Offensive Security
Penetration Test Report for Internal Lab and Exam

v1.0

mailme@example.com

OSID: OS-XXXXX

Contents

1 High-Level Summary

Someone Student was tasked with performing an internal penetration test towards Offensive Security Labs. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal lab systems – the THINC.local domain. Someone's overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Offensive Security's network. When performing the attacks, Someone was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, Someone had administrative level access to multiple systems. All systems were successfully exploited and local access granted. Some machines were granted root/administrator access. These systems as well as a brief description on how access was obtained are listed below:

- 10.11.1.1 – Got in through SomeWebService <= 1.3.3 RCE (CVE-2001-10000)
- 127.0.0.1 – Got in through Black Magic 0.9 RCE (CVE-2018-1234)

1.1 Recommendations

Someone recommends patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date. Developing your own software is a big risk. Let you developers be more security aware and have the software and source code audited by security specialists.

2 Methodologies

Someone utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments are secure. Below is a breakout of how Someone was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

2.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, Someone was tasked with exploiting the lab and exam network. The specific IP addresses were:

Network: 10.11.1.1, 127.0.0.1

2.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

IP Address	Open ports
10.11.1.1	**TCP:** 1, 2, 3, 4 **UDP:** 5, 6, 7, 8
127.0.0.1	**TCP:** 1, 2, 3, 4 **UDP:** 5, 6, 7, 8

Table 1: Service enumeration

2.3 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, Someone was able to successfully gain access to 2 out of the 2 systems. On 1 out of the 2 machines, Someone gained root/administrator privileges.

Vulnerability Exploited: SomeWebService <= 1.3.3 RCE (CVE-2001-10000)

Sytem Vulnerable: 10.11.1.1

Vulnerability Explanation SomeWebService is subject to a path traversal vulnerability leading to remote code execution. This vulnerability affects SomeWebService versions 1.3.3 and below. This is caused by an unchecked "hackme" parameter that is used to override the flux capacitor bypass.

Vulnerability Fix Install the latest version of SomeWebService, current latest version 2.1.

Severity Critical

Proof of Concept Code The only modifications made to the proof of concept code is where the reverse shell should connect to. See marked in red below.

The exploit can be found here: https://www.exploitdb.com/exploits/1337

```
#!/bin/bash
echo argument 1: $1
echo argument 2: $2
ping -c 5 127.0.0.1
echo all arguments: $@
```

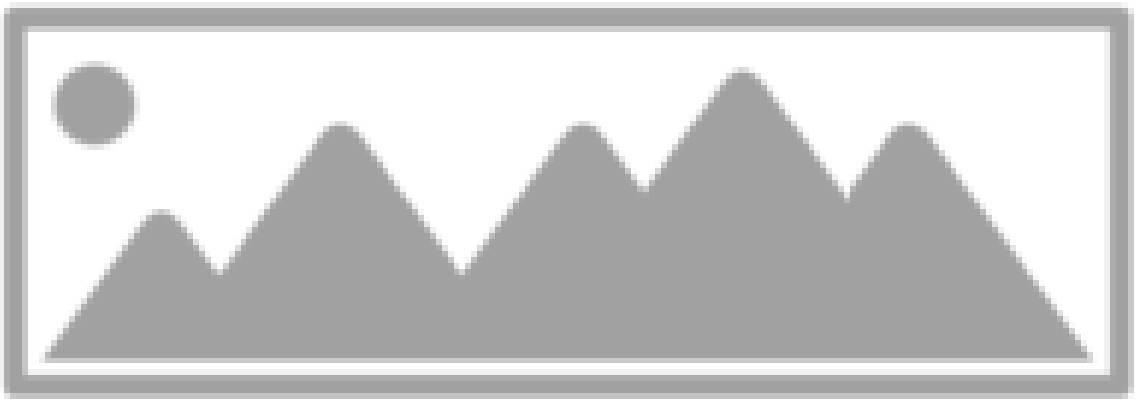
Information gathering, with nmap scan

```
Starting Nmap 7.80 ( https://nmap.org ) at 2001-01-01 13:37 CET
NSE: Loaded 45 scripts for scanning.
Initiating Ping Scan at 09:23
Scanning 10.11.1.1 [4 ports]
Completed Ping Scan at 09:24, 0.16s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:24
Completed Parallel DNS resolution of 1 host. at 09:24, 8.02s elapsed
Initiating SYN Stealth Scan at 09:24
Scanning 10.11.1.1 [65535 ports]
Discovered open port 1/tcp on 10.11.1.1
...
Completed SYN Stealth Scan at 13:38, 41.69s elapsed (65535 total ports)
Initiating Service scan at 13:38
Scanning 13 services on 10.11.1.1
Service scan Timing: About 53.85% done; ETC: 09:26 (0:00:48 remaining)
Completed Service scan at 09:25, 55.42s elapsed (13 services on 1 host)
NSE: Script scanning 10.11.1.1.
Initiating NSE at 09:25
Completed NSE at 09:25, 0.66s elapsed
Initiating NSE at 09:25
Completed NSE at 09:25, 0.48s elapsed
Nmap scan report for 10.11.1.1
Host is up, received echo-reply ttl 127 (0.12s latency).
Not shown: 63673 closed ports, 1849 filtered ports
Reason: 63673 resets and 1849 no-responses
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT STATE SERVICE REASON VERSION
1/tcp open msrpc syn-ack ttl 127 Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 1995 R1 - 2000; CPE: cpe:/o:microsoft:windows
Read data files from: /usr/bin/../share/nmap
```

```
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 107.01 seconds
Raw packets sent: 85444 (3.760MB) | Rcvd: 74052 (2.979M
```

