Sri Lanka Institute of Information Technology

Bug Bounty - Report 10

Vulnerable JS Library

DOMPurify.version="3.0.3"

Student Name – Wanasinghe N.K

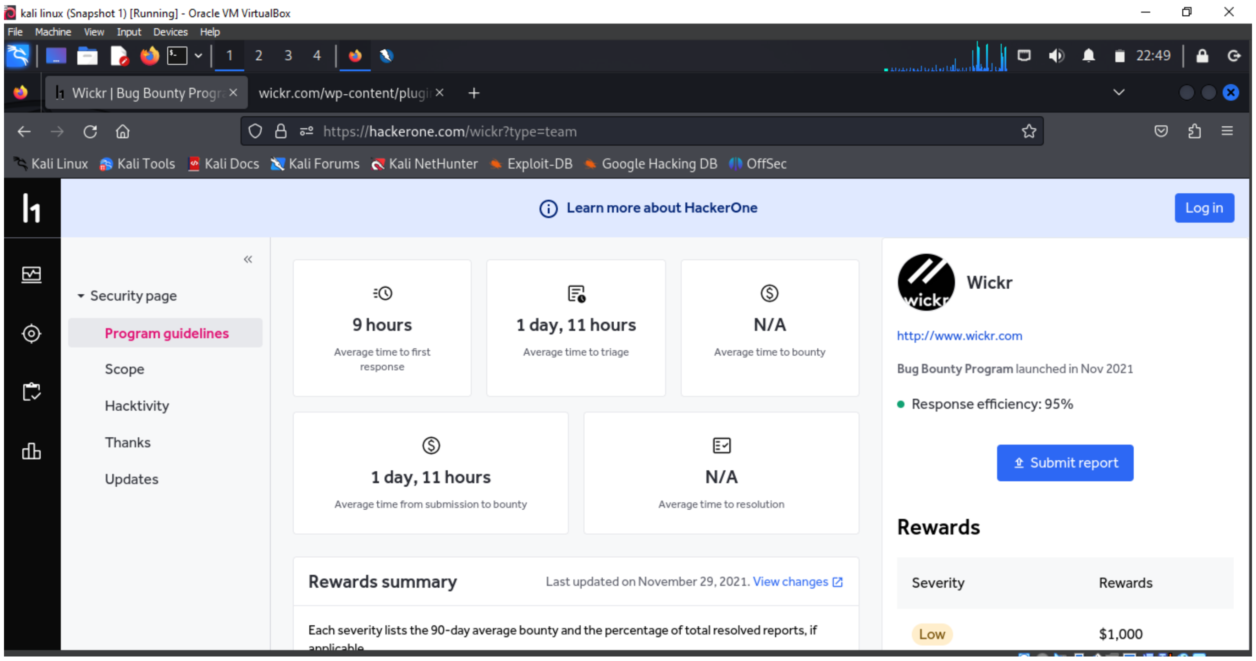
Student ID – IT23221000

IE2062 - Web Security

B.Sc. (Hons) in information Technology Specializing in Cyber Security

Report 10 – wickr.com (Hackerone)

Main domain – www.wickr.com/



9 hours
Average time to first response

1 day, 11 hours
Average time to triage

N/A
Average time to bounty

1 day, 11 hours
Average time from submission to bounty

N/A
Average time to resolution

Rewards summary — Last updated on November 29, 2021. View changes

Each severity lists the 90-day average bounty and the percentage of total resolved reports, if applicable.

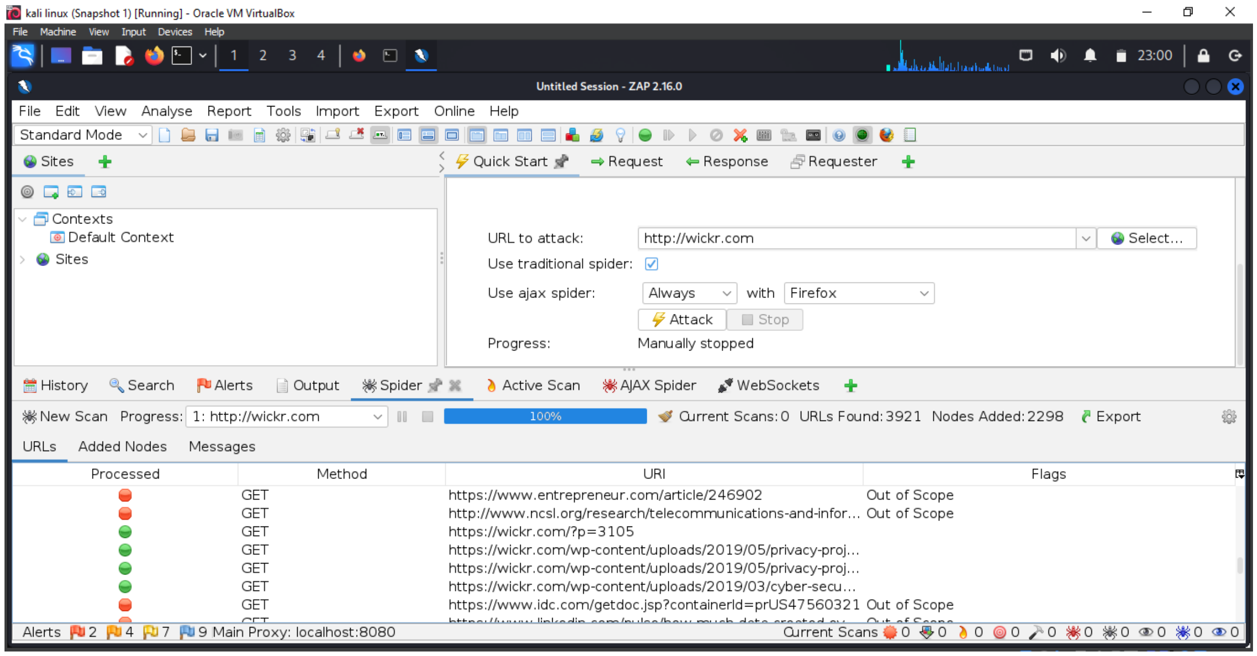
Wickr
http://www.wickr.com
Bug Bounty Program launched in Nov 2021
Response efficiency: 95%

Rewards

Severity	Rewards
Low	$1,000

I used OWASP ZAP tool to scan the website.

I found all the in scope and out of scope domains by running a spider attack.



kali linux (Snapshot 1) [Running] - Oracle VM VirtualBox
Untitled Session - ZAP 2.16.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Contexts
Default Context
Sites
History Search Alerts Output Spider Active Scan AJAX Spider WebSockets
New Scan
Export
URLs Added Nodes Messages

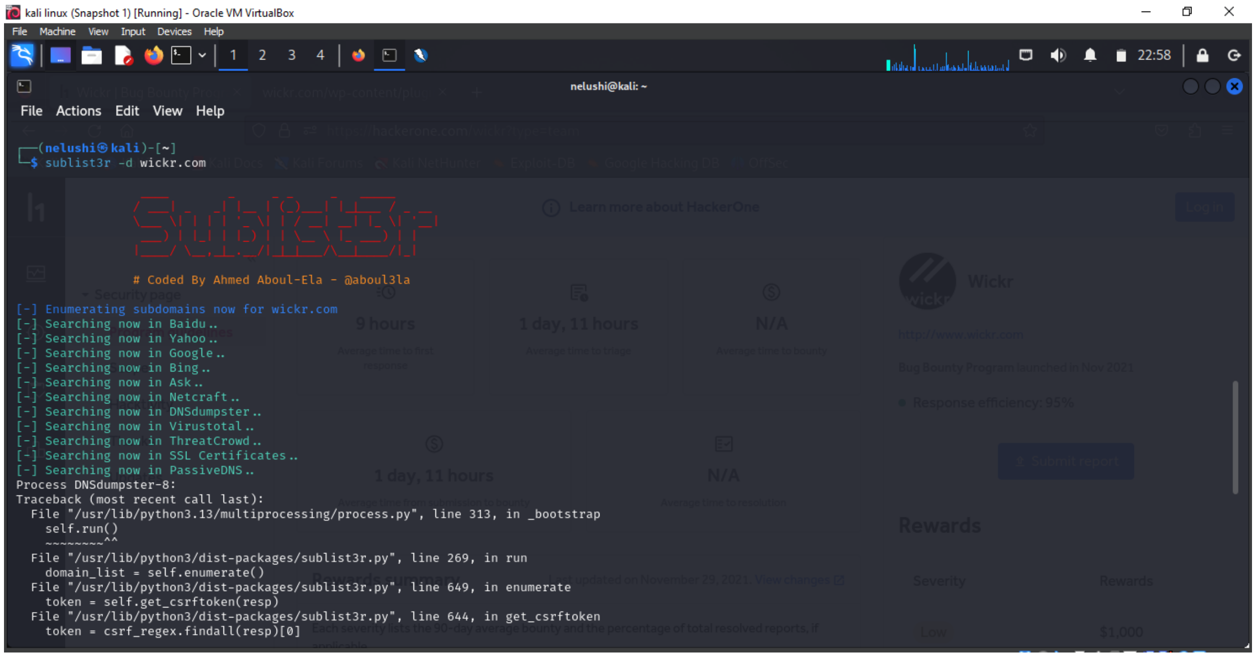
URL to attack: http://wickr.com
Use traditional spider: ✓
Use ajax spider: Always with Firefox
Attack Stop
Progress: Manually stopped

