PENETRATION TEST REPORT

for

Sitting Duck B.V.

V1.0
Amsterdam
January 26th, 2015

Document Properties

Client	Sitting Duck B.V.
Title	PENETRATION TEST REPORT
Targets	bla fishinabarrel.sittingduck.com bla fishinabarrel.sittingduck.com
Version	1.0
Pentesters	Melanie Rieback, Aristotle, George Boole, William of Ockham, Ludwig Josef Johann Wittgenstein
Authors	Patricia Piolon, Ernest Hemingway, JRR Tolkien, Arthur Conan Doyle
Reviewed by	Melanie Rieback
Approved by	Melanie Rieback

Version control

Version	Date	Author	Description
0.1	January 19th, 2015	Patricia Piolon	Initial draft
0.2	January 20th, 2015	Ernest Hemingway	Structure & contents revision
0.3	January 21st, 2015	Patricia Piolon	Added threat levels and recommendations
0.4	January 22nd, 2015	Patricia Piolon, JRR Tolkien	Revision
0.5	January 23rd, 2015	Patricia Piolon	Revision
1.0	January 26th, 2015	Arthur Conan Doyle	Finalizing

Contact

For more information about this Document and its contents please contact Radically Open Security B.V.

Name	Melanie Rieback
Address	Zieseniskade 21 1017 RT Amsterdam The Netherlands
Phone	+31 6 10 21 32 40
Email	info@radicallyopensecurity.com

Table of Contents

1 Executive Summary

1.1 Introduction

Sitting Duck B.V. (“Sitting Duck”) has assigned the task of performing a 6-day Penetration Test of the FishInABarrel Web Application to Radically Open Security BV (hereafter “ROS”). Sitting Duck has made this request to better evaluate the security of the application and to identify application level vulnerabilities in order to see whether the FishInABarrel Web Application is ready, security-wise, for production deployment.

This report contains our findings as well as detailed explanations of exactly how ROS performed the penetration test.

1.2 Scope of work

The scope of the Sitting Duck penetration test was limited to the following target:

- bla
  fishinabarrel.sittingduck.com
- bla
  fishinabarrel.sittingduck.com

The penetration test was carried out from a black box perspective: no information regarding the system(s) tested was provided by Sitting Duck or FishInABarrel, although FishInABarrel did provide ROS with two test user accounts.

1.3 Project objectives

The objective of the security assessment is to gain insight into the security of the host and the FishInABarrel Web Application.

1.4 Timeline

The FishInABarrel Security Audit took place between January 14 and January 16, 2015.

1.5 Results in a Nutshell

During this pentest, we found quite a number of different security problems – Cross-site Scripting (XSS) vulnerabilities, both stored and reflected, Cross-site Request Forgery (CSRF) vulnerabilities, information disclosures (multiple instances), and lack of brute force protection.

1.6 Summary of Findings

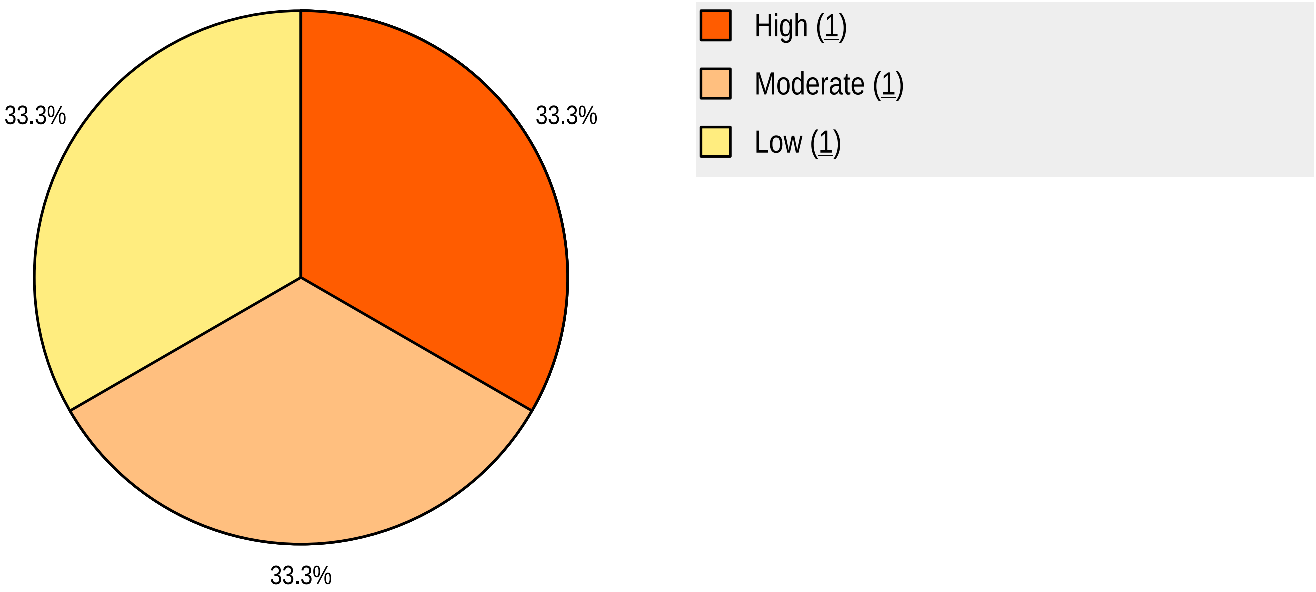
ID	Type	Description	Threat level
SID-002	XSS	A general description of the problem.	High
SID-001	Information Leak	The phpinfo() function of the PHP language is readable, resulting in a listing of all the runtime information of the environment, thus disclosing potentially valuable information to attackers.	Moderate
SID-003	XSS	A description of the problem.	Low

