Protection against Penetration Attacks using Metasploit

Himanshu Gupta
Senior Faculty Member
Amity Institute of Information Technology,
Amity University, Noida, India
E-mail: hgupta@amity.edu

Rohit Kumar
P.G. Student
Amity Institute of Information Technology,
Amity University, Noida, India
E-mail: rohit.work@live.in

***Abstract*— A script based attack framework is a type of web attack program written in scripting language. It has many attack scripts for various vulnerabilities of many systems. It supports quick development of latest attack scripts that are able to exploit zero day vulnerabilities. Such tools present a challenge for the defense side as traditional malware and spy-ware analysis can't catch up with this speed of new attack scripts. In this paper, we propose a system to counter the attacks by these frameworks, especially Metasploit. It involves proposal of a system which is able to block the metasploit attacks in specific cases otherwise alert the administrator. Previous research shows that many IDS and antivirus are ineffective against Metasploit. The proposed system uses a network monitoring application which is able to monitor the connection attempted to the host system and respond accordingly by using algorithm used in the system.**

Keywords: **Vulnerabilities, Metasploit, IDS**

I. INTRODUCTION

The fast evolution of attacking techniques has led to emergence of script based attack frameworks and it has become a big threat. A script based web attack framework is an attack launching platform written in languages like Ruby, C, C++ or even Python. Such framework is able to carry numerous attack scripts, many of which are able to exploit vulnerabilities of a specific application across many versions. With this high productivity, hackers can now easily develop early attack scripts to exploit existing or even newer vulnerabilities.

For launching the attack, the hacker runs the attack -script on the framework remotely. By investigating a vulnerable target, the script composes the attack payload, and delivers the payload to the target for exploiting the vulnerability. This web attack framework also ensures provision of numerous built-in modules that support quick development of fresh attack scripts. When the zero day vulnerability is discovered, a new script is rapidly developed and is distributed among hackers in various hacking related communities, where many other hackers and even the script kiddies are able to directly download this new script for launching attacks and exploiting this early vulnerability.

A very well known example of this type of script based web attack frameworks is Metasploit [1], one among the most popular Ruby language based web penetration framework. It has more or less 800 attack scripts and counting, targeting various vulnerable servers, services and applications running on different operating systems. It also has the provision of built-in modules for creating fresh attack scripts. Metasploit, the penetration testing framework was originally developed for penetration testing using proof of concept scripts (POCs). But eventually with many improvements, it has now become a fully-fledged web attack framework. As it is available open source, Metasploit is easily obtained and used by hackers for purposes other than penetration testing, mainly illegal hacking. E.g. The reported well known "Conficker" worm used the payload which was generated by the Metasploit framework, to spread [2]. The Metasploit attack script was quickly distributed in hacking communities soon after a zero day vulnerability was found in Java version 7 [3]. A 4 year study proves genuine malicious worldwide web traffic in relation to Metasploit. This study also shows that many of these Metasploit attack scripts are used by hackers soon after these scripts get distributed [4].

II. BACKGROUND & PROBLEM STATEMENT

This gives the background of how this attack script actually works. The Metasploit framework is intended for penetration testing and investigation purposes. The sole intention of using Metasploit and creation of Metasploit though is for good purposes i.e. for ensuring the security and integrity of the server. Unfortunately, Metasploit has been used for hacking and exploiting systems for monitory benefits and also to cause intentional damage to the server which is targeted. It is even used by everyday computer hobbyists to launch attacks out of sheer excitement. Usually, when the attack-script runs from any attack framework, this attack-script performs 4 important steps to launch the attack. (1) The script investigates the version number and the runtime environment of the target over the web. (2) Based on the result gathered and through the script's hard coded information base, this script then identifies the very specific

978-1-4673-7231-2/15/$31.00 ©2015 IEEE

vulnerability which is existing on the target. This information base usually contains the information (e.g. the return address) of the target that the script can attack. (3) The script then composes the payload used to attack which is customized for the target system. (4) Lastly, this script forwards the payload to the target system for exploiting the detected vulnerability.

