



---

# User Guide

Release 4.9

Last Updated: March 21, 2014

# TABLE OF CONTENTS

## About this Guide

Target Audience .....	1
Organization .....	1
Document Conventions .....	1
Support .....	2
Support for Metasploit Pro and Metasploit Express .....	2
Support for the Metasploit Framework and Metasploit Community .....	2

## Overview

Product Overview .....	3
Metasploit Community Components .....	3
Metasploit Framework .....	3
Services .....	4
Modules .....	4
User Interface .....	4
Service Listeners .....	4
Metasploit Implementation .....	5
Common Metasploit Terminology .....	5
Database .....	5
Discovery Scan .....	5
Exploit .....	6
Listener .....	6
Meterpreter .....	6
Module .....	6
Payload .....	6
Project .....	7
Shell .....	7
Shellcode .....	7
Target .....	7
Task .....	7
Vulnerability .....	7
Metasploit Workflow .....	8
Supported Browsers .....	8

Support for IPv6 Targets .....	9
--------------------------------	---

## Features Overview

Features Overview .....	10
The Dashboard .....	10
Navigational Tour .....	11
Administration Tour .....	12
Project Management.....	12
Global Settings .....	12
System Management.....	13
Features Tour .....	13
Host Scan .....	14
Exploitation .....	14

## Administration

Administration Overview .....	16
User Account Management .....	16
Creating a User Account.....	16
Editing a User Account .....	17
Changing a User Account Password .....	17
Resetting a User Account Password on Windows.....	17
Resetting a User Account Password on Linux.....	17
Deleting a User Account.....	18
System Management.....	18
Product News .....	18
Configuring Global Settings.....	18
Managing License Keys.....	20
Managing the System .....	21
Project Management .....	22
Configuring Project Settings .....	23

## Projects

Project Overview .....	24
Project Components .....	24
Working with a Project.....	25
Creating a Project.....	25
Editing a Project.....	25

Showing a List of All Projects .....	26
--------------------------------------	----

## Discovering Hosts

Discovery Overview .....	27
Discovery Scan.....	27
IPv6 Addresses for Target Hosts.....	28
Discovery Scan Options .....	28
Discovering Hosts.....	30
Discovering Virtual Hosts.....	30
Scanning the Network for H.323 Video Conferencing Systems .....	31
Defining Nmap Arguments.....	31
Nexpose Scan .....	32
Nexpose Scan Options.....	32
Configuring a Nexpose Console .....	34
Running a Nexpose Scan .....	35
Running a Nexpose Scan with a Custom Scan Template .....	35
Passing the Hash from Metasploit Community.....	36
Purging Scan Data.....	37
Imported Scan and Vulnerability Data .....	37
Supported Scan Data Formats .....	37
Importing Data .....	38
Host Data.....	39
Viewing Host Notes .....	39
Viewing Host Services .....	39
Viewing Host Evidence.....	39
Viewing Host Vulnerabilities .....	39
Vulnerability Management .....	40
Adding a Vulnerability .....	40
Exploiting a Known Vulnerability.....	40
Editing a Vulnerability .....	41
Deleting a Vulnerability .....	41
Host Management .....	41
Adding a Host .....	41
Deleting a Host .....	42
Host Badges .....	42

## Exploitation

Exploitation .....	43
Modules .....	43

Module Types .....	43
Module Search.....	44
Module Statistics.....	46
IPv6 Payloads.....	46
Exploits .....	47
Manual Exploits .....	47
Post-Exploitation.....	48
Post-Exploitation Modules .....	48
Post-Exploitation Macros .....	49
Listeners .....	50

# ABOUT THIS GUIDE

This guide provides information and instructions for Metasploit Community. The following sections describe the audience, organization, and conventions used within this guide.

## Target Audience

This guide is for IT and security professionals who use Metasploit Community as a penetration testing solution.

## Organization

This guide includes the following chapters:

- About this Guide
- Overview
- Metasploit Community Tour
- Administration
- Projects
- Discovering Hosts
- Gaining Access
- Index

## Document Conventions

The following table describes the conventions and formats that this guide uses:

Convention	Description
Command	Indicates buttons, UI controls, and fields. For example, " <b>Click Projects &gt; New Project.</b> "
Code	Indicates command line, code, or file directories. For example, "Enter the following: <code>chmod +x Desktop/metasploit-3.7.1-linux-x64-installer.</code> "
Title	Indicates the title of a document or chapter name. For example, "For more information, see the <i>Metasploit Pro Installation Guide.</i> "

Convention	Description
Note	Indicates there is additional information about the topic.

## Support

Rapid7 and the community strive to provide you with a variety of support options. For a list of support options that are available, view the support section for the Metasploit product that you are using.

### Support for Metasploit Pro and Metasploit Express

You can visit the Customer Center or e-mail the Rapid7 support team to obtain support for Metasploit Pro and Metasploit Express. To log in to the Customer Center, use the e-mail and password provided by Rapid7.

The following table describes the methods you can use to contact the Rapid7 support team.

Support Method	Contact Information
Customer Center	<a href="https://www.rapid7.com/for-customers/">https://www.rapid7.com/for-customers/</a>

### Support for the Metasploit Framework and Metasploit Community

An official support team is not available for the Metasploit Framework or for Metasploit Community. However, there are multiple support channels available for you to use, such as the IRC channel and mailing list.

You can visit the [Metasploit Community](#) to submit your question to the community or you can visit the [help page](#) to view the support options that are available.

# OVERVIEW

This chapter covers the following topics:

- [Product Overview 3](#)
- [Metasploit Community Components 3](#)
- [Service Listeners 4](#)
- [Common Metasploit Terminology 5](#)
- [Supported Browsers 8](#)
- [Support for IPv6 Targets 9](#)

## Product Overview

Metasploit Community is an all-inclusive exploitation tool that helps you divide the penetration testing workflow into smaller and more manageable tasks. With Metasploit Community, you can leverage the power of the Metasploit Framework and its exploit database through a web based user interface to perform security assessments and vulnerability verification.

Metasploit Community automates the process of discovery and exploitation and provides you with the necessary tools to perform the manual testing phase of a penetration test. You can use Metasploit Community to perform tasks like scan for open ports and services, exploit vulnerabilities, and collect evidence.

Ultimately, Metasploit Community helps you identify the weakest point to exploit a target and prove that a vulnerability or security issue exists.

## Metasploit Community Components

Metasploit Community consists of multiple components that work together to provide you with a complete penetration testing tool. The following components make up Metasploit Community.

## Metasploit Framework

An open source penetration testing and development platform that provides you with access to every module that Metasploit Community needs to perform tasks. The Metasploit Framework contains an exploit database that provides you with the latest exploit code for various applications, operating systems, and platforms. You can leverage the power of the Metasploit Framework to create additional custom security tools or write your own exploit code for new vulnerabilities. The Metasploit team regularly releases weekly updates that contain



new modules and bi-weekly updates that contain fixes and enhancements for known issues with Metasploit Community.

## Services

Metasploit Community uses PostgreSQL, Ruby on Rails, and Pro Service. PostgreSQL runs the database that Metasploit Community uses to store data from a project. Ruby on Rails runs the web Metasploit Community web interface. Pro service, or the Metasploit service bootstraps Rails, the Metasploit Framework, and the Metasploit RPC server.

## Modules

A prepackaged collection of code from the Metasploit Framework that performs a specific task, such as run a Nmap scan or an exploit. Every task in Metasploit Community uses modules. Some tasks, like a bruteforce attack or discovery scan, use multiple modules, whereas an exploit uses a single module.

## User Interface

The component that you use to interact with Metasploit Community. To launch the user interface, open a web browser and go to <https://localhost:3790>.

## Service Listeners

Metasploit Community uses the following service listeners to provide the user interface:

- 0.0.0:3790 – Apache SSL Service – Metasploit Community utilizes Apache as a front end web server for the Rails UI application. This is the primary service you will be interacting with when you use Metasploit Community.
- 127.0.0.1:3001 –Thin Rails Server (bound to localhost) – Metasploit Community utilizes Ruby on Rails, and Thin is used as the glue layer between Apache and Rails.
- 127.0.0.1:7337 – PostgreSQL Database (bound to localhost) – Metasploit Community uses PostgreSQL as the host for the Pro datastore. PostgreSQL was chosen for performance reasons.
- 127.0.0.1:50505 – Metasploit RPC Service (bound to localhost) – The RPC service is similar to that provided with the Metasploit Framework, with additional functionality added. This service makes it possible to communicate directly with the Metasploit Community system through RPC. The Rails UI utilizes RPC on this port to communicate with the Metasploit Community engine.

# Metasploit Implementation

Rapid7 distributes Metasploit Community as an executable file for Linux and Windows operating systems. Download and run the executable to install Metasploit Community on your local machine or on a remote host, like a web server. Regardless of where you install Metasploit Community, you always access the user interface through a web browser. Metasploit Community uses a secure connection to connect to the server or machine that runs it.

If you install Metasploit Community on a web server, users can use a web browser to access the user interface from any location. Users will need the address and port for the server that Metasploit Community uses. By default, the Metasploit service uses port 3790. You can change the port that Metasploit uses during the installation process. So, for example, if Metasploit Community runs on 192.168.184.142 and port 3790, users can use <https://192.168.184.142:3790> to launch the user interface.

If Metasploit Community runs on your local machine, you can use localhost and port 3790 to access Metasploit Community. For example, type <https://localhost:3790> in the browser URL box to load the user interface.

If you have not installed Metasploit Community, you can download the installer from the [Rapid7 website](#). You will need a license key to activate the product. If you do not have a license key, please contact the [Rapid7 support team](#).

## Common Metasploit Terminology

The following sections describe the most commonly used terms in Metasploit.

### Database

The database stores target host data, system logs, collected evidence, and report data.

### Discovery Scan

A discovery scan is the Metasploit internal scanner that combines Nmap and several Metasploit modules to scan and fingerprint targets. If you do not have Nexpose or scan data to import into Metasploit Community, you can run a discovery scan to gather information about the target. There are several scan speeds that you can configure for a discovery scan. The scan speed determines the method that the discovery scan uses to perform the discovery process.