Progress: 1: http://wickr.com 100% Current Scans: 0 URLs Found: 3921 Nodes Added: 2298

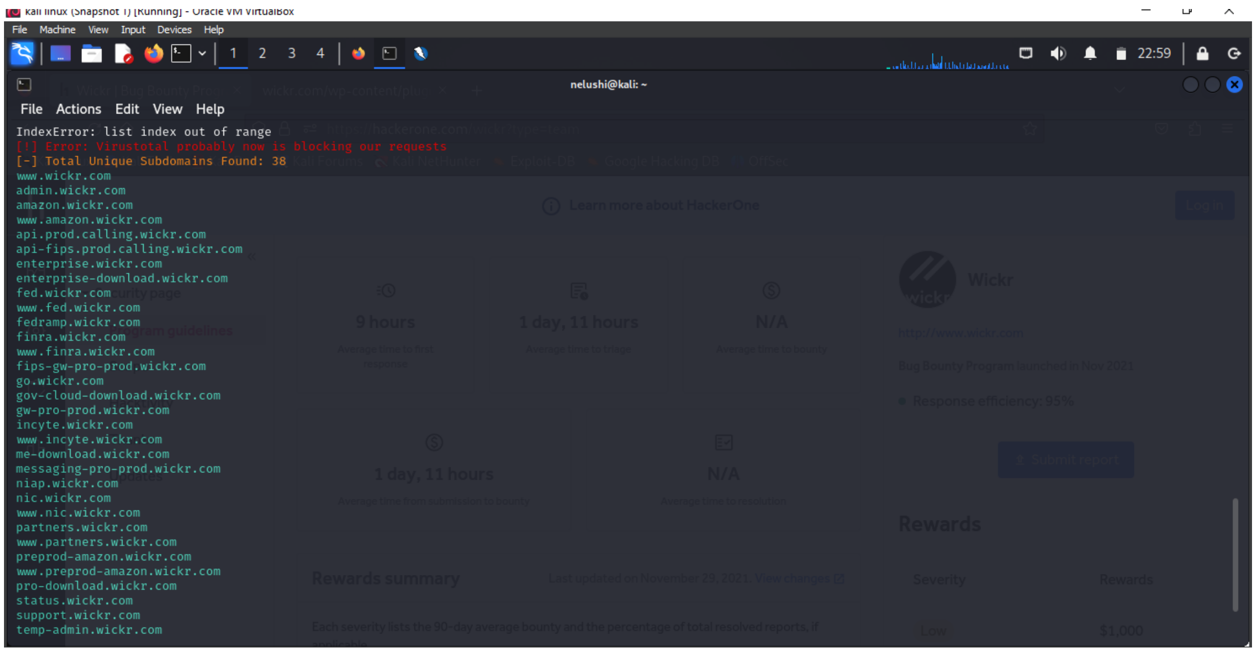
Processed	Method	URI	Flags
🟥	GET	https://www.entrepreneur.com/article/246902	Out of Scope
🟥	GET	http://www.ncsl.org/research/telecommunications-and-infor…	Out of Scope
🟩	GET	https://wickr.com/?p=3105	
🟩	GET	https://wickr.com/wp-content/uploads/2019/05/privacy-proj…	
🟩	GET	https://wickr.com/wp-content/uploads/2019/05/privacy-proj…	
🟩	GET	https://wickr.com/wp-content/uploads/2019/03/cyber-secu…	
🟥	GET	https://www.idc.com/getdoc.jsp?containerId=prUS47560321	Out of Scope

Alerts 2 4 7 9 Main Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0 0

Then I used **sublist3r tool** to find the subdomains of the webpage.


kali linux (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
22:58
nelushi@kali: ~
File Actions Edit View Help

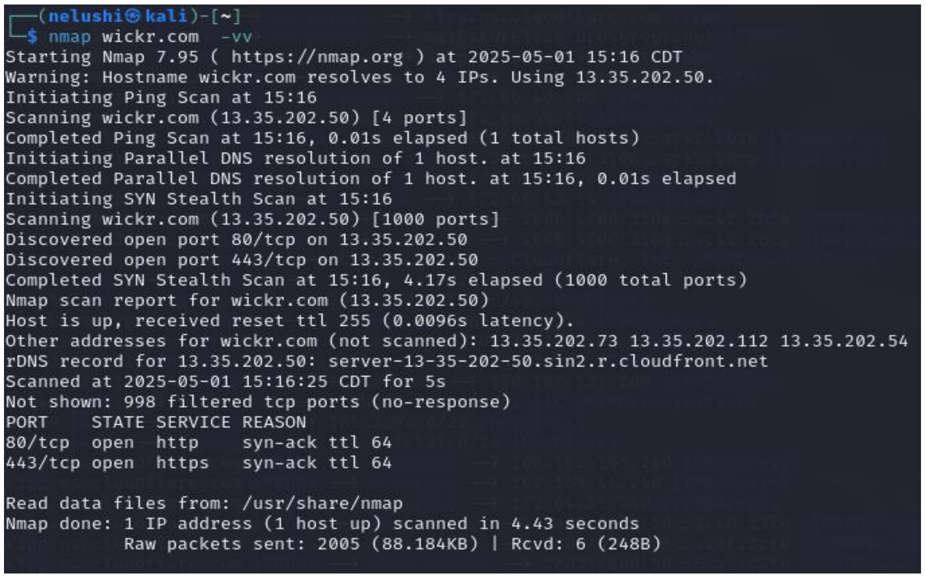
```
┌──(nelushi㉿kali)-[~]
└─$ sublist3r -d wickr.com

                 # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for wickr.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..
Process DNSdumpster-8:
Traceback (most recent call last):
  File "/usr/lib/python3.13/multiprocessing/process.py", line 313, in _bootstrap
    self.run()
    ~~~~~~~~^^
  File "/usr/lib/python3/dist-packages/sublist3r.py", line 269, in run
    domain_list = self.enumerate()
  File "/usr/lib/python3/dist-packages/sublist3r.py", line 649, in enumerate
    token = self.get_csrftoken(resp)
  File "/usr/lib/python3/dist-packages/sublist3r.py", line 644, in get_csrftoken
    token = csrf_regex.findall(resp)[0]
```


kali linux (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
22:59
nelushi@kali: ~
File Actions Edit View Help

```
IndexError: list index out of range
[!] Error: Virustotal probably now is blocking our requests
[-] Total Unique Subdomains Found: 38
www.wickr.com
admin.wickr.com
amazon.wickr.com
www.amazon.wickr.com
api.prod.calling.wickr.com
api-fips.prod.calling.wickr.com
enterprise.wickr.com
enterprise-download.wickr.com
fed.wickr.com
www.fed.wickr.com
fedramp.wickr.com
finra.wickr.com
www.finra.wickr.com
fips-gw-pro-prod.wickr.com
go.wickr.com
gov-cloud-download.wickr.com
gw-pro-prod.wickr.com
incyte.wickr.com
www.incyte.wickr.com
me-download.wickr.com
messaging-pro-prod.wickr.com
niap.wickr.com
nic.wickr.com
www.nic.wickr.com
partners.wickr.com
www.partners.wickr.com
preprod-amazon.wickr.com
www.preprod-amazon.wickr.com
pro-download.wickr.com
status.wickr.com
support.wickr.com
temp-admin.wickr.com
```

