Sri Lanka Institute of Information Technology

Web Security - IE2062

Bug Bounty Report 7

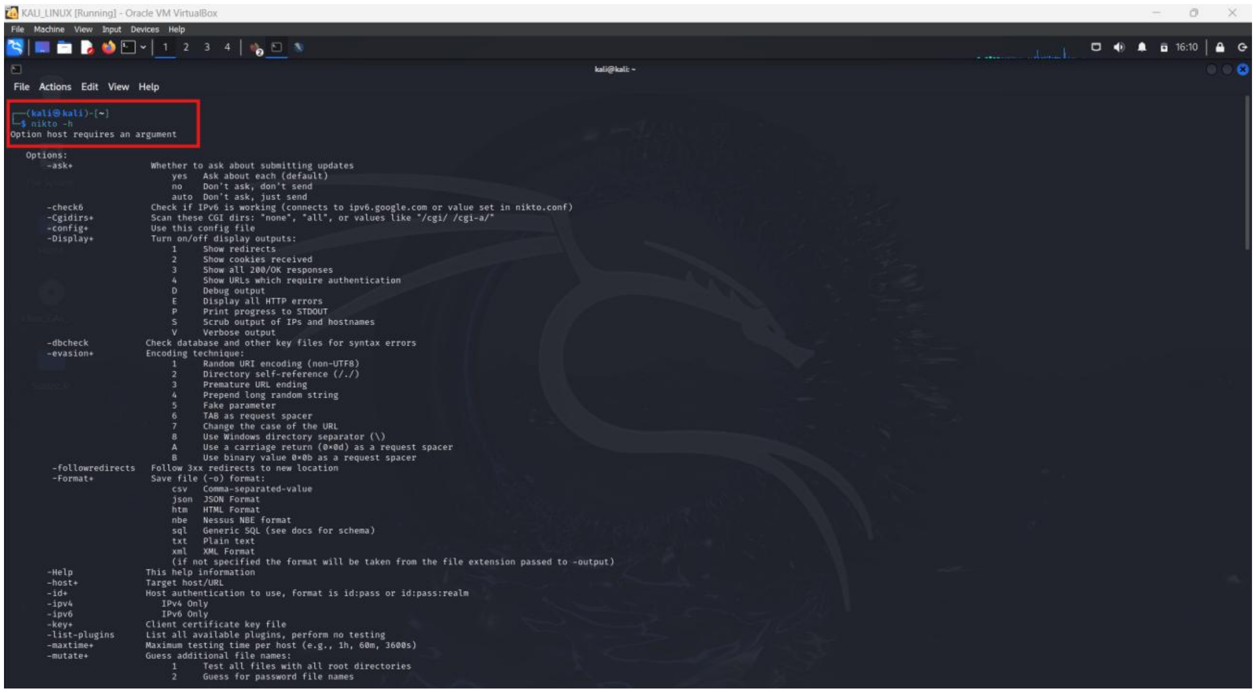
PERERA A.P.J

IT22280992

Group Y2S2.CS

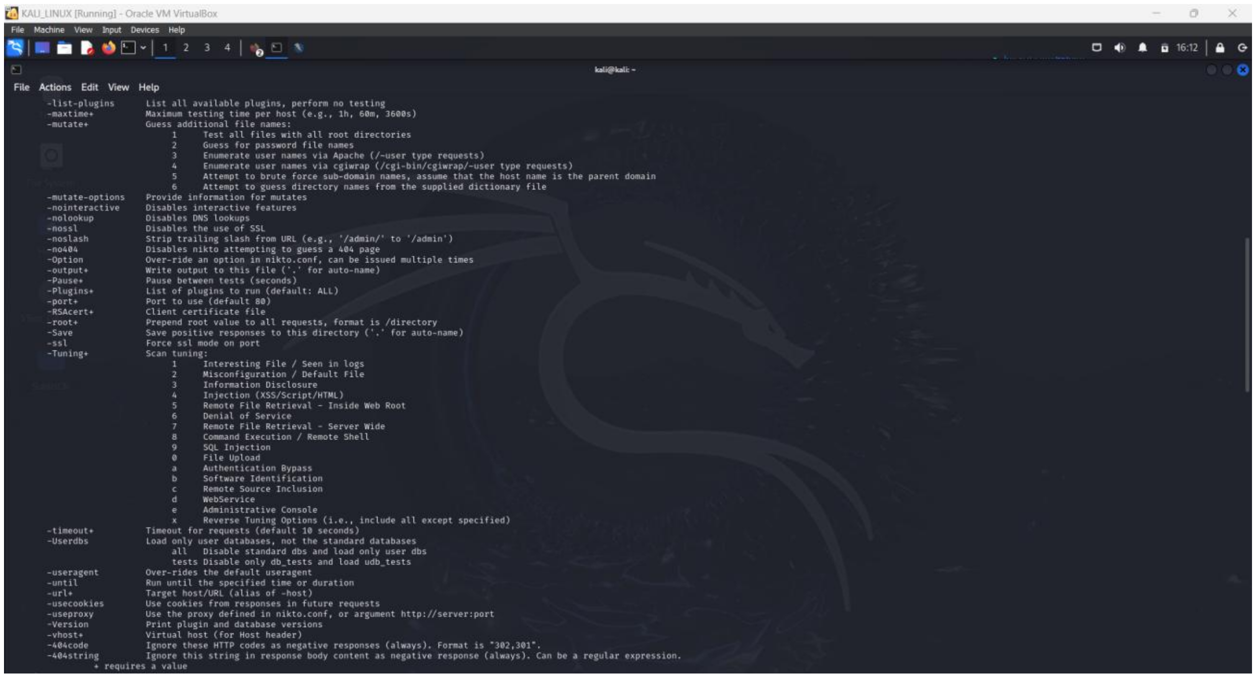
Table of Contents

1 Using NIKTO tool we can find vulnerabilities.


KALI_LINUX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help

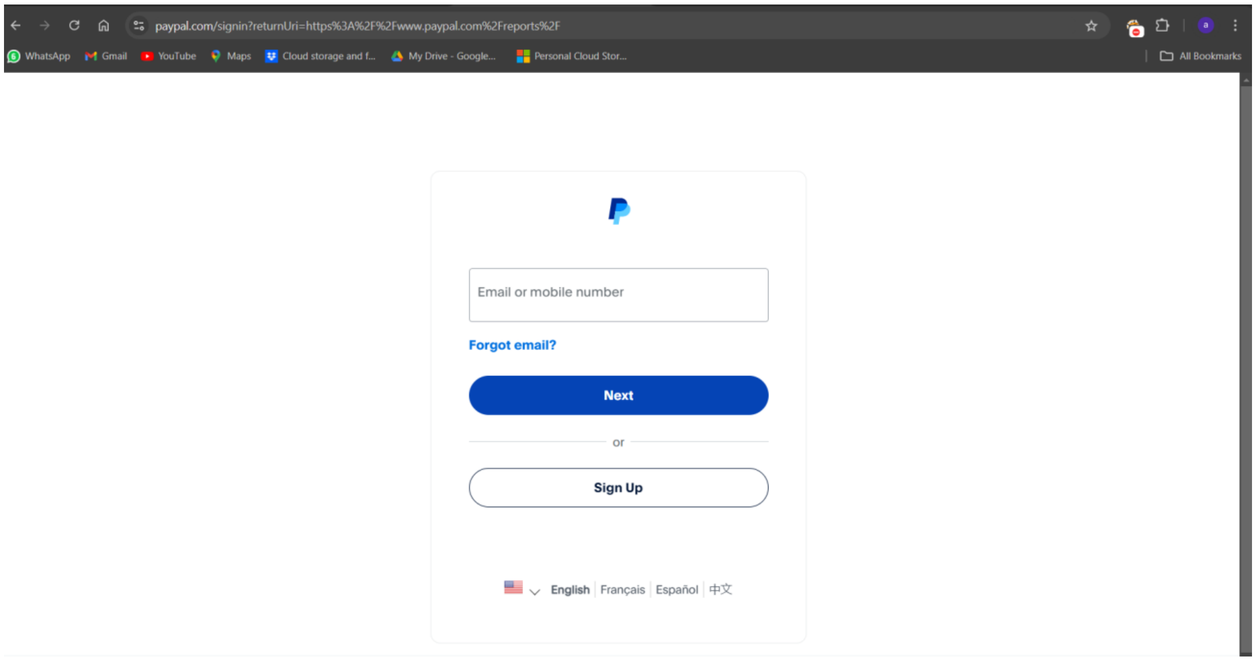
```
┌──(kali㉿kali)-[~]
└─$ nikto -h
Option host requires an argument

   Options:
       -ask+               Whether to ask about submitting updates
                               yes   Ask about each (default)
                               no    Don't ask, don't send
                               auto  Don't ask, just send
       -check6             Check if IPv6 is working (connects to ipv6.google.com or value set in nikto.conf)
       -Cgidirs+           Scan these CGI dirs: "none", "all", or values like "/cgi/ /cgi-a/"
       -config+            Use this config file
       -Display+           Turn on/off display outputs:
                               1     Show redirects
                               2     Show cookies received
                               3     Show all 200/OK responses
                               4     Show URLs which require authentication
                               D     Debug output
                               E     Display all HTTP errors
                               P     Print progress to STDOUT
                               S     Scrub output of IPs and hostnames
                               V     Verbose output
       -dbcheck           Check database and other key files for syntax errors
       -evasion+          Encoding technique:
                               1     Random URI encoding (non-UTF8)
                               2     Directory self-reference (/./)
                               3     Premature URL ending
                               4     Prepend long random string
                               5     Fake parameter
                               6     TAB as request spacer
                               7     Change the case of the URL
                               8     Use Windows directory separator (\)
                               A     Use a carriage return (0x0d) as a request spacer
                               B     Use binary value 0x0b as a request spacer
        -followredirects   Follow 3xx redirects to new location
        -Format+           Save file (-o) format:
                               csv   Comma-separated-value
                               json  JSON Format
                               htm   HTML Format
                               nbe   Nessus NBE format
                               sql   Generic SQL (see docs for schema)
                               txt   Plain text
                               xml   XML Format
                               (if not specified the format will be taken from the file extension passed to -output)
       -Help              This help information
       -host+             Target host/URL
       -id+               Host authentication to use, format is id:pass or id:pass:realm
       -ipv4                 IPv4 Only
       -ipv6                 IPv6 Only
       -key+              Client certificate key file
       -list-plugins      List all available plugins, perform no testing
       -maxtime+          Maximum testing time per host (e.g., 1h, 60m, 3600s)
       -mutate+           Guess additional file names:
                               1     Test all files with all root directories
                               2     Guess for password file names
```


