Web application security

Week1

Timo Lehosvuo, M3426
TTV18S1

Harjoitustyö
Web application security, Heikki Salo, Joni Ahonen
22.2.2021
Tekniikan ala

1. Reflection

I wish to get a better understanding about vulnerabilities and learn how to hack better. Hacking interest me so I want to get better at it and understanding vulnerabilities better will help me become better at hacking. These two go hand in hand.

2. Reading report

2.1 vulnerabilities and bug bounties

I was aware that companies have bug bounties, but I did not know that some companies have vulnerability disclosure programs (VDP) instead of bug bounties. It surprised me that VDPs don’t offer any money for finding possible vulnerabilities since you would think that you would want to encourage people to test your systems.

2.2 Server response

I did not know that there isn’t a strict enforcement of how servers implement its use of HTTP status code. I have always thought that it was a strict rule for:

- 1xx information response
- 2xx success
- 3xx redirect
- 4xx client error
- 5xx server error

but apparently these can be changed.

2.3 HTTP requests

I did not know all the HTTP request methods. I only knew GET, HEAD, POST and PUT, but there are also DELETE, TRACE, CONNECT and OPTIONS.

3. Issue report

3.1 Wrongly configured robots.txt

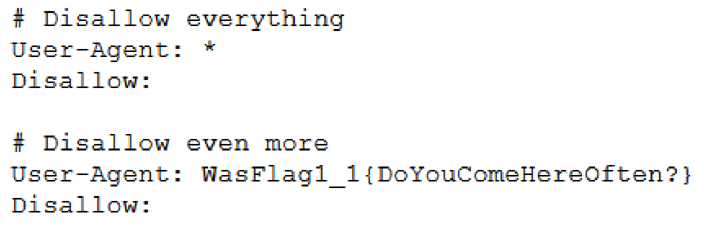
Description: Robots.txt contains wrong or useless configurations. This is not really a vulnerability instead more of a bad configuration by the user.

Steps to produce:

- Set up “wasdat” following this guide: https://gitlab.labranet.jamk.fi/ethical-hacking-module/was-students/-/blob/master/INSTRUCTIONS.md
- Visit http://localhost:8080/ on your host machine and you will see the robots.txt configurations.
- There are multiple sections with wrong configuration

Mitigation:

- Change the lines “Disallow: “to “Disallow: / “. This will exclude all robots from the server. Currently it is allowing all robots complete access except for /*/*.git and /backup/.
- Giving a single robot “WasFlag1_1{DoYouComeHereOften?} access is useless since everybody has already been given access to the server so removing it is advisable.
- You can also change the comment “Disallow everything” to “Allow everything” if you don’t want to change the code and want to restrict only /*/*.git$ and /backup/ pages.



```
# Disallow everything
User-Agent: *
Disallow:

# Disallow even more
User-Agent: WasFlag1_1{DoYouComeHereOften?}
Disallow:
```

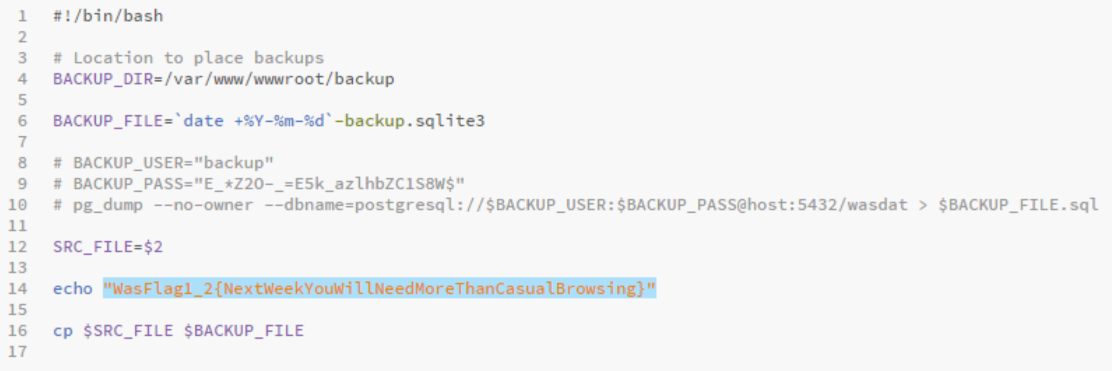
Figure 1: Robots.txt configurations.