Nmap – Network scanning and enumeration

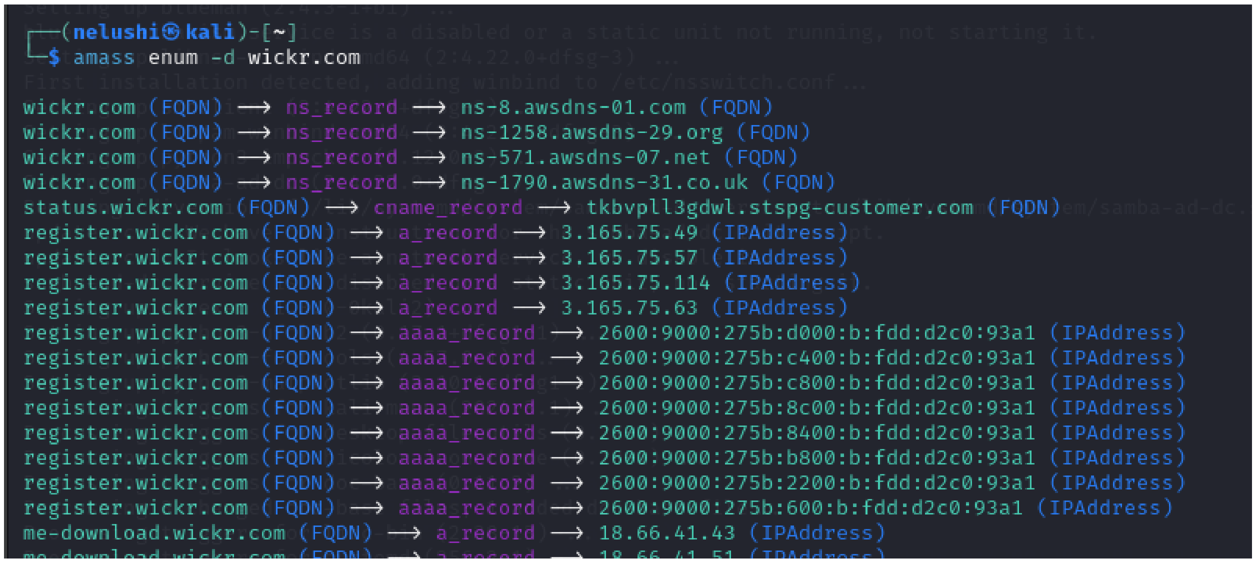
I found all the open ports and detected the running services on the target server using Nmap.



```
┌──(nelushi㉿kali)-[~]
└─$ nmap wickr.com -vv
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-01 15:16 CDT
Warning: Hostname wickr.com resolves to 4 IPs. Using 13.35.202.50.
Initiating Ping Scan at 15:16
Scanning wickr.com (13.35.202.50) [4 ports]
Completed Ping Scan at 15:16, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:16
Completed Parallel DNS resolution of 1 host. at 15:16, 0.01s elapsed
Initiating SYN Stealth Scan at 15:16
Scanning wickr.com (13.35.202.50) [1000 ports]
Discovered open port 80/tcp on 13.35.202.50
Discovered open port 443/tcp on 13.35.202.50
Completed SYN Stealth Scan at 15:16, 4.17s elapsed (1000 total ports)
Nmap scan report for wickr.com (13.35.202.50)
Host is up, received reset ttl 255 (0.0096s latency).
Other addresses for wickr.com (not scanned): 13.35.202.73 13.35.202.112 13.35.202.54
rDNS record for 13.35.202.50: server-13-35-202-50.sin2.r.cloudfront.net
Scanned at 2025-05-01 15:16:25 CDT for 5s
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE REASON
80/tcp  open  http    syn-ack ttl 64
443/tcp open  https   syn-ack ttl 64

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 4.43 seconds
           Raw packets sent: 2005 (88.184KB) | Rcvd: 6 (248B)
```

Amass – Subdomain and DNS mapping

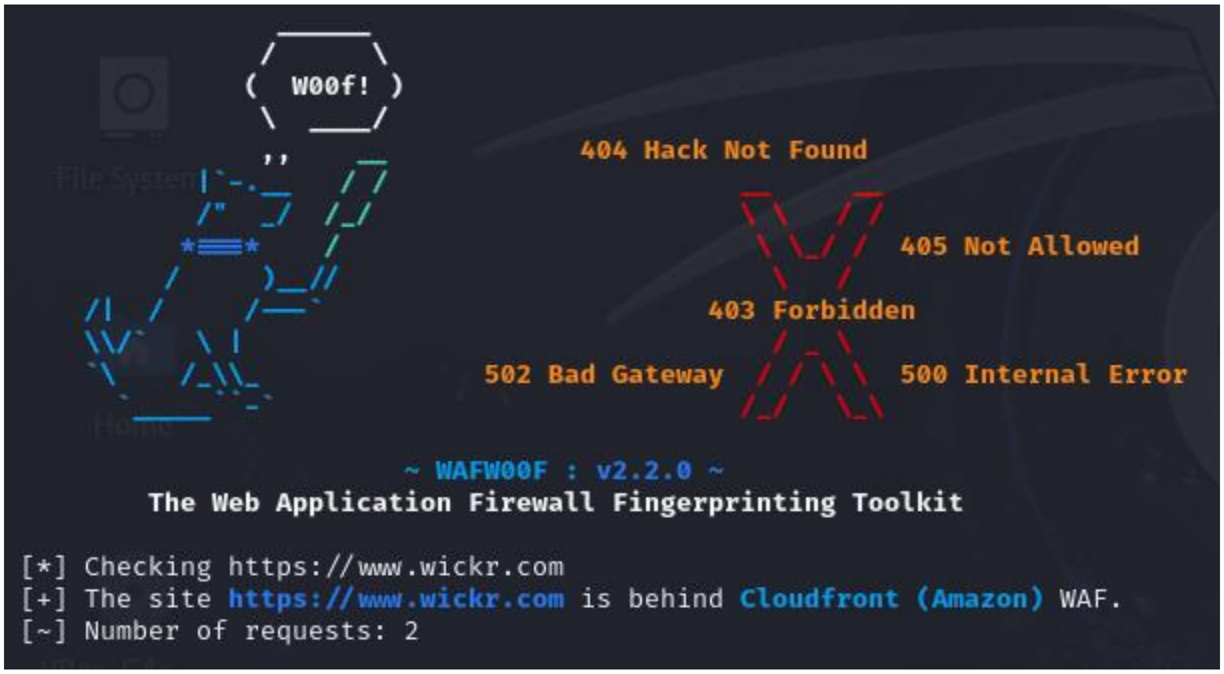
I found all the subdomains related to the target domain using Amass.



```
┌──(nelushi㉿kali)-[~]
└─$ amass enum -d wickr.com
wickr.com (FQDN) ⟶ ns_record ⟶ ns-8.awsdns-01.com (FQDN)
wickr.com (FQDN) ⟶ ns_record ⟶ ns-1258.awsdns-29.org (FQDN)
wickr.com (FQDN) ⟶ ns_record ⟶ ns-571.awsdns-07.net (FQDN)
wickr.com (FQDN) ⟶ ns_record ⟶ ns-1790.awsdns-31.co.uk (FQDN)
status.wickr.com (FQDN) ⟶ cname_record ⟶ tkbvpll3gdwl.stspg-customer.com (FQDN)
register.wickr.com (FQDN) ⟶ a_record ⟶ 3.165.75.49 (IPAddress)
register.wickr.com (FQDN) ⟶ a_record ⟶ 3.165.75.57 (IPAddress)
register.wickr.com (FQDN) ⟶ a_record ⟶ 3.165.75.114 (IPAddress)
register.wickr.com (FQDN) ⟶ a_record ⟶ 3.165.75.63 (IPAddress)
register.wickr.com (FQDN) ⟶ aaaa_record ⟶ 2600:9000:275b:d000:b:fdd:d2c0:93a1 (IPAddress)
register.wickr.com (FQDN) ⟶ aaaa_record ⟶ 2600:9000:275b:c400:b:fdd:d2c0:93a1 (IPAddress)
register.wickr.com (FQDN) ⟶ aaaa_record ⟶ 2600:9000:275b:c800:b:fdd:d2c0:93a1 (IPAddress)
register.wickr.com (FQDN) ⟶ aaaa_record ⟶ 2600:9000:275b:8c00:b:fdd:d2c0:93a1 (IPAddress)
register.wickr.com (FQDN) ⟶ aaaa_record ⟶ 2600:9000:275b:8400:b:fdd:d2c0:93a1 (IPAddress)
register.wickr.com (FQDN) ⟶ aaaa_record ⟶ 2600:9000:275b:b800:b:fdd:d2c0:93a1 (IPAddress)
register.wickr.com (FQDN) ⟶ aaaa_record ⟶ 2600:9000:275b:2200:b:fdd:d2c0:93a1 (IPAddress)
register.wickr.com (FQDN) ⟶ aaaa_record ⟶ 2600:9000:275b:600:b:fdd:d2c0:93a1 (IPAddress)
me-download.wickr.com (FQDN) ⟶ a_record ⟶ 18.66.41.43 (IPAddress)
me-download.wickr.com (FQDN) ⟶ a_record ⟶ 18.66.41.51 (IPAddress)
```

Wafw00f – Firewall Detection

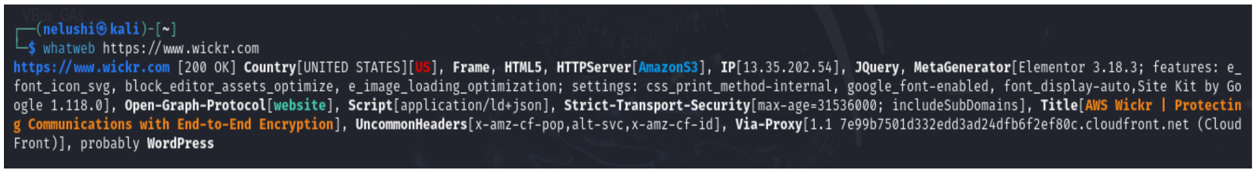
Command used – wafw00f https://www.wickr.com



```
                ______
               /      \
              (  W00f! )
               \  ____/
               ,,    __            404 Hack Not Found
           |`-.__   / /                      __     __
           /"  _/  /_/                       \ \   / /         405 Not Allowed
          *===*    /                          \ \_/ /
         /        )__//                        \   /           403 Forbidden
     /|  /       /---`                          / \
     \\/`    \ |              502 Bad Gateway  / /\ \          500 Internal Error
      `\    /_\\_                             /_/  \_\
           `----`

                    ~ WAFW00F : v2.2.0 ~
    The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://www.wickr.com
[+] The site https://www.wickr.com is behind Cloudfront (Amazon) WAF.
[~] Number of requests: 2
```

Whatweb – to identify the technologies used by the site.