## Exploit

An exploit is a program that takes advantage of a specific vulnerability and provides an attacker with access to the target system. An exploit typically carries a payload and delivers the payload to the target system. For example, one of the most common exploits is windows/smb/s08-067\_netapi, which targets a Windows Server Service vulnerability that could allow remote code execution. You can run this exploit against a machine that has the ms0-067 vulnerability to remotely take control of the system.

## Listener

A listener waits for an incoming connection from either the exploited target or the attacking machine and manages the connection when it receives it.

## Meterpreter

Meterpreter is an advanced multi-function payload that provides you an interactive shell. From the Meterpreter shell, you can do things like download a file, obtain the password hashes for user accounts, and pivot into other networks. Meterpreter runs on memory, so it is undetectable by most intrusion detection systems.

## Module

A module is a standalone piece of code, or software, that extends functionality of the Metasploit Framework. Modules automate the functionality that the Metasploit Framework provides and enables you to perform tasks with Metasploit Community.

A module can be an exploit, auxiliary, payload, no operation payload (NOP), or post-exploitation module. The module type determines its purpose. For example, any module that opens a shell on a target is an exploit module.

## Payload

A payload is the actual code that executes on the target system after an exploit successfully executes.

A payload can be a reverse shell payload or a bind shell payload. The major difference between these payloads is the direction of the connection after the exploit occurs.

## Bind Shell Payload

A bind shell attaches a listener on the exploited system and waits for the attacking machine to connect to the listener.

## Reverse Shell Payload

A reverse shell connects back to the attacking machine as a command prompt.

## Project

A project is a container for the targets, tasks, reports, and data that are part of a penetration test. A project represents the workspace that you use to create a penetration test and configure tasks. Every penetration test runs from within a project.

## Shell

A shell is a console-like interface that provides you with access to a remote target.

## Shellcode

Shellcode is the set of instructions that an exploit uses as the payload.

## Target

A target is the system that you want to exploit. The term target can represent a single host, multiple hosts, a network range, or an entire network.

## Task

A task represents an action that Metasploit Community can perform, such as a scan, bruteforce attack, exploit, or report generation.

## Vulnerability

A vulnerability is a security flaw or weakness in an application or system that enables an attacker to compromise the target system. A compromised system can result in privilege escalation, denial-of-service, unauthorized data access, stolen passwords, and buffer overflows.

# Metasploit Workflow

The overall process of penetration testing can be broken down into a series of steps or phases. Depending on the methodology that you follow, there can be anywhere between four and seven phases in a penetration test. The names of the phases can vary, but they generally include reconnaissance, scanning, exploitation, post-exploitation, maintaining access, reporting, and cleaning up.

The Metasploit Community workflow follows the general steps of a penetration test. Besides reconnaissance, you can perform the other penetration testing steps from Metasploit Community.

- 1.) **Information Gathering**- Use the Discovery scan, Nexpose scan, or import tool to supply Metasploit Community with a list of targets and the running services and open ports associated with those targets.
- 2.) **Exploitation** - Use smart exploits or manual exploits to launch attacks against target machines. Additionally, you can run bruteforce attacks to escalate account privileges and to gain access to exploited machines.
- 3.) **Post-Exploitation** - Use post-exploitation modules or interactive sessions to interact gather more information from compromised targets. Metasploit Community provides you with several tools that you can use to interact with open sessions on an exploited machine. For example, you can view shared file systems on the compromised target to identify information about internal applications. You can leverage this information to obtain even more information about the
- 4.) **Reporting** - Use the reporting engine to create a report that details the findings of the penetration test. Metasploit Community provides several types that let you to determine the type of information that the report includes.
- 5.) **Cleaning Up** - Use the Clean Up tool to close any open sessions on an exploited target and to remove any evidence of any data used during the penetration test. This step restores the original settings on the target system.

## Supported Browsers

The following browsers support Metasploit Community:

- Chrome 8+
- Firefox 4+
- Internet Explorer 9+

**Note:** Windows XP does not support Internet Explorer 9. Therefore, Windows XP users should use Chrome or Firefox to access Metasploit Community.

# Support for IPv6 Targets

IPv6 is the latest version of the Internet Protocol designed by the Internet Engineering Task Force to replace the current version of IPv4. The implementation of IPv6 predominantly impacts addressing, routing, security, and services.

An IPv6 address consists of 128 bits and contains eight groups of hexadecimal numbers separated by colons. For example, you can define a full IPv6 address as `fe80:0:0:0:200:f8ff:fe21:67cf`. To save space, you can use a double colon (::) to replace groups of leading zeros. In this example, you can enter `fe80:0:0:0:200:f8ff:fe21:67cf` as `fe80::200:f8ff:fe21:67cf`.

For more information on IPv6, visit <http://ipv6.com/articles/general/ipv6-the-next-generation-internet.htm>.

In Metasploit Community, you can define IPv6 addresses for target hosts. For example, when you perform a discovery scan, scan a web application, execute a bruteforce attack, or run a module, you can define an IPv6 address for the target hosts. For modules, Metasploit Community provides several payloads that provide IPv6 support for Windows x86, Linux x86, BSD x86, PHP, and cmd.

**Note:** you can import IPv6 addresses from a text file or you can manually add them to your project. If you import IPv6 addresses from a text file, you must separate each address with a new line. Metasploit Community does not support IPv6 for link local broadcast discovery or pivoting.

# FEATURES OVERVIEW

This chapter covers the following topics:

- [Features Overview](#) 10
- [The Dashboard](#) 10
- [Navigational Tour](#) 11
- [Administration Tour](#) 12
- [Features Tour](#) 13

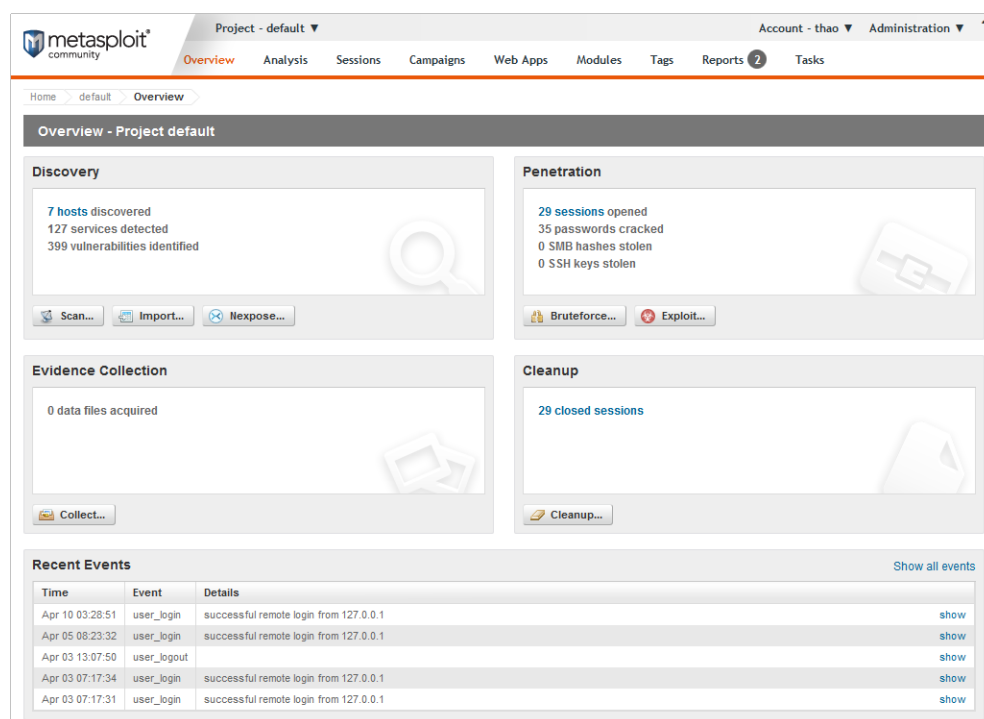
## Features Overview

Metasploit Community provides a comprehensive and intuitive workspace that you can use to perform administrative tasks and to configure penetration tests.

## The Dashboard

The Dashboard provides access to quick tasks and displays a project overview. The project overview shows a numerical breakdown of discovered hosts, opened and closed sessions, and collected evidence. Use the Dashboard for a high level overview of the project.

The following figure shows the Dashboard:



## Navigation Tour

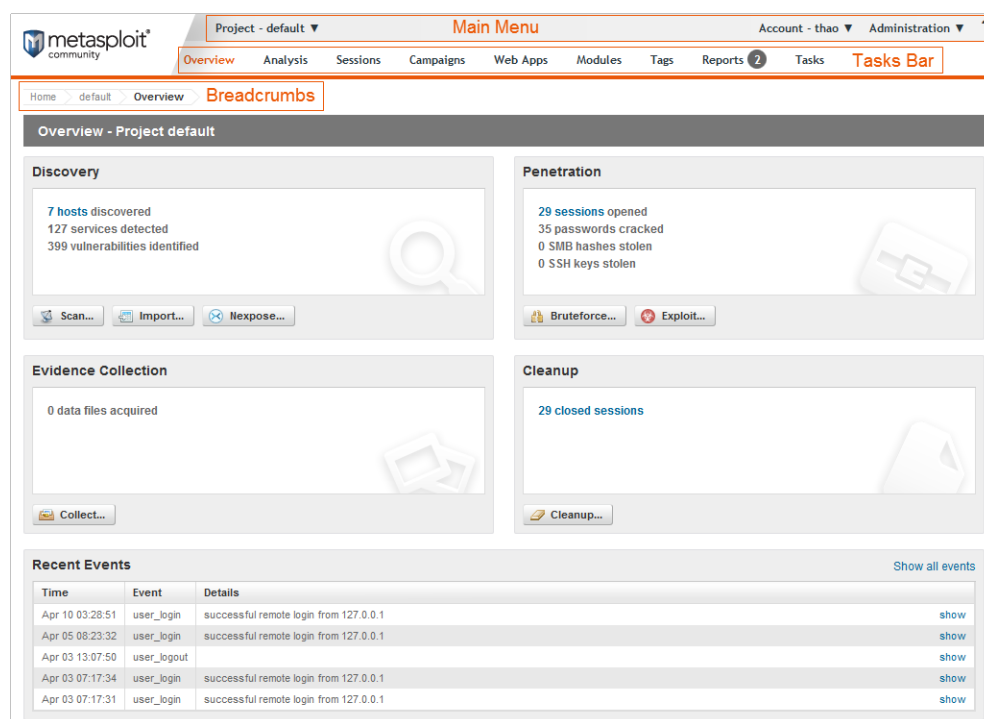
You can use the navigational features to navigate between the different areas of Metasploit Community.

