Sri Lanka Institute of Information Technology

Web Security – IE2062

Topic: Bug Bounty Report 1

Y2S2.WE.CS

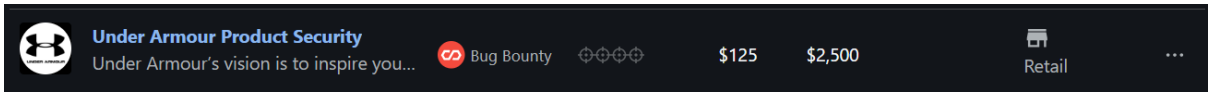
Name: S.D.W.Gunaratne

(IT23241978)

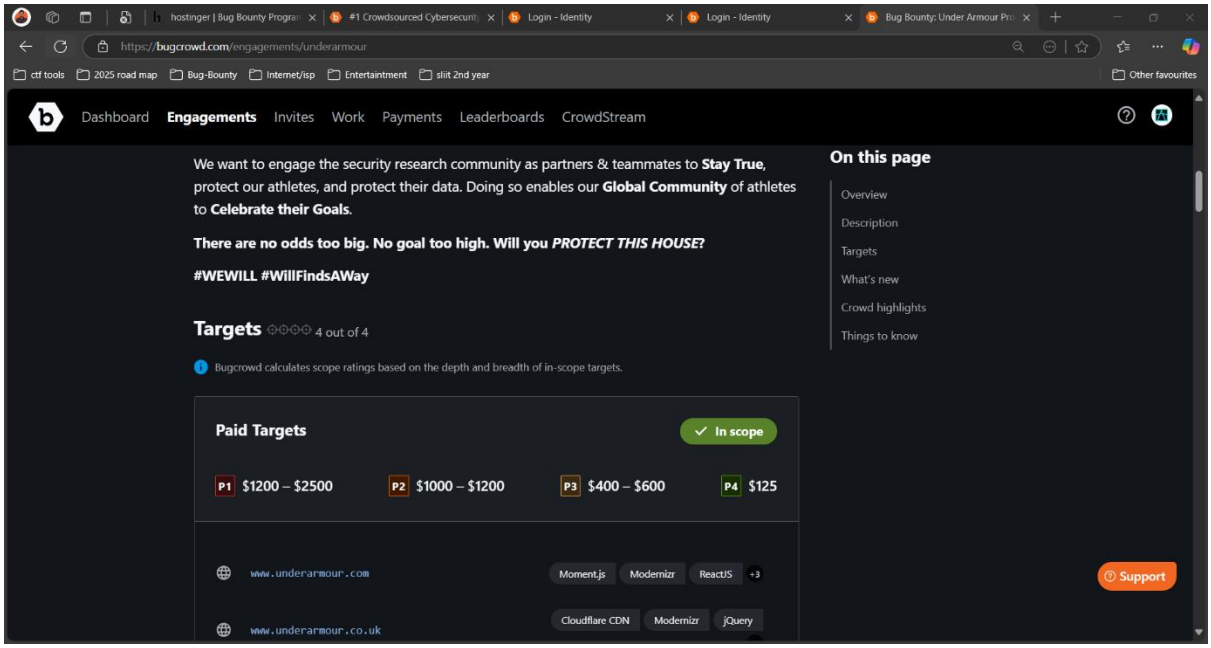
Table of Content

How I started?

