             ^^^^^^^
NameError: name 'unicode' is not defined
Process DNSdumpster-8:
Traceback (most recent call last):
  File "/usr/lib/python3.13/multiprocessing/process.py", line 313, in _bootstrap
    self.run()
    ~~~~~~~~^^
  File "/home/kali/Desktop/Sublist3r/sublist3r.py", line 268, in run
    domain_list = self.enumerate()
  File "/home/kali/Desktop/Sublist3r/sublist3r.py", line 647, in enumerate
    token = self.get_csrftoken(resp)
  File "/home/kali/Desktop/Sublist3r/sublist3r.py", line 641, in get_csrftoken
    token = csrf_regex.findall(resp)[0]
            ~~~~~~~~~~~~~~~~~~~~~~~^^^
IndexError: list index out of range
HTTPSConnectionPool(host='searchdns.netcraft.com', port=443): Max retries exceeded with url: /?restriction=site+end
s+with&host=hostinger.com (Caused by NameResolutionError("<urllib3.connection.HTTPSConnection object at 0x7efd9287e
270>: Failed to resolve 'searchdns.netcraft.com' ([Errno -3] Temporary failure in name resolution)"))
Process NetcraftEnum-7:
Traceback (most recent call last):
  File "/usr/lib/python3.13/multiprocessing/process.py", line 313, in _bootstrap
```

www.hostinger.com
academy.hostinger.com
d.account.hostinger.com
e.account.hostinger.com
mg.account.hostinger.com
se.account.hostinger.com
affiliates.hostinger.com
www.affiliates.hostinger.com
affs-stats.hostinger.com
aktivalas.hostinger.com
ambassador.hostinger.com
api.hostinger.com
apstiprina.hostinger.com
autenticacion.hostinger.com
autenticar.hostinger.com
autentificacion.hostinger.com
autentificar.hostinger.com
bekraeft.hostinger.com
ping.bnk.hostinger.com
builder.hostinger.com
cdn.hostinger.com
confirmar.hostinger.com
connect.hostinger.com
convalida.hostinger.com
cpanel.hostinger.com
design.hostinger.com
dogrula.hostinger.com
domains.hostinger.com
d.email.hostinger.com
e.email.hostinger.com
mg.email.hostinger.com
se.email.hostinger.com
epikurwsi.hostinger.com
flockcalendar.hostinger.com
www.flockcalendar.hostinger.com
flockcontacts.hostinger.com
www.flockcontacts.hostinger.com
flockmail.hostinger.com
www.flockmail.hostinger.com
help.hostinger.com
hpanel.hostinger.com
imapproxy-test.hostinger.com
kontroll.hostinger.com
mail.hostinger.com
mailer.hostinger.com
mailstorage-test.hostinger.com
marketing.hostinger.com
www.marketing.hostinger.com
mg.notifications.hostinger.com
ovjeriti.hostinger.com
partners.hostinger.com
patvirtinti.hostinger.com
payments.hostinger.com

ii. Subfindre: [subfinder_result_hostinger.txt]

**Tool** : Subfinder
**Code** : subfinder -d hostinger.com -o subfinder_result.txt
**Explanation:**
*subfinder* - run subfinder too
