
OSMR Exam Documentation

OffSec

your@email.tld, OSID: XXXXX

2023-03-29

Contents

1	Offsec OSMR Exam Documentation	1
1.1	Objective	1
1.2	Requirements	1
2	High-Level Summary	2
3	Assignment 1	3
3.1	Proof.txt	3
3.2	Initial Analysis	3
3.3	Vulnerability Discovery	3
3.4	Exploit/Bypass Creation	3
3.5	Screenshots	4
4	Assignment 2	5
4.1	Proof.txt	5
4.2	Initial Analysis	5
4.3	Vulnerability Discovery	5
4.4	Exploit/Bypass Creation	5
4.5	Screenshots	6
5	Assignment 3	7
5.1	Proof.txt	7
5.2	Initial Analysis	7
5.3	Vulnerability Discovery	7
5.4	Exploit/Bypass Creation	7
5.5	Screenshots	8
6	Assignment 4	9
6.1	Proof.txt	9
6.2	Initial Analysis	9
6.3	Vulnerability Discovery	9
6.4	Exploit/Bypass Creation	9

6.5	Screenshots	10
7	Appendix I: Assignment Files	11
8	Appendix II: proof.txt files	12

1 Offsec OSMR Exam Documentation

The OffSec OSMR exam documentation contains all efforts that were conducted in order to pass the OffSec macOS Researcher 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 OffSec macOS Researcher certification.

1.1 Objective

The objective of this exam is to solve four 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.
- Screenshots of proofs.

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 and discovery of the application and/or the environment. 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

```
1 #!/usr/bin/env bash
2
3 ...
```

3.3 Vulnerability Discovery

Provide relevant analysis steps to locate vulnerability inside the application or environment, this includes results from static analysis and/or dynamic analysis. Explain the vulnerability identified.

Only the steps that ended up working are required.

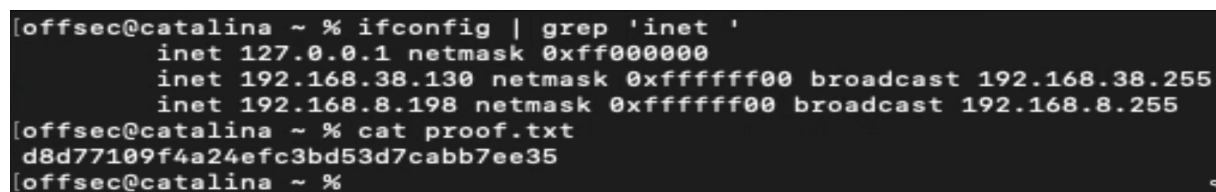
3.4 Exploit/Bypass Creation

Provide a description of steps to create the exploit or security control bypass. At the end of this section the full exploit (or bypass) code should be developed while an explanation of each step should be performed.

3.5 Screenshots

The exam control panel contains a section available to submit your proof files. The contents of the local.txt, proof.txt or secret.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 local.txt, proof.txt or secret.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.

A terminal window screenshot with a black background and white text. The prompt is [offsec@catalina ~ %]. The first command is 'ifconfig | grep 'inet '' which outputs three lines of IP configuration: 'inet 127.0.0.1 netmask 0xff000000', 'inet 192.168.38.130 netmask 0xffffffff broadcast 192.168.38.255', and 'inet 192.168.8.198 netmask 0xffffffff broadcast 192.168.8.255'. The second command is 'cat proof.txt' which outputs a single line of a hexadecimal hash: 'd8d77109f4a24efc3bd53d7cabb7ee35'. The prompt returns to [offsec@catalina ~ %].

```
[offsec@catalina ~ % ifconfig | grep 'inet '  
    inet 127.0.0.1 netmask 0xff000000  
    inet 192.168.38.130 netmask 0xffffffff broadcast 192.168.38.255  
    inet 192.168.8.198 netmask 0xffffffff broadcast 192.168.8.255  
[offsec@catalina ~ % cat proof.txt  
d8d77109f4a24efc3bd53d7cabb7ee35  
[offsec@catalina ~ %
```

Figure 3.1: proof.txt

4 Assignment 2

4.1 Proof.txt

`proof.txt: xxxx`

4.2 Initial Analysis

Provide relevant techniques and methods used to perform enumeration and discovery of the application and/or the environment. 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

```
1  #include<stdio.h>
2
3  int main() {
4      // ...
5      return 0;
6  }
```

4.3 Vulnerability Discovery

Provide relevant analysis steps to locate vulnerability inside the application or environment, this includes results from static analysis and/or dynamic analysis. Explain the vulnerability identified.