The following list describes the navigational options:

- 1.) Main menu - Use the main menu to manage project settings, configure user account information, and perform administration tasks.
- 2.) Task bar - Use the task bar to navigate between task pages.
- 3.) Navigational breadcrumbs - Use the navigational breadcrumbs to switch between task pages.



The following figure shows the navigational features:



## Administration Tour

Administrators can perform administrative tasks, like manage projects, accounts, global settings, and software updates, from the main menu.

## Project Management

A Metasploit Community project contains the penetration test that you want to run. A project defines the target systems, network boundaries, modules, and web campaigns that you want to include in the penetration test. Additionally, within a project, you can use discovery scan to identify target systems and bruteforce to gain access to systems.

## Global Settings

Global settings define settings that all projects use. You can access global settings from the Administration menu.

From the global settings, you can set the payload type for the modules and enable access to the diagnostic console through a web browser.

The following figure shows the global settings area:

Value	Category	Setting	Description
<input type="checkbox"/>	Payloads	payload_prefer_https	Allow HTTPS-based payloads whenever possible (less reliable, but more stealthy)
<input type="checkbox"/>	Payloads	payload_prefer_http	Allow HTTP-based payloads whenever possible (mostly reliable, traverses proxies)
<input type="checkbox"/>	Debugging	allow_console_access	Allow access to the unsupported diagnostic console through the web browser (less secure)
<input checked="" type="checkbox"/>	Updates	automatically_check_updates	Automatically check for available updates
<input type="checkbox"/>	Updates	use_http_proxy	Connect to the Internet via http proxy to check for software updates

**SMTP Settings**

Address:

Port:

Use SSL? ☐

Domain:

Username:

Password:

Authentication:

Update Settings

## System Management

As an administrator, you can update the license key and perform software updates. You can access the system management tools from the Administration menu.

The following figure shows the license key management area:

**Activate Your Metasploit License**

**1. Get Trial Key for Metasploit Pro**

Get a free, fully-featured trial version of Metasploit Pro. If you already have a community, trial or full license product key, you can skip this step.

[GET METASPLOIT PRO TRIAL](#)

**2. Enter Product Key You've Received by Email**

Paste in the product key that was sent to the email address you registered with and click the ACTIVATE LICENSE button.

☐ Use an HTTP Proxy to reach the internet?

[ACTIVATE LICENSE](#)

[Offline Activation](#)

**Revert to Previous License**

A previous license has been found. If you would like to switch to the old license, simply click the Revert License button below.

[Revert License](#)

## Features Tour

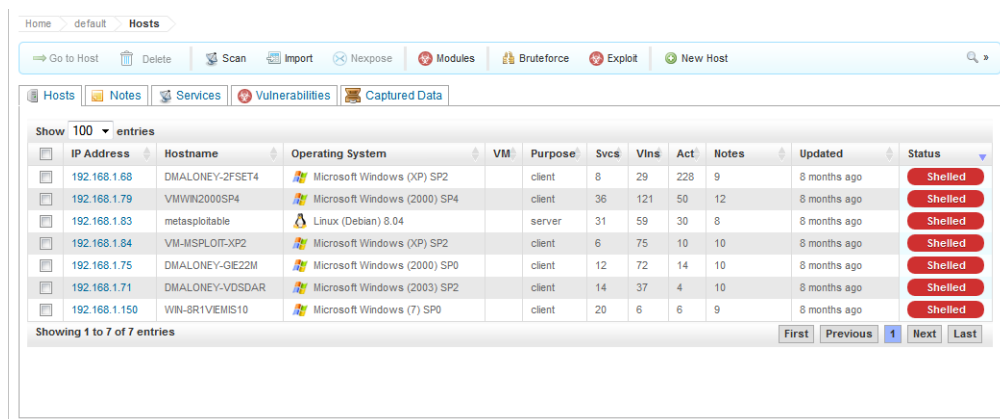
Metasploit Community provides a comprehensive penetration testing system that you can use to scan for target hosts, open and control sessions, exploit vulnerabilities, and generate reports.

## Host Scan

A host scan identifies vulnerable systems within the target network range that you define. When you perform a scan, Metasploit Community provides information about the services, vulnerabilities, and captured evidence for hosts that the scan discovers. Additionally, you can add vulnerabilities, notes, tags, and tokens to identified hosts.

You can scan target systems and view discovered host information from the Analysis tab.

The following figure shows the features that you can access from the Analysis tab:



The screenshot shows the Metasploit Community interface with the 'Hosts' tab selected. The interface includes a top navigation bar with links like 'Home', 'default', and 'Hosts'. Below this is a toolbar with buttons for 'Go to Host', 'Delete', 'Scan', 'Import', 'Nexpose', 'Modules', 'Bruteforce', 'Exploit', and 'New Host'. The main content area displays a table of hosts with the following columns: IP Address, Hostname, Operating System, VM, Purpose, Svcs, Vins, Act, Notes, Updated, and Status. The table contains 7 entries, all of which are marked as 'Shelled'.

IP Address	Hostname	Operating System	VM	Purpose	Svcs	Vins	Act	Notes	Updated	Status
192.168.1.68	DIMALONEY-2FSET4	Microsoft Windows (XP) SP2		client	8	29	228	9	8 months ago	Shelled
192.168.1.79	VMWIN2000SP4	Microsoft Windows (2000) SP4		client	36	121	50	12	8 months ago	Shelled
192.168.1.83	metasploitable	Linux (Debian) 8.04		server	31	59	30	8	8 months ago	Shelled
192.168.1.84	VM-MSPLOIT-XP2	Microsoft Windows (XP) SP2		client	6	75	10	10	8 months ago	Shelled
192.168.1.75	DIMALONEY-GIE22M	Microsoft Windows (2000) SP0		client	12	72	14	10	8 months ago	Shelled
192.168.1.71	DIMALONEY-VDSDAR	Microsoft Windows (2003) SP2		client	14	37	4	10	8 months ago	Shelled
192.168.1.150	WIN-SR1VEMIS10	Microsoft Windows (7) SP0		client	20	6	6	9	8 months ago	Shelled

## Exploitation

Modules expose and exploit vulnerabilities and security flaws in target systems. Metasploit Community offers access to a comprehensive library of exploit modules, auxiliary modules, and post-exploitation modules. You can run automated exploits or manual exploits.

Automated exploitation uses the minimum reliability option to determine the set of exploits to run against the target systems. You cannot select the modules or define evasion options that Metasploit Community uses.

Manual exploitation provides granular control over the exploits that you run against the target systems. You run one exploit at a time, and you can choose the modules and evasion options that you want to use.















The following figure shows the modules area:

Home > default > Modules

Search Modules

Module Statistics [show](#) Search Keywords [show](#)

Found 10 matching modules

Module Type	OS	Module	Disclosure Date	Module Ranking	CVE	BID	OSVDB	EDB
Client Exploit	 	Honeywell HSC Remote Deployer ActiveX Remote Code Execution	February 21, 2013	★★★★★	2013-0108	58134	90583	
Server Exploit		Kordil EDMS v2.2.60rc3 Unauthenticated Arbitrary File Upload Vulnerability	February 21, 2013	★★★★★				
Server Exploit		OpenEMR PHP File Upload Vulnerability	February 12, 2013	★★★★★		37314	90222	
Client Exploit	 	MS13-009 Microsoft Internet Explorer SLayoutRun Use-After-Free	February 12, 2013	★★	2013-0025			
Server Exploit		Glossword v1.8.8 - 1.8.12 Arbitrary File Upload Vulnerability	February 4, 2013	★★★★★				24456
Auxiliary		OpenSSL TLS 1.1 and 1.2 AES-NI DoS	February 4, 2013	★★	2012-2686			
Auxiliary		D-Link DIR-600 / DIR-300 Unauthenticated Remote Command Execution	February 3, 2013	★★			89861	24453
Server Exploit	 	SCADA 3S CoDeSys Gateway Server Directory Traversal	February 1, 2013	★★★★★	2012-4705			
Server Exploit	 	Firebird Relational Database CNCT Group Number Buffer Overflow	January 30, 2013	★★	2013-2492			
Client Exploit	 	Novell GroupWise Client gwcls1.dll ActiveX Remote Code Execution	January 29, 2013	★★	2012-0439	57658	89700	

# ADMINISTRATION

This chapter covers the following topics:

- [Administration Overview 16](#)
- [User Account Management 16](#)
- [System Management 18](#)
- [Project Management 22](#)

## Administration Overview

As an administrator, you manage user accounts, perform system maintenance, and manage projects.

## User Account Management

A user account can be a basic user account or an administrator account. A basic user account cannot add, modify, or remove user accounts or configure global settings and network boundaries for the system. An administrator account has unrestricted access to Metasploit Community features.

## Creating a User Account

- 1.) Click **Administrator > User Administration** from the main menu.
- 2.) Click **New User**.
- 3.) Enter a user name.
- 4.) Enter the first and last name in the **Full Name** field.
- 5.) Enter a password. Use mixed case, punctuation, numbers, and at least six characters to create a strong password. You must create a strong password because Metasploit Community runs as root.
- 6.) Reenter the password in the **Password Confirmation** field.
- 7.) Select a role for the user. If you do not choose “Administrator,” the default user role is basic.
- 8.) Save the changes to the user account.

## Editing a User Account

- 1.) Click **Account > User Settings** from the main menu.
- 2.) Edit the **Full Name**, **Email**, **Organization**, or **Time Zone** fields for the user account.
- 3.) Save the changes.

## Changing a User Account Password

- 1.) Click **Administration > User Administration** from the main menu.
- 2.) Click the user account that you want to modify.
- 3.) Enter a new password for the user account. Use mixed case, punctuation, numbers, and at least six characters to create a strong password. You must create a strong password because Metasploit Community runs as root.
- 4.) Reenter the new password.
- 5.) Apply the changes to the password.