*l -d hostinger.com* - Mention the target website
*-o subfinder_result.txt* – Mention the output file



www.roadmap.hostinger.com
se.account.hostinger.com
mail.hostinger.com
autenticacion.hostinger.com
bekraeft.hostinger.com
auth.hostinger.com
marketing.hostinger.com
www.titancontacts.hostinger.com
validate10.hostinger.com
validate3.hostinger.com
validate8.hostinger.com
websites-api.hostinger.com
rest-hosting.hostinger.com
webhooks.hostinger.com
mx2.hostinger.com
validate12.hostinger.com
api.hostinger.com
sso.hostinger.com
hostinger.com
mg.store.hostinger.com
validate1.hostinger.com
autodiscover.mail.hostinger.com
ns2.hostinger.com
titancontacts.hostinger.com
apstiprina.hostinger.com
zweryfikuj2.hostinger.com
imapproxy-test.hostinger.com
ns1.hostinger.com
dogrula.hostinger.com
kontroll.hostinger.com
validasikan.hostinger.com
validera.hostinger.com
smtp.hostinger.com
aktivalas.hostinger.com
validate2.hostinger.com
support.hostinger.com
statuspage.hostinger.com
status.hostinger.com
domains.hostinger.com
pop-mpa.hostinger.com
mg.account.hostinger.com
e.email.hostinger.com
flockmail.hostinger.com
autentificar.hostinger.com
auth-db196.hostinger.com
connect.hostinger.com
any2.hostinger.com
rdns2.hostinger.com
confirmar.hostinger.com
affs-stats.hostinger.com
www.affiliates.hostinger.com
titancalendar.hostinger.com
www.flockmail.hostinger.com
validate4.hostinger.com
frontend-event-api.hostinger.com
ambassador.hostinger.com
auth-db191.hostinger.com
ns3.hostinger.com

## b. Live Subdomain Discovery

**Tool :** httpx: livesub_results.txt

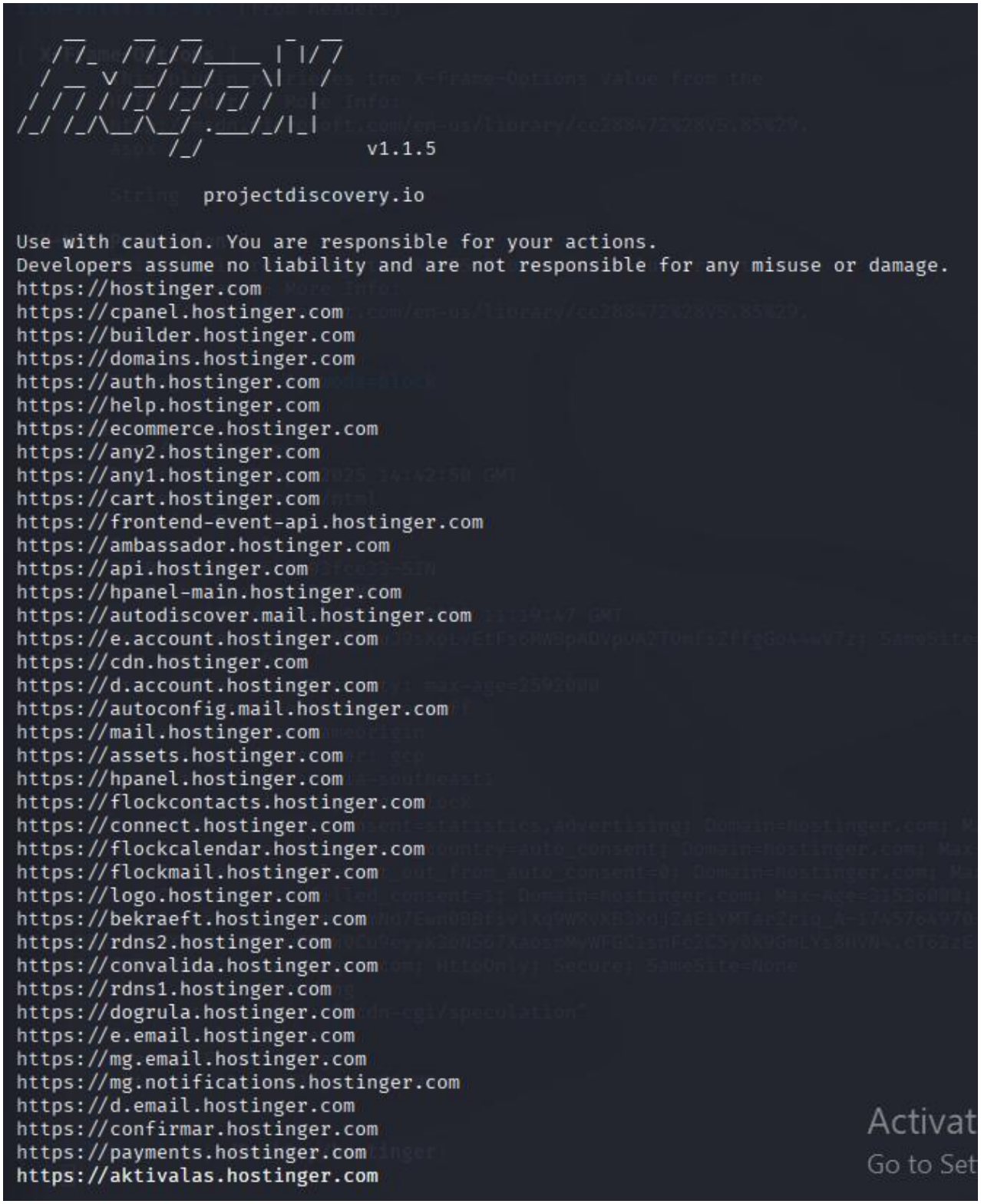
**Code :** httpx-toolkit -l subfinder_result.txt -o livesub_results.txt

**Explanation:**

*httpx-toolkit* - run the httpx tool

*-l subfinder_result.txt* – mention the file containing input

*-o livesub_results.txt* – mention the file which should write the output



https://hostinger.com
https://cpanel.hostinger.com
https://builder.hostinger.com
https://domains.hostinger.com
https://auth.hostinger.com
https://help.hostinger.com
https://ecommerce.hostinger.com
https://any2.hostinger.com
https://any1.hostinger.com
https://cart.hostinger.com
https://frontend-event-api.hostinger.com
https://ambassador.hostinger.com
https://api.hostinger.com
https://hpanel-main.hostinger.com
https://autodiscover.mail.hostinger.com
https://e.account.hostinger.com
https://cdn.hostinger.com
https://d.account.hostinger.com
https://autoconfig.mail.hostinger.com
https://mail.hostinger.com
https://assets.hostinger.com
https://hpanel.hostinger.com
https://flockcontacts.hostinger.com
https://connect.hostinger.com