Commans used – whatweb https://www.wickr.com



```
┌──(nelushi㉿kali)-[~]
└─$ whatweb https://www.wickr.com
https://www.wickr.com [200 OK] Country[UNITED STATES][US], Frame, HTML5, HTTPServer[AmazonS3], IP[13.35.202.54], JQuery, MetaGenerator[Elementor 3.18.3; features: e_
font_icon_svg, block_editor_assets_optimize, e_image_loading_optimization; settings: css_print_method-internal, google_font-enabled, font_display-auto,Site Kit by Go
ogle 1.118.0], Open-Graph-Protocol[website], Script[application/ld+json], Strict-Transport-Security[max-age=31536000; includeSubDomains], Title[AWS Wickr | Protectin
g Communications with End-to-End Encryption], UncommonHeaders[x-amz-cf-pop,alt-svc,x-amz-cf-id], Via-Proxy[1.1 7e99b7501d332edd3ad24dfb6f2ef80c.cloudfront.net (Cloud
Front)], probably WordPress
```

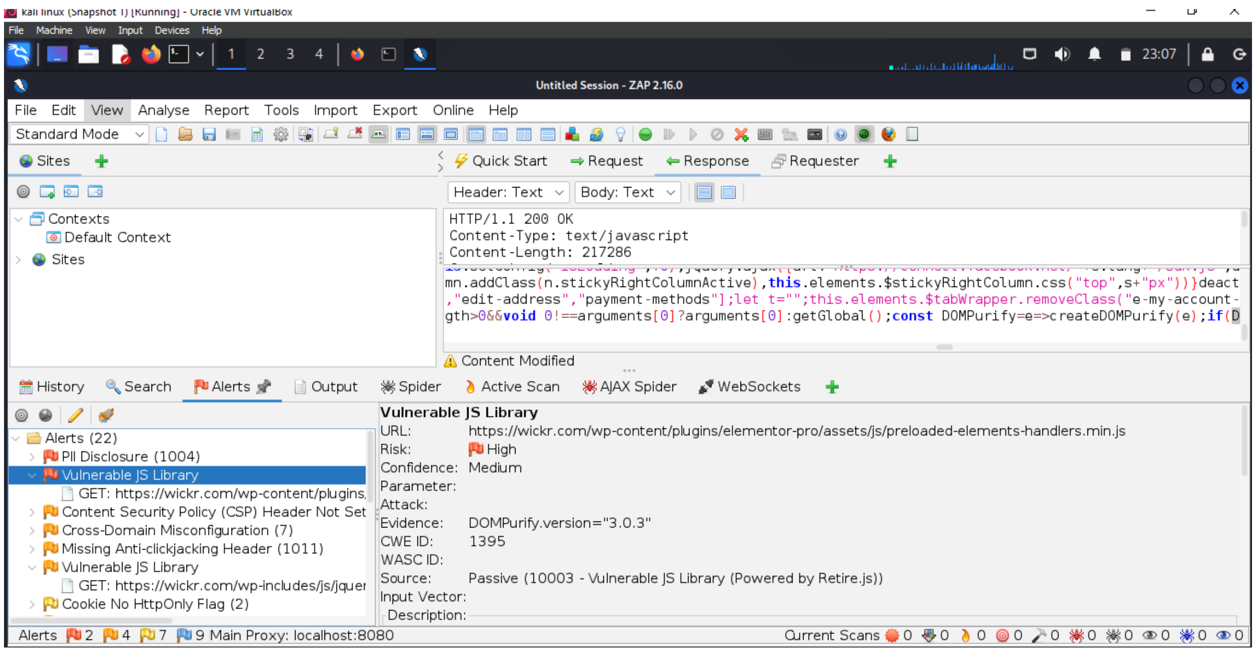
Vulnerability 01

Domain

https://wickr.com/wp-content/plugins/elementor-pro/assets/js/preloaded-elements-handlers.min.js

Vulnerability title

Vulnerable JS Library - DOMPurify.version="3.0.3"


kali linux (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
23:07
Untitled Session - ZAP 2.16.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Header: Text
Body: Text
Contexts
Default Context
Sites
History
Search
Alerts
Output
Spider
Active Scan
AJAX Spider
WebSockets

```
HTTP/1.1 200 OK
Content-Type: text/javascript
Content-Length: 217286
mn.addClass(n.stickyRightColumnActive),this.elements.$stickyRightColumn.css("top",s+"px"))}deact
,"edit-address","payment-methods"];let t="";this.elements.$tabWrapper.removeClass("e-my-account-
gth>0&&void 0!==arguments[0]?arguments[0]:getGlobal();const DOMPurify=e=>createDOMPurify(e);if(D
```

Content Modified

```
Alerts (22)
  PII Disclosure (1004)
  Vulnerable JS Library
    GET: https://wickr.com/wp-content/plugins,
  Content Security Policy (CSP) Header Not Set
  Cross-Domain Misconfiguration (7)
  Missing Anti-clickjacking Header (1011)
  Vulnerable JS Library
    GET: https://wickr.com/wp-includes/js/jquer
  Cookie No HttpOnly Flag (2)
```

```
Vulnerable JS Library
URL:          https://wickr.com/wp-content/plugins/elementor-pro/assets/js/preloaded-elements-handlers.min.js
Risk:         High
Confidence:   Medium
Parameter:
Attack:
Evidence:     DOMPurify.version="3.0.3"
CWE ID:       1395
WASC ID:
Source:       Passive (10003 - Vulnerable JS Library (Powered by Retire.js))
Input Vector:
Description:
```

```
Alerts 2 4 7 9 Main Proxy: localhost:8080        Current Scans 0 0 0 0 0 0 0 0 0 0
```

Vulnerability description

Vulnerable JS Library Vulnerability

The website third party is using an old or vulnerable JavaScript library for the web application. This would initiate this type of Vulnerable JS Library exposure.

Different types of security issues can arise due to this, but the most important ones are cross-site scripting (XSS), output data leakage, and unauthorized access.

This kind of attack may be performed when an attacker injects a malicious script inside the web application and affects either the user or the system in question with that particular implementation.

Vulnerability in DOMPurify v3.0.3:

It is a JavaScript library used for HTML sanitization and XSS protection by purifying input given by users.

DOMPurify version 3.0.3 was found to have some vulnerability.

Possible Issue: DOMPurify fails to cleanse some user inputs and exposes the application to XSS attacks where scripts can be executed directly on the user's browser.

Attack: Attackers can insert scripts into inputs or URLs and harvest user data and execute code without any permission in the browser.

Key Points:

Although it is meant for input sanitization, version 3.0.3 of DOMPurify often fails to consider some edge cases or inputs and it could give ways for attack through injection of malicious scripts leading to data theft, session hijacking or performance of other malicious activities.

Upgrading to the latest patched version of DOMPurify will eliminate the risk and ensure sanitization procedures are performed accurately.

Affected Components

- DOMPurify v3.0.3

The vulnerable version is capable of sanitizing HTML input and prone to an XSS attack.

- Web Applications

All web applications that use DOMPurify to sanitize user input will be affected.

For example, form submissions, file uploads, and comments.

- Content Management Systems

If using DOMPurify v3.0.3 for sanitization, WordPress, Joomla, etc., will also be vulnerable.

- Web frameworks

React, Angular, Vue.js apps that sanitize HTML with the help of DOMPurify.

- Frontend Libraries

A UI library that depends on this version of DOMPurify for sanitization.

- Custom Applications

Any application that carries out unconventional HTML sanitization procedures which refer back to DOMPurify may be compromised.

Impact assessment

Severity – High

- Security implications

XSS vulnerabilities can be employed to enable attackers to inject harmful client-side scripts on the trusted sites and, as a result, compromise the user data.

- Data theft

The malicious scripts can steal sensitive data such as user login details, personal data, or even session cookies.

- Negative impact on reputation

The vulnerability can negatively impact the reputation of all the concerned organizations by triggering distrust against their platforms.

- Legal risks and compliance issues

Disclosure of any user information or personally identifiable information (PII) is sure to trigger offenses under the personal data protection legislations (ex: GDPR, CCPA) with penalties as a punishment.

- Disruption of Services

XSS attacks may also lead to a denial-of-service attack (DoS) by crashing the site or extreme malfunctioning of the user interface.

- Access Control Issues

Attackers use unauthorized access to restricted areas in the web application by elevating themselves to higher privileges.