## Resetting a User Account Password on Windows

If you forget the Metasploit Community user account password, you can reset the password. The system resets the password to a random value, which you can change after you log back in to Metasploit Community.

To reset the password, you must be logged in to Windows as an administrator.

- 1.) From the Start menu, choose **All Programs > Metasploit > Password Reset**. The Password Reset window appears. Wait for the environment to load and prompt you to continue.
- 2.) Type **yes** to continue. The system resets the password to a random value.
- 3.) Copy the password and use the password the next time you log in to Metasploit Community.
- 4.) Exit the **Password Reset** window.

## Resetting a User Account Password on Linux

- 1.) In the console, execute the following command: `sudo /path/to/metasploit/diagnostic_shell`.
- 2.) Next, execute `/path/to/metasploit/apps/pro/ui/script/resetpw`.
- 3.) Copy the password and use the password the next time you log into Metasploit Community. You can change the password after you log in to Metasploit Community.
- 4.) Exit the console.

## Deleting a User Account

Users with administrator privileges can delete user accounts.

- 1.) Click **Administration > User Administration** from the main menu.
- 2.) Click the user account that you want to delete.
- 3.) Click **Delete**.
- 4.) Click **OK** to confirm that you want to delete the account.

## System Management

The administrator can configure the global settings for projects, create API keys, manage license keys, and update the system.

## Product News

When you access the Projects page, the Product News displays and lists the latest blog posts from the Metasploit Community site. You can click on any of the blog links to access the blog entry.

The figure below shows the Product News:

The screenshot displays the Metasploit Community interface. At the top, there's a navigation bar with 'Home' and 'Projects'. Below it, a section titled 'Quick Start Wizards' asks 'What do you want to do?' and offers three options: 'Quick PenTest', 'Phishing Campaign', and 'Web App Test'. The main content area is divided into two parts. On the left, 'Project Listing' shows a table of projects with columns for Name, Hosts, Active Sessions, Tasks, Owner, Members, Updated, and Description. The table lists three projects: 'Test', 'default', and 'Phishing'. On the right, the 'Product News' panel features two articles: 'Compromising Embedded Linux Routers with Metasploit' and 'Weekly Update: Minecraft RAT Attacks, PHP Shell Games, and MongoDB'.

Name	Hosts	Active Sessions	Tasks	Owner	Members	Updated	Description
Test	0	0	0	system	0	8 days ago	
default	7	0	0	system	0	8 days ago	
Phishing	5	0	0	system	0	10 days ago	

## Configuring Global Settings

Metasploit Community applies global settings to all projects. Use global settings to set HTTP and HTTPS payloads and to access diagnostic data through a Web browser. Additionally, you

can configure an HTTP proxy so that the system can alert you when updates are available for Metasploit Community.

The following image shows the Global Settings:

**Global Settings**  
This section defines options that are applicable across all projects.

Value	Category	Setting	Description
<input type="checkbox"/>	Payloads	payload_prefer_https	Allow HTTPS-based payloads whenever possible (less reliable, but more stealthy)
<input type="checkbox"/>	Payloads	payload_prefer_http	Allow HTTP-based payloads whenever possible (mostly reliable, traverses proxies)
<input type="checkbox"/>	Debugging	allow_console_access	Allow access to the unsupported diagnostic console through the web browser (less secure)
<input checked="" type="checkbox"/>	Updates	automatically_check_updates	Automatically check for available updates
<input type="checkbox"/>	Updates	use_http_proxy	Connect to the Internet via http proxy to check for software updates

**SMTP Settings**

Address:

Port:

Use SSL? ☐

Domain:

Username:

Password:

Authentication:

## Setting HTTP Payloads

- 1.) Select **Administration > Global Settings** from the main menu.
- 2.) Select or deselect **payload\_prefer\_http** from the Global Settings.
- 3.) Update the settings.

## Setting HTTPS Payloads

- 1.) Choose **Administration > Global Settings** from the main menu.
- 2.) Choose **payload\_prefer\_https** from the Global Settings.
- 3.) Update the settings.

## Accessing Diagnostic Data

- 1.) Choose **Administration > Global Settings** from the main menu.
- 2.) Choose **allow\_console\_access** from the Global Settings.
- 3.) Update the settings.

## Setting Automatic Checks for Updates

- 1.) Choose **Administration > Global Settings** from the main menu.
- 2.) Choose **automatically\_check\_updates** from the Global Settings.



- 3.) Update the settings.

## Setting HTTP Proxy Settings for Update Notifications

- 1.) Choose **Administration > Global Settings** from the main menu.
- 2.) Choose **use\_http\_proxy** from the Global Settings.
- 3.) Enter the settings for the HTTP proxy server. You must define the IP address, port, user name, and password for the proxy server.
- 4.) Update the settings. The settings that you define automatically fill the HTTP proxy server settings when you perform an update.

## Managing License Keys

License keys define the product edition and the registered owner of Metasploit Community. Metasploit Community uses the license key to identify the number of days that remain on the license.

### Updating License Keys

- 1.) Select **Administration > Software Licenses** from the main menu.
- 2.) Enter the license key in the **Product Key** field.
- 3.) Activate the license.

### Performing an Offline Activation

If you do not have network access, use the offline activation file to activate Metasploit Community. To obtain an offline activation file, contact customer support.

- 1.) Select **Administration > Software Licenses** from the main menu. The **Offline Activation** window appears.
- 2.) Browse to the location of the activation file.
- 3.) Select the activation file.
- 4.) Click **Activate Product** to complete the activation.

### Reverting to a Previous License Key

You can revert to a previous license key if Metasploit Community detects that a previous license key exists on the system. Use license key reversion to switch between different versions of Metasploit products. For example, if you install a trial version of a Metasploit product, use license key reversion to switch back to the full version.

- 1.) Select **Administration > Software Licenses** from the main menu.
- 2.) Click **Change Key**.
- 3.) Click **Revert License**. The **License Details** window appears if Metasploit Community reverts to the previous version.

## Managing the System

Administrators can update, maintain, and uninstall Metasploit Community.

### Updating the System

If you are an administrator, you must regularly check for available updates to Metasploit Community. When you check for updates, Metasploit Community alerts you when a newer version is available for you to install. If a newer version of Metasploit Community is not available, the system notifies you that you have the latest version.

- 1.) Click **Administration > Software Updates** from the main menu. The **Software Updates** window appears.
- 2.) Select **Use an HTTP Proxy to reach the internet** if you want to use an HTTP proxy server to check for updates. If you select this option, the proxy settings appear. Configure the settings for the HTTP proxy that you want to use.
- 3.) Check for updates.

After the update completes, Metasploit Community prompts you to restart the back end services. If you restart the services, Metasploit Community terminates active sessions and requires up to five minutes to restart.

### Maintaining the System

Metasploit Community uses log files to store system information.

The log file sizes can become large over time because there is no automatic rotation for log files. To reduce the amount of disk space the log files consume, regularly review and clear log files.

The following table describes the log files that are available:

Log File	Log File Location
Database log	\$INSTALL_ROOT/postgres/postgresql.log
Web server error log	\$INSTALL_ROOT/apache2/logs/error_log
Web server access log	\$INSTALL_ROOT/apache2/logs/access_log
Rails log	\$INSTALL_ROOT/apps/pro/ui/log/production.log

Log File	Log File Location
Rails server log	\$INSTALL_ROOT/apps/pro/ui/log/thin.log
Metasploit Framework log	\$INSTALL_ROOT/apps/pro/engine/config/logs/framework.log
Metasploit RPC log	\$INSTALL_ROOT/apps/pro/engine/prosvc.log
Task log	\$INSTALL_ROOT/apps/pro/engine/tasks
License log	\$INSTALL_ROOT/apps/pro/engine/license.log

## Uninstalling Metasploit Community on Linux

When you uninstall Metasploit Community, you remove the components and modules from the system and the data stored within the penetration tests.

- 1.) Navigate to the root installation directory and enter `./ctlscript.sh.stop` to stop all Metasploit Community services.
- 2.) Enter `./uninstall`.
- 3.) Click **Yes** to confirm that you want to uninstall Metasploit Community components and modules.
- 4.) Click **Yes** to confirm that you want to delete the data saved in the penetration tests. If you click **No**, the `$INSTALLER_ROOT/apps` directory remains intact, and you can access Metasploit Community data stored in this directory.

## Uninstalling Metasploit Community on Windows

- 1.) Navigate to **Start > All Programs > Metasploit**.
- 2.) Click **Uninstall Metasploit**.
- 3.) Click **Yes** to confirm that you want to delete all saved data from the penetration tests.
- 4.) Click **OK** when the uninstall completes.

## Project Management

A project is a penetration test. Use projects to define the target systems that you want to test and to configure tasks for the penetration test.

You want to create multiple projects to test different networks or different components of a single network. For example, if you want to perform an internal and external penetration test, create separate projects for each penetration test.

# Configuring Project Settings

Project settings define the project name, description, network range, and user account access.

## Defining the Network Range

When you create a project, you can define optional network boundaries that Metasploit Community enforces on the penetration test. Use network boundaries to maintain the scope of a project. If you enforce network boundaries, you ensure that you do not target devices outside the range of targeted devices. Additionally, the network range defines the default range that all tasks use.

Administrators and project owners can define the network range for a project.

- 1.) Open the project.
- 2.) Click **Project > Project Settings** from the main menu.
- 3.) Define the network address range.

**Note:** Metasploit Community supports IPv4 and IPv6 addresses. You can use standard IPv6 addressing to define individual IPv6 addresses. For example, use `fe80::202:b3ff:fe1e:8329` for single addresses and `2001:db8::/32` for CIDR notations. For link local addresses, you must append the interface ID to the address. For example, enter `fe80::1%eth0` for a link local address.

- 4.) Update the project.

## Restricting the Network Range

Restrict the network range to enforce network boundaries on a project. When you restrict the network range for a project, a user cannot run the penetration test unless the network range for the project falls within network range that you define.

Before you restrict the network range, you must define the network range.

- 1.) Open the project.
- 2.) Click **Project > Project Settings**.
- 3.) Select **Restrict to Network Range**.
- 4.) Update the project.