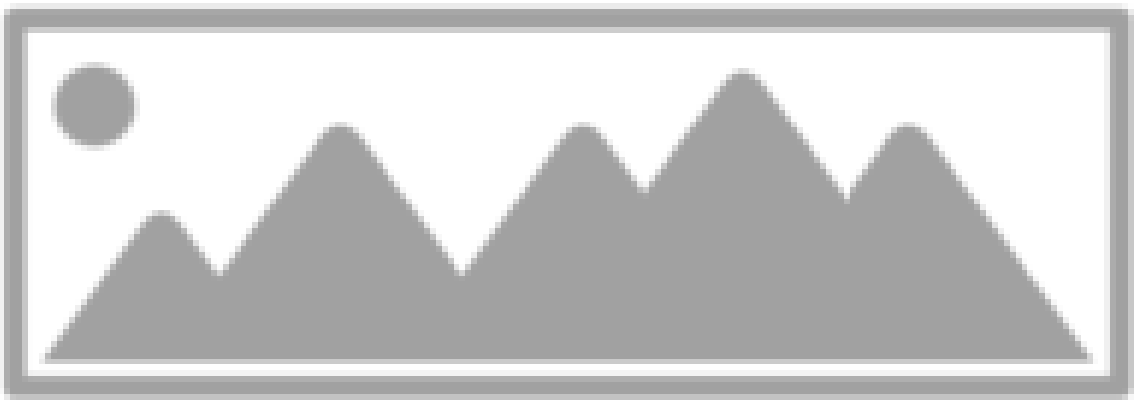
Check the robots.txt

Checking the robots.txt on the server, reveals some interesting urls.



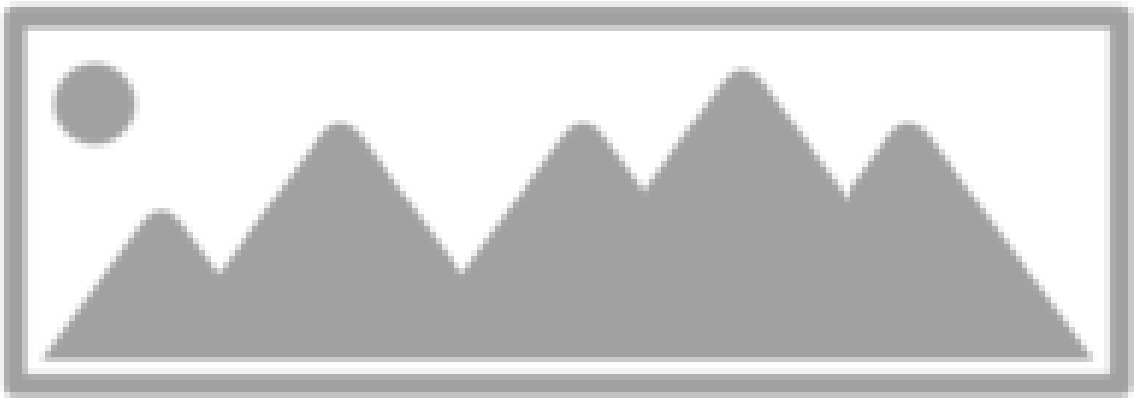
Check the url /SomeWebService/

The url /SomeWebService/ reveals a very interesting web application.