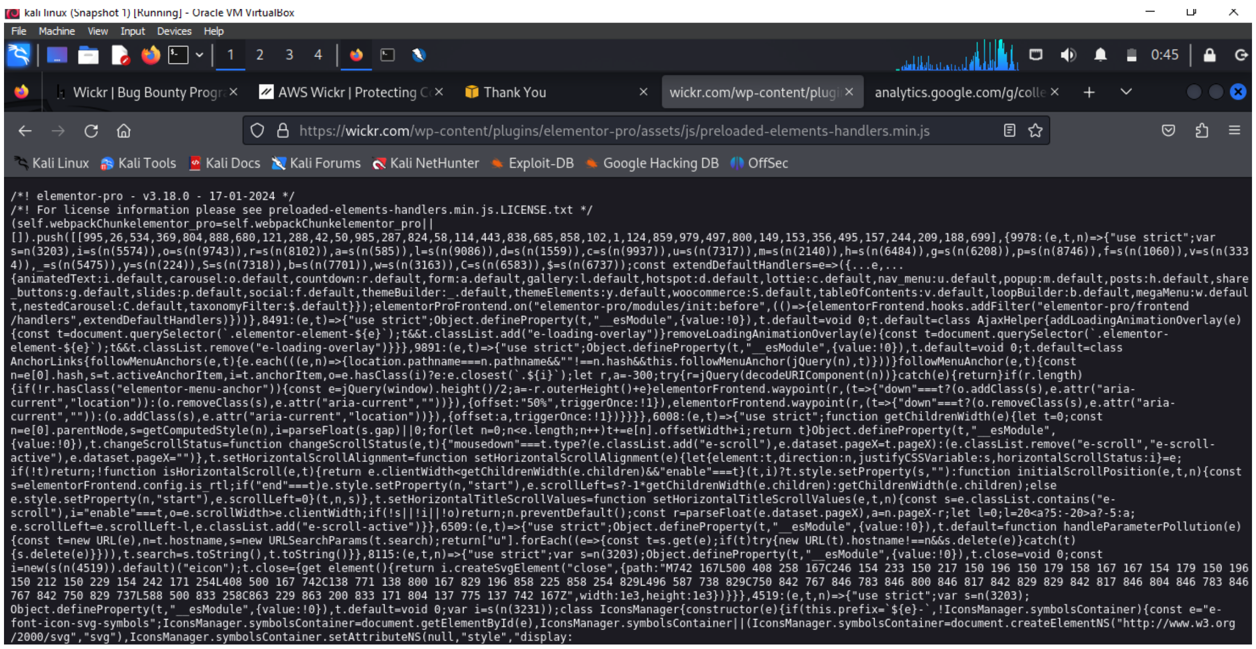
Overall Effect:

The vulnerability can result in extreme security compromise, loss of information, and negative impact. It should be fixed as soon as possible, updating to the latest version of DOMPurify, is highly recommended.

Steps to reproduce with Proof of Concept (poc)

1. First, I ensured that the js file is being included correctly on the page.

To check that I open the JS file URL (https://wickr.com/wp-content/plugins/elementor-pro/assets/js/preloaded-elements-handlers.min.js).

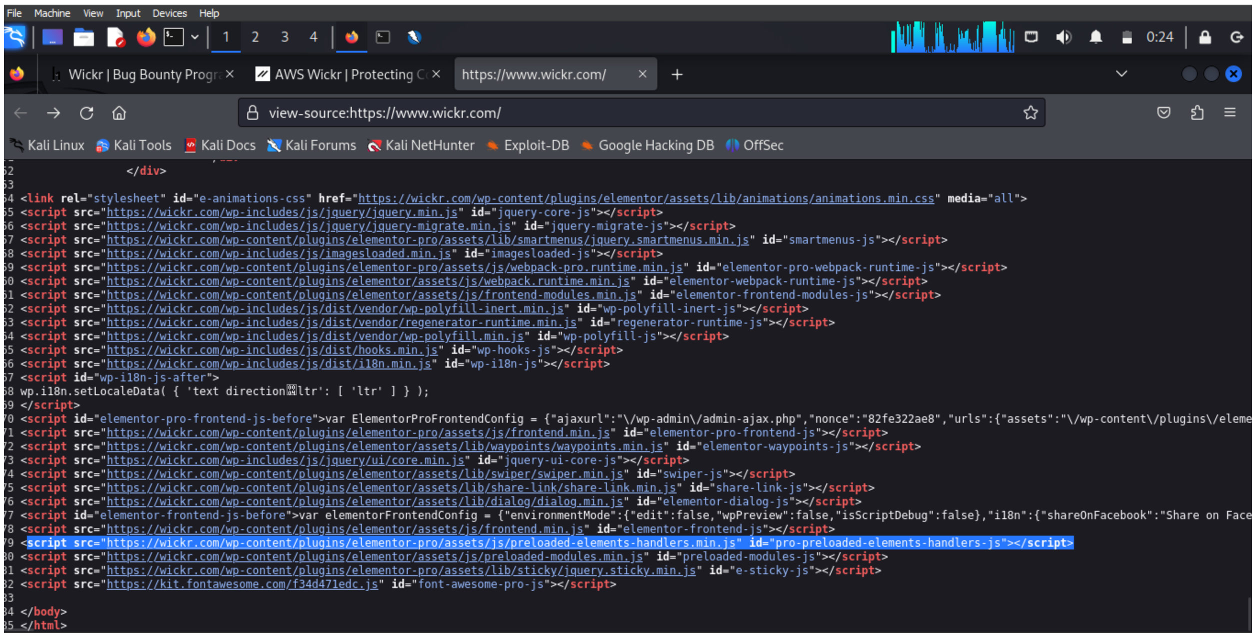

kali linux (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
0:45
Wickr | Bug Bounty Progr
AWS Wickr | Protecting C
Thank You
wickr.com/wp-content/plugi
analytics.google.com/g/colle
https://wickr.com/wp-content/plugins/elementor-pro/assets/js/preloaded-elements-handlers.min.js
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec

```
/*! elementor-pro - v3.18.0 - 17-01-2024 */
/*! For license information please see preloaded-elements-handlers.min.js.LICENSE.txt */
(self.webpackChunkelementor_pro=self.webpackChunkelementor_pro||
[]).push([[995,26,534,369,804,888,680,121,288,42,50,985,287,824,58,114,443,838,685,858,102,1,124,859,979,497,800,149,153,356,495,157,244,209,188,699],{9978:(e,t,n)=>{"use strict";var
s=n(3203),i=s(n(5574)),o=s(n(9743)),r=s(n(8102)),a=s(n(585)),l=s(n(9086)),d=s(n(1559)),c=s(n(9937)),u=s(n(7317)),m=s(n(2140)),h=s(n(6484)),g=s(n(6208)),p=s(n(8746)),f=s(n(1060)),v=s(n(333
4)),_=s(n(5475)),y=s(n(224)),S=s(n(7318)),b=s(n(7701)),w=s(n(3163)),C=s(n(6583)),$=s(n(6737));const extendDefaultHandlers=e=>({...e,...
{animatedText:i.default,carousel:o.default,countdown:r.default,form:a.default,gallery:l.default,hotspot:d.default,lottie:c.default,nav_menu:u.default,popup:m.default,posts:h.default,share
_buttons:g.default,slides:p.default,social:f.default,themeBuilder:_.default,themeElements:y.default,woocommerce:S.default,tableOfContents:v.default,loopBuilder:b.default,megaMenu:w.defaul
t,nestedCarousel:C.default,taxonomyFilter:$.default}});elementorProFrontend.on("elementor-pro/modules/init:before",(()=>{elementorFrontend.hooks.addFilter("elementor-pro/frontend
/handlers",extendDefaultHandlers)}))},8491:(e,t)=>{"use strict";Object.defineProperty(t,"__esModule",{value:!0}),t.default=void 0;t.default=class AjaxHelper{addLoadingAnimationOverlay(e)
{const t=document.querySelector(`.elementor-element-${e}`);t&&t.classList.add("e-loading-overlay")}removeLoadingAnimationOverlay(e){const t=document.querySelector(`.elementor-
element-${e}`);t&&t.classList.remove("e-loading-overlay")}}},9891:(e,t)=>{"use strict";Object.defineProperty(t,"__esModule",{value:!0}),t.default=void 0;t.default=class
AnchorLinks{followMenuAnchors(e,t){e.each(((e,n)=>{location.pathname===n.pathname&&""!==n.hash&&this.followMenuAnchor(jQuery(n),t)}))}followMenuAnchor(e,t){const
n=e[0].hash,s=t.activeAnchorItem,i=t.anchorItem,o=e.hasClass(i)?e:e.closest(`.${i}`);let r,a=-300;try{r=jQuery(decodeURIComponent(n))}catch(e){return}if(r.length)
{if(!r.hasClass("elementor-menu-anchor")){const e=jQuery(window).height()/2;a=-r.outerHeight()+e}elementorFrontend.waypoint(r,(t=>{"down"===t?(o.addClass(s),e.attr("aria-
current","location")):(o.removeClass(s),e.attr("aria-current",""))}),{offset:"50%",triggerOnce:!1}),elementorFrontend.waypoint(r,(t=>{"down"===t?(o.removeClass(s),e.attr("aria-
current","")):(o.addClass(s),e.attr("aria-current","location"))}),{offset:a,triggerOnce:!1})}}}},6008:(e,t)=>{"use strict";function getChildrenWidth(e){let t=0;const
n=e[0].parentNode,s=getComputedStyle(n),i=parseFloat(s.gap)||0;for(let n=0;n<e.length;n++)t+=e[n].offsetWidth+i;return t}Object.defineProperty(t,"__esModule",
{value:!0}),t.changeScrollStatus=function changeScrollStatus(e,t){"mousedown"===t.type?(e.classList.add("e-scroll"),e.dataset.pageX=t.pageX):(e.classList.remove("e-scroll","e-scroll-
active"),e.dataset.pageX="")},t.setHorizontalScrollAlignment=function setHorizontalScrollAlignment(e){let{element:t,direction:n,justifyCSSVariable:s,horizontalScrollStatus:i}=e;
if(!t)return;!function isHorizontalScroll(e,t){return e.clientWidth<getChildrenWidth(e.children)&&"enable"===t}(t,i)?t.style.setProperty(s,""):function initialScrollPosition(e,t,n){const
s=elementorFrontend.config.is_rtl;if("end"===t)e.style.setProperty(n,"start"),e.scrollLeft=s?-1*getChildrenWidth(e.children):getChildrenWidth(e.children);else
e.style.setProperty(n,"start"),e.scrollLeft=0}(t,n,s)},t.setHorizontalTitleScrollValues=function setHorizontalTitleScrollValues(e,t,n){const s=e.classList.contains("e-
scroll"),i="enable"===t,o=e.scrollWidth>e.clientWidth;if(!s||!i||!o)return;n.preventDefault();const r=parseFloat(e.dataset.pageX),a=n.pageX-r;let l=0;l=20<a?5:-20>a?-5:a;
e.scrollLeft=e.scrollLeft-l,e.classList.add("e-scroll-active")}},6509:(e,t)=>{"use strict";Object.defineProperty(t,"__esModule",{value:!0}),t.default=function handleParameterPollution(e)
{const t=new URL(e),n=t.hostname,s=new URLSearchParams(t.search);return["u"].forEach((e=>{const t=s.get(e);if(t)try{new URL(t).hostname!==n&&s.delete(e)}catch(t)
{s.delete(e)}})),t.search=s.toString(),t.toString()}},8115:(e,t,n)=>{"use strict";var s=n(3203);Object.defineProperty(t,"__esModule",{value:!0}),t.close=void 0;const
i=new(s(n(4519)).default)("eicon");t.close={get element(){return i.createSvgElement("close",{path:"M742 167L500 408 258 167C246 154 233 150 217 150 196 150 179 158 167 167 154 179 150 196
150 212 150 229 154 242 171 254L408 500 167 742C138 771 138 800 167 829 196 858 225 858 254 829L496 587 738 829C750 842 767 846 783 846 800 846 817 842 829 829 842 817 846 804 846 783 846
767 842 750 829 737L588 500 833 258C863 229 863 200 833 171 804 137 775 137 742 167Z",width:1e3,height:1e3})}}},4519:(e,t,n)=>{"use strict";var s=n(3203);
Object.defineProperty(t,"__esModule",{value:!0}),t.default=void 0;var i=s(n(3231));class IconsManager{constructor(e){if(this.prefix=`${e}-`,!IconsManager.symbolsContainer){const e="e-
font-icon-svg-symbols";IconsManager.symbolsContainer=document.getElementById(e),IconsManager.symbolsContainer||(IconsManager.symbolsContainer=document.createElementNS("http://www.w3.org
/2000/svg","svg"),IconsManager.symbolsContainer.setAttributeNS(null,"style","display:
```

