# OSED Exam Documentation

Offensive Security

your@email.tld, OSID: XXXXX

2022-01-21

# Contents

# 1 Offensive-Security OSED Exam Documentation

The Offensive Security OSED exam documentation contains all efforts that were conducted in order to pass the Offensive Security Exploit Developer exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has the technical knowledge required to pass the qualifications for the Offensive Security Exploit Developer certification.

## 1.1 Objective

The objective of this exam is to solve three given assignments as described in the control panel. The student is tasked with following a methodical approach in analyzing and solving the assignments. The exam report is meant to be a writeup of the steps taken to solve the assignment, including any analysis performed and code written.

## 1.2 Requirements

The student will be required to fill out this exam documentation fully and to include the following sections:

- High-Level summary of assignment solutions.
- Methodology walkthrough and detailed outline of steps taken through analysis and all written code.
- Each finding with included screenshots, walkthrough, sample code or reference.
- Screenshot of proof.txt.

# 2 High-Level Summary

A brief description of the assignments that were solved, including the overall exploitation steps.

# 3 Assignment 1

## 3.1 Proof.txt

`proof.txt`: `xxxx`

## 3.2 Initial Analysis

Provide relevant techniques and methods used to perform enumeration of the application, including network ports, security mitigations etc. The steps taken should be reproducible and easy to understand. Include any custom code or references to public tools.

**Listing 3.1:** Initial PoC for Assignment 1

```
#!/usr/bin/python

def main():
    # ...


if __name__ == "__main__":
    main()
```

## 3.3 Application Analysis

Provide a description of of the analysis performed against the application, this includes both dynamic and static analysis.

The analysis should include any reverse engineering performed to understand network protocols or file formats as well as how the application may be triggered to dispatch available commands.

## 3.4 Vulnerability Discovery

Provide relevant analysis steps to locate vulnerabilities inside the application, this includes both results from static analysis and dynamic analysis.

As part of the documentation, proof of concept Python3 code must be created and explained that triggers the vulnerabilities.

Only the steps that ended up working are required.

## 3.5 Exploit Creation

Provide a description of steps to create the exploit, this includes how to combine vulnerabilities, how to bypass DEP and how to write any custom shellcode. At the end of this section the full exploit code should be developed while an explanation of each step should be performed.

## 3.6 Screenshots

The exam control panel contains a section available to submit your proof files. The contents of the proof.txt files obtained from your exam machines must be submitted in the control panel before your exam has ended. Note that the control panel will not indicate whether the submitted proof is correct or not.

Each proof.txt found must be shown in a screenshot that includes the contents of the file, as well as the IP address of the target by using ipconfig.

```
C:\Users\Administrator\Desktop>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : localdomain
   IPv4 Address. . . . . . . . . . . : 192.168.13.37
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
                                       192.168.13.1

C:\Users\Administrator\Desktop>more proof.txt
xxxx
```

**Figure 3.1:** proof.txt

# 4 Assignment 2

## 4.1 Proof.txt

`proof.txt`: `xxxxx`

## 4.2 Initial Analysis

Provide relevant techniques and methods used to perform enumeration of the application, including network ports, security mitigations etc. The steps taken should be reproducible and easy to understand. Include any custom code or references to public tools.

**Listing 4.1:** Initial PoC for Assignment 2

```
#!/usr/bin/python

def main():
    # ...


if __name__ == "__main__":
    main()
```

## 4.3 Application Analysis

Provide a description of of the analysis performed against the application, this includes both dynamic and static analysis.

The analysis should include any reverse engineering performed to understand network protocols or file formats as well as how the application may be triggered to dispatch available commands.

## 4.4 Vulnerability Discovery

Provide relevant analysis steps to locate vulnerabilities inside the application, this includes both results from static analysis and dynamic analysis.

As part of the documentation, proof of concept Python3 code must be created and explained that triggers the vulnerabilities.

Only the steps that ended up working are required.

## 4.5 Exploit Creation

Provide a description of steps to create the exploit, this includes how to combine vulnerabilities, how to bypass DEP and how to write any custom shellcode. At the end of this section the full exploit code should be developed while an explanation of each step should be performed.

## 4.6 Screenshots

The exam control panel contains a section available to submit your proof files. The contents of the proof.txt files obtained from your exam machines must be submitted in the control panel before your exam has ended. Note that the control panel will not indicate whether the submitted proof is correct or not.

Each proof.txt found must be shown in a screenshot that includes the contents of the file, as well as the IP address of the target by using ipconfig.

```
C:\Users\Administrator\Desktop>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : localdomain
   IPv4 Address. . . . . . . . . . . : 192.168.13.37
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
                                       192.168.13.1

C:\Users\Administrator\Desktop>more proof.txt
xxxx
```

**Figure 4.1:** proof.txt

# 5 Assignment 3

## 5.1 Proof.txt

`proof.txt`: `xxxxx`

## 5.2 Initial Analysis

Provide a description of of the analysis performed against the application, this includes both dynamic and static analysis.

The analysis should include any reverse engineering performed to understand network protocols or file formats as well as how the application may be triggered to dispatch available commands.

**Listing 5.1:** Initial PoC for Assignment 3

```
#!/usr/bin/python

def main():
    # ...


if __name__ == "__main__":
    main()
```

## 5.3 Application Analysis

Provide a description of of the analysis performed against the application, this includes both dynamic and static analysis.

The analysis should include any reverse engineering performed to understand network protocols or file formats as well as how the application may be triggered to dispatch available commands.

## 5.4 Vulnerability Discovery

Provide relevant analysis steps to locate vulnerabilities inside the application, this includes both results from static analysis and dynamic analysis.

As part of the documentation, proof of concept Python3 code must be created and explained that triggers the vulnerabilities.

Only the steps that ended up working are required.

## 5.5 Exploit Creation

Provide a description of steps to create the exploit, this includes how to combine vulnerabilities, how to bypass DEP and how to write any custom shellcode. At the end of this section the full exploit code should be developed while an explanation of each step should be performed.

## 5.6 Screenshots

The exam control panel contains a section available to submit your proof files. The contents of the proof.txt files obtained from your exam machines must be submitted in the control panel before your exam has ended. Note that the control panel will not indicate whether the submitted proof is correct or not.

Each proof.txt found must be shown in a screenshot that includes the contents of the file, as well as the IP address of the target by using ipconfig.

```
C:\Users\Administrator\Desktop>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : localdomain
   IPv4 Address. . . . . . . . . . . : 192.168.13.37
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
                                       192.168.13.1

C:\Users\Administrator\Desktop>more proof.txt
xxxx
```

**Figure 5.1:** proof.txt

# 6 Appendix I - proof.txt files

**Table 6.1:** Proofs summary

| ID | IP | Proofs |
|---|---|---|
| **1** | x.x.x.x | **proof.txt** : xxxxxxxxxxxxxxxxxxxxxxxxxxxx |
| **2** | x.x.x.x | **proof.txt** : xxxxxxxxxxxxxxxxxxxxxxxxxxxx |
| **3** | x.x.x.x | **proof.txt** : xxxxxxxxxxxxxxxxxxxxxxxxxxxx |