1. Once I search from Bug crowd, I saw a Under Armour product security bug bounty program.

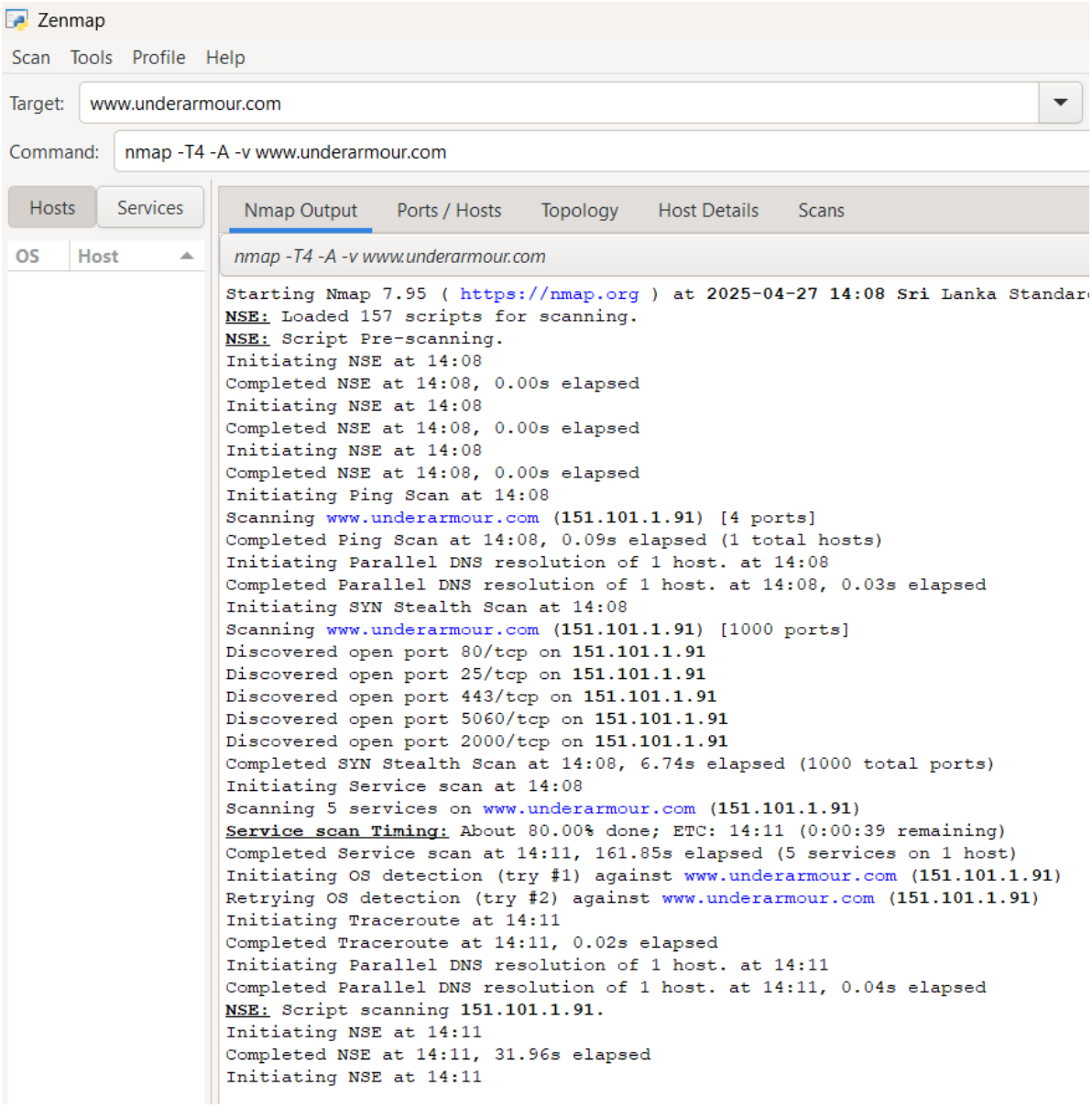


2. Then, I discovered full main domain allowed for scope, so that I choose https://www.underarmour.com/ .



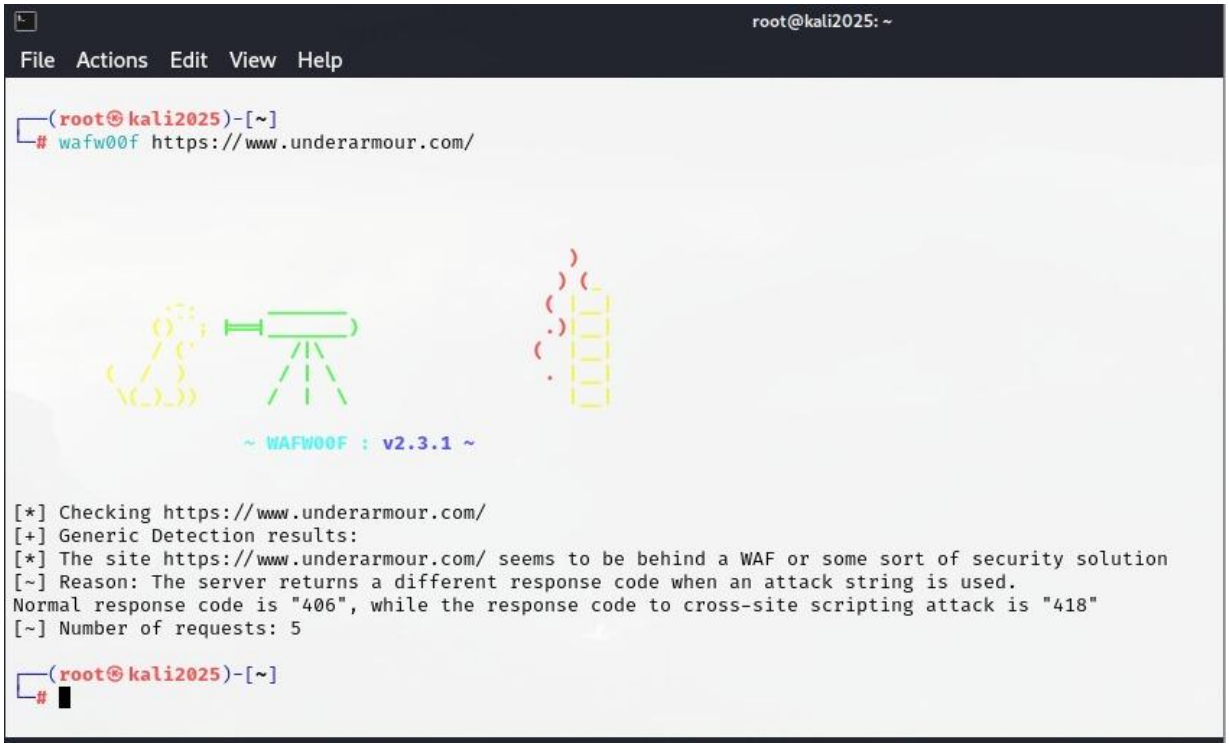
3. I use several methods/tools to do penetration testing.