https://flockcalendar.hostinger.com
https://flockmail.hostinger.com
https://logo.hostinger.com
https://bekraeft.hostinger.com
https://rdns2.hostinger.com
https://convalida.hostinger.com
https://rdns1.hostinger.com
https://dogrula.hostinger.com
https://e.email.hostinger.com
https://mg.email.hostinger.com
https://mg.notifications.hostinger.com
https://d.email.hostinger.com
https://confirmar.hostinger.com
https://payments.hostinger.com
https://aktivalas.hostinger.com
https://autentificacion.hostinger.com
https://apstiprina.hostinger.com
https://autenticar.hostinger.com
https://rvlclick.hostinger.com
https://status.hostinger.com
https://autentificar.hostinger.com
https://rest-hosting.hostinger.com
https://autenticacion.hostinger.com
https://epikurwsi.hostinger.com
https://validate6.hostinger.com
https://validate7.hostinger.com
https://validera.hostinger.com
https://validere.hostinger.com
https://valideren.hostinger.com
https://validate8.hostinger.com
https://www.flockcalendar.hostinger.com
https://validate9.hostinger.com
https://validieren.hostinger.com
https://validate5.hostinger.com

## c. IP Discovery

**Tool:** nslookup: nslookup_result.txt

**Code:** since we whole file with subdomains, to find IP addresses using "nslookup" we need to make a loop until all the Ips of all the subdomains are found.

```
while read sub; do
 echo "Looking up: $sub" >> nslookup_result.txt
  nslookup "$sub" | awk '/^Name:|^Address:/' >> nslookup_result.txt
  echo "------------------------" >> nslookup_result.txt
done < livesub_results.txt
```

**Explanation:**

*While read sub; do* - start of the loop

*Echo "Looking up: $sub">>nslookup_result.txt* - print message "Looking up: subdomain" into the file "*nslookup_result*.txt"

*nslookup "$sub" | awk '/^Name:|^Address:/' >> nslookup_result.txt* - run the nslookup command

*echo "_____" >> nslookup_result.txt* - separate one subdomain details from another

*done < livesub_results.txt* - End the loop and continue until the lines in the livesub_results.txt

```
┌──(kali㉿kali)-[~/Desktop/hostinger]
└─$ ./nslookup_script.sh

┌──(kali㉿kali)-[~/Desktop/hostinger]
└─$ cat nslookup_result.txt
Looking up: https://hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://cpanel.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://builder.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://domains.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://auth.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://help.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://ecommerce.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://any2.