1.7 Summary of Recommendations

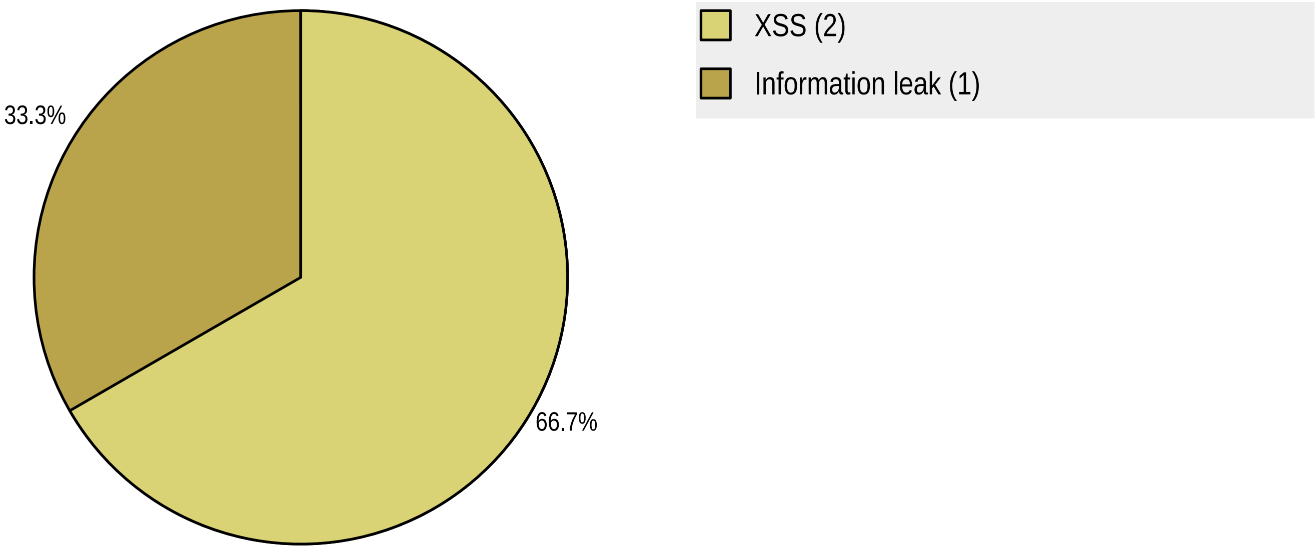
ID	Type	Recommendation
SID-001	Information Leak	Here is where we write some tips to solve the problem.
SID-002	XSS	This is where we solve everything and the sun starts shining again.
SID-003	XSS	A ready solution.

1.8 Charts

1.8.1 Findings by Threat Level



1.8.2 Findings by Type



2 Methodology

2.1 Planning

Our general approach during this penetration test was as follows:

1. **Reconnaissance**
   We attempted to gather as much information as possible about the target. Reconnaissance can take two forms: active and passive. A passive attack is always the best starting point as this would normally defeat intrusion detection systems and other forms of protection, etc., afforded to the network. This would usually involve trying to discover publicly available information by utilizing a web browser and visiting newsgroups etc. An active form would be more intrusive and may show up in audit logs and may take the form of a social engineering type of attack.
2. **Enumeration**
   We used varied operating system fingerprinting tools to determine what hosts are alive on the network and more importantly what services and operating systems they are running. Research into these services would be carried out to tailor the test to the discovered services.
3. **Scanning**
   Through the use of vulnerability scanners, all discovered hosts would be tested for vulnerabilities. The result would be analyzed to determine if there any vulnerabilities that could be exploited to gain access to a target host on a network.
4. **Obtaining Access**
   Through the use of published exploits or weaknesses found in applications, operating system and services access would then be attempted. This may be done surreptitiously or by more brute force methods.