4. First, I used Nmap. It helps me to find what are the open ports, Identify the web technologies such as webservers.

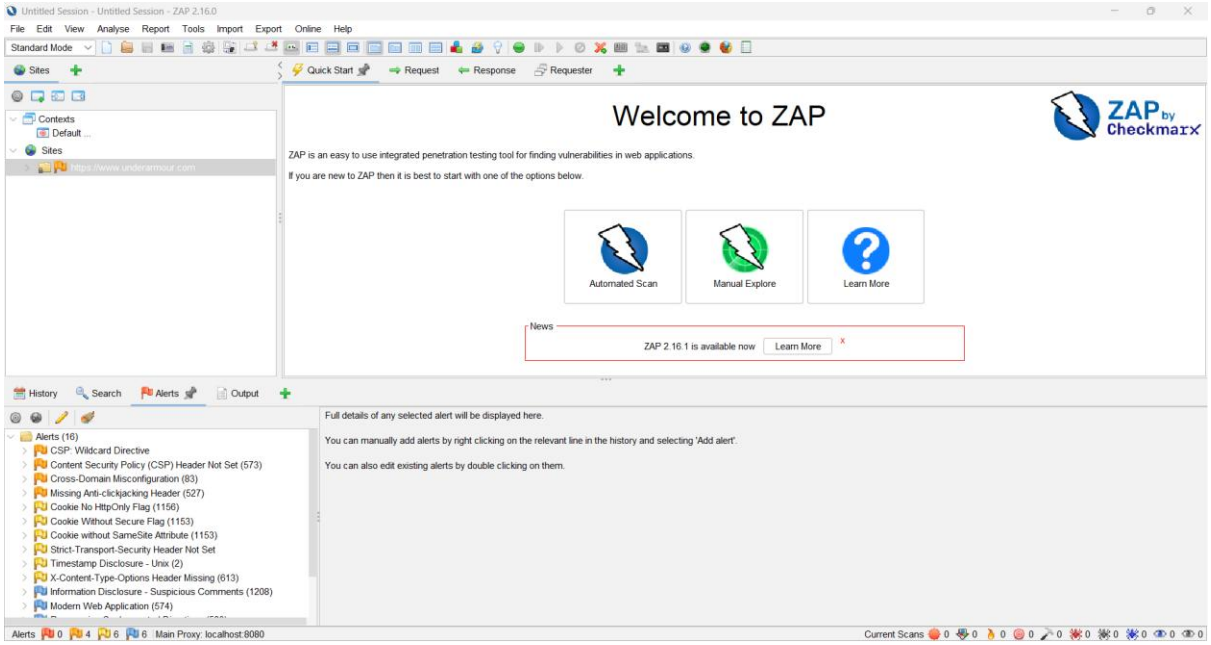


5. Secondly, I used Subfider tool to find hidden or forgotten web asserts. Because hidden web assert can have poor security, unpatched vulnerabilities.