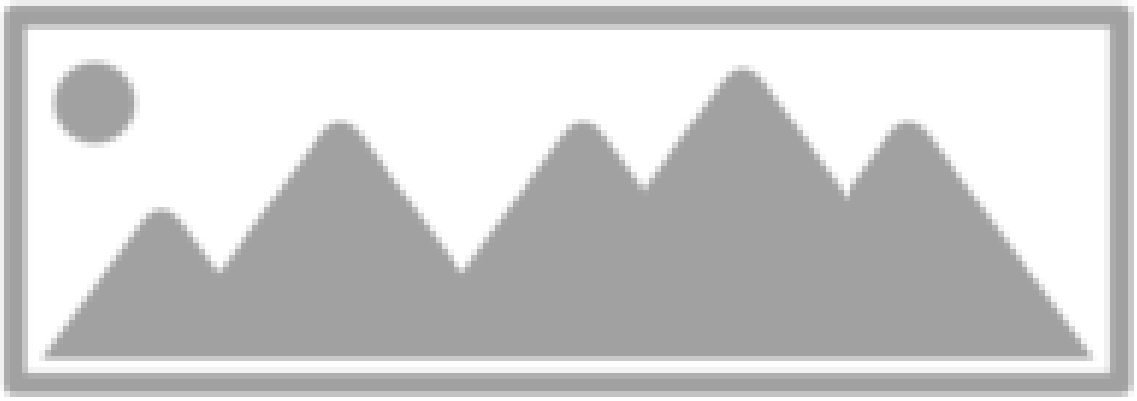


Check version

The source of the page reveals the version is 1.3.3.7:



Default credentials
Searching for default credentials works and we find “admin/admin”:

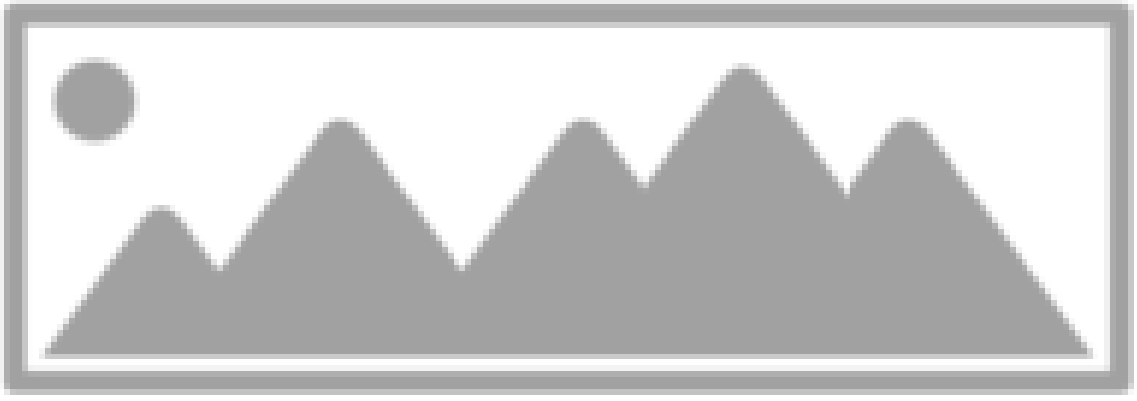


Exploit searching
A quick search finds the following exploit: https://www.exploit-db.com/exploits/1337