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://any1.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://cart.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://frontend-event-api.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://ambassador.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://api.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://hpanel-main.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://autodiscover.mail.hostinger.com
Address:        192.168.0.1#53
------------------------
Looking up: https://e.account.hostinger.com
Address:        192.168.0.1#53
------------------------
```

**IP list:**

Looking up: https://hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://cpanel.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://builder.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://domains.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://auth.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://help.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://ecommerce.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://any2.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://any1.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://cart.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://frontend-event-api.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://ambassador.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://api.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://hpanel-main.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://autodiscover.mail.hostinger.com
Address: 192.168.0.1#53
---
Looking up: https://e.account.hostinger.com
Address: 192.168.0.1#53
---

### d. Open Ports

**Tool:** nmap: nmap_result.txt

**Code:** nmap -sV -A -v -O hostinger.com -oN nmap_results.txt

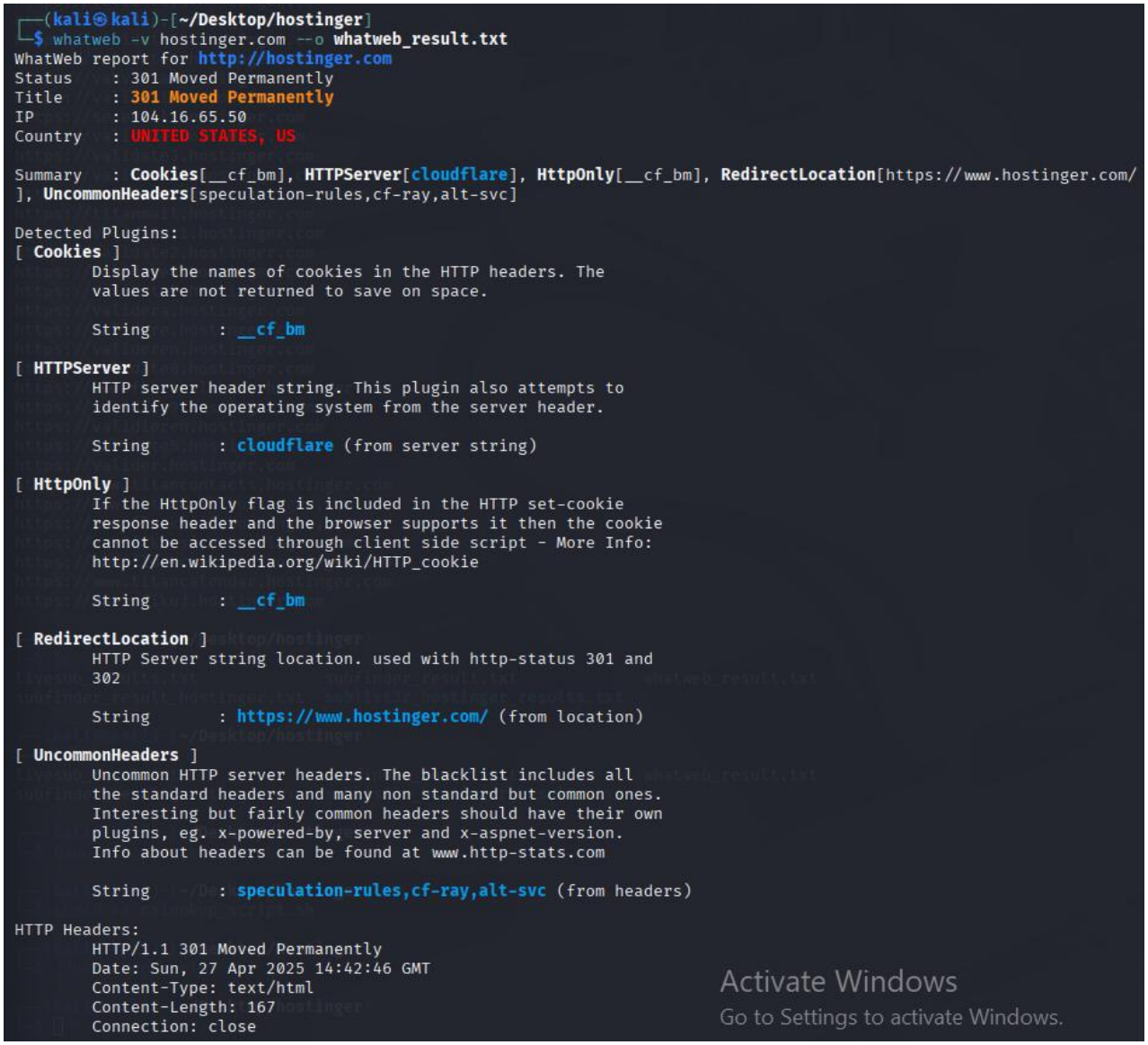
**Explanation:**

*nmap* - start the tool
*-sV* - Service and version detection
*-A* - OS detection, version detection, script scanning
*-v* - increase verbosity level
*-O* - Os detection
- *hostinger.com* - target website
*-oN nmap_results.txt* - result in an output text file

```
┌──(kali㉿kali)-[~/Desktop/hostinger]
└─$ nmap -sV -A -v -O hostinger.com -oN nmap_result.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-27 20:18 +0530
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 20:18
Completed NSE at 20:18, 0.00s elapsed
Initiating NSE at 20:18
Completed NSE at 20:18, 0.00s elapsed
Initiating NSE at 20:18
Completed NSE at 20:18, 0.00s elapsed
Initiating Ping Scan at 20:18
Scanning hostinger.com (104.16.65.50) [4 ports]