6. Thirdly, I used Wafwoof tool to find website is protected by a WAF (web application firewall). Because if WAF is active, so pen tester do their test without blocked, and they can do their testing with bypass WAF.



7.Finaly, I use OWASP zap to automatically find the vulnerabilities.

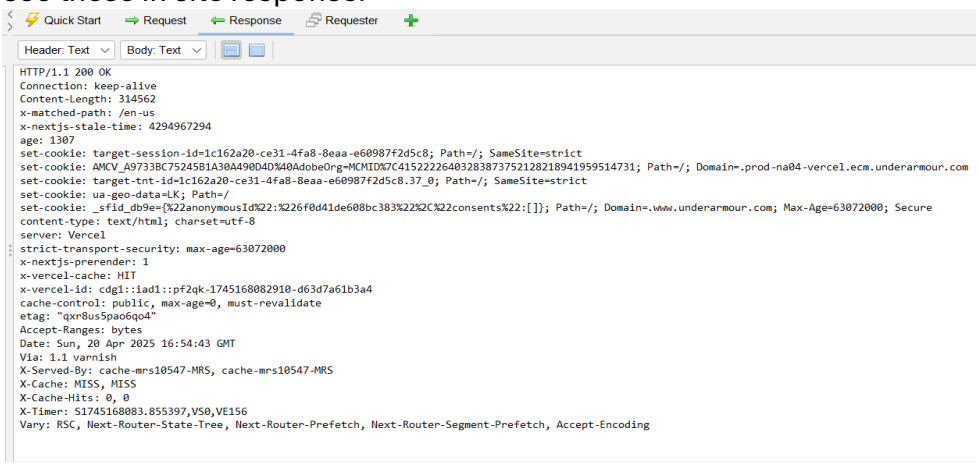


With getting these tool's support, I found below details about vulnerability.

2) Introduction

2.1 Domain	https://www.underarmour.com/
2.2 Severity	• Medium

3) Vulnerability

3.1 Vulnerability title	Missing Anti-clickjacking Header OWASP_2021_A05 CWE-1021
3.2 Vulnerability description	The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.
3.3 Affected components	x-frame-options In this case effectuated component is https://prod-na04-vercel.ecm.underarmour.com/en-us To prevent Anti-clickjacking, we need X-Frame-Options Or Content-Security-Policy: frame-ancestors. But in this site, we can't see these in site response.  Quick Start Request Response Requester Header: Text Body: Text HTTP/1.1 200 OK Connection: keep-alive Content-Length: 314562 x-matched-path: /en-us x-nextjs-stale-time: 4294967294 age: 1307 set-cookie: target-session-id=1c162a20-ce31-4fa8-8eaa-e60987f2d5c8; Path=/; SameSite=strict set-cookie: AMCV_A9733BC75245B1A30A490D4D%40AdobeOrg=MCMID%7C41522226403283873752128218941959514731; Path=/; Domain=.prod-na04-vercel.ecm.underarmour.com set-cookie: target-tnt-id=1c162a20-ce31-4fa8-8eaa-e60987f2d5c8.37_0; Path=/; SameSite=strict set-cookie: ua-geo-data=LK; Path=/ set-cookie: _sfid_db9e={%22anonymousId%22:%226f0d41de608bc383%22%2C%22consents%22:[]}; Path=/; Domain=.www.underarmour.com; Max-Age=63072000; Secure content-type: text/html; charset=utf-8 server: Vercel strict-transport-security: max-age=63072000 x-nextjs-prerender: 1 x-vercel-cache: HIT x-vercel-id: cdg1::iad1::pf2qk-1745168082910-d63d7a61b3a4 cache-control: public, max-age=0, must-revalidate etag: "qxr8us5pao6qo4" Accept-Ranges: bytes Date: Sun, 20 Apr 2025 16:54:43 GMT Via: 1.1 varnish X-Served-By: cache-mrs10547-MRS, cache-mrs10547-MRS X-Cache: MISS, MISS X-Cache-Hits: 0, 0 X-Timer: S1745168083.855397,VS0,VE156 Vary: RSC, Next-Router-State-Tree, Next-Router-Prefetch, Next-Router-Segment-Prefetch, Accept-Encoding
3.4 Impact assessment	Because the headers don't exist, attackers can embed this page inside an iframe on an evil website. This makes Clickjacking attacks possible, where users are tricked into clicking on buttons or links that they do not see potentially leading to unauthorized operations or theft (data loss)

