Python for Application Security Testing

Zaki Akhmad
za@indocisc.co.id

February 28, 2014

We have deployed network firewall, but why our application still get exploited from SQL injection vulnerability?

(Application Security)

1 Introduction

Network security vs application security. Both are at the different level. Network firewall could never be able to prevent and detect attacks from the application layer. That's why application security testing is a must, as one of approach to ensure the security level of the application.

Python is really suitable language for penetration testers to perform security testing. From using simple Python function, writing simple script, writing exploit script, and using sophisticated tool written in python. Python is a **swiss army knife** to perform application security testing.

2 Application Security Testing

There are many kinds of application security vulnerabilities. OWASP ranked the top ten application security risks at OWASP Top Ten Project. Then CWE listed top 25 most dangerous software errors.

2.1 Identifying Password Hashes Algorithm

The first thing to do when cracking password hashes is find out what's the password hash algorithm. From using built-in Python standard function, `len()`, we may reveal the password hash algorithm.

Assuming you have the hashed password: `202cb962ac59075b964b07152d234b70`. It contains a-f characters and 0-9 numbers. How much is the length? Python here to help.

```
a = '202cb962ac59075b964b07152d234b70'
len(a)=32
```

When the hashed password has 32 length, a-f characters, 0-9 numbers, you may guess this password hash is using MD5 algorithm.

This just a simple example to identify algorithm from hashed password. This example shows how Python could really helpful for penetration testers. To build a more complete algorithm identifier from hashed password, we could start writing a more advanced Python script. root@blackploit.com had written Python script to identify algorithm for hashed password hash-identifier.

Another well known tool to perform brute force attack is hydra. Hydra can only take one file argument as the dictionary while performing brute force attack using known words (or called dictionary attack). The problem is when we have many dictionaries file. Writing a simple Python script might help us to solve this problem. Even Python supports threading so we could perform brute force attacks in thread mode.

```
#!/usr/bin/python

import threading
import os
from subprocess import call

def listdir_fullpath(d):
    return [os.path.join(d, f) for f in os.listdir(d)]

def main():
    wordlists = listdir_fullpath("/home/za/tools/wordlist")
    for wordlist in wordlists:
        print 'in progress using %s wordlist' % wordlist
        call(['hydra', '-l', 'admin', '-P', wordlist, '192.168.99.66',
          'mssql', '-v', '-t', '128'])

    print 'done'

if __name__ == '__main__':
    main()

# for improvement: use threading
```

2.2 Finding SQL Injection

OWASP Top 10, put **injection** as the number one risks. If an application has SQL injection vulnerability, an attacker could read the data in the database. Including confidential information and hashed passwords (or worse, the application keeps the passwords in plain text).

Finding (and even exploiting!) SQL injection is never been this easy. sqlmap is an automated tool for finding and exploiting SQL injection vulnerabilities written in

Python. What penetration testers should find manually, are the application entry points which accept inputs from user and this input is performing query to the database and the application didn't sanitize this user's input.

sqlmap is a really powerful tool. It could find SQL injection vulnerability using these technique: boolean-based blind, error-based, time-based, UNION query-based and stacked queries.

2.3 Proxying Network Traffic

One of the techniques to find out how an application works is by seeing the traffic. By standing between the server and the client, we could analyze the network traffic. This technique called proxying or MiTM (Man in The Middle). mitmproxy written in Python by Aldo Cortesi is really helpful to implement this technique. With `h, j, k, l` navigation like vim, mitmproxy might be the perfect choice for vim hackers. mitmproxy is able to intercept and modify the request/response.

2.4 Network Forensic

Python not only useful for application security testing, but also to perform network forensic from SQL injection attacks. By using Python scapy library, we could identify when/where/how the attacker performs SQL injection. With the help of Python scapy library, we analyze the network packet pcap files.

3 Final Words

With 25-minute talk duration, this is just an introduction talk about how Python is used for application security testing. I hope this talk could inspire the audience to start explore deeper.