Only the steps that ended up working are required.

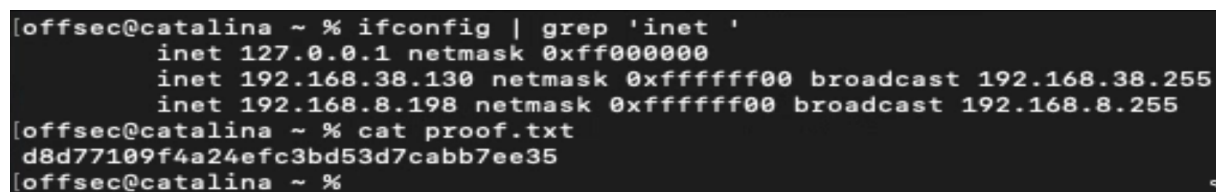
4.4 Exploit/Bypass Creation

Provide a description of steps to create the exploit or security control bypass. At the end of this section the full exploit (or bypass) code should be developed while an explanation of each step should be performed.

4.5 Screenshots

The exam control panel contains a section available to submit your proof files. The contents of the local.txt, proof.txt or secret.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 local.txt, proof.txt or secret.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.

A screenshot of a terminal window with a black background and white text. The prompt is [offsec@catalina ~ %]. The first command is 'ifconfig | grep 'inet '' which outputs three lines of network configuration: 'inet 127.0.0.1 netmask 0xff000000', 'inet 192.168.38.130 netmask 0xffffffff00 broadcast 192.168.38.255', and 'inet 192.168.8.198 netmask 0xffffffff00 broadcast 192.168.8.255'. The second command is 'cat proof.txt' which outputs a single line of hexadecimal: 'd8d77109f4a24efc3bd53d7cabb7ee35'. The prompt returns to [offsec@catalina ~ %].

```
[offsec@catalina ~ % ifconfig | grep 'inet '  
    inet 127.0.0.1 netmask 0xff000000  
    inet 192.168.38.130 netmask 0xffffffff00 broadcast 192.168.38.255  
    inet 192.168.8.198 netmask 0xffffffff00 broadcast 192.168.8.255  
[offsec@catalina ~ % cat proof.txt  
d8d77109f4a24efc3bd53d7cabb7ee35  
[offsec@catalina ~ %
```

Figure 4.1: proof.txt

5 Assignment 3

5.1 Proof.txt

proof.txt: xxxx

5.2 Initial Analysis

Provide relevant techniques and methods used to perform enumeration and discovery of the application and/or the environment. The steps taken should be reproducible and easy to understand. Include any custom code or references to public tools.

Listing 5.1: Initial PoC for Assignment 3

```
1  #import <Foundation/Foundation.h>
2
3  int main() {
4      NSLog(@"...");
5      // ...
6      return 0;
7  }
```

5.3 Vulnerability Discovery

Provide relevant analysis steps to locate vulnerability inside the application or environment, this includes results from static analysis and/or dynamic analysis. Explain the vulnerability identified.

Only the steps that ended up working are required.

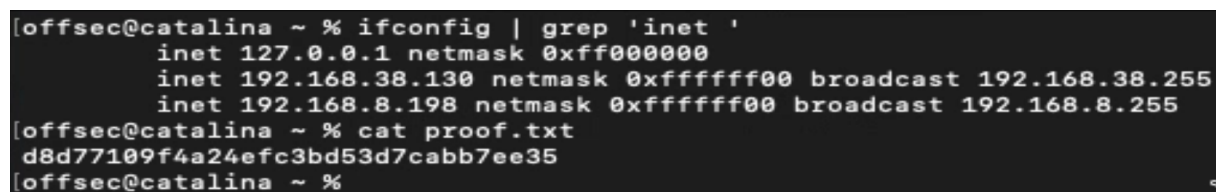
5.4 Exploit/Bypass Creation

Provide a description of steps to create the exploit or security control bypass. At the end of this section the full exploit (or bypass) code should be developed while an explanation of each step should be performed.

5.5 Screenshots

The exam control panel contains a section available to submit your proof files. The contents of the local.txt, proof.txt or secret.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 local.txt, proof.txt or secret.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.