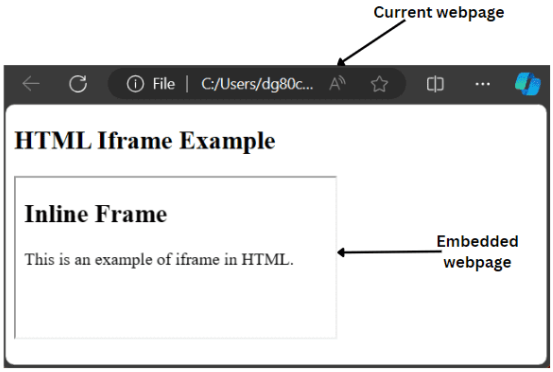


Fig: An example of Iframe in HTML

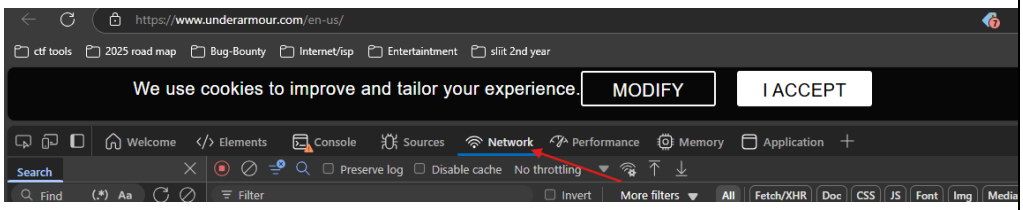
3.5 Steps to reproduce

This how to reproduce: -

1. Search URL(https://www.underarmour.com/) on our browser

2. Go to developer tools

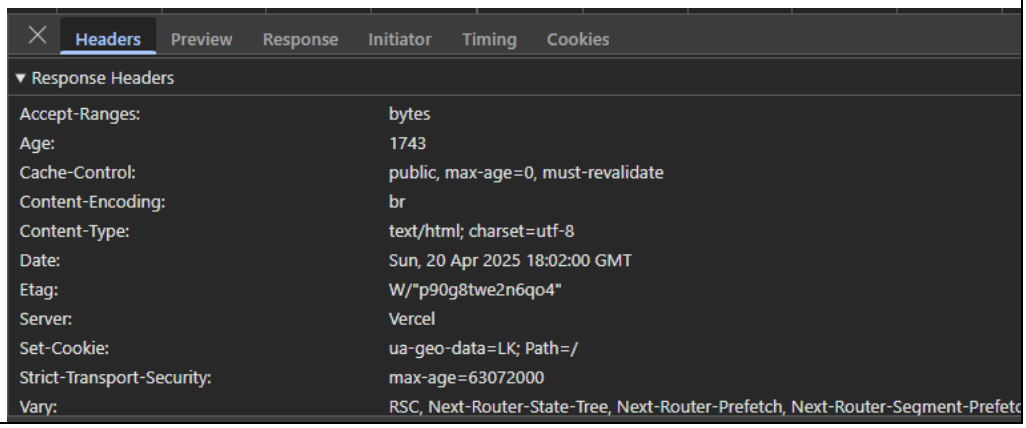
3. Select Network option



4. Refresh the page and check respond header

3.6 Proof of concept

5. While checking we can see, X-Frame-Options
Or Content-Security-Policy with frame-ancestors are missing (not there)



X-Cache:	MISS, MISS
X-Cache-Hits:	0, 0
X-Matched-Path:	/en-us
X-Nextjs-Prerender:	1
X-Nextjs-Stale-Time:	4294967294
X-Served-By:	cache-mrs10525-MRS, cache-mrs10525-MRS
X-Timer:	S1745172120.406166,VS0,VE188
X-Vercel-Cache:	HIT
X-Vercel-Id:	cdg1::iad1::h9nx2-1745172120472-8af511764291

Abou Clickjacking: -
It enables the attacker to trick the users into clicking something on the website without knowing it — this is referred to as Clickjacking.
This one does by a small window on a website that can show another website inside it.