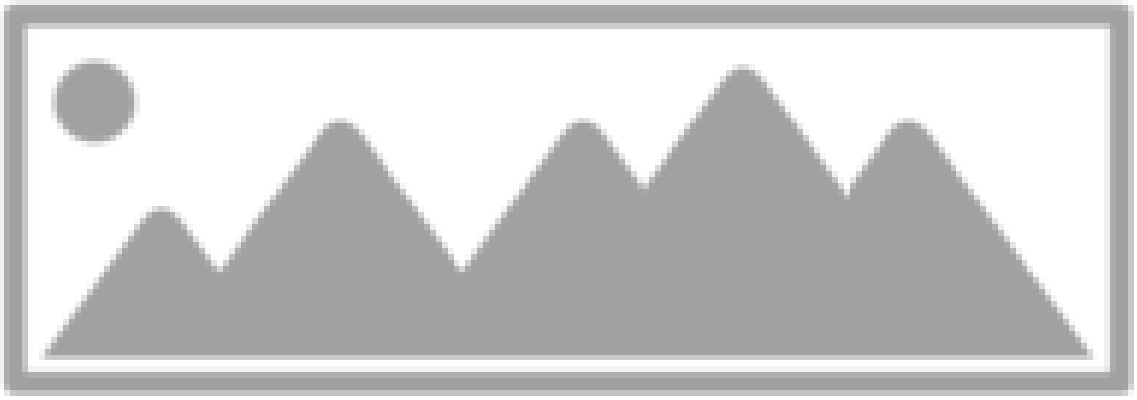
Exploiting We setup a reverse shell and trigger the following url:

http://10.11.1.1/SomeWebService/?flux=GimmeReverseShell

Which results in a reverse shell connecting back to our machine



Local Proof
The reverse shell alows us to retrieve the local.txt file.



Vulnerability Exploited: snapd < 2.37 (Ubuntu) - 'dirty_sock' Local Privilege Escalation

Sytem Vulnerable: 10.11.1.1

Vulnerability Explanation This exploit bypasses access control checks to use a restricted API function (POST /v2/create-user) of the local snapd service. This queries the Ubuntu SSO for a username and public SSH key of a provided email address, and then creates a local user based on these value.

Vulnerability Fix Install a version greater than 2.37 of snapd.

Severity Critical

Proof of Concept Code No modifications made to the proof of concept code.

The exploit can be found here: https://www.exploit-db.com/exploits/46361

Further exploitation During the exam we ran linuxprivchecker.py to find a vulnerability:

https://github.com/sleventyeleven/linuxprivchecker/blob/master/linuxprivchecker.py

The output quickly reveals that snapd is vulnerable.

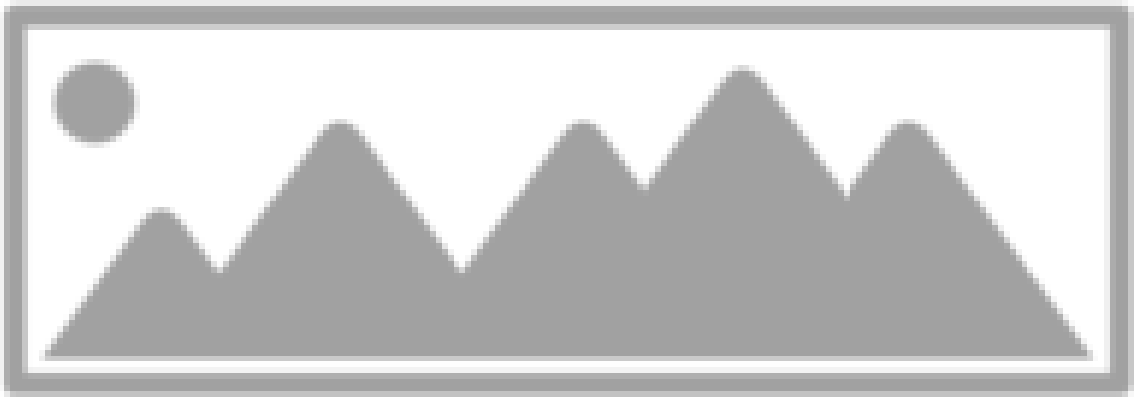
```
<snip>

[+] snapd is vulnerable!

<snip>
```

Exploit searching
A quick search finds the following exploit: https://www.exploit-db.com/exploits/46361