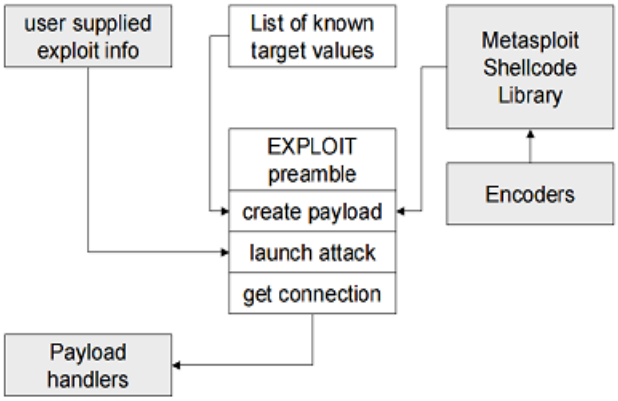

Figure 1: Working of Metasploit

```
1.  exploit_def()
2.  connection()
3.  exploit_preamble =  "\x00\x00\x01"
4.  version_find =  probing_ver()
5.  if (version equals 5)
6.  attack_payload =  prepare_payload5()
7.  else
8.  attack_payload =  prepare_payload4()
9.  end
10. exploit_preamble << payload_length
11. socket.put(exploit_preamble) //Reqd by the protocol
12. socket.get_once()
13. socket.put(attack_payload)  //sendingthe  attack
                payload
14. socket.get_once()
15. ... # triggering vulnerability
16. end
17. def prepare_payload5()
18. attack_payload =  shellcode
19. attack_payload << rand_alpha(payload.length)
20. attack_payload <<  "\x010" +  [-117].pack("X")
21. attack_payload <<  "\xe\xf"
22. attack_payload  <<  get_target_ret(5)  //  Target
                Version: 5
23. attack_payload <random_alpha(409)
24. return attack_payload
25. end
```

Figure 2: Code Snippet from a Metasploit Script

Figure 1 shows the working of Metasploit, the user supplied exploit information is used to launch attack by creating the connection and launching the attack. Metasploit generates the shellcode (using Metasploit Shellcode Library) by making use of the parameters that is specified. Depending on the strategy and type of vulnerability, various scripts can have various different behaviors of attack when performing these shown steps. E.g., a brute force attack attempt can keep creating and sending the attack payloads with the guessed up values until the target system is hacked or compromised, while the stealth attack may also clean up the traces left in the target system's log after delivering the payload. In these mentioned steps, creating and sending the attack payload are the major steps for launching any attack. An at- tack payload is usually bytes of string created with these elements: (a) special bytes that can exploit certain vulnerability; (b) a shellcode that hackers execute after that vulnerability is exploited. This shellcode is usually variable, especially when it is obfuscated; (c) random padding (for example NOP 0x90). [A NOP sled is simply the processor architecture 'no operation' instruction. In buffer overflows it is used to allocate a lot of space before the payload itself, to allow for a reliable return address in memory (instead of knowing the exact location of the start of the payload, just hit the NOP sled instead and it will return) or to align the registers]. That makes the attack payload robust; (d) format bytes required by network protocols. With help of the libraries of the scripting languages and built-in modules which are provided by the web attack framework, the attack script calls APIs of the related libraries or modules to help perform every step, especially creation of the attack payload.

Figure 2 shows a code snip-pet from a real Metasploit script exploiting some vulnerable application.. In this example, this script has two methods. **exploit_def()** is the main method that performs the steps to launch this attack. **prepare_payload5()** is one of the methods that compose payload. When this script runs on Metasploit framework, firstly it connects to the target system over the network (2^{nd} Line), and then discovers the target system's version (4^{th} Line). Here both connection and probing ver are methods of the builtin network protocol module... Based on the discovered version, it then calls the respective method to start creating the attack payload specific to that version of the target (Line 5-9).