KALI_LINUX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help

```
       -list-plugins      List all available plugins, perform no testing
       -maxtime+          Maximum testing time per host (e.g., 1h, 60m, 3600s)
       -mutate+           Guess additional file names:
                               1     Test all files with all root directories
                               2     Guess for password file names
                               3     Enumerate user names via Apache (/~user type requests)
                               4     Enumerate user names via cgiwrap (/cgi-bin/cgiwrap/~user type requests)
                               5     Attempt to brute force sub-domain names, assume that the host name is the parent domain
                               6     Attempt to guess directory names from the supplied dictionary file
       -mutate-options    Provide information for mutates
       -nointeractive     Disables interactive features
       -nolookup          Disables DNS lookups
       -nossl             Disables the use of SSL
       -noslash           Strip trailing slash from URL (e.g., '/admin/' to '/admin')
       -no404             Disables nikto attempting to guess a 404 page
       -Option            Over-ride an option in nikto.conf, can be issued multiple times
       -output+           Write output to this file ('.' for auto-name)
       -Pause+            Pause between tests (seconds)
       -Plugins+          List of plugins to run (default: ALL)
       -port+             Port to use (default 80)
       -RSAcert+          Client certificate file
       -root+             Prepend root value to all requests, format is /directory
       -Save              Save positive responses to this directory ('.' for auto-name)
       -ssl               Force ssl mode on port
       -Tuning+           Scan tuning:
                               1     Interesting File / Seen in logs
                               2     Misconfiguration / Default File
                               3     Information Disclosure
                               4     Injection (XSS/Script/HTML)
                               5     Remote File Retrieval - Inside Web Root
                               6     Denial of Service
                               7     Remote File Retrieval - Server Wide
                               8     Command Execution / Remote Shell
                               9     SQL Injection
                               0     File Upload
                               a     Authentication Bypass
                               b     Software Identification
                               c     Remote Source Inclusion
                               d     WebService
                               e     Administrative Console
                               x     Reverse Tuning Options (i.e., include all except specified)
       -timeout+          Timeout for requests (default 10 seconds)
       -Userdbs           Load only user databases, not the standard databases
                               all   Disable standard dbs and load only user dbs
                               tests Disable only db_tests and load udb_tests
       -useragent         Over-rides the default useragent
       -until             Run until the specified time or duration
       -url+              Target host/URL (alias of -host)
       -usecookies        Use cookies from responses in future requests
       -useproxy          Use the proxy defined in nikto.conf, or argument http://server:port
       -Version           Print plugin and database versions
       -vhost+            Virtual host (for Host header)
       -404code           Ignore these HTTP codes as negative responses (always). Format is "302,301".
       -404string         Ignore this string in response body content as negative response (always). Can be a regular expression.
             + requires a value
```

2. TARGET : http://business.paypal.com



paypal.com/signin?returnUri=https%3A%2F%2Fwww.paypal.com%2Freports%2F

Email or mobile number

Forgot email?