Completed Ping Scan at 20:18, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 20:18
Completed Parallel DNS resolution of 1 host. at 20:18, 0.13s elapsed
Initiating SYN Stealth Scan at 20:18
Scanning hostinger.com (104.16.65.50) [1000 ports]
Discovered open port 443/tcp on 104.16.65.50
Discovered open port 8080/tcp on 104.16.65.50
Discovered open port 25/tcp on 104.16.65.50
Discovered open port 80/tcp on 104.16.65.50
Completed SYN Stealth Scan at 20:18, 6.08s elapsed (1000 total ports)
Initiating Service scan at 20:18
Scanning 4 services on hostinger.com (104.16.65.50)
Completed Service scan at 20:18, 14.91s elapsed (4 services on 1 host)
Initiating OS detection (try #1) against hostinger.com (104.16.65.50)
Retrying OS detection (try #2) against hostinger.com (104.16.65.50)
Initiating Traceroute at 20:18
Completed Traceroute at 20:18, 0.07s elapsed
Initiating Parallel DNS resolution of 2 hosts. at 20:18
Completed Parallel DNS resolution of 2 hosts. at 20:18, 0.08s elapsed
NSE: Script scanning 104.16.65.50.
Initiating NSE at 20:18
Completed NSE at 20:19, 32.28s elapsed
Initiating NSE at 20:19
Completed NSE at 20:19, 30.14s elapsed
Initiating NSE at 20:19
Completed NSE at 20:19, 0.01s elapsed
Nmap scan report for hostinger.com (104.16.65.50)
Host is up (0.0094s latency).
Other addresses for hostinger.com (not scanned): 104.16.66.50 2606:4700::6810:4132 2606:4700::6810:4232
Not shown: 996 filtered tcp ports (no-response)
PORT    STATE SERVICE VERSION
25/tcp  open  smtp?
|_smtp-commands: Couldn't establish connection on port 25
| fingerprint-strings:
|   NULL:
|_    421 service not available (connection to blocklisted host (104.16.65.50 - DNSBL))
80/tcp  open  http    Cloudflare http proxy
|_http-title: Did not follow redirect to https://www.hostinger.com/
443/tcp open  https?
| ssl-cert: Subject: commonName=*.hostinger.com
| Subject Alternative Name: DNS:*.hostinger.com, DNS:hostinger.com
```

## e. Used Technologies

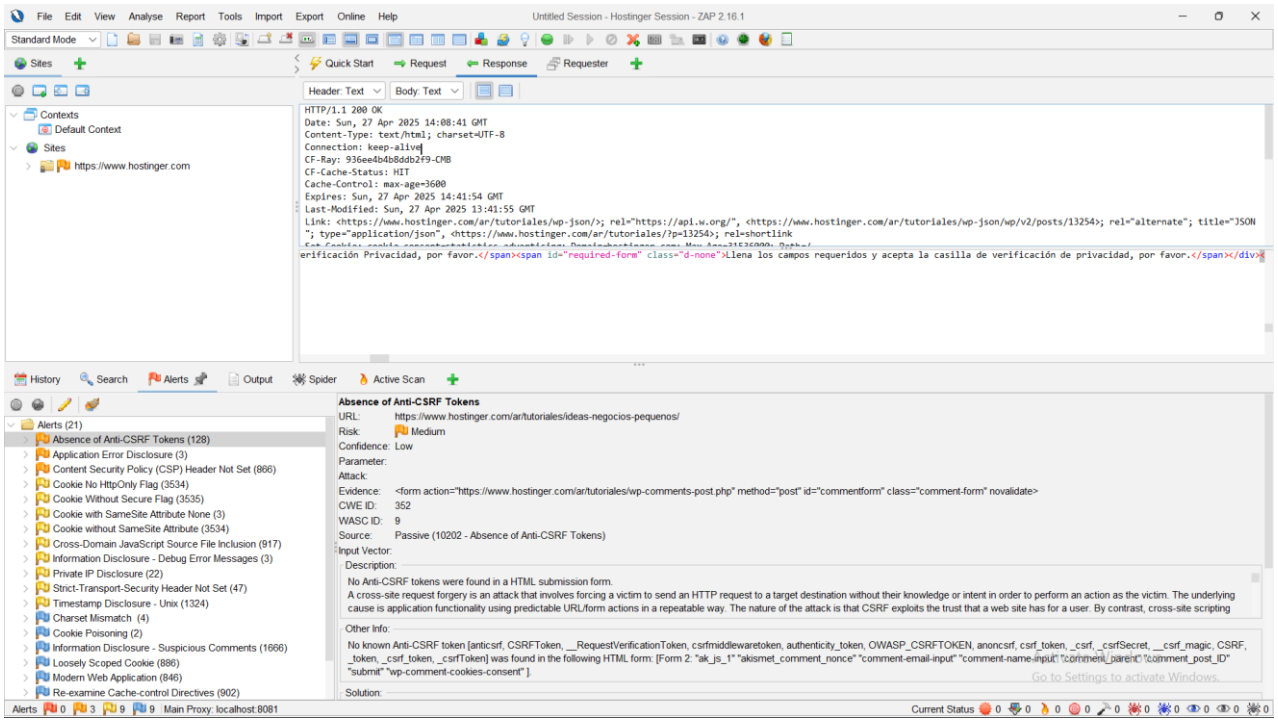
**Tool:** whatweb - whatweb_result.txt
**Code:** whatweb -v hostinger.com > whatweb_result.txt
**Explanation:**
*whatweb* - start whatweb tool
*-v* - verbose
*hostinger.com* - target website
*> whatweb_result.txt* - file with the output



# 3. Step 02: Scanning and vulnerability identification

## a. Identify Potential Vulnerabilities

**Tool** : OWASP ZAP
**Vulnerability** : Absence of Anti-CSRF Tokens




## Absence of Anti-CSRF Tokens:

URL: https://www.hostinger.com/ar/tutoriales/ideas-negocios-pequenos/
Risk: Medium
Confidential: Low
Parameter:
Attack:
Evidence: <form action="https://www.hostinger.com/ar/tutoriales/wp-comments-post.php" method="post" id="commentform" class="comment-form" novalidate>
CWE ID: 352
WASC ID: 9
Source: Passive (10202 - Absence of Anti-CSRF Tokens)
Input Vector:

- **Description**: No Anti-CSRF tokens were found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf. CSRF attacks are effective in a number of situations, including:
    - The victim has an active session on the target site.
    - The victim is authenticated via HTTP auth on the target site.
    - The victim is on the same local network as the target site.

  CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.

- **Other Info**: No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 2: "ak_js_1" "akismet_comment_nonce" "comment-email-input" "comment-name-input" "comment_parent" "comment_post_ID" "submit" "wp-comment-cookies-consent" ].

- **Solution**: Phase: Architecture and Design
  Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
  For example, use anti-CSRF packages such as the OWASP CSRFGuard.
  Phase: Implementation
  Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script.
  Phase: Architecture and Design
  Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330).
  Note that this can be bypassed using XSS.
  Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.
  Note that this can be bypassed using XSS.
  Use the ESAPI Session Management control.
  This control includes a component for CSRF.
  Do not use the GET method for any request that triggers a state change.
  Phase: Implementation
  Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons..

- **Reference**:
  - https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
  - https://cwe.mitre.org/data/definitions/352.html

- **Alert Tags**:
  - OWASP_2021_A01: https://owasp.org/Top10/A01_2021-Broken_Access_Control/
  - WSTG-v42-SESS-05: https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/06-Session_Management_Testing/05-Testing_for_Cross_Site_Request_Forgery
  - OWASP_2017_A05: https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control.html
  - CWE-352: https://cwe.mitre.org/data/definitions/352.html

## b. Absence of Anti-CSRF Tokens

**Absence of Anti-CSRF (Cross-Site Request Forgery) Tokens** means that a web application does not implement unique, unpredictable tokens to verify the legitimacy of user-initiated actions. Without these protections, attackers can trick authenticated users into performing unwanted actions, such as changing account settings, transferring money, or making unauthorized purchases, by exploiting the trust that a website has in the user's session..

.