When **prepare_payload5()** is called, the payload is 1st assigned by the shellcode module, which returns a shellcode which is configured.(18^{th} Line). The shellcode can be independently chosen as well. The shellcode module offers many shellcodes for different purpose. The payload is then appended (<<) with given contents (Line 19 to 23). Random alphabet padding is generated by random alpha to extend the payload to the required size of the network protocol. The concrete bytes represent the assembly code that goes to the shellcode. **pack("X")** converts the integer to byte as the

offset of one JMP. **Get_ target_ ret** is another attack framework API that queries the script's information base. After the payload is created, the script sends a exploit_preamble packet to the target system, which is followed by the payload packet to exploit the vulnerability (Line 11 to 13).

III. PROPOSED SYSTEM

The attack frameworks provide many built-in module encompassing various network protocols, Operating Systems, and providing many shellcode and NOP instructions, which enable the hackers to quickly develop early attack scripts to exploit various targets. Furthermore, advanced hackers can create even complex attack scripts, which can have many execution paths performing various attack behaviors and various payloads. Some of them may be triggered only in unique attack conditions.

In this paper, I propose *a* system which provides a fix to the systems for which metasploit has payloads, also the proposed system should have the capability of updating so that future upgrades are possible. Particularly, the system gives the *first aid* to vulnerabilities whose security patches are not updated or available while the attack scripts that exploit them are already distributed. The system will be evaluated using real-world attack scripts. The system initially will be prepared to counter few attacks using real-world Metasploit attack scripts from the website exploit-db.com**.**

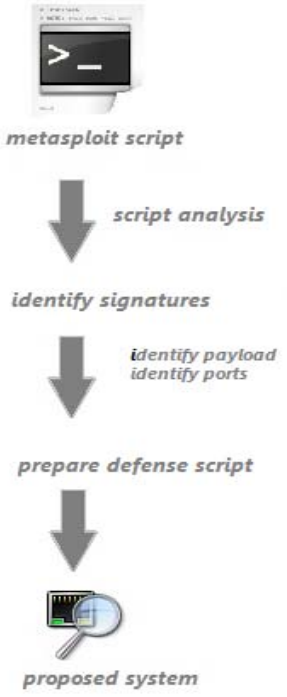

Figure 3: Flow of the Proposed System

The required metasploit script is downloaded and the script is analyzed. The signature is identified; the source and destination ports are identified from the payload. The other signatures include the expected code that could be executed on the victim machines. The combinations of these are used to prepare the defense script that is used in the proposed system.

The proposed system, thus, essentially is a network monitor that looks for the occurrence of the signatures that can be used to hack into the victim's system.

Following is a simple diagram that explains how the proposed system is prepared.

In summary, three contributions can be identified:

1. Security issues of script-based attacks, and propose a scalable system that counters attack scripts and defends.

2. A practical system for countering Metasploit attack framework that defends against newly distributed Metasploit attack scripts from Day One.