# PROJECTS

This chapter covers the following topics:

- [Project Overview 24](#)
- [Working with a Project 25](#)

## Project Overview

A project contains the workspace that you use to perform the different steps for a penetration test and store the data that you collect from the target. Projects are useful tools that you can use to set up tests and organize the data that you gather from target machines. You can create as many projects as you need, and you can switch between projects while tasks are in progress.

From within a project, you define the targets that you want to test and configure the tasks that you want to run against those targets. You can scan targets for active services and hosts, attempt to exploit vulnerabilities, collect data from exploited machines, and generate reports that detail your findings.

You can create projects to separate an engagement into logical groupings. Oftentimes, you may have different requirements for the various departments, or subnets, within an organization. Therefore, it may be more efficient for you to have different projects to represent those requirements.

For example, you may want to create a project for the human resources department and another project for the IT department. Your requirements for these departments may vary greatly, so it would be logical for you to separate the targets into different projects. At the end of the engagement, you can generate separate reports for each department to perform a comparative analysis and present your findings to your organization or client.

## Project Components

Use the following components to create a project:

- **Name** - Provides a unique identifier for the project.
- **Description** - Describes the purpose and scope of the project.
- **Network range** - Defines the default network range for the project. When you create a project, Metasploit Community automatically populates the default target range with the network range that you define for the project. Metasploit Community does not force the project to use the network range unless you enable the **network range restriction** option.

**Network range restriction** - An option that restricts a project to a specific network range. Enable this option if you want to ensure that the test does not target devices outside the scope of the engagement. If you enable this option, Metasploit Community will not run tasks against a target whose address does not fall within the network range.

## Working with a Project

A project consists of a name, description, and network boundaries. Network boundaries define the scope of the project and ensure that you do not target devices outside of the range of intended devices. You use network boundaries to enforce a default network range for all tasks. You can restrict a project to a single network range or multiple network ranges.

Within a project, you can scan for hosts, open and take control of sessions, and generate reports.

You create a project when you want to test multiple networks or different components of a single network. For example, if you want to perform an internal and external penetration test, you create a separate project for each test. Each project generates a separate report for each test scenario that you can use to compare test results.

## Creating a Project

- 1.) Select **Project > Create New Project** from the main menu.
- 2.) Enter the project name.
- 3.) Enter a description for the project.
- 4.) Define an optional network range. To enter multiple network ranges, use a comma to separate each range.
- 5.) Select **Restrict to network range** if you want to enforce network boundaries on the project.
- 6.) Create the project.

## Editing a Project

- 1.) Select **Project > Project Settings** from the main menu.
- 2.) Edit the project name, description, network range, or network range restriction.
- 3.) Update the project.

## Showing a List of All Projects

To view a list of all projects, select **Project > Show All Projects** from the main menu.



# DISCOVERING HOSTS

This chapter covers the following topics:

- [Discovery Overview 27](#)
- [Discovery Scan 27](#)
- [Nexpose Scan 32](#)
- [Imported Scan and Vulnerability Data 37](#)
- [Host Data 39](#)
- [Vulnerability Management 39](#)
- [Host Management 41](#)
- [Host Badges 42](#)

## Discovery Overview

Before you can begin the exploitation phase of a penetration test, you must add host data to the project. Host data refers to the IP addresses of the systems that you want to exploit and the active ports, services, and vulnerability information associated with those systems. To add host data to a project, you can either run a discovery scan or you can import scan data from a vulnerability scanner, such as Nexpose or Nessus. If you import data from vulnerability analysis tool, or some other third party vendor, you should still run a discovery scan to identify new or additional information for those hosts.

A discovery scan is the port scanner included with Metasploit Community. It combines Nmap with several modules to identify the systems that are alive and to uncover the open ports and services. A port is a data connection that serves as a gateway for communication and enables traffic to travel between systems. Network services, like SSH, telnet, and HTTP, typically run on standard port numbers and can indicate the purpose of the system. You can use the results to filter the list of attackable targets.

For example, if you discover a service that allows remote code execution, like VNC, you can bruteforce the service to attempt to log into the system.

## Discovery Scan

A discovery scan queries network services to identify and fingerprint valid hosts. You can perform a discovery scan to identify the details of the hosts within a target address range and to enumerate the listener ports. To perform a discovery scan, you must supply Metasploit Community with a valid target range.



## IPv6 Addresses for Target Hosts

Metasploit Community does not automatically detect IPv6 addresses during a discovery scan. For hosts with IPv6 addresses, you must know the individual IP addresses that are in use by the target devices and specify those addresses to Metasploit Community. To identify individual IPv6 addresses, you can use SNMP, Nmap, or thc-alive6, which is part of the thc-ipv6 tool kit.

After you identify the IPv6 addresses for the target devices, you can either import a text file that contains the host addresses into a project or manually add the hosts to a project. If you choose to import the addresses, the text file that you use must list one IPv6 address on each line.

To import a host address file, select **Analysis > Hosts > Import**. The **Import Data** window appears. Browse to the location of the host address file and import the host address file.

To manually add a host, select **Analysis > Hosts > New Host**.

## Discovery Scan Options

The following table describes the settings that you can configure for a discovery scan:

Option	Description
Perform initial portscan	Performs a portscan before the discovery scan performs service version verification.
Custom Nmap arguments	Sends flags and commands to the Nmap executable. Discovery scan supports most Nmap options except for:  -o -i -resume -script -datadir -stylesheet
Additional TCP ports	Appends additional TCP ports to the existing Nmap scan ports. Discovery scan appends the ports to -p.
Excluded TCP ports	Excludes the TCP ports from service discovery, which includes all Nmap options.

Option	Description
Custom TCP port range	<p>Specifies a range of TCP ports for the discovery scan to use instead of the default ports.</p> <p>For example, if you specify ports 1-20, the following Nmap command is returned:</p> <pre>/nmap -sS -PS1-20 -PA1-20 -PU51094 -PP -PE -PM -PI -p1-20 --host-timeout=5m -O --max-rtt-timeout=300 --initial-rtt-timeout=100 --max-retries=2 --stats-every 10s --min-rate=200</pre> <p>Note: UDP Service Discovery or Identify Unknown Services run even if you configure a custom TCP port range.</p>
Custom TCP source port	Specifies the TCP source port that the discovery scan uses instead of the default port. Use this option to test firewall rules.
Fast detect: Common TCP ports only	Performs a scan on the most common TCP ports, which reduces the number of ports that the discovery scan scans.
Portscan speed	<p>Controls the Nmap timing option (-T). Choose from the following timing templates::</p> <p><b>Insane (5)</b> - Speeds up the scan. Assumes that you are on a fast network and sacrifices accuracy for speed. Scan delay is less than 5 ms.</p> <p><b>Aggressive (4)</b> - Speeds up the scan. Assumes that you are on a fast and reliable network. Scan delay is less than 10 ms.</p> <p><b>Normal (3)</b> - The default portscan speed. Does not affect the scan.</p> <p><b>Polite (2)</b> - Uses less bandwidth and target resources to slow the scan.</p> <p><b>Sneaky (1)</b> - Use this portscan speed for IDS evasion.</p> <p>Paranoid (0) - Use this portscan speed for IDS evasion.</p>
Portscan timeout	Determines the amount of time Nmap spends on each host. Default value is 5 minutes.
UDP service discovery	Sets the discovery scan to find all services that are on the network.
Scan SNMP community strings	Launches a background task that scans for devices that respond to a variety of community strings.
Enumerate users via finger	Queries user names when the discovery scan detects fingers.

Option	Description
Identify unknown services	Sets the discovery scan to find all unknown services and applications on the network.
Single scan: scan hosts individually	Runs a scan on individual hosts. The discovery scan scans the first host entirely and stores the information in the database before it moves onto the next host.
Dry run: only show scan information	Prepares the Nmap command line, but does not execute the command line.
SMB user name	Defines the user name that the Metasploit SMB enumeration modules use.
SMB password	Defines the password that the Metasploit enumeration modules use.
SMB domain	Defines the domain that the Metasploit enumeration modules use.

## Discovering Hosts

- 1.) Create or open a project to run a discovery scan.
- 2.) Click **Scan**. The **New Discovery Scan** window displays.
- 3.) Enter the target addresses that you want to include in the scan. Enter a single address, an address range, or a CIDR notation.
- 4.) Click **Show Advanced Options** to verify and configure the advanced options for the scan. If you do not configure additional options, Metasploit Community uses the default configuration for the scan.
- 5.) Run the scan.

## Discovering Virtual Hosts

When you perform a discovery scan, Metasploit Community automatically discovers guest operating systems on the target system. Metasploit Community displays a list of virtual machines on the host page and denotes the virtual machine with a VM icon. For example, a machine that runs VMware ESX displays the VMware icon and the guest operating system and version.

Virtualization support enables you to easily differentiate between actual machines and virtual machines. This ability becomes useful when you plan the scope of a penetration test.

## Supported Guest Operating Systems

Metasploit Community supports the following guest operating systems:

- VMware
- Xen
- BreakingPoint
- Virtual PC
- Virtual Iron
- QEMU
- VirtualBox

## Supported Host VM Servers

Metasploit Community supports the following host VM servers:

- VMware ESXi 3.5, 4.0, 4.1, and 5.0
- VMware ESX 1.5, 2.5, 3.0, and 4.0
- vCenter

## Compromised Virtual Systems

If you gain access to a target system that runs a virtual environment, Metasploit Community captures screenshots of the guest operating systems on the host system. To view the screenshots of the guest operating systems, go to **Analysis > Hosts > Captured Evidence**. The **Captured Evidence** tab displays a list of looted evidence, such as screenshots from virtual machines.

## Scanning the Network for H.323 Video Conferencing Systems

- 1.) Create or open a project.
- 2.) Click **Scan**.
- 3.) Click **Show Advanced Options**.
- 4.) Enter 1720 for the Custom TCP source port.
- 5.) Clear the **UDP service discovery** option.
- 6.) Select the **Scan H.323 video endpoints** option.
- 7.) Run the scan.

## Defining Nmap Arguments

Administrators can define a list of command line arguments to the Nmap executable for a discovery scan. The command line arguments take precedence over any internal system settings. You can use Nmap arguments to perform custom scan techniques, alternate configurations, and modify scan speeds.