I saw that I can access it.

2. Then, I inspect whether this file is included in the page's HTML <script> tags (in the source code).

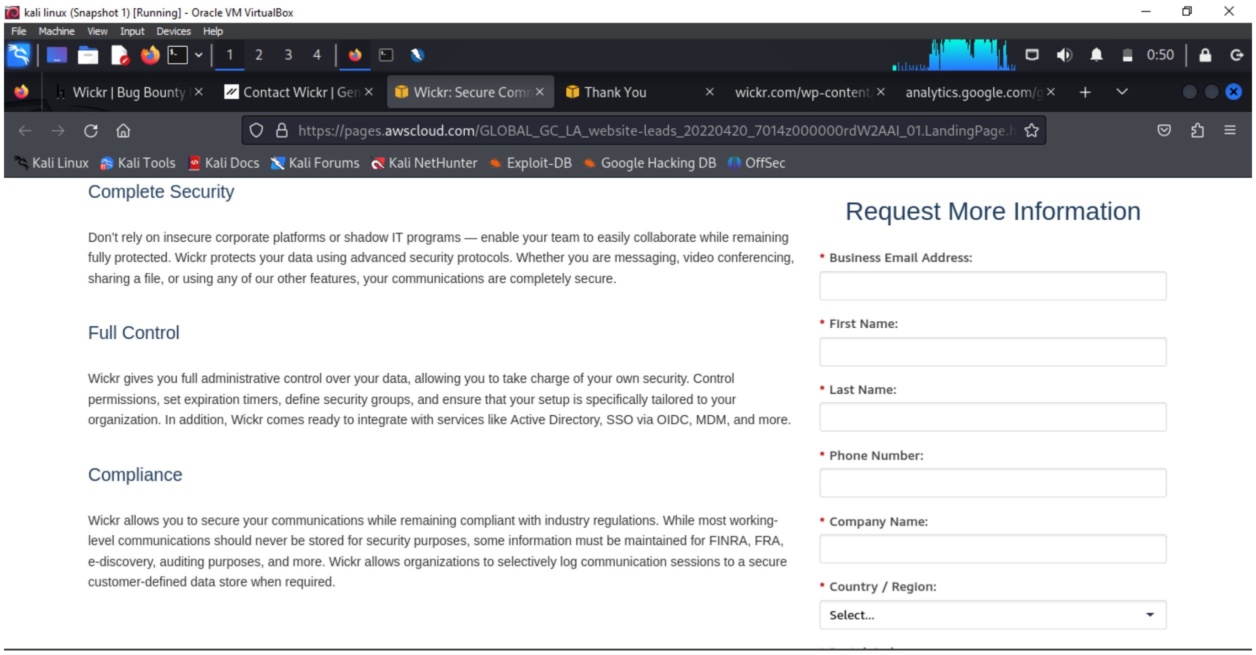
And it is included.



```
            </div>

<link rel="stylesheet" id="e-animations-css" href="https://wickr.com/wp-content/plugins/elementor/assets/lib/animations/animations.min.css" media="all">
<script src="https://wickr.com/wp-includes/js/jquery/jquery.min.js" id="jquery-core-js"></script>
<script src="https://wickr.com/wp-includes/js/jquery/jquery-migrate.min.js" id="jquery-migrate-js"></script>
<script src="https://wickr.com/wp-content/plugins/elementor-pro/assets/lib/smartmenus/jquery.smartmenus.min.js" id="smartmenus-js"></script>
<script src="https://wickr.com/wp-includes/js/imagesloaded.min.js" id="imagesloaded-js"></script>
<script src="https://wickr.com/wp-content/plugins/elementor-pro/assets/js/webpack-pro.runtime.min.js" id="elementor-pro-webpack-runtime-js"></script>
<script src="https://wickr.com/wp-content/plugins/elementor/assets/js/webpack.runtime.min.js" id="elementor-webpack-runtime-js"></script>
<script src="https://wickr.com/wp-content/plugins/elementor/assets/js/frontend-modules.min.js" id="elementor-frontend-modules-js"></script>
<script src="https://wickr.com/wp-includes/js/dist/vendor/wp-polyfill-inert.min.js" id="wp-polyfill-inert-js"></script>
<script src="https://wickr.com/wp-includes/js/dist/vendor/regenerator-runtime.min.js" id="regenerator-runtime-js"></script>
<script src="https://wickr.com/wp-includes/js/dist/vendor/wp-polyfill.min.js" id="wp-polyfill-js"></script>
<script src="https://wickr.com/wp-includes/js/dist/hooks.min.js" id="wp-hooks-js"></script>
<script src="https://wickr.com/wp-includes/js/dist/i18n.min.js" id="wp-i18n-js"></script>
<script id="wp-i18n-js-after">
wp.i18n.setLocaleData( { 'text direction ltr': [ 'ltr' ] } );
</script>
<script id="elementor-pro-frontend-js-before">var ElementorProFrontendConfig = {"ajaxurl":"\/wp-admin\/admin-ajax.php","nonce":"82fe322ae8","urls":{"assets":"\/wp-content\/plugins\/eleme
<script src="https://wickr.com/wp-content/plugins/elementor-pro/assets/js/frontend.min.js" id="elementor-pro-frontend-js"></script>
<script src="https://wickr.com/wp-content/plugins/elementor/assets/lib/waypoints/waypoints.min.js" id="elementor-waypoints-js"></script>
<script src="https://wickr.com/wp-includes/js/jquery/ui/core.min.js" id="jquery-ui-core-js"></script>
<script src="https://wickr.com/wp-content/plugins/elementor/assets/lib/swiper/swiper.min.js" id="swiper-js"></script>
<script src="https://wickr.com/wp-content/plugins/elementor/assets/lib/share-link/share-link.min.js" id="share-link-js"></script>
<script src="https://wickr.com/wp-content/plugins/elementor/assets/lib/dialog/dialog.min.js" id="elementor-dialog-js"></script>
<script id="elementor-frontend-js-before">var elementorFrontendConfig = {"environmentMode":{"edit":false,"wpPreview":false,"isScriptDebug":false},"i18n":{"shareOnFacebook":"Share on Face
<script src="https://wickr.com/wp-content/plugins/elementor/assets/js/frontend.min.js" id="elementor-frontend-js"></script>
<script src="https://wickr.com/wp-content/plugins/elementor-pro/assets/js/preloaded-elements-handlers.min.js" id="pro-preloaded-elements-handlers-js"></script>
<script src="https://wickr.com/wp-content/plugins/elementor/assets/js/preloaded-modules.min.js" id="preloaded-modules-js"></script>
<script src="https://wickr.com/wp-content/plugins/elementor-pro/assets/lib/sticky/jquery.sticky.min.js" id="e-sticky-js"></script>
<script src="https://kit.fontawesome.com/f34d471edc.js" id="font-awesome-pro-js"></script>

</body>
</html>
```

3. I looked for an input field on the page where users can submit data. This could be a search bar, comment box, or form field.