Next

or

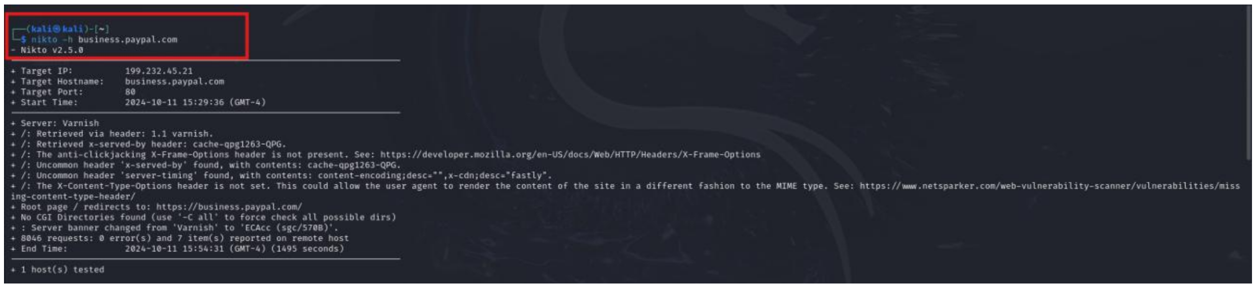
Sign Up

English | Français | Español | 中文

3. Vulnerability

3.1 Vulnerability title

- Anti clickjacking x – frame-option header not found.



```
┌──(kali㉿kali)-[~]
└─$ nikto -h business.paypal.com
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          199.232.45.21
+ Target Hostname:    business.paypal.com
+ Target Port:        80
+ Start Time:         2024-10-11 15:29:36 (GMT-4)
---------------------------------------------------------------------------
+ Server: Varnish
+ /: Retrieved via header: 1.1 varnish.
+ /: Retrieved x-served-by header: cache-qpg1263-QPG.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: Uncommon header 'x-served-by' found, with contents: cache-qpg1263-QPG.
+ /: Uncommon header 'server-timing' found, with contents: content-encoding;desc="",x-cdn;desc="fastly".
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/miss
ing-content-type-header/
+ Root page / redirects to: https://business.paypal.com/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ : Server banner changed from 'Varnish' to 'ECAcc (sgc/570B)'.
+ 8046 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2024-10-11 15:54:31 (GMT-4) (1495 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

3.2 **Vulnerability description**

- The ‘Anti-clickjacking X-Frame-Options header not found’ security flaw means that a webpage does not have the X-Frame-Options header hence it is susceptible to clickjacking. Click jacking is a method where a real and trusted web page is planted within an iframe of a different malicious web page and certain elements are placed above the embedded page such that the user is made to interact with the embedded page without his/her knowledge. It causes things like changes to a user’s account or even fraudulent purchases using that account. The X-Frame-Options header is set in order to protect the users from clickjacking by restricting the framing of the webpages in the iframes by the web browsers. The x-frame-options header adds the following options. DENY option prevents any iframe embedding, SAMEORIGIN allows only same origins embedding, ALLOW-FROM URI embeds allowed sources. In the absence of this header, it becomes easier for the miscreants to persuade the users to perform any intrusive actions against a legitimate website and hence the security and confidence of the user is at a risk.

3.3 **Affected components**

- The "Anti-clickjacking X-Frame-Options header not found" vulnerability primarily affects the following components of a website:

    1. User Interface (UI): Since clickjacking often involves overlaying malicious elements on a legitimate UI, it impacts the visible components of the webpage. Users may unknowingly click on hidden or misrepresented elements, leading to unauthorized actions.

    2. HTTP Headers: The absence of the X-Frame-Options header in HTTP responses is the core issue. Without this header, browsers do not restrict iframe embedding, allowing the page to be loaded in iframes on other sites.

    3. Web Application Server: If the application server does not add the necessary header, it becomes the source of vulnerability. The server needs configuration updates to ensure the X-Frame-Options header is included in responses.

    4. User Sessions and Accounts: User session data and accounts are at risk since clickjacking attacks can lead to actions such as forced clicks or submissions, potentially allowing attackers to compromise or alter user information.

    5. Security Controls: Other security mechanisms like Content Security Policy (CSP) directives may also be impacted if not properly configured to block iframe embedding from untrusted sources, making the vulnerability more severe.

3.4 **Impact assessment**