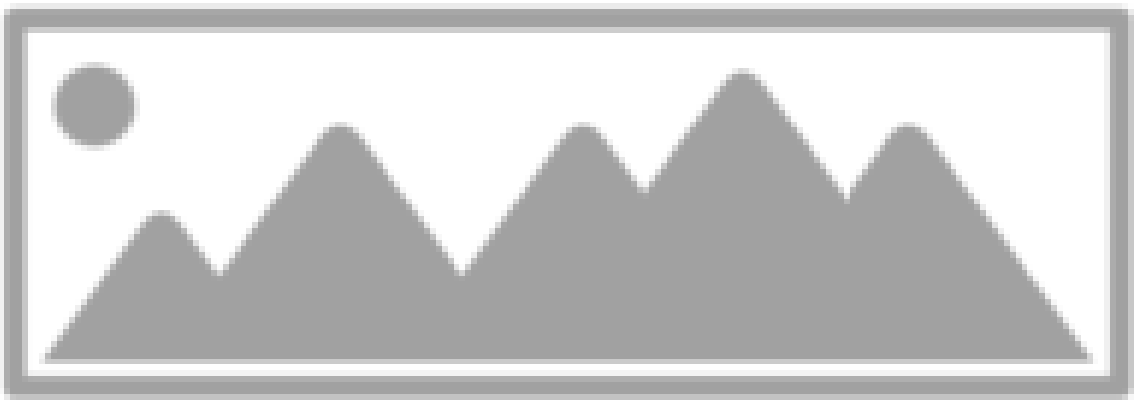
Exploiting We setup a reverse shell and we run the script.

```
dirty_sock.py 10.11.1.1
```

Which results in a reverse shell connecting back to our machine



Proof
The reverse shell alows us to retrieve the proof.txt file.



Vulnerability Exploited: Black Magic 0.9 RCE (CVE-2018-1234)

Sytem Vulnerable: 127.0.0.1

Vulnerability Explanation
Black Magic is subject to vulnerability leading to remote code execution. This vulnerability affects all Black Magic versions prior to 1.0. This is caused by a junior programmer who makes many big mistakes.

Vulnerability Fix
Install the latest version of Black Magic, current latest version 1.0.

Severity Critical

Proof of Concept Code The only modifications made to the proof of concept code is where the reverse shell should connect to. See marked in red below. The exploit can be found here: https://www.exploitdb.com/exploits/1337

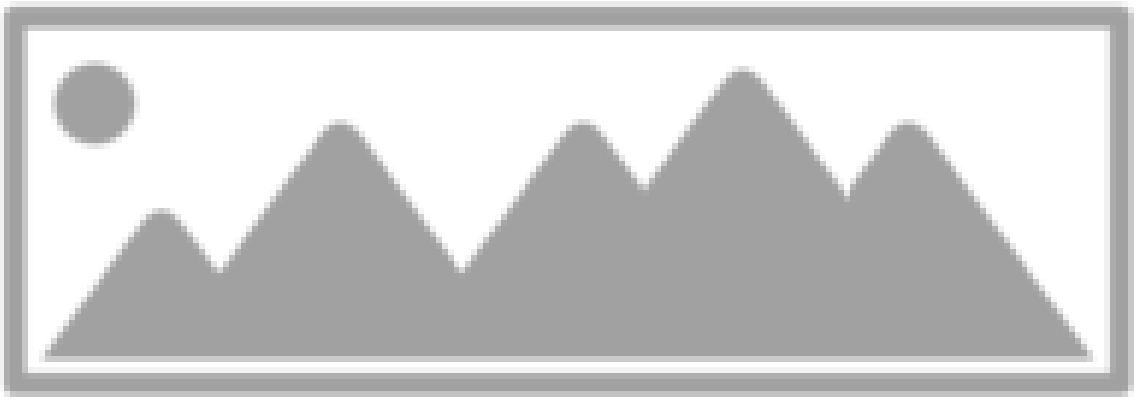
```
#!/bin/bash
echo argument 1: $1
echo argument 2: $2
echo all arguments: $@
```