Cause of Absence of Anti-CSRF Tokens in a website:

- Failure to implement CSRF protection mechanisms during form submissions or sensitive actions
- Relying solely on cookie-based authentication without validating request origins
- Lack of security awareness during development or rapid deployment cycles
- Using outdated web frameworks that do not automatically implement CSRF protection
- Incorrectly configured or missing CSRF middleware in modern frameworks
- Belief that using HTTPS alone is enough to prevent CSRF attacks

Propositions to Mitigation or Fix:

- Implement CSRF Tokens: Generate unique, unpredictable tokens for each user session and validate them on each sensitive request
- Use Secure Frameworks: Use modern frameworks (like Django, Laravel, or Spring Security) that offer built-in CSRF protection
- Double-Submit Cookies Strategy: Implement token validation both in cookies and in the request body or headers
- SameSite Cookie Attribute: Set cookies with SameSite=Strict or SameSite=Lax to limit cross-origin requests
- Validate Request Origins: Check the Origin and Referer headers for sensitive actions to ensure requests come from trusted sources
- Educate Developers: Make sure development teams understand the importance and proper implementation of CSRF defenses
- Conduct Regular Security Testing: Perform penetration testing and vulnerability scanning to detect CSRF vulnerabilities

# 4. Step 03: Exploitation and Validation

Request:

```
GET https://www.hostinger.com/ar/tutoriales/ideas-negocios-pequenos/ HTTP/1.1
host: www.hostinger.com
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
pragma: no-cache
cache-control: no-cache
referer: https://www.hostinger.com/ar/tutoriales/search/
Cookie: __cf_bm=Z8IgbIXV6FXU9nGO97IaTMbppfodr8QWsYnerJYIbos-1745762912-1.0.1.1-WMxeplaDqvhiH9Ju.pGo6vpe5948LJ0jlPbS6hQ_SeyKrwllNxKbpBtvmuA3um.
unJ1HQkvjZJITxwoKNkh65m70glG5w0I0GClqknXGFI4; __cflb=02DiuJ9sKpLvEtFs6MWBpADVpUA2TUmfiScCyRF73f32t; auto_filled_consent=1; cookie_consent=statistics,advertising; cookie_consent_
country=auto_consent; possible_opt_out_from_auto_consent=0
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 27 Apr 2025 14:08:41 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
CF-Ray: 936ee4b4b8ddb2f9-CMB
CF-Cache-Status: HIT
Cache-Control: max-age=3600
Expires: Sun, 27 Apr 2025 14:41:54 GMT
Last-Modified: Sun, 27 Apr 2025 13:41:55 GMT
Link: <https://www.hostinger.com/ar/tutoriales/wp-json/>; rel="https://api.w.org/", <https://www.hostinger.com/ar/tutoriales/wp-json/wp/v2/posts/13254>; rel="alternate"; title="JSON
"; type="application/json", <https://www.hostinger.com/ar/tutoriales/?p=13254>; rel=shortlink
Set-Cookie: cookie_consent=statistics,advertising; Domain=hostinger.com; Max-Age=31536000; Path=/
Strict-Transport-Security: max-age=2592000
Vary: X-Forwarded-Proto,Accept-Encoding
Set-cookie: cookie_consent_country=auto_consent; Domain=hostinger.com; Max-Age=31536000; Path=/
Set-cookie: possible_opt_out_from_auto_consent=0; Domain=hostinger.com; Max-Age=31536000; Path=/
Set-cookie: auto_filled_consent=1; Domain=hostinger.com; Max-Age=31536000; Path=/
x-content-type-options: nosniff
x-frame-options: sameorigin
x-hostinger-datacenter: gcp
x-hostinger-node: asia-southeast1
x-pingback: https://www.hostinger.com/ar/tutoriales/xmlrpc.php
x-xss-protection: 1; mode=block
speculation-rules: "/cdn-cgi/speculation"
Server: cloudflare
alt-svc: h3=":443"; ma=86400
content-length: 211119
```

```
erificación Privacidad, por favor.</span><span id="required-form" class="d-none">Llena los campos requeridos y acepta la casilla de verificación de privacidad, por favor.</span></div>
```

# 5. Step 04: Mitigation / Fix

Immediate Mitigation Actions:
1. Add CSRF Tokens to Forms.
2. For custom forms generate a unique token per session.

Long Term Prevention:
1. Use built-in CSRF protections
2. Developer Training