   My goal was to input data that might be reflected or displayed back on the page without being sanitized.



Complete Security

Don't rely on insecure corporate platforms or shadow IT programs — enable your team to easily collaborate while remaining fully protected. Wickr protects your data using advanced security protocols. Whether you are messaging, video conferencing, sharing a file, or using any of our other features, your communications are completely secure.

Full Control

Wickr gives you full administrative control over your data, allowing you to take charge of your own security. Control permissions, set expiration timers, define security groups, and ensure that your setup is specifically tailored to your organization. In addition, Wickr comes ready to integrate with services like Active Directory, SSO via OIDC, MDM, and more.

Compliance

Wickr allows you to secure your communications while remaining compliant with industry regulations. While most working-level communications should never be stored for security purposes, some information must be maintained for FINRA, FRA, e-discovery, auditing purposes, and more. Wickr allows organizations to selectively log communication sessions to a secure customer-defined data store when required.

Request More Information

* Business Email Address:

* First Name:

* Last Name:

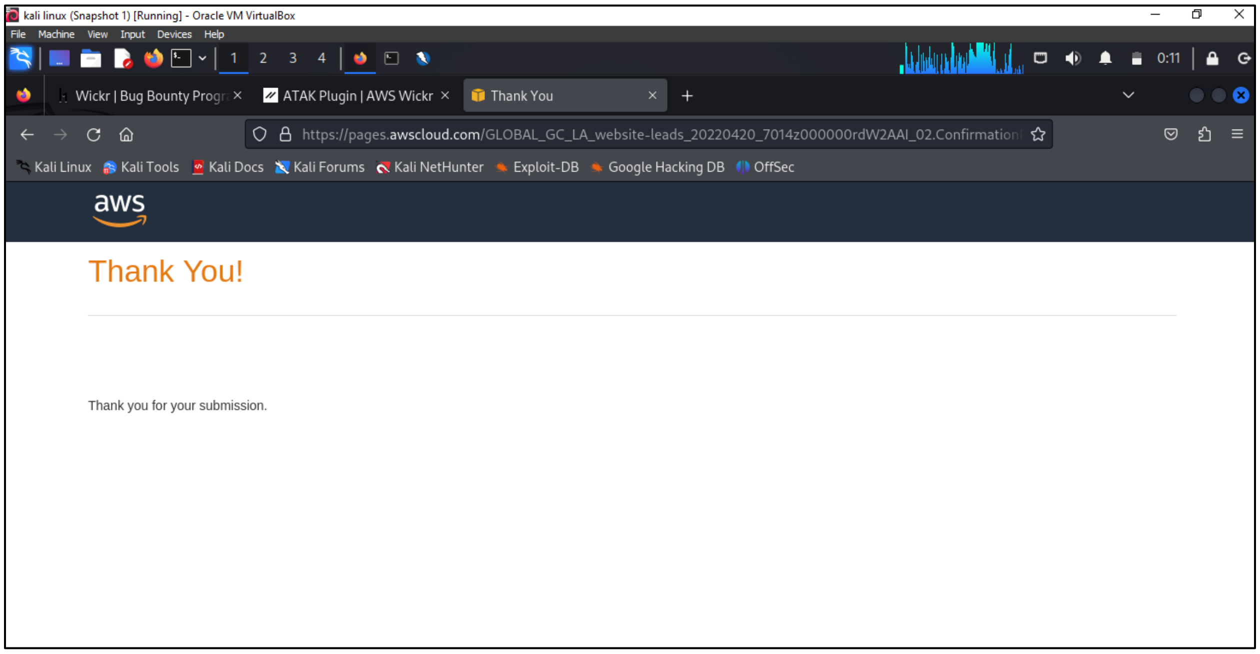
* Phone Number:

* Company Name:

* Country / Region:
Select...

4. In the input field, I tried to inject a simple XSS payload to see if the application reflects it. *<svg/onload=alert('XSS')>*

I submitted the form and checked if the payload is reflected back on the page.



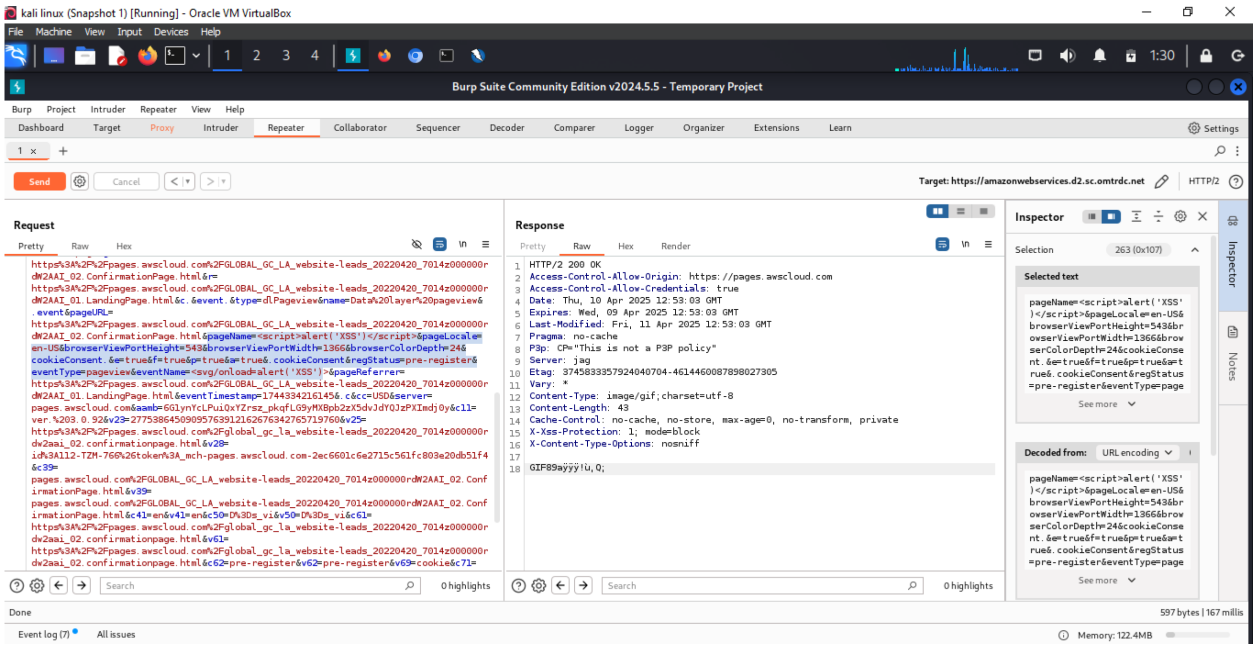
https://pages.awscloud.com/GLOBAL_GC_LA_website-leads_20220420_7014z000000rdW2AAI_02.Confirmation

Thank You!

Thank you for your submission.

I saw "Thank you for your submission" but no alert, the payload is not executed. But the server processes my details even they are invalid.

5. I tried using burp suite for deeper testing. I submitted the form and captured the request.I tried to modify parameters like **pageName, eventName** in the request with the payload, but it didn't work. The server blocks my payload by enabling XSS protection.



Target: https://amazonwebservices.d2.sc.omtrdc.net HTTP/2

Request

```
https%3A%2F%2Fpages.awscloud.com%2FGLOBAL_GC_LA_website-leads_20220420_7014z000000r
dW2AAI_02.ConfirmationPage.html&r=
https%3A%2F%2Fpages.awscloud.com%2FGLOBAL_GC_LA_website-leads_20220420_7014z000000r
dW2AAI_01.LandingPage.html&c.&event.&type=dlPageview&name=Data%20layer%20pageview&
.event&pageURL=
https%3A%2F%2Fpages.awscloud.com%2FGLOBAL_GC_LA_website-leads_20220420_7014z000000r
dW2AAI_02.ConfirmationPage.html&pageName=<script>alert('XSS')</script>&pageLocale=
en-US&browserViewPortHeight=543&browserViewPortWidth=1366&browserColorDepth=24&
cookieConsent.&e=true&f=true&p=true&a=true&.cookieConsent&regStatus=pre-register&
eventType=pageview&eventName=<svg/onload=alert('XSS')>&pageReferrer=
https%3A%2F%2Fpages.awscloud.com%2FGLOBAL_GC_LA_website-leads_20220420_7014z000000r
dW2AAI_01.LandingPage.html&eventTimestamp=1744334216145&.c&cc=USD&server=
pages.awscloud.com&aamb=6G1ynYcLPuiQxYZrsz_pkqfLG9yMXBpb2zX5dvJdYQJzPXImdj0y&c11=
ver.%203.0.92&v23=27753864509095763912162676342765719760&v25=
https%3A%2F%2Fpages.awscloud.com%2Fglobal_gc_la_website-leads_20220420_7014z000000r
dw2aai_02.confirmationpage.html&v28=
id%3A112-TZM-766%26token%3A_mch-pages.awscloud.com-2ec6601c6e2715c561fc803e20db51f4
&c39=
pages.awscloud.com%2FGLOBAL_GC_LA_website-leads_20220420_7014z000000rdW2AAI_02.Conf
irmationPage.html&v39=
pages.awscloud.com%2FGLOBAL_GC_LA_website-leads_20220420_7014z000000rdW2AAI_02.Conf
irmationPage.html&c41=en&v41=en&c50=D%3Ds_vi&v50=D%3Ds_vi&c61=
https%3A%2F%2Fpages.awscloud.com%2Fglobal_gc_la_website-leads_20220420_7014z000000r
dw2aai_02.confirmationpage.html&v61=
https%3A%2F%2Fpages.awscloud.com%2Fglobal_gc_la_website-leads_20220420_7014z000000r
dw2aai_02.confirmationpage.html&c62=pre-register&v62=pre-register&v69=cookie&c71=
```