- The impact assessment of the "Anti-clickjacking X-Frame-Options header not found" vulnerability can vary depending on the nature of the website and its functionalities, but the primary risks include:

1. Unauthorized Actions by Users: Attackers can manipulate users into unknowingly performing actions such as submitting forms, clicking buttons, or changing account settings. This can lead to actions that the user did not intend, including transactions or modifications within their account.

2. Data Theft and Account Compromise: Clickjacking can lead to user data being exposed or stolen. For instance, users could be tricked into disclosing sensitive information by interacting with seemingly legitimate UI elements that have been overlaid with malicious content. Additionally, if the attacker can force actions within an authenticated session, they might gain unauthorized access to user accounts.

3. Damage to User Trust and Reputation: A successful clickjacking attack can damage user trust in the website, especially if it results in unauthorized actions that the user perceives as being facilitated by the site itself. This can harm the website's reputation, particularly if customers suffer financial or personal losses due to the attack.

4. Financial Loss and Compliance Issues: For e-commerce or financial sites, clickjacking could result in fraudulent transactions, leading to financial losses. Moreover, if personal data is exposed due to clickjacking, the site may face compliance violations under data protection regulations like GDPR or CCPA.

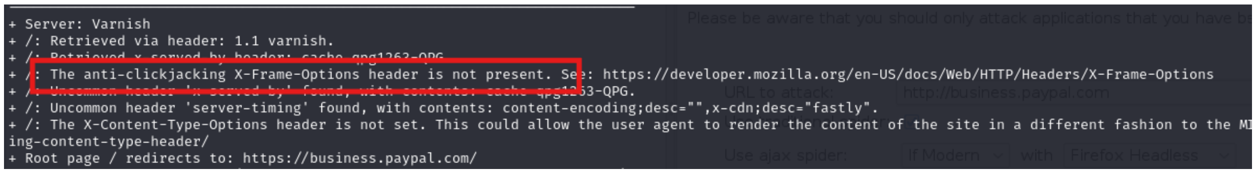
5. Increased Risk of Broader Attacks: If an attacker successfully uses clickjacking to perform actions on behalf of a user, they may be able to escalate the attack to gain further control or conduct more invasive attacks, such as session hijacking or account takeovers.

3.5 **Steps to reproduce**

- Steps

1. Open the Target Page: Go to the website where you suspect the vulnerability exists.
2. Check for the X-Frame-Options Header:
    - Open the browser's Developer Tools (usually accessible by pressing F12 or Ctrl + Shift + I).
    - Go to the Network tab and reload the page.
    - Click on the main request (the page URL) to view the response headers.
    - Look for the X-Frame-Options header in the response headers section.
    - If the header is missing, the site is potentially vulnerable.
3. Create a Simple HTML File for Testing:
    - On your local machine, create a new HTML file with the following code:
    - Replace https://target-website.com with the URL of the target site.
4. Open the HTML File in a Browser:
    - Open the file you created in a browser.
    - If the target website loads within the <iframe>, this indicates that the X-Frame-Options header is either missing or not correctly configured to prevent iframe embedding.
5. Attempt to Interact with the Embedded Content:
    - Try clicking or interacting with elements in the embedded iframe.
    - If successful, this confirms that the website is vulnerable to clickjacking, as the lack of the X-Frame-Options header allows its content to be loaded and potentially manipulated within an iframe.

3.6 **Proof of concept**



```
+ Server: Varnish
+ /: Retrieved via header: 1.1 varnish.
+ /: Retrieved x-served-by header: cache-qpg1263-QPG.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: Uncommon header 'x-served-by' found, with contents: cache-qpg1263-QPG.
+ /: Uncommon header 'server-timing' found, with contents: content-encoding;desc="",x-cdn;desc="fastly".
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MI
ing-content-type-header/
+ Root page / redirects to: https://business.paypal.com/
```

3.7 **Proposed mitigation or fix**

- Implement the x-frame option header.
- Use Control Security Policy.
- Regular Security Audits.
- User Education.
- Web Application Firewall.