3.2 Unrestricted access to backup files

Description: Anonymous users have free access to browse and download backup files located in the “http://localhost/8080/backup/.

Steps to produce:

- Visit the site http://localhost:8080/robots.txt. There you can find a line that says “/backup/.
- Go to the site “http://localhost:8080/backup/ and you have free access to download and browse backup files.
- There are multiple backups but choose the one that has bytes greater than zero.
- extract the file and open the file “backup.sh” for example in brackets



```
#!/bin/bash

# Location to place backups
BACKUP_DIR=/var/www/wwwroot/backup

BACKUP_FILE=`date +%Y-%m-%d`-backup.sqlite3

# BACKUP_USER="backup"
# BACKUP_PASS="E_*Z20-_=E5k_azlhbZC1S8W$"
# pg_dump --no-owner --dbname=postgresql://$BACKUP_USER:$BACKUP_PASS@host:5432/wasdat > $BACKUP_FILE.sql

SRC_FILE=$2

echo "WasFlag1_2{NextWeekYouWillNeedMoreThanCasualBrowsing}"

cp $SRC_FILE $BACKUP_FILE

```

Figure 2: Backup.sh.

Mitigation:

- Do not store backups in a webserver.
- Restrict access to backup files for example require a login to browse backup files.
- Be careful what areas you restrict/allow in robots.txt since it’s a public file.

3.3 Flags

- WasFlag1_1{DoYouComeHereOften?}

- WasFlag1_2{NextWeekYouNeedMoreThanCasualBrowsing}

4. Reflection

I started browsing then site and looking for possible clues where the juice shop scoreboard might be. I checked all the products but that was a dead-end and simply tested if the page can we found from “http://localhost:3000/scoreboard” but it wasn’t there. After a while to the top left corner appeared a helper that said that I should check the javascript code and so I did. I used “inspect element” to see the “js” code and found out multiple matches for the scoreboard from the “main-es5.js” code. None of these matches for “scoreboard” helped so I change my search to “/score” and found a new hint “/score-board”. I went back to the juice shop page and tested if “http://localhost:3000/score-board would work but it didn’t so I tried “http://localhost:3000/#/score-board and there it was:

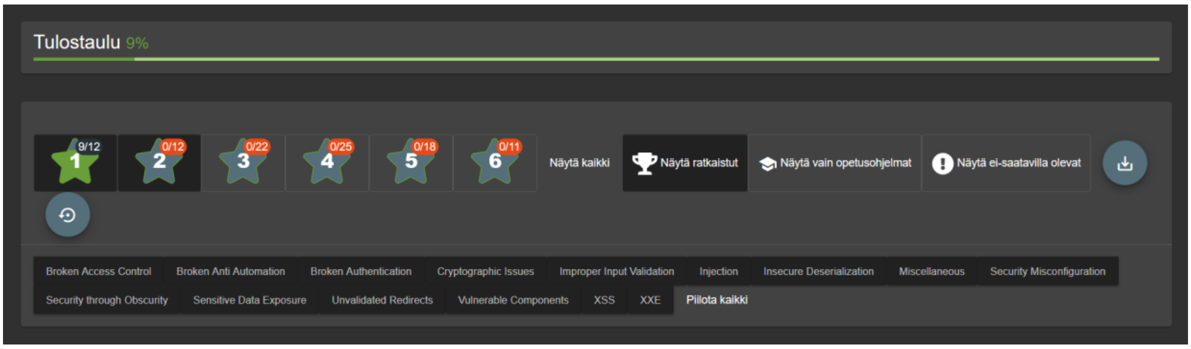


Figure 3: Scoreboard.