The discovery scan supports most Nmap options except for -o, -i, -resume, -datadir, and -stylesheet.

- 1.) Open a project and launch a discovery scan. The **New Discovery Scan** window appears.
- 2.) Click **Show Advanced Options**.
- 3.) Enter the Nmap arguments in the **Custom Nmap arguments** field.
- 4.) Configure any additional options for the scan.
- 5.) Run the scan.

## Nexpose Scan

You can use the Community and Enterprise editions of Nexpose to discover and scan devices for known vulnerabilities. After you complete a Nexpose scan, you can import the scan data into Metasploit Community. Metasploit Community imports the scan data and enables you to validate and test the scan results.

Metasploit Community provides a connector that allows you to run and automatically import the results of a Nexpose scan into a project.

Before you can run a Nexpose scan, you must download, install, and configure Nexpose. Additionally, you must configure a Nexpose console through Metasploit Community.

Metasploit Community only supports the number of hosts that you have licenses for in Nexpose. If you provide more hosts than you have licenses for, the scan fails. For example, if you have a Community license, the most number of hosts Nexpose supports is 32. If you provide 35 hosts, the scan fails.

You can download the Community edition of Nexpose from <http://www.rapid7.com/vulnerability-scanner.jsp>. For more information on how to install and configure Nexpose, visit <http://community.rapid7.com>.

## Nexpose Scan Options

The following table describes the settings that you can configure for a discovery scan:

Option	Description
Nexpose scan targets	Defines the target address range for the Nexpose scan.
Scan Template: Penetration Test Audit	Uses safe checks to perform an in-depth penetration test of the target systems. Enables host discovery and network penetration options, which allows Nexpose to dynamically discover additional systems in the target network.

Option	Description
Scan Template: Full Audit	Uses safe checks to perform a full network audit of all target systems. The network audit includes network-based vulnerability checks, patch/hot fix checks, and application layer audits. The Full Audit scan only scans default ports. Policy checking is disabled, which makes the Full Audit scan perform faster than the Exhaustive scan.
Scan Template: Exhaustive Audit	Uses safe checks to perform an exhaustive network audit of all target systems and services. The network audit includes network-based vulnerability checks, patch/hot fix checks, and application layer audits. An Exhaustive scan can take several hours or days to complete.
Scan Template: Discovery	Identifies live devices on the network, which includes the host name and operating system for each host. The Discover scan does not perform any additional enumeration or policy/vulnerability scanning.
Scan Template: Aggressive Discovery	Performs a fast and cursory scan to identify live devices on high speed networks. The discovery scan identifies the host name and operating system for each host. The discovery scan sends packets at a high rate, which may trigger IPS and IDS sensors, SYN flood protection, and exhaust states on stateful firewalls. The Aggressive Discovery scan does not perform any additional enumeration or policy/vulnerability scanning.
Scan Template: DoS Audit	Uses safe and unsafe checks to perform a basic audit of all target systems. The DoS Audit scan does not perform any additional enumeration or policy/vulnerability scanning.
Purge scan results upon completion	Removes the results from the scan from the Nexpose console after the scan completes.
Specify additional scan credentials	Defines the credentials that the Nexpose scan uses. Multiple credentials are not supported. You must use Nexpose to configure multiple credential support.
Pass the LM/NTLM hash credentials	Enables a Nexpose scan to use the password hashes that Metasploit Community collects to authenticate against the host.

Option	Description
Hash credentials	Defines the hash credentials that you want to use to authenticate against a target. The hash credentials are populated with the hash values that Metasploit Community collects from the target. If you need to modify the hash list, use the following format to add or modify hash credentials: <user name>:LM:NTLM.
Type	Use Windows/CIFS, Secure Shell/SSH, Telnet, HTTP, FTP, SNMP, or POP3. This option appears if you select that you want to specify additional scan credentials.
User	Defines the user name for the scan credentials. This option appears if you select that you want to specify additional scan credentials.
Password	Defines the password for the scan credentials. This option appears if you select that you want to specify additional scan credentials.

## Configuring a Nexpose Console

Before you can run a Nexpose scan, you must add a Nexpose console to the system. You can manage Nexpose consoles globally. Connections to the Nexpose console act as a persistent connections that you can use to import individual sites into a project.

After you set up the Nexpose console, you can access and use the console for a Nexpose scan. Configured Nexpose consoles are automatically available for you to use.

- 1.) Open a project.
- 2.) Click **Administration > Global Settings** from the main menu.
- 3.) Scroll down to the Nexpose Consoles area.
- 4.) Click **Configure a Nexpose Console**.
- 5.) Enter a console name.
- 6.) Enter the console address.
- 7.) Enter the console port.
- 8.) Enter the console user name.
- 9.) Enter the console password.
- 10.) Save the Nexpose console configuration.

## Running a Nexpose Scan

- 1.) Open a project.
- 2.) Click the **Analysis** tab.
- 3.) Click **Nexpose** from the Quick Tasks menu.
- 4.) Select a Nexpose console. The list shows Nexpose consoles that you have added to the project.
- 5.) Enter the addresses for the scan targets. You can specify an IP address or a host name. There can be one address on each line.

**Note:** You can use standard IPv6 addressing to define individual IPv6 addresses. For example, use `fe80::202:b3ff:fe1e:8329` for single addresses and `2001:db8::/32` for CIDR notations. For link local addresses, you must append the interface ID to the address. For example, enter `fe80::1%eth0` for a link local address.

- 6.) Select a scan template.
- 7.) Click **Show Advanced Options** to configure additional options for the scan.
- 8.) Launch the Nexpose scan.

## Running a Nexpose Scan with a Custom Scan Template

To use a custom scan template for a Nexpose scan, you must supply the scan template ID, not the scan template name. To identify the scan template ID, log into the Nexpose Security Console, select **Administration > Scan Templates**, and choose the scan template that you want to use.

When the Scan Template Configuration page displays, locate the URL address box at the top of the Nexpose Console. The URL address box displays the address and the template ID for the scan template. For example, in the following address, <https://my.console.address:3780/admin/wizard/scan-template.html?templateid=dos-audit>, the template id is `dos-audit`.

For more information on scan template IDs, visit the [Nexpose documentation](#).

- 1.) Open a project.
- 2.) Click the **Analysis** tab.
- 3.) Click **Nexpose** from the Quick Tasks menu.
- 4.) Select a Nexpose console. The list shows Nexpose consoles that you have added to the project.
- 5.) Enter the addresses for the scan targets. You can specify an IP address or a host name. There can be one address on each line.

**Note:** You can use standard IPv6 addressing to define individual IPv6 addresses. For example, use `fe80::202:b3ff:fe1e:8329` for single addresses and `2001:db8::/32` for CIDR notations. For link local addresses, you must append the interface ID to the



address. For example, enter `fe80::1%eth0` for a link local address.

- 6.) Click the Scan Template list. Choose **Custom**, which enables you to select a custom scan template.
- 7.) Click **Show Advanced Options**.
- 8.) From the **Advanced Nexpose Scan Settings** area, enter the scan ID for the that you want to use in the **Custom scan template name** field.

**Note:** Scan template IDs cannot contain a hyphen. If the scan template ID contains a hyphen, replace the hyphen with an underscore. If the scan template ID changes, the Nexpose scan does not update the scan template ID. You must update the Nexpose scan to use the new scan template ID.

- 9.) Launch the Nexpose scan.

## Passing the Hash from Metasploit Community

Passing the hash is a technique that enables attackers to use the NTLM and LM of a user's password to authenticate to a remote server or service. During exploitation, Metasploit Community collects data, such as password hashes, from the exploited system. After Metasploit Community collects password hashes from a target system, you can pass the hash and run a Nexpose scan to perform a credentialed scan.

Before you can pass the hash in Metasploit Community, you must configure a Nexpose console from the Global Settings. After you configure a Nexpose console, you can launch a Nexpose scan from the Metasploit Community interface to pass the hash to the Nexpose scan.

- 1.) Open a project.
- 2.) Click the **Analysis** tab.
- 3.) Click **Nexpose** from the Quick Tasks menu.
- 4.) Select a Nexpose console. The list shows Nexpose consoles that are available for the project.
- 5.) Enter addresses for the scan targets. You can specify an IP address or a host name. There can be one address on each line.

**Note:** Metasploit Community supports IPv4 and IPv6 addresses. You can use standard IPv6 addressing to define individual IPv6 addresses. For example, use `fe80::202:b3ff:fe1e:8329` for single addresses and `2001:db8::/32` for CIDR notations. For link local addresses, you must append the interface ID to the address. For example, enter `fe80::1%eth0` for a link local address.

- 6.) Select a scan template.
- 7.) Click **Show Advanced Options** to configure additional options for the scan.
- 8.) Select **Pass the LM/NTLM hash credentials**. The **Hash Credentials** box displays. Metasploit Community automatically populates the **Hash Credentials** box with a list of looted hashes. You can modify or add hashes to the hash list.

- 9.) Launch the Nexpose scan.

## Purging Scan Data

A purge removes all scan data from the Nexpose console and ensures optimal performance from the Nexpose scanner.

If you enable the purge scan option, Nexpose automatically deletes the scan data when the scan completes.

- 1.) Open a project.
- 2.) Click the **Analysis** tab.
- 3.) Click **Nexpose** from the Quick Tasks menu.
- 4.) Select a Nexpose console. The list shows Nexpose consoles available for the project.
- 5.) Enter addresses for the scan targets. You can specify an IP address or a host name. There can be one address on each line.

**Note:** Metasploit Community supports IPv4 and IPv6 addresses. You can use standard IPv6 addressing to define individual IPv6 addresses. For example, use `fe80::202:b3ff:fe1e:8329` for single addresses and `2001:db8::/32` for CIDR notations. For link local addresses, you must append the interface ID to the address. For example, enter `fe80::1%eth0` for a link local address.

- 6.) Select a scan template.
- 7.) Click **Show Advanced Options** to configure additional options for the scan.
- 8.) Select **Purge Scan results** upon completion.
- 9.) Launch the Nexpose scan.

## Imported Scan and Vulnerability Data

You can import scan data into Metasploit Community. When you import scan data, you import the hosts, ports, and services that the scan report contains.