This is How the attacker tricks the user: -
1. The hacker will make his own fake website.

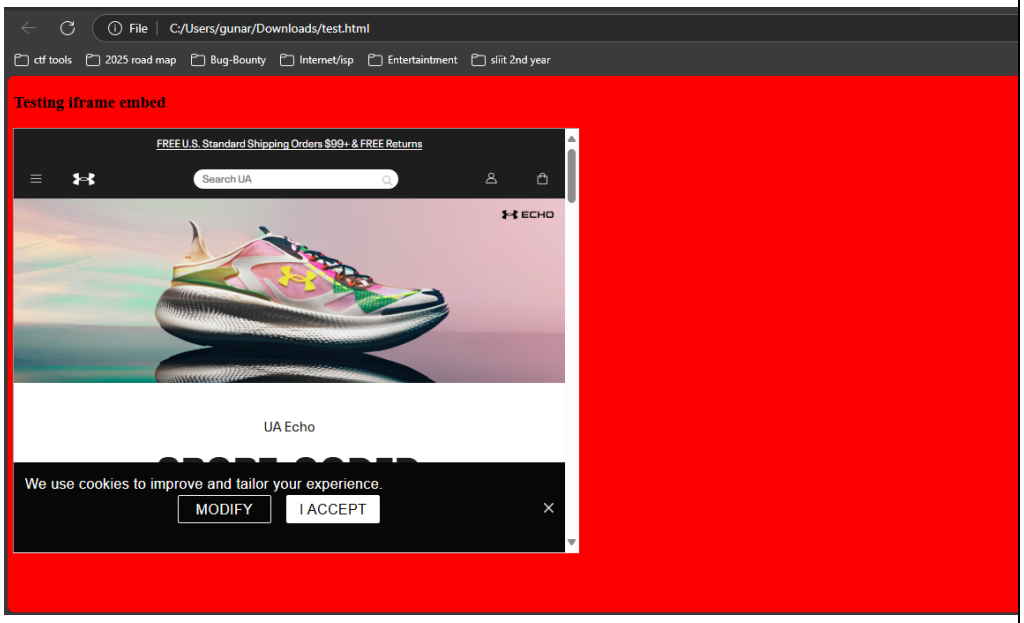
2. In their site, they quietly load your true website inside a hidden window (a frame).

3. This is dangerous because, user might click links or button with their account without acknowledgement.

This is **how to do pen testing** this concept:

Use below code and make web site first and link our tested site :-
https://prod-na04-vercel.ecm.underarmour.com/en-us

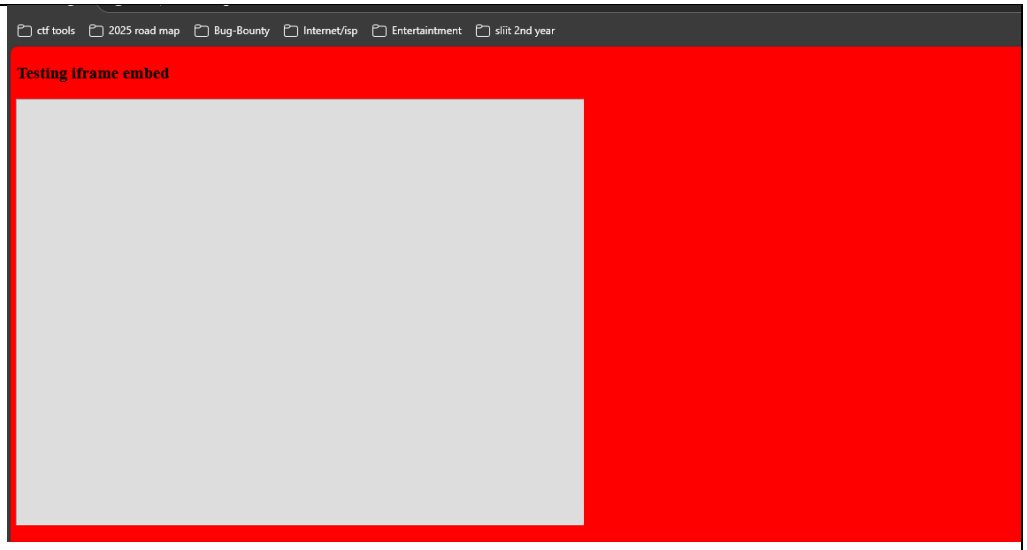
```
<!DOCTYPE html>
<html>
 <head><title>Clickjacking Test</title></head>
 <body>
  <h3>Testing iframe embed</h3>
  <iframe src="https://prod-na04-vercel.ecm.underarmour.com/en-us"
width="800" height="600"></iframe>
 </body>
</html>
```

<table>
<tr>
<td></td>
<td>


In this scenario we can see this site loaded to our tested web page so, this is not safe.
Ok,
</td>
</tr>
<tr>
<td>3.7 Proposed mitigation or fix</td>
<td>
Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.