3. The effectiveness of system using recent Metasploit attack scripts in real-world attack environments.

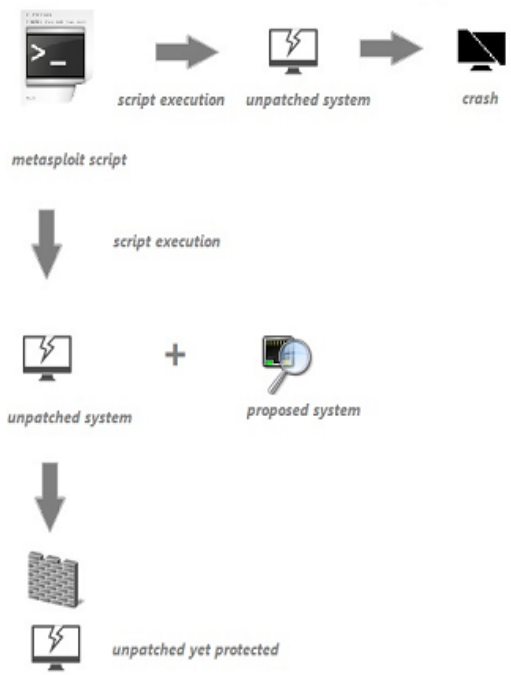

Figure 4: Implementation of the Proposed System

Once, the metasploit script is analysed for signatures and a defence script is prepared, it can be used in a system which can utilize the defence script to prepare a shield which helps to protect the system which even in unpatched state is not vulnerable to the metasploit exploit.

Figure 3 and 4 demonstrates how the use of the proposed system on the unpatched systems can make the unpatched system ready to thwart attacks by metasploit. This would be most effective when combating the zero day exploits as the zero day exploits when launched can create havoc on the systems which remain unpatched for quite some time because the Operating Systems, Software makers or the service providers do not launch the patch immediately. The patch comes after thorough and rigorous testing so that it does not introduce more bugs.

```
def exploit
  ...
  trigger = '/ldap://localhost/%3fA%3fA%3
      fCCCCCCCCCC%3fC%3f%90'
  # Sending payload
  send_request_raw({
    'uri' => '/' + rewrite_path() + trigger +
      shellcode(),
    'version' => '1.0',
    }, 2)
  ...
end
```

The code snippet from a Metasploit attack script apache_mod_rewrite_ldap.rb

Figure 5: Code Snippet from a Metasploit Script

Figure 5 shows one of the scripts from the metasploit frame-work and Figure 6 shows the counter script prepared after analysing the metasploit script. The counter script is prepared by studying the metasploit script and by using the "content" part which happens to be the payload against which detection will take place. The script is prepared in the "**Snort**" format. Snort is the leading Intrusion Detection System tool which is used in the Industry for attempts of Intrusion.

```
alert tcp any any -> any 80 (
  msg:"Metasploit apache_mod_rewrite_ldap,
  Target:[Apache 1.3/2.0/2.2],
  Behavior:[HTTP request with Vul-specific
      bytes]";
  content:"GET";
  content:"/ldap://localhost/%3fA%3fA%3
      fCCCCCCCCCC%3fC%3f%90";
  content:"|20|HTTP/1.0|0D 0A|Host|3A 20|";
  reference:cve,2006-3747;
  sid:5000539; rev:0;)
```

Figure 6:The counter script prepared by studying the exploit

Rest of the contents, apart from the "content" is static and can be common in some cases of preparation of other counter scripts. This part is subject to change if the exploit uses an entirely different script. This counter script is fed into the SNORT Intrusion Detection System to thwart the exploits used by metasploit.

IV. CONCLUSION

The preparation of the counter script to defend the attacks by analysing the attack script is beneficial in many cases, most importantly in the case of Zero Day Exploits. The vulnerabilities when found take a good amount of time to get a fix, This gives enough time to the hackers to create exploits for them and use metasploit to execute these exploits. The script used to counter the Metasploit exploits in this study can be fed to Snort IDS to get the defence mechanism working. This gives an efficient solution to the problem of delayed provision of patch by the software companies or service providers.

REFERENCES

[1] A brief description of Metasploit and its features accessed through web link as http://www.metasploit.com.

[2] An article "How Conficker worm uses metasploit payload to spread" published by SANS institute and accessed through http://www.sans.org/securityresources/malwarefaq/confickerworm.php

[3] Web contents provided in an article "Top Vulnerability Tools" through web link as http://sectools.org/tag/sploits

[4] An article titled as "How Conficker worm uses metasploit payload to spread" published as web contents.

[5] Tysen Leckie and Alec Yasinsac, "Metadata for Anomaly Based Security Protocol Attack Detection", IEEE Transactions on Knowledge and Data Engineering, Volume 16, Number 9, September 2004, pp. 1157-68