Information gathering, with nmap scan

```
Starting Nmap 7.80 ( https://nmap.org ) at 2001-01-01 13:37 CET
NSE: Loaded 45 scripts for scanning.
Initiating Ping Scan at 09:23
Scanning 127.0.0.1 [4 ports]
Completed Ping Scan at 09:24, 0.16s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:24
Completed Parallel DNS resolution of 1 host. at 09:24, 8.02s elapsed
Initiating SYN Stealth Scan at 09:24
Scanning 127.0.0.1 [65535 ports]
Discovered open port 1/tcp on 127.0.0.1
...
Completed SYN Stealth Scan at 13:38, 41.69s elapsed (65535 total ports)
Initiating Service scan at 13:38
Scanning 13 services on 127.0.0.1
Service scan Timing: About 53.85% done; ETC: 09:26 (0:00:48 remaining)
Completed Service scan at 09:25, 55.42s elapsed (13 services on 1 host)
NSE: Script scanning 127.0.0.1.
Initiating NSE at 09:25
Completed NSE at 09:25, 0.66s elapsed
Initiating NSE at 09:25
Completed NSE at 09:25, 0.48s elapsed
Nmap scan report for 127.0.0.1
Host is up, received echo-reply ttl 127 (0.12s latency).
Not shown: 63673 closed ports, 1849 filtered ports
Reason: 63673 resets and 1849 no-responses
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT STATE SERVICE REASON VERSION
1/tcp open msrpc syn-ack ttl 127 Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 1995 R1 - 2000; CPE: cpe:/o:microsoft:windows
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 107.01 seconds
Raw packets sent: 85444 (3.760MB) | Rcvd: 74052 (2.979M
```

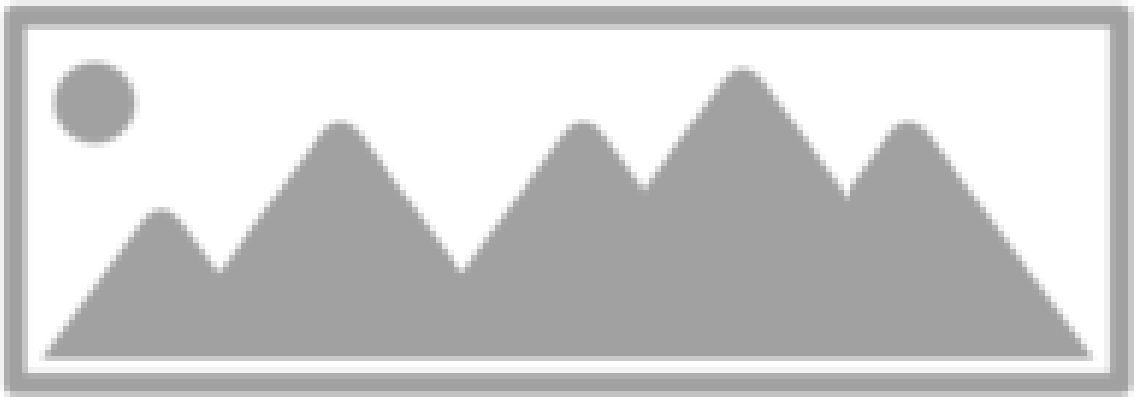
Check the robots.txt

The robots.txt file shows some interesting urls.



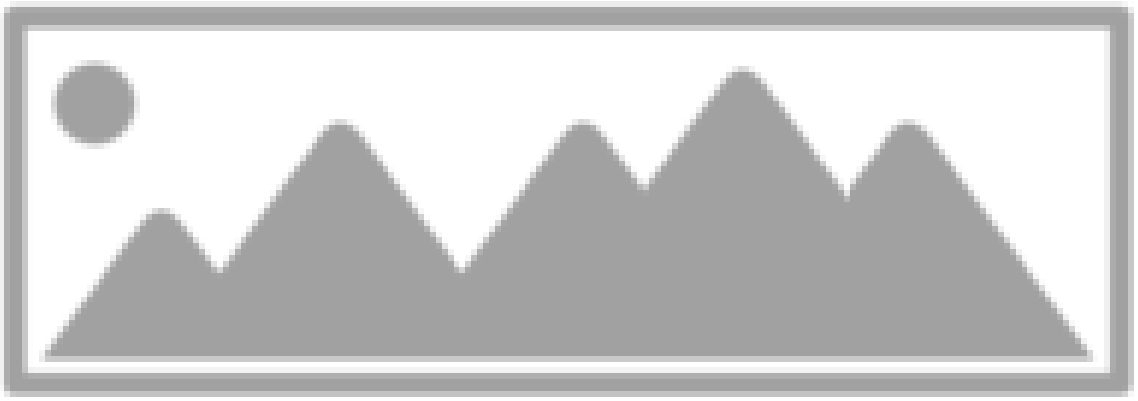
Check the url /SomeWebService/

The url reveals a web application.