Response

```
HTTP/2 200 OK
Access-Control-Allow-Origin: https://pages.awscloud.com
Access-Control-Allow-Credentials: true
Date: Thu, 10 Apr 2025 12:53:03 GMT
Expires: Wed, 09 Apr 2025 12:53:03 GMT
Last-Modified: Fri, 11 Apr 2025 12:53:03 GMT
Pragma: no-cache
P3p: CP="This is not a P3P policy"
Server: jag
Etag: 3745833357924040704-4614460087898027305
Vary: *
Content-Type: image/gif;charset=utf-8
Content-Length: 43
Cache-Control: no-cache, no-store, max-age=0, no-transform, private
X-Xss-Protection: 1; mode=block
X-Content-Type-Options: nosniff

GIF89aÿÿÿ!ù,Q;
```

Inspector — Selection 263 (0x107)

Selected text

```
pageName=<script>alert('XSS'
)</script>&pageLocale=en-US&
browserViewPortHeight=543&br
owserViewPortWidth=1366&brow
serColorDepth=24&cookieConse
nt.&e=true&f=true&p=true&a=t
rue&.cookieConsent&regStatus
=pre-register&eventType=page
```

Decoded from: URL encoding

```
pageName=<script>alert('XSS'
)</script>&pageLocale=en-US&
browserViewPortHeight=543&br
owserViewPortWidth=1366&brow
serColorDepth=24&cookieConse
nt.&e=true&f=true&p=true&a=t
rue&.cookieConsent&regStatus
=pre-register&eventType=page
```

Done

597 bytes | 167 millis

Event log (7) All issues Memory: 122.4MB

Reflection of the report

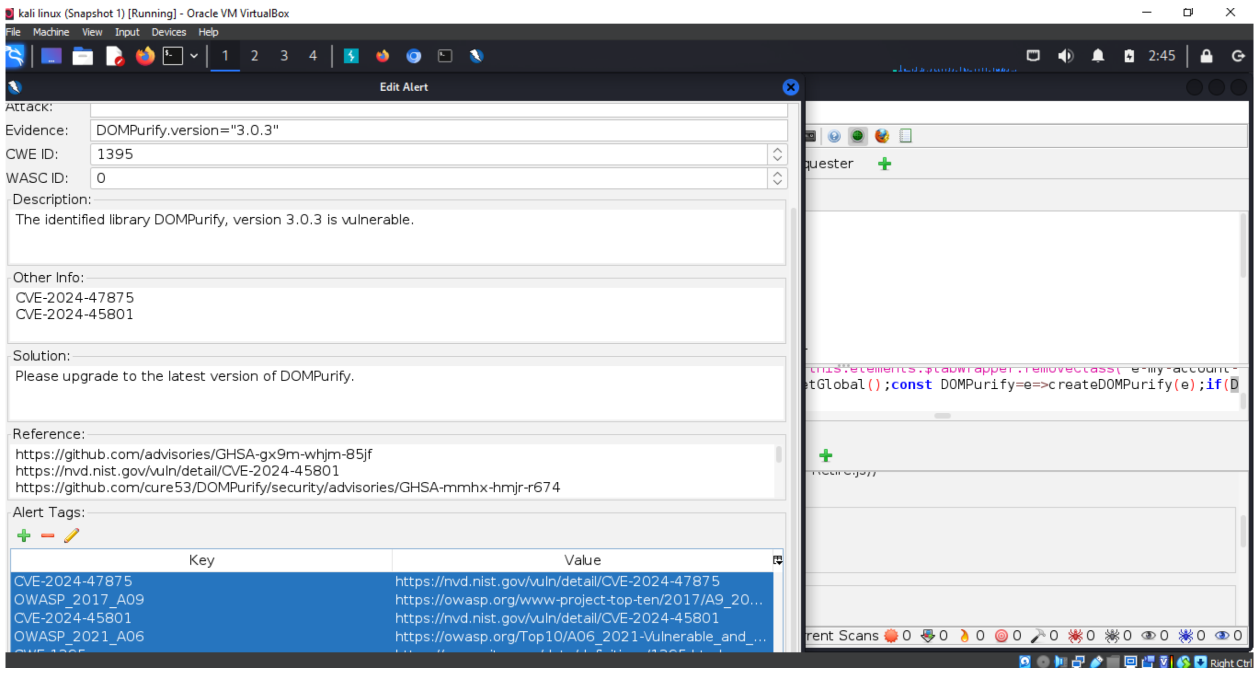
XSS injection tests I conducted against the DOMPurify library under the tag of vulnerable version 3.0.3 included user input fields were particularly unsuccessful, we must keep in mind that the DOMPurify version 3.0.3 is vulnerable and the consideration of this should not be ignored.

The payload not executing in my case, could be a result of the input sanitization working in this case but does not take away from the idea that some other scenarios or configurations may well exist where that same payload might succeed.

Even if exploitation was not successful with my testing, I think the issue should be fixed for maximum security, as there are exploit scenarios for DOMPurify 3.0.3 under certain configurations that can allow attacks to succeed. So, I strongly recommend an upgrade to the latest version of DOMPurify to eliminate any possible risk.

Keys and values of the alert

CVE-2024-47875	https://nvd.nist.gov/vuln/detail/CVE-2024-47875
OWASP_2017_A09	https://owasp.org/www-project-top-ten/2017/A9_2017-Using_Components_with_Known_Vulnerabilities.html
OWASP_2021_A06	https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/
CWE-1395	https://cwe.mitre.org/data/definitions/1395.html


kali linux (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
2:45
Edit Alert
Attack:
quester
Right Ctrl

Evidence: DOMPurify.version="3.0.3"
CWE ID: 1395
WASC ID: 0
Description:
The identified library DOMPurify, version 3.0.3 is vulnerable.
Other Info:
CVE-2024-47875
CVE-2024-45801
Solution:
Please upgrade to the latest version of DOMPurify.
Reference:
https://github.com/advisories/GHSA-gx9m-whjm-85jf
https://nvd.nist.gov/vuln/detail/CVE-2024-45801
https://github.com/cure53/DOMPurify/security/advisories/GHSA-mmhx-hmjr-r674
Alert Tags:

Key	Value
CVE-2024-47875	https://nvd.nist.gov/vuln/detail/CVE-2024-47875
OWASP_2017_A09	https://owasp.org/www-project-top-ten/2017/A9_20…
CVE-2024-45801	https://nvd.nist.gov/vuln/detail/CVE-2024-45801
OWASP_2021_A06	https://owasp.org/Top10/A06_2021-Vulnerable_and_…

```
etGlobal();const DOMPurify=e=>createDOMPurify(e);if(D
```

rent Scans 0 0 0 0 0 0 0 0 0 0

This vulnerability is defined as **OWASP 2021 A06: Vulnerable and Outdated Components** and **OWASP 2017 A09: Using Components with Known Vulnerabilities**. The application is running with an old version of the DOMPurify library, which is vulnerable to security issues. Although the XSS attack configuration set by me failed, it adds risk to future attacks due to dependence on obsolete components. It is recommended to update that library to a more secure version and renew regular patching of all third-party components.

Proposed mitigation or fix

- Update DOMPurify - The first thing to do is to upgrade DOMPurify to the latest version—new releases would have patched against such security vulnerabilities and introduced improvements in input sanitization.

- Configuration Hardening - Properly configure DOMPurify to ensure it blocks any form of malicious content getting into the system, especially when **dealing with HTML or SVG inputs.**

- Content Security Policy (CSP): Implementation of a content security policy could serve as an added control mechanism to restrict any executable content that could be loaded.
  This would increase the odds of blocking a successful XSS attack even when DOMPurify might fail to appropriately sanitize the inputs.

- Continued Validation and Auditing - Both the client and server-side input sanitization have to be checked for validity at regular intervals to ensure malicious scripts are not being executed.

- Regular Monitoring and Patching - Security patching of libraries like DOMPurify is part of general security practice so monitoring updates and vulnerabilities in third-party libraries is very important.