## Supported Scan Data Formats

Metasploit Community supports the following data file formats:

- Metasploit PWDump Export
- Metasploit XML (all versions)
- Metasploit ZIP (all versions)
- NeXpose Simple XML or XML
- NeXpose Raw XML or XML Export
- Foundstone Network Inventory XML
- Microsoft MBSA SecScan XML

- nCircle IP360 (XMLv3 and ASPL)
- NetSparker XML
- Nessus NBE
- Nessus XML (v1 and v2)
- Qualys Asset XML
- Qualys Scan XML
- Burp Session XML
- Acunetix XML
- AppScan XML
- Nmap XML
- Retina XML
- Amap Log
- IP Address List
- Libcap

Raw XML is only available in commercial editions of Nexpose and includes additional vulnerability information.

**Note:** Metasploit Community does not import service and port information from Qualys Asset files. If you import a Qualys Asset file, you need to run a discovery scan on the imported hosts to enumerate services and ports that are active on those hosts.

## Importing Data

- 1.) Open or create a project.
- 2.) Click the **Analysis** tab. The **Host** window appears.
- 3.) Click **Import**. The **Import Data** window appears.
- 4.) Click **Browse** to choose a file to import. The **File Upload** window appears.
- 5.) Navigate and choose a file to import. Click **Open** after you select the file.
- 6.) Enter the target addresses that you want to exclude.

**Note:** Metasploit Community supports IPv4 and IPv6 addresses. You can use standard IPv6 addressing to define individual IPv6 addresses. For example, use `fe80::202:b3ff:fe1e:8329` for single addresses and `2001:db8::/32` for CIDR notations. For link local addresses, you must append the interface ID to the address. For example, enter `fe80::1%eth0` for a link local address.

- 7.) Select **Do not change existing hosts** if you do not want the imported information to affect the existing hosts.
- 8.) Select if you want Metasploit Community to automatically tag hosts with their OS as the system imports them. Enable any additional tags that you want to use.
- 9.) Import the data.

## Host Data

During a scan, Metasploit Community collects additional host information that you can view from the Analysis page. Metasploit Community collects information from notes, services, vulnerabilities, and captured evidence.

You can view host data through a grouped view or an individual view. The grouped view shows the information grouped together by service type, vulnerability type, and evidence type. The individual view lists all services, vulnerabilities, and evidence.

## Viewing Host Notes

- 1.) Open a project.
- 2.) Click the **Analysis** tab. The **Host** window appears.
- 3.) Click the **Notes** tab. A list of all notes appears.

## Viewing Host Services

- 1.) Open a project.
- 2.) Click the **Analysis** tab. The **Host** window appears.
- 3.) Click the **Services** tab. A list of all services appears.

## Viewing Host Evidence

- 1.) Open a project.
- 2.) Click the **Analysis** tab. The **Host** window appears.
- 3.) Click the **Captured Evidence** tab. A list of all captured evidence appears.

## Viewing Host Vulnerabilities

- 1.) Open a project.
- 2.) Click the **Analysis** tab. The **Host** window appears.
- 3.) Click the **Vulnerabilities** tab. A list of all vulnerabilities appears.

## Vulnerability Management

When Metasploit Community scans target systems, it identifies and fingerprints hosts as well as determines the details of the hosts within a target address range. During the scanning process, Metasploit Community identifies any known vulnerabilities for the target hosts.

If Metasploit Community does not identify a known vulnerability during a scan, you can add the vulnerability to a target host.

**Note:** Before you modify or add a vulnerability, you must run a discovery scan for the project.

## Adding a Vulnerability

- 1.) Open a project.
- 2.) Click the **Analysis** tab. The **Host** window appears.
- 3.) Click on a host IP address to open the host details window.
- 4.) Click the **Vulnerabilities** tab.
- 5.) Click **New Vuln**. The **New Vuln** window appears.
- 6.) Enter the vulnerability name. For example, `exploit/windows/smb/psexec`.
- 7.) Enter reference information for the vulnerability (CVE identifier, OSVDBID). Use the **Add Reference** button to add a new line of information.
- 8.) Save the vulnerability.

## Exploiting a Known Vulnerability

After Metasploit Community identifies the vulnerabilities that exist on a host, you can access and run the exploit for each vulnerability directly from the host page. If you want to view more information about the vulnerability, you can click the reference number that Metasploit Community lists for each vulnerability.

- 1.) Open a project.
- 2.) Click the **Analysis** tab. The **Host** window appears.
- 3.) Click on a host IP address to open the host details window.
- 4.) Click the **Vulnerabilities** tab. The tab displays the vulnerabilities for the host.
- 5.) Click the exploit name. The module page appears. Configure the options that you want the exploit to use.
- 6.) Run the exploit.

## Editing a Vulnerability

- 1.) Open a project.
- 2.) Click the **Analysis** tab. The **Host** window appears.
- 3.) Click the **Vulnerabilities** tab.
- 4.) Locate the vulnerability that you want to edit and click **Edit**.
- 5.) Edit the settings and reference information.

- 6.) Save the changes.

## Deleting a Vulnerability

- 1.) Open a project.
- 2.) Click the **Analysis** tab. The **Host** window appears.
- 3.) Click on a host IP address to open the host details page.
- 4.) Click the **Vulnerabilities** tab.
- 5.) Locate the vulnerability that you want to delete and click **Delete**.

## Host Management

You can manually configure a host if there is a host that you want to add to the project. You can configure the details for the host, which includes the network, operating system, and service information. You can also delete any hosts that you no longer need to access for the project.

### Adding a Host

- 1.) Open a project.
- 2.) Click the **Analysis** tab. The **Hosts** window appears.
- 3.) Click **New Host**.
- 4.) Enter a name for the host.
- 5.) Enter an IP address for the host.

**Note:** Metasploit Community supports IPv4 and IPv6 addresses. You can use standard IPv6 addressing to define individual IPv6 addresses. For example, use `fe80::202:b3ff:fe1e:8329` for single addresses and `2001:db8::/32` for CIDR notations. For link local addresses, you must append the interface ID to the address. For example, enter `fe80::1%eth0` for a link local address.

- 6.) Enter an optional Ethernet address for the host.
- 7.) Enter an optional OS system for the host. For example, enter `Windows XP`.
- 8.) Enter an optional OS version for the host. For example, enter `SP2`.
- 9.) Enter an optional OS flavor for the host.
- 10.) Enter an optional purpose for the host. For example, enter `client` or `server`.
- 11.) Select **Lock edited host attributes** if you do not want import, discovery scan, or Nexpose scan to change the host on subsequent scans.
- 12.) Click **Add Service** if you want to add a service to the host. If you add a service, enter the name, port, protocol, and state for the service.
- 13.) Save the host.

## Deleting a Host

- 1.) Open a project.
- 2.) Click the **Analysis** tab. The **Hosts** window appears.
- 3.) Select the hosts that you want to delete.
- 4.) Click **Delete**.
- 5.) Confirm that you want to delete the host.

## Host Badges

A host badge identifies the status of each discovered host. Use the host badge to determine whether Metasploit Community has scanned, cracked, shelled, or looted the host.

You can view the host badge for a host from the **Status** column on the **Analysis** window.

The following table describes the host badges:

Host Badge	Description
Scanned	The discovery scan discovered the host.
Cracked	The bruteforce was successful, but the system could not open a session.
Shelled	The system opened a session on the target device.
Looted	The system collected evidence from the device.

# EXPLOITATION

This chapter covers the following topics:

- [Modules 43](#)
- [Modules 43](#)
- [Exploits 46](#)
- [Post-Exploitation 48](#)

## Exploitation

An exploit executes a sequence of commands that target a specific vulnerability found in a system or application to provide the attacker with access to the system. Exploits include buffer overflow, code injection, and web application exploits.

With Metasploit Community, you have the ability to run manual exploits against a target system. A manual exploit is a module that you can select, configure, and run individually. In order to manually run exploits, you must know the vulnerabilities and security flaws that exist on the target system. This knowledge helps you determine which module would be most effective against the system.

For example, if you know that the host runs Windows Service Pack 1, you can run an exploit that targets Windows Service Pack 1 vulnerabilities. Or if you know that the target system has a specific vulnerability that you want to test, you can run the exploit that targets that particular weakness.

The options and instructions that you perform for manual exploits vary based on the exploit that you choose to run. Therefore, use the following instructions as a guideline to manually run exploits.

## Modules

A module is the component that Metasploit Community uses to perform an attack or a specific action. The attack or action that the module performs depends on the module type.

## Module Types

The Metasploit Framework categorizes modules based on the action that the module performs.



The following are modules types that are available:

- Exploit - A module that targets and exploits the vulnerabilities that the vulnerability scanners discover.
- Auxiliary - A module that performs tasks other than exploitation, such as fuzzing and scanning.
- Post-Exploitation - A module that runs after Metasploit Community compromises a target system.

## Excluded Modules

Most modules that are available in the Metasploit Framework are available in Metasploit Community. However, some modules may be excluded if their dependencies are unavailable.

Modules that are currently excluded are modules that depend on the following libraries:

- Oracle - Affects modules that target Oracle.
- Lorcon2 - Affects modules that target wireless systems.
- Libpcap - Affects modules that target sniffers.
- DECT - Affects modules that target telephony.

## Module Search

The module search engine searches the module database for the keyword expression and returns a list of results that match the query. Use the module search engine to find the module that you want to run against a target system.

## Keyword Tags

You can use keyword tags to define a keyword expression.

The following table describes keyword tags:

Keyword Tag	Description
name	Searches for the keyword expression within the module descriptive name.
path	Searches for the keyword expression within module path name.
platform	Searches for the modules that affect the platform or target that you define in the keyword expression.
type	Searches for the modules that belong to the module type that you define in the keyword expression. For example, use exploit, auxiliary, or post.
app	Searches for modules that are either a client or server attack.

Keyword Tag	Description
author	Searches for modules by author.
cve	Searches for modules by CVE ID.
bid	Search for modules by Bugtraq ID.
osvdb	Search for modules by OSVDB ID.

## Defining a Keyword Expression

A keyword expression consists of a keyword tag and the keyword.