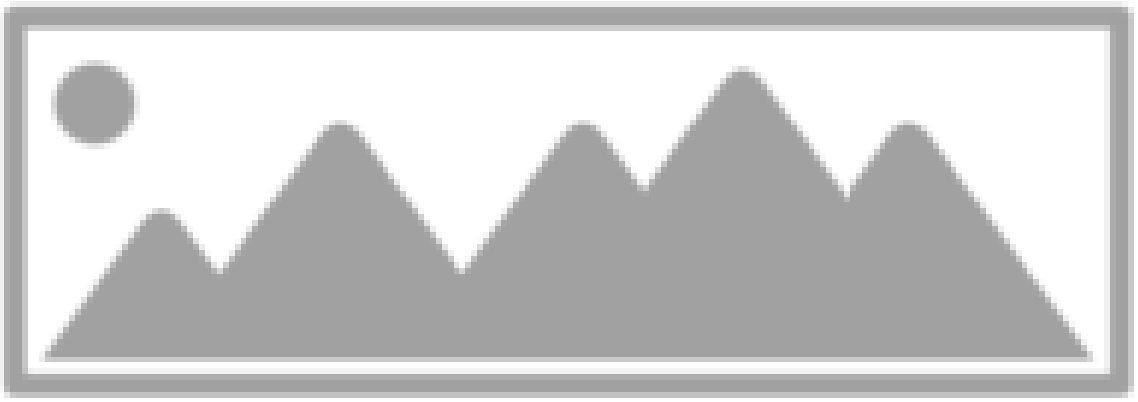
Check version

The source of the page reveals the version is 0.9.9.9:



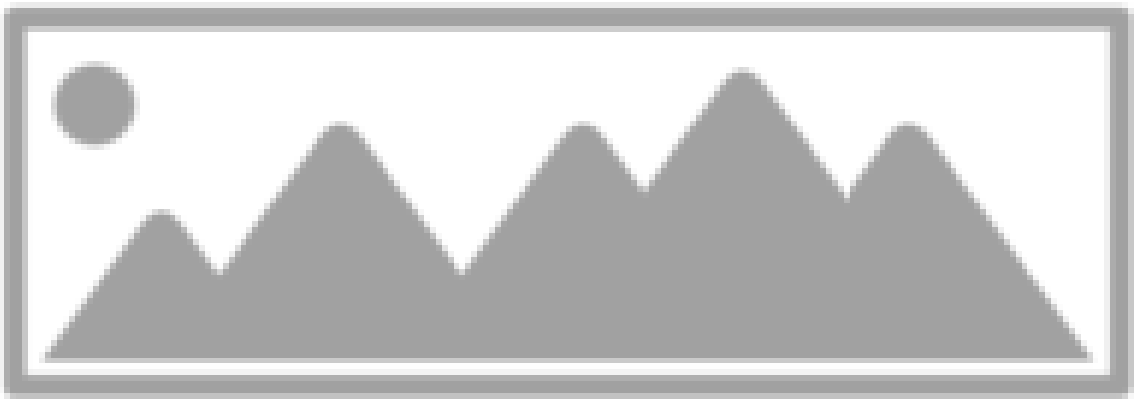
Default credentials

Searching for default credentials works and we find “admin/admin”:



Exploit searching
A quick search finds the following exploit: https://www.exploit-db.com/exploits/1337

Exploiting We setup a reverse shell and trigger the following url: http://127.0.0.1/SomeWebService/?flux=GimmeReve
Which results in a reverse shell connecting back to our machine



Proof
Here we are able to get the proof.txt.