If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.

If we successfully mitigate this problem our expected out look like this:
</td>
</tr>
</table>

	 If we successfully mitigate this, page cannot not load this frame.

Extra: - For this vulnerability I report this issue for bug crowd site, and I get this respond,

Dashboard Engagements Invites **Work** Payments Leaderboards CrowdStream

Submissions Workflows

← Go back

Look! Your site can be shown inside another site — that's not safe.

Submitted 3 days ago • Last activity 3 days ago

ID	21974992-356f-4a5f-8b30-f0e1bd72423c
Submitted	29 Apr 2025 14:57:41 UTC
Target Location	www.underarmour.com
Target category	Web App
VRT	Server Security Misconfiguration > Lack of Security Headers > X-Frame-Options
Priority	P5
Bug URL	https://prod-na04 vercel.ecm.underarmour.com/en-us
Description	This report describes a Missing Anti-clickjacking Header vulnerability. Full technical details and PoC are included in the attached PDF All information includes in pdf file. The real reason to do this bug bounty program is gaining new knowledge about web security as a beginner.
Files attached	Under%20Armour%20-%20Bug%20bounty.pdf (418 KB)

Activity

ApolloCipher created the submission
3 days ago (29 Apr 2025 14:57:42 UTC)

teapot_bugcrowd sent a message
3 days ago (29 Apr 2025 15:08:38 UTC)

Hi

Thank you for your submission, however, this issue is considered to be a P5 (Informational) finding as per Bugcrowd's Vulnerability Rating Taxonomy (VRT). Therefore, this finding typically does not qualify for a reward.

Typically, this is the case when an issue lacks a demonstrated risk and is considered security best practice. While that will apply regularly when we assign a submission a P5 rating, if you are able to exploit this finding further (or chain it with other findings) to meet the definition of another item within the VRT, please do submit a new report. We look forward to reading it!

As you progress with bug bounties it's important to consider not just the vulnerability but also the impact that this vulnerability has, so we encourage you to always explore any finding to better understand the impact it may have. Each submission should aim to answer the question as an attacker I could....

If you're unsure of the next steps to take this with submission, we recommend the Bugcrowd University as a starting point for learning how you can escalate bugs from a P5, into P4s or even P3 findings!

Best regards.
-Bugcrowd Security Operations Team

teapot_bugcrowd changed the state to **Informational**
3 days ago (29 Apr 2025 15:08:40 UTC)

Send a message — Write Preview

To: Everyone

Your message to everyone here...

Embed images by dragging & dropping, **selecting**, or pasting them.

Add attachments

You can attach up to 20 files. Please keep individual upload size under 400MiB.

You can embed attachments (.jpg/.gif/.png, smaller than 5MB) into the Markdown fields. You can copy the embed code using the 'Copy as Markdown' button.

Send message

Status

Informational

This submission was reproducible but is considered informational. It may be that the issue is a best practice recommendation, low risk, has existing mitigations in place, or is an accepted business risk for the customer.

Reward

VRT version

1.15.1

Support

Engagement

Under Armour Product Security

Closed on

29 Apr 2025

CrowdStream visibility

Choose to show your details with this submission in CrowdStream when accepted.

Please note: your username will always be shown with your submission if it is disclosed.

Show username

Show reward

Please note: This engagement does not currently disclose submission activity in CrowdStream.

Disclosure policy

Please note: This engagement does **not allow** disclosure. You may not release information about vulnerabilities found in this engagement to the public.

Need help?

Issues with a submission? Please visit Bugcrowd Support and create a support ticket.

Terms & Conditions Privacy Policy Security Do Not Sell My Information

Docs FAQ Resources Blog Contact Get Help

bugcrowd

This is my first achievement from bug crowd.