The following table contains examples of keyword expressions:

Key Tag	KeyWord Expression Example
name	name:Java
path	path:windows/smb
platform	platform:linux
type	type:exploit
app	app:client
author	author:todb
cve	cve:2009
bid	bid:10078
osvdb	osvdb:875

## Searching for Modules

- 1.) Open a project.
- 2.) Click the **Modules** tab.
- 3.) Enter a keyword expression to search for a specific module. Use the keyword tags to define the keyword expression.
- 4.) Press **Enter** to perform a search.

## Module Statistics

Module statistics show the total number of modules that are available and show the number of modules that are available for each type of module. Module types include exploit modules, auxiliary modules, server-side exploits, and client-side exploits.

## Viewing Module Statistics

- 1.) Open a project.
- 2.) Click the **Modules** tab. You can view the module statistics from the **Module Statistics** area.

## IPv6 Payloads

The following table describes the IPv6 payloads that are available for Windows, Linux, BSD, Shell, and PHP targets. If the IPv6 payload successfully executes on the target machine, then a session opens on the target machine.

IPv6 Target	Payloads
Windows x86	stagers/windows/reverse_ipv6_http stagers/windows/reverse_ipv6_https stagers/windows/reverse_ipv6_tcp stagers/windows/bind_ipv6_tcp
Linux x86	singles/linux/x86/shell_bind_ipv6_tcp stagers/linux/x86/reverse_ipv6_tcp stagers/linux/x86/bind_ipv6_tcp
BSD x86	singles/bsd/x86/shell_reverse_tcp_ipv6 singles/bsd/x86/shell_bind_tcp_ipv6 stagers/bsd/x86/reverse_ipv6_tcp stagers/bsd/x86/bind_ipv6_tcp
Shell	singles/cmd/windows/bind_perl_ipv6 singles/cmd/unix/bind_netcat_ipv6 singles/cmd/unix/bind_perl_ipv6 singles/cmd/unix/bind_ruby_ipv6
PHP	singles/php/bind_perl_ipv6 singles/php/bind_php_ipv6 stagers/php/bind_tcp_ipv6

## Exploits

An exploit executes a sequence of commands to target a specific vulnerability found in a system or application. An exploit takes advantage of a vulnerability to provide the attacker with access to the target system. Exploits include buffer overflow, code injection, and web application exploits.

# Manual Exploits

A manual exploit is a module that you can select and run individually. You perform a manual exploit when you want to exploit a known vulnerability.

You choose the exploit module based on the information you have about the host. For example, if you know that the host runs Windows Service Pack 1, you can run an exploit that targets Windows Service Pack 1 vulnerabilities. Or if you know that the target system has a specific vulnerability that you want to test, you can run the exploit that targets that particular weakness.

The options and instructions that you perform for manual exploits vary based on the exploit that you choose to run. Therefore, use the following instructions as a guideline to manually run exploits.

## Manual Exploits Overview

- Create a list of system targets.
- Create a map of all available exploits using references, ports, and service names.
- Create a match table of exploits for systems, but do not include devices that are fragile or devices that cannot be exploited.
- Create a prioritized queue of exploit modules based on reliability and interleave exploits between hosts.
- Execute exploit modules until Metasploit Community obtains a session.

## Running a Manual Exploit

- 1.) Open a project.
- 2.) Click the **Modules** tab.
- 3.) Use the search engine to find a specific module. Use the keyword tags to define the search term.
- 4.) Click on a module name to select the module. The **Module** window appears.
- 5.) Define the target hosts that you want to include or exclude from the exploit.
- 6.) Define the payload options, if the options are available.
- 7.) Define the module options. Module options vary between modules. Use the in-product help to view descriptions for each option.
- 8.) Define the advanced options. Advanced options vary between modules. Use the in-product help to view descriptions for each option.
- 9.) Define the evasion options. Evasion options vary between modules. Use the in-product help to view descriptions for each option.
- 10.) Run the module.

# Post-Exploitation

After you gain access to a target system, you can run scripts through the command shell or run post-exploitation modules to take control of the system.

## Post-Exploitation Modules

A post-exploitation module provides a standardized interface that you can use to perform post-exploit attacks. The post-exploitation phase enables you to collect further information about a target system and to gain further access to the network. During the post-exploitation phase, you can identify things like additional subnets, routers, server names, network services, and installed applications.

After you obtain a session on the target system, you can view the post-exploitation modules that are applicable for that session.

## Running Post-Exploitation Modules

- 1.) Open a project.
- 2.) Click the **Sessions** tab. The **Sessions** window appears.
- 3.) Click on a session name from the **Active Sessions** column.
- 4.) Click the **Post-Exploitation Modules** tab. The **Module** window appears.
- 5.) Click on a module name from the **Module Name** column. The module information appears.
- 6.) Select the module options you want to use.
- 7.) Define the advanced options for the module.
- 8.) Run the module.

## Post-Exploitation Modules for Virtual Targets

After you gain access to a virtual target, you can utilize post-exploitation modules to interact with the virtual machines. The post-exploitation modules that are available for virtual machines enable you to log into VMware and terminate user sessions and enumerate VirtualBox machines on the target machine.

The following are post-exploitation modules that you can use for virtual machines:

- post/multi/gather/find\_vmx
- post/multi/gather/enum\_vbox

## Post-Exploitation Macros

A post-exploitation macro is a set of predefined actions that deploy when Metasploit Community obtains an active session. The session can be an existing session or a session that a task creates, like a campaign task. You can use a post-exploitation macro to automate the events that occur after Metasploit Community opens a session on a target system.

A post-exploitation macro automatically runs after a target system runs an exploits and connects the post-exploitation macro to a listener. Therefore, before you can execute a post-exploitation macro, you must create a listener and assign the listener to the post-exploitation macro.

To create a listener, you can define a global listener, or you can assign a macro to a campaign. If you create a macro through a campaign, the campaign automatically creates a listener and connects the macro to the listener.

You can manage post-exploitation macros and persistent listeners from the global settings area of the project.

### Creating a Post-Exploitation Macro

- 1.) Open a project.
- 2.) Click **Administration > Global Settings** from the main menu. The **Global Settings** window appears.
- 3.) Click **New Macro**, which is located under Post-Exploitation Macros. The **Macros Settings** window appears.
- 4.) Enter a name for the post-exploitation macro.
- 5.) Enter a description for the post-exploitation macro.
- 6.) Enter a time limit, in seconds, for the post-exploitation macro.
- 7.) Save the post-exploitation macro. After you save the post-exploitation macro, a list of available actions displays.
- 8.) Search through the list of modules and find the module that you want to add to the post-exploitation macro.
- 9.) Add the module. The **Module Configuration** window appears.
- 10.) Configure the options for the module. Options vary between modules. Refer to the in-product help for descriptions of the options.
- 11.) Repeat the previous step for each module that you want to add to the post-exploitation macro. Add the modules in the order in which you want the modules to execute.

## Listeners

After an exploit successfully compromises a target system, Metasploit Community uses a listener to wait for an incoming connection from the exploited system. The listener is the component that handles persistent agents from exploited systems.

When you create a listener, you associate the listener to a specific project. Therefore, when an exploited target makes a connection with the listener, you see an active session open in the project.

**Note:** You can create global listeners that you can use across multiple projects. However, only one project can use the listener at a time.

You assign a post-exploitation macro to each listener. When the exploited system makes a connection with the attacking system, Metasploit Community launches the post-exploitation macro. Listeners stop after you delete a project or you manually stop a listener.

## Creating a Listener

When you create a listener, Metasploit Community uses the listener address and port to assign a listener name. For example, if the listener address is 10.10.10.1, and the port is 47385, then the port name is 10:10:10:1:47385.

- 1.) Open a project.
- 2.) Click **Administration > Global Settings** from the main menu.
- 3.) Click New Listener, which is located under Persistent Listeners. The **Create a Listener** window appears.
- 4.) Choose an associated project for the listener.
- 5.) Define the listener payload type.
- 6.) Enter an IP address for the listener.

**Note:** Metasploit Community supports IPv4 and IPv6 addresses.

- 7.) Enter a port for the listener.
- 8.) Choose a post-exploitation macro to deploy after the listener connects to the target system.
- 9.) Enable the listener.
- 10.) Save the listener.

## Enabling and Disabling a Listener

- 1.) Open a project.
- 2.) Select **Administration > Global Settings** from the main menu. The **Global Settings** window appears.

- 3.) Click on a listener from the **Scope** column.
- 4.) Select or deselect the Enabled option.
- 5.) Update the listener.

## Stopping a Listener

To stop a listener, you can either delete the listener from the system or you can stop the listener from the Task screen.

- 1.) Open a project.
- 2.) Click the **Tasks** tab.
- 3.) Find the listening tasks.
- 4.) Click the **Stop** button in the **Timestamp/Duration** column.



# INDEX

## A

auxiliary 44

## D

Dashboard 10  
data file formats 37  
discovery scan 27

## E

exploit 6, 43, 44, 47

## G

global settings 12, 18

## H

H.323 31  
hash 36  
host  
    add 41  
    management 41  
host badge 42  
host data 39  
host notes 39  
host services 39  
HTTP payloads 19  
HTTPS payloads 19

## K

keyword expression 44, 45  
keyword tags 44

## L

license key  
    revert 20  
    update 20  
license keys 20  
listener 50  
    create 50  
LM 36  
log files 21

## M

manual exploits 43, 47  
module 43  
module statistics 46  
modules 14

## N

network boundaries 23  
network range 23  
    restrict 23  
Nexpose console 34  
Nexpose scan 32  
Nmap arguments 31  
NTLM 36

## O

offline activation file 20

## P

payload 6  
post-exploitation macro 49  
post-exploitation module 44  
post-exploitation modules 48  
project 22  
    create 25  
    edit 25  
project settings 23

## S

scan template 33  
    aggressive discovery 33  
    discovery 33  
    DoS Audit 33  
    exhaustive audit 33  
    full audit 33  
service listeners 4  
system updates 21

## T

task 7

## U

uninstall  
    Metasploit 22  
updates 21  
user account 16  
    delete 18  
    edit 17  
    reset 17

## V

vulnerability 7, 40  
    delete 41  
    edit 41  
    management 40