2.2 Risk Classification

Throughout the document, each vulnerability or risk identified has been labeled and categorized as:

- **Extreme**
  Extreme risk of security controls being compromised with the possibility of catastrophic financial/ reputational losses occurring as a result.
- **High**
  High risk of security controls being compromised with the potential for significant financial/ reputational losses occurring as a result.
- **Elevated**
  Elevated risk of security controls being compromised with the potential for material financial/ reputational losses occurring as a result.
- **Moderate**

Moderate risk of security controls being compromised with the potential for limited financial/ reputational losses occurring as a result.

- **Low**
  Low risk of security controls being compromised with measurable negative impacts as a result.

Please note that this risk rating system was taken from the Penetration Testing Execution Standard (PTES). For more information, see: http://www.pentest-standard.org/index.php/Reporting. 6

3 Reconnaissance and Fingerprinting

Through automated scans we were able to gain the following information about the software and infrastructure. Detailed scan output can be found in the sections below.

Fingerprinted Information
Windows XP Microsoft IIS 6.0 PHP 5.4.29 jQuery 1.7.2 Mailserver XYZ FTPserver ABC

3.1 Automated Scans

As part of our active reconnaissance we used the following automated scans:

- nmap – http://nmap.org
- skipfish - https://code.google.com/p/skipfish/
- sqlmap – http://sqlmap.org
- Wapiti – http://wapiti.sourceforge.net

Of these, only the output of nmap turned out to be useful; consequently only nmap and output will be discussed in this section.

3.2 nmap

Command:

```
$ nmap -vvvv -oA fishinabarrel.sittingduck.com_complete -sV -sC -A -p1-65535 -T5
```

```
fishinabarrel.sittingduck.com
```

Outcome:

```
 Nmap scan report for fishinabarrel.sittingduck.com (10.10.10.1)
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2013-11-11 15:43 EST
Initiating ARP Ping Scan against 10.10.10.1 [1 port] at 15:43
The ARP Ping Scan took 0.01s to scan 1 total hosts.
Initiating SYN Stealth Scan against fishinabarrel.sittingduck.com (10.10.10.1) [1680 ports] at 15:43
Discovered open port 22/tcp on 10.10.10.1
Discovered open port 80/tcp on 10.10.10.1
Discovered open port 8888/tcp on 10.10.10.1
Discovered open port 111/tcp on 10.10.10.1
Discovered open port 3306/tcp on 10.10.10.1
Discovered open port 957/tcp on 10.10.10.1
The SYN Stealth Scan took 0.30s to scan 1680 total ports.
Host fishinabarrel.sittingduck.com (10.10.10.1) appears to be up ... good.
Interesting ports on fishinabarrel.sittingduck.com (10.10.10.1):
Not shown: 1674 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
110/tcp  open  pop3
111/tcp  open  rpcbind
957/tcp  open  unknown
3306/tcp open  mysql
4000/tcp open  dangerous service

Nmap finished: 1 IP address (1 host up) scanned in 0.485 seconds
Raw packets sent: 1681 (73.962KB) | Rcvd: 1681 (77.322KB)
```

The scan revealed a very large number of open services on this machine, which greatly increases the attack surface; see SID-002 (page 10) for more information on the security risk.

4 Pentest Technical Summary

4.1 Findings

We have identified the following issues:

4.1.1 SID-001 — PHPInfo Disclosure

Vulnerability ID: SID-001

Vulnerability type: Information Leak

Threat level: Moderate

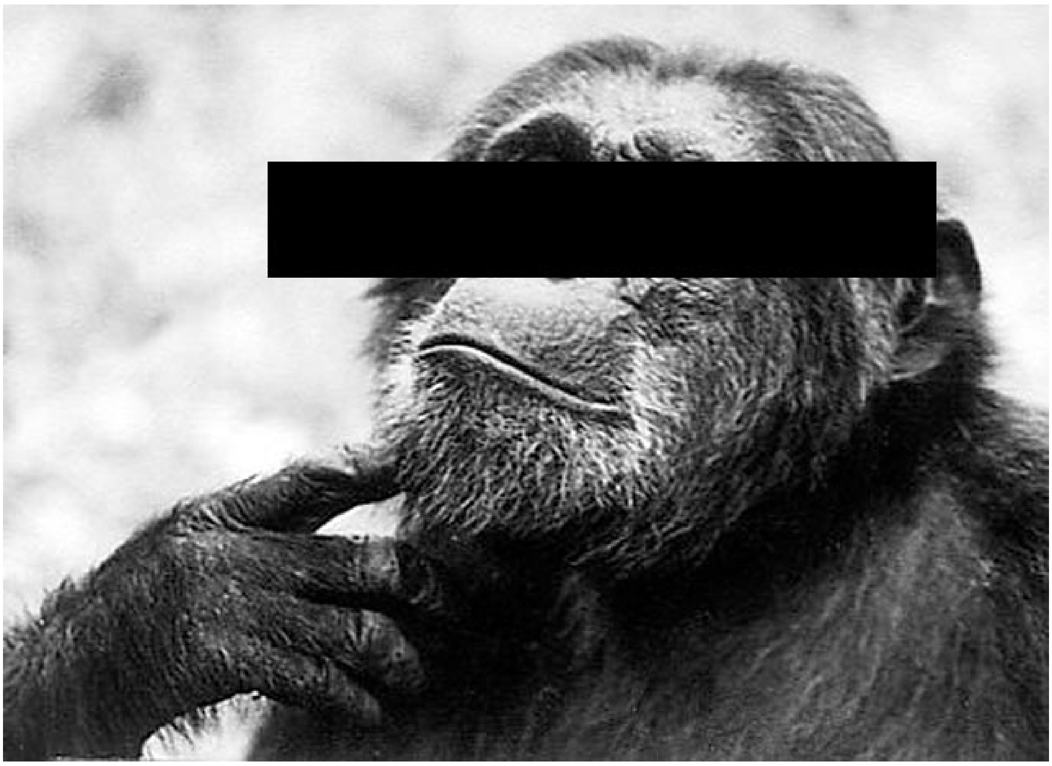
Description:

