DEPARTMENT OF INFORMATICS, UNIVERSITY OF BERGEN

INF226 — Software (in)security 101

Benjamin Chetioui, Håkon Gylterud

August 21, 2019

This is one of three mandatory assignments for INF226, autumn 2019. This assignment counts 10% of your final grade. There are four exercises, scored individually.
The assignment is due 15th of September, and is to be handed in as a PDF report, including any source code you write, through mitt.uib.no.

1 Pwn — 5 pts

The purpose of this section is to perform basic binary exploitation.

The report should, for each exercise, contain:

1. a short description of the vulnerability,
2. a Python script showing the exploit using **pwntools**,
3. a description of how the exploit works, and
4. the string found in flag.txt.

For each exercise, a compiled ELF 64-bit binary file as well as the source code is provided. Each binary is running as a service on the server **shepherd.ii.uib.no**. The relevant port is specified within the description of the exercise. The exploit you submit must exploit the vulnerability directly on the remote server.

1.1 0x01 — 3 pts

This exercise is running on port 9001 on the remote server.

Listing 1: Source code of exercise 0x01

```
#include <assert.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv) {

    char buffer[32];
    int32_t check = 0xdeadbeef;

    printf("Try to get past me!\n");
    fflush(stdout);

    assert(fgets(buffer, 1024, stdin) != NULL);

    if (check == 0xc0cac01a) {
        printf("Congratulations, you win!\n");
        fflush(stdout);
        system("cat flag.txt");
    }
    else {
        printf("You lose! Bye.\n");
    }

    return 0;
}
```

1.2 0x02 — 2 pts

This exercise is running on port 9002 on the remote server.

Listing 2: Source code of exercise 0x02

```c
#include <assert.h>
#include <stdio.h>
#include <stdlib.h>

void do_system() {
    system("cat flag.txt");
}

int main(int argc, char **argv) {

    char buffer[32];

    printf("Try to get past me!\n");
    fflush(stdout);

    assert(fgets(buffer, 1024, stdin) != NULL);

    return 0;
}
```

2 SQL INJECTIONS — 5 PTS

The purpose of the second section of the assignment is to perform basic Web exploitation through SQL injections.

The report should, for each exercise, contain:

1. a short description of the vulnerability,
2. a description of how you exploited the vulnerability,
3. any script you used in the exploitation, and
4. the string found in flag.txt.

Each exercise is running as a service on the server **shepherd.ii.uib.no**. Each port is specified within the description of the exercise. The exploit you submit must exploit the vulnerability directly on the remote server.

2.1 0x03 — 3 pts

An HTTP server is running on port 8001 on the remote server for this exercise.

Listing 3: Source code of exercise 0x03

```
from flask import Flask, json, render_template, request, redirect, url_for
import mysql.connector
import os
import socket

flag = <SNIPPED>
app = Flask(__name__)

def opendb():
    config = <SNIPPED>

    connection = mysql.connector.connect(**config)
    cursor = connection.cursor()

    return (connection, cursor)

def closedb(connection, cursor):
    cursor.close()
    connection.close()

@app.route("/")
def main():
    return redirect(url_for('showSignIn'))

@app.route('/showSignIn')
def showSignIn():
    return render_template('signin.html')

@app.route('/signIn', methods=['POST'])
def signIn():
    _name       = request.form.get('inputName')
    _password = request.form.get('inputPassword')

    if not _name or not _password:
        error = {'message': '<span>Some fields are missing</span>'}
        return json.dumps(error)

    if _name and _password:
        (connection, cursor) = opendb()

        req = "SELECT * FROM tbl_user WHERE login='{}' AND password='{}'"
        cursor.execute(req.format(_name, _password))

        data = [(login, password) for (login, password) in cursor]
        closedb(connection, cursor)

        if len(data) != 0 and data[0][0] == 'admin':
            return json.dumps({'gg': 'GG! Flag: {}!'.format(flag)})
        else:
            error = {'error': 'No user found with these credentials.'}
            return json.dumps(error)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=8001)
```

2.2 0x04 — 2 pts

Note: this challenge is the most time-consuming.

An HTTP server is running on port 8002 on the remote server for this exercise. No source code provided, you're on your own!

Hint: you first have to find the correct table.