Twitter Penetration Test Report

BI-EHA Semestral work

Jan Koníř, Tadeáš Kouba

FIT CTU, 2023

Table of Contents

Introduction

Team Info

- Jan Koníř
  - konirjan@fit.cvut.cz
  - Student of CTU, Faculty of Information Technology

- Tadeáš Kouba
  - koubatad@fit.cvut.cz
  - Student of CTU, Faculty of Information Technology

Project Overview

This report is part of semestral work for subject Ethical hacking of the Information Security program on FIT, CTU.
The focus is to assert skills gained during the semester as well as use them in real-world scenario. The website twitter.com has been chosen for this purpouse as it offers a bug bounty and is therefore eligible for ethical hackers to try and hack.

Scope Description

We will be focusing on the web application on domain twitter.com.
While the bug bounty posted by twitter puts some vulnerabilities out scope for financial reward, we will still be trying to perform these attacks as to deepen our ethical hacking skills.

Pentesting Methodology

As a pentesting methodology we have chosen the OWASP Web Application Security Testing guide - version 4.2, which is the latest stable version. To be more precise, we will be covering following topics:

- Configuration and Deployment Management Testing
- Identity Management Testing
- Authentication Testing
- Authorization Testing
- Session Management Testing
- Input Validation Testing
- Testing for Error Handling
- Testing for Weak Cryptography
- Business Logic Testing
- Client-side Testing

Scoring System

We will be using the widely used Common Vulnerability Scoring System (CVSS) standard for evaulating the threat levels of vulnerabilities found in this report. In particular we will use the CVSS 3.1 version. The standard defines multiple metrics (base, temporal, environmental), however we will use only the base metric since we are performing a one-time pentesting as external entity so the temporal and environmental metrics are not that important and available to us.
We will be scoring each vulnerability on several factors such as access complexity, confidentiality impact, integrity impact and others, to calculate the final score. This will determine the final rating of that vulnerability as seen in the following table.

Rating	CVSS Score
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Executive Summary

TBD

List of Findings

TBD

Pre-engagement

Threat Model

Twitter does not openly disclose how its backend communication works.
Since twitter is a web application it is obviously exposed to the internet which naturally puts it in proximity of multitude of threats. On top of that it is definitely an attractive target for hackers, since it stores personal information of about 350 million people.
Some vulnerable points might be the client-server-database communication for authentication or tweets being stored in database and showing to completely random users' feeds.

Intelligence-gathering Outcomes

Firstly we ran port scan with **nmap**. We used the following command to find the open ports and possibly the version of service running on them:

```
nmap -sV twitter.com
```

```
nmap -sV twitter.com
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-27 15:04 EDT
Nmap scan report for twitter.com (104.244.42.65)
Host is up (0.034s latency).
Other addresses for twitter.com (not scanned): 104.244.42.193
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE    VERSION
80/tcp  open  tcpwrapped
443/tcp open  ssl/https  tsa_o

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 32.38 seconds
```

Full output can be seen in *outputs/nmap_version.txt*.
We can see that there are two open tcp ports (80, 443).

We then ran **nmap** with intention to determine the OS running, version and traceroute with:

```
nmap -A twitter.com
```

```
PORT    STATE SERVICE   VERSION
80/tcp  open  http      tsa_o
|_http-title: Did not follow redirect to https://twitter.com/
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     HTTP/1.1 400 Bad Request
|     content-length: 11
|     content-type: text/plain
|     x-connection-hash: e41ea0468e7b5be750935d042454e8711e3b3961c1737992a4cc017f6081eee2
|     date: Thu, 27 Apr 2023 19:07:40 GMT
|     server: tsa_o
|     connection: close
|     Request
|   FourOhFourRequest:
|     HTTP/1.0 400 Bad Request
|     x-connection-hash: 02d1cd22645d11f2bf49cdace64f2604c7e3a924a2cfc0bc51ef47d0e2cb5f6b
|     date: Thu, 27 Apr 2023 19:07:33 GMT
|     server: tsa_o
|     connection: close
|     content-length: 0
|   GetRequest:
|     HTTP/1.0 400 Bad Request
|     x-connection-hash: f18259468620a4e8ad658e4dc615da660dc1215ad1a001d84c1e96c4e3d25137
|     date: Thu, 27 Apr 2023 19:07:29 GMT
|     server: tsa_o
|     connection: close
|     content-length: 0
|   HTTPOptions:
|     HTTP/1.0 400 Bad Request
|     x-connection-hash: 53ab8f7fb60e3463465249882514a86891e613e9036689229e062378af384b46
|     date: Thu, 27 Apr 2023 19:07:30 GMT
|     server: tsa_o
|     connection: close
|     content-length: 0
```

```
443/tcp open  ssl/https tsa_o
| ssl-cert: Subject: commonName=twitter.com/organizationName=Twitter, Inc./stateOrProvinceName=
| Subject Alternative Name: DNS:twitter.com, DNS:www.twitter.com
| Not valid before: 2023-02-05T00:00:00
|_Not valid after:  2024-02-05T23:59:59
|_http-server-header: tsa_o
|_ssl-date: TLS randomness does not represent time
| http-robots.txt: 13 disallowed entries
| /search/realtime /search/users /search/*/grid /*?
| /*/followers /*/following /account/deactivated
|_/settings/deactivated /oauth /1/oauth /i/streams /i/hello /i/u
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     HTTP/1.1 400 Bad Request
|     content-length: 11
|     content-type: text/plain
|     x-connection-hash: 4a220339f0d48a93266c291c45e2d2945f95f6e4c4d9e0be7a4b8c0e7af86e2a
|     date: Thu, 27 Apr 2023 19:07:47 GMT
|     server: tsa_o
|     connection: close
|     Request
|   FourOhFourRequest:
|     HTTP/1.0 400 Bad Request
|     x-connection-hash: 562463708a4194b26a6cd538525aeb5c4ed5db1011d43b9d221accfdab4b191d
|     date: Thu, 27 Apr 2023 19:07:37 GMT
|     server: tsa_o
|     connection: close
|     content-length: 0
|   GetRequest:
|     HTTP/1.0 400 Bad Request
|     x-connection-hash: 8627003e329bef15b9d472e46b19be7b09d9ed3834bdd984dde2be922899bd26
|     date: Thu, 27 Apr 2023 19:07:35 GMT
|     server: tsa_o
|     connection: close
|     content-length: 0
```