A screenshot of a terminal window with a black background and white text. The prompt is [offsec@catalina ~ %]. The first command is 'ifconfig | grep 'inet '' which outputs three lines of IP configuration: 'inet 127.0.0.1 netmask 0xff000000', 'inet 192.168.38.130 netmask 0xffffffff00 broadcast 192.168.38.255', and 'inet 192.168.8.198 netmask 0xffffffff00 broadcast 192.168.8.255'. The second command is 'cat proof.txt' which outputs a single line of a hexadecimal hash: 'd8d77109f4a24efc3bd53d7cabb7ee35'. The prompt returns to [offsec@catalina ~ %].

```
[offsec@catalina ~ % ifconfig | grep 'inet '  
    inet 127.0.0.1 netmask 0xff000000  
    inet 192.168.38.130 netmask 0xffffffff00 broadcast 192.168.38.255  
    inet 192.168.8.198 netmask 0xffffffff00 broadcast 192.168.8.255  
[offsec@catalina ~ % cat proof.txt  
d8d77109f4a24efc3bd53d7cabb7ee35  
[offsec@catalina ~ %
```

Figure 5.1: proof.txt

6 Assignment 4

6.1 Proof.txt

`proof.txt: xxxx`

6.2 Initial Analysis

Provide relevant techniques and methods used to perform enumeration and discovery of the application and/or the environment. The steps taken should be reproducible and easy to understand. Include any custom code or references to public tools.

Listing 6.1: Initial PoC for Assignment 4

```
1  #include<stdio.h>
2
3  int main() {
4      // ...
5      return 0;
6  }
```

6.3 Vulnerability Discovery

Provide relevant analysis steps to locate vulnerability inside the application or environment, this includes results from static analysis and/or dynamic analysis. Explain the vulnerability identified.

Only the steps that ended up working are required.

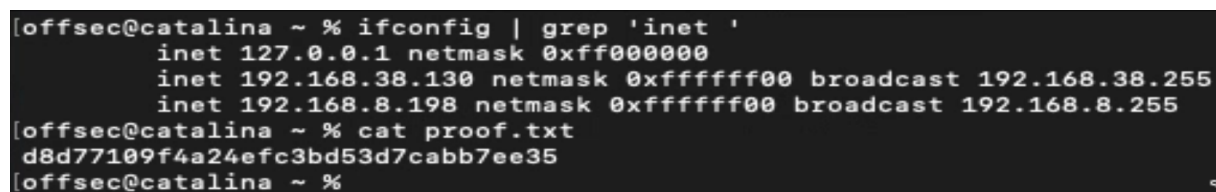
6.4 Exploit/Bypass Creation

Provide a description of steps to create the exploit or security control bypass. At the end of this section the full exploit (or bypass) code should be developed while an explanation of each step should be performed.

6.5 Screenshots

The exam control panel contains a section available to submit your proof files. The contents of the local.txt, proof.txt or secret.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 local.txt, proof.txt or secret.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.

A terminal window screenshot with a black background and white text. The prompt is [offsec@catalina ~ %]. The first command is 'ifconfig | grep 'inet '' which outputs three lines of IP configuration: 'inet 127.0.0.1 netmask 0xff000000', 'inet 192.168.38.130 netmask 0xffffffff00 broadcast 192.168.38.255', and 'inet 192.168.8.198 netmask 0xffffffff00 broadcast 192.168.8.255'. The second command is 'cat proof.txt' which outputs a single line of a hexadecimal hash: 'd8d77109f4a24efc3bd53d7cabb7ee35'. The prompt returns to [offsec@catalina ~ %].

```
[offsec@catalina ~ % ifconfig | grep 'inet '  
    inet 127.0.0.1 netmask 0xff000000  
    inet 192.168.38.130 netmask 0xffffffff00 broadcast 192.168.38.255  
    inet 192.168.8.198 netmask 0xffffffff00 broadcast 192.168.8.255  
[offsec@catalina ~ % cat proof.txt  
d8d77109f4a24efc3bd53d7cabb7ee35  
[offsec@catalina ~ %
```

Figure 6.1: proof.txt

7 Appendix I: Assignment Files

The following is a table of all assignment files that are attached with this report.

Table 7.1: Attached assignment files

Task	Assignment file
1	assignment1.sh
2	assignment2.c
3	assignment3.m
4	assignment4.c

8 Appendix II: proof.txt files

Table 8.1: Proofs summary

ID	IP	Proofs
1	X.X.X.X	proof.txt : XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2	X.X.X.X	proof.txt : XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3	X.X.X.X	proof.txt : XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4	X.X.X.X	proof.txt : XXXXXXXXXXXXXXXXXXXXXXXXXXXXX