The phpinfo() function of the PHP language is readable, resulting in a listing of all the runtime information of the environment, thus disclosing potentially valuable information to attackers.

Technical description:

This is where the good stuff goes. We give a detailed technical description of the problem.

Illustrative picture of an evil hacker pondering dark deeds:



Impact:

This is where we explain how the sh*t is hitting the fan, exactly.

Recommendation:

Here is where we write some tips to solve the problem.

4.1.2 SID-002 — A terrible XSS issue

Vulnerability ID: SID-002

Vulnerability type: XSS

Threat level: High

Description:

A general description of the problem.

Technical description:

This is we go into great detail about the vulnerability.

Impact:

This is where we explain why this vulnerability is a problem.

Recommendation:

This is where we solve everything and the sun starts shining again.

4.1.3 SID-003 — A not quite so terrible XSS issue

Vulnerability ID: SID-003

Vulnerability type: XSS

Threat level: Low

Description:

A description of the problem.

Technical description:

Vulnerability described in detail.

Impact:

Impact on security.

Recommendation:

A ready solution.

4.2 Non-Findings

In this section we list some of the things that were tried but turned out to be dead ends.

4.2.1 NF-001 — FTP

The server was running FTPserver ABC, the most recent version of this particular piece of software. Anonymous login was turned off and no relevant vulnerabilities or exploits were found.

4.2.2 NF-002 — Mail Server

The server was running Mailserver XYZ, the most recent version of this particular piece of software. No relevant vulnerabilities or exploits were found.

4.2.3 NF-003 — SQL Code Injection

The following parameters are not vulnerable to SQL injection.

All parameters have been checked manually.

```
-file1.php
-file2.php
-file3.php
```

4.2.4 NF-004 — Heartbleed

System was not vulnerable to heartbleed.

4.2.5 NF-005 — Windows XP

The host is running Windows XP. As we all know, Windows XP is bulletproof.

5 Conclusion

In the course of this penetration test, we have demonstrated that the FishInABarrel Web Application faces a range of security issues which makes it vulnerable to a number of different attacks. Vulnerabilities found included: cross-site scripting (both stored and reflected), cross-site request forgery, information disclosure and lack of brute force protection.

Our conclusion is that there are a number of things that FishInABarrel BV has to fix before Sitting Duck should use their software. A number of the security issues highlighted in this report have fairly simple solutions, but these should nevertheless be fixed before use of the FishInABarrel Web App continues.

We finally want to emphasize that security is a process – and this penetration test is just a one-time snapshot. Security posture must be continuously evaluated and improved. Regular audits and ongoing improvements are essential in order to maintain control of your corporate information security. We hope that this pentest report (and the detailed explanations of our findings) will contribute meaningfully towards that end. Don't hesitate to let us know if you have any further questions or need further clarification of anything in this report.

Appendix 1 Testing team

Aristotle	Greek philosopher and scientist born in the Macedonian city of Stagira, Chalkidice, on the northern periphery of Classical Greece.
George Boole	English mathematician, philosopher and logician. Works in the fields of differential equations and algebraic logic, and is now best known as the author of The Laws of Thought.
William of Ockham	English Franciscan friar and scholastic philosopher and theologian. Considered to be one of the major figures of medieval thought. At the centre of some major intellectual and political controversies.
Ludwig Josef Johann Wittgenstein	Austrian-British philosopher who works primarily in logic, the philosophy of mathematics, the philosophy of mind, and the philosophy of language.
Melanie Rieback	Melanie Rieback is a former Asst. Prof. of Computer Science from the VU, who is also the co-founder/CEO of Radically Open Security.