Full output can be seen in *outputs/nmap_A.txt*.
Here it is specified that on port 80 there is http running. Also all requests sent are returned with HTTP 400 so service as well as OS could not be recognized.

Then we ran **dnsenum** to try and gather some infromation about the domain:

```
dnsenum --enum twitter.com
```

```
Host's addresses:
__________________

twitter.com.                            1007    IN    A        104.244.42.193


Name Servers:
_____________

a.u06.twtrdns.net.                      91      IN    A        204.74.66.101
b.r06.twtrdns.net.                      97      IN    A        205.251.196.198
b.u06.twtrdns.net.                      10      IN    A        204.74.67.101
c.r06.twtrdns.net.                      103     IN    A        205.251.194.151
c.u06.twtrdns.net.                      55      IN    A        204.74.110.101
a.r06.twtrdns.net.                      91      IN    A        205.251.192.179
b.r06.twtrdns.net.                      97      IN    A        205.251.196.198
b.u06.twtrdns.net.                      10      IN    A        204.74.67.101
c.r06.twtrdns.net.                      103     IN    A        205.251.194.151
c.u06.twtrdns.net.                      55      IN    A        204.74.110.101
a.r06.twtrdns.net.                      91      IN    A        205.251.192.179
a.u06.twtrdns.net.                      91      IN    A        204.74.66.101
d.r06.twtrdns.net.                      11      IN    A        205.251.199.195
d.u06.twtrdns.net.                      12      IN    A        204.74.111.101


Mail (MX) Servers:
__________________

aspmx3.googlemail.com.                  224     IN    A        142.250.150.26
aspmx2.googlemail.com.                  221     IN    A        142.251.9.26
aspmx.l.google.com.                     71      IN    A        173.194.76.27
alt2.aspmx.l.google.com.                193     IN    A        142.251.9.26
alt1.aspmx.l.google.com.                194     IN    A        142.250.153.27
```

Full output can be seen in *outputs/dnsenum.txt*.
Which listed us a lot of subdomains and IP addresses of twitter.

As stated before the twitter developers don't share much information about twitter. However through developer blog they sometimes discuss what actually happens on twitter's backend. Through that we were able to find out that twitter used to run on MySQL database, but has switched to distributed database NoSQL system called Manhattan, that is custom developed for twitter. Manhattan itself utilizes multiple open-source technologies like Apache Cassandra or Hadoop. This information might prove useful for performing some SQL Injection attacks.

As for the front-end part of Twitter it mostly consists of HTML, CSS and Javascript, particularly React.js. Of those, React.js is obviously the one to be concentrated on when pentesting. Some of its most common vulnerabilities are Cross-site scripting, SQL injection, Cross-site request forgery, Vulnerability in packages and dependencies, Broken authentication, Zip slip or XML external entities.

Last but not least we tried so called Google Hacking/Dorking but it didn't disclose any important information.

Testing process

4.1 Information Gathering

This topic is covered in the Pre-engagement section.

4.2 Configuration and Deployment Management Testing

4.3 Identity Management Testing

4.4 Authentication Testing

4.5 Authorization Testing

4.6 Session Management Testing

4.7 Input Validation Testing

4.8 Testing for Error Handling

4.9 Testing for Weak Cryptography

4.10 Business Logic Testing

4.11 Client-side Testing

Sources

- https://blog.twitter.com/engineering/en_us/topics/infrastructure/2017/the-infrastructure-behind-twitter-scale
- https://www.first.org/cvss
- https://blog.twitter.com/engineering/en_us/topics/
- http://www.pentest-standard.org
- https://www.thirdrocktechkno.com/blog/react-security-vulnerabilities/