metasploit®

4.11

USER GUIDE

Getting Started

First things first. If you haven't installed Metasploit yet, check out this these instructions if you're a commercial user. Otherwise, if you already have Metasploit installed, congratulations! You've come to the right place to get started.

What's Metasploit?

Metasploit is a penetration testing platform that enables you to find, exploit, and validate vulnerabilities. The platform includes the Metasploit Framework and its commercial counterparts: Metasploit Pro, Express, Community, and Nexpose Ultimate.

Metasploit Framework

The Metasploit Framework is the foundation on which the commercial products are built. It is an open source project that provides the infrastructure, content, and tools to perform penetration tests and extensive security auditing. Thanks to the open source community and Rapid7's own hard working content team, new modules are added on a regular basis, which means that the latest exploit is available to you as soon as it's published.

There are quite a few resources available online to help you learn how to use the Metasploit Framework; however, we highly recommend that you take a look at the Metasploit Framework Wiki, which is maintained by Rapid7's content team, to ensure that you have the most up to date information available. You can also use the sidebar navigation on the left to view the documentation that is available on this site; just click on the **Metasploit Framework** topic or search for the topic you want. Either way, if you are unable to find what you need, let us know, and we will add it to the documentation back log.

Metasploit Pro and Other Commercial Editions

The commercial editions of Metasploit, which include Pro, Express, Community, and Nexpose Ultimate, are available to users who prefer to use a web interface to pentest. In addition to a web interface, some of the commercial editions provide features that are unavailable in the Metasploit Framework. Most of the additional features are targeted towards automating and streamlining common pentest tasks, such as vulnerability validation, social engineering, custom payload generation, and bruteforce attacks.

! All features documented are available in Metasploit Pro. Certain features may not be available in other editions. For a comparison of features available in different editions, check out this handy page that breaks down the features in each edition.

If you are command line user, but still want access to the commercial features, don't worry. Metasploit Pro includes its very only console, which is very much like msfconsole, except it gives you access to most of the features in Metasploit Pro via command line.

Metasploit Implementation

Rapid7 distributes the commercial editions of Metasploit as an executable file for Linux and Windows operating systems.

You can download and run the executable to install Metasploit Pro on your local machine or on a remote host, like a web server. Regardless of where you install Metasploit Pro, you can access the user interface through a web browser. Metasploit Pro uses a secure connection to connect to the server that runs it.

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

! You must have a license key to activate the product. If you do not have a license key, please contact the Rapid7 sales team at sales@rapid7.com.

Metasploit Pro Components

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

Metasploit Framework

An open source penetration testing and development platform that provides you with access to every module that Metasploit Pro 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 Pro.

Modules

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 Pro.

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.

Services

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

Web Interface

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

Understanding Basic Concepts and Terms

To help familiarize you with Metasploit Pro, the following are some basic terms and concepts that you should understand.

Project

All work in Metasploit Pro must be done inside of a project. A project is basically a container for the data you use and collect during a penetration test.

Workspace

A workspace is the same thing as a project, except it's only used when referring to the Metasploit Framework.

Task

Everything you do in Metasploit is a task. A task is basically any action that you can perform in Metasploit, like running a scan or exploit.

Module

Most of the tasks that you perform in Metasploit require the use of a module, which is a standalone piece of code that extends the functionality of the Metasploit Framework. A module can be an exploit, auxiliary or post-exploitation module. The module type determines its purpose. For example, any module that can open a shell on a target is considered an exploit module. A popular exploit module is MS08-067.

Exploit Module

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

Auxiliary Module

An auxiliary module does not execute a payload and perform arbitrary actions that may not be related to exploitation. Examples of auxiliary modules include scanners, fuzzers, and denial of service attacks.

Post-Exploitation Module

A post-exploitation module enables you to gather more information or to gain further access to an exploited target system. Examples of post-exploitation modules include hash dumps and application and service enumerators.

Payload

A payload is the shell code that runs after an exploit successfully compromises a system. The payload enables you to define how you want to connect to the shell and what you want to do to the target system after you take control of it. A payload can open a Meterpreter or command shell. Meterpreter is an advanced payload that allows you to write DLL files to dynamically create new features as you need them.

Bind Shell Payload

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

Database

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

Discovery Scan

A discovery scan is a Metasploit scan that combines Nmap and several Metasploit modules to enumerate and fingerprint targets.

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 it to a target. 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.

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.

Modules

A prepackaged collection of code from the Metasploit Framework that performs a specific task, such as run a scan or launch an exploit.

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.

Project

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

Reverse Shell Payload

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

Shellcode

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

Shell

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

Task

A task is an action that Metasploit Pro can perform. Examples of tasks include performing a scan, running a bruteforce attack, exploiting a vulnerable target, or generating a report.

Vulnerability

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

Metasploit Pro 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 Pro workflow can be tailored based on the various phases of penetration testing. Generally, the Metasploit Pro workflow includes the following steps:

1. **Create a project** - Create a project to store the data that you collect from your targets.
2. **Gather information**- Use the discovery scan, Nexpose scan, or import tool to supply Metasploit Pro with a list of targets and the running services and open ports associated with those targets.
3. **Exploit** - 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.
4. **Perform post-exploitation** - Use post-exploitation modules or interactive sessions to interact gather more information from compromised targets. Metasploit Pro 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 compromised systems.
5. **Clean up open sessions** - Use the Clean Up option 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.
6. **Generate reports** - Use the reporting engine to create a report that details the findings of the penetration test. Metasploit Pro provides several types that let you to determine the type of information that the report includes.

Accessing Metasploit Pro

To access the web interface for Metasploit Pro, open a browser and go to `https://localhost:3790` if Metasploit Pro runs on your local machine. If Metasploit Pro runs on a remote machine, you need to replace `localhost` with the address of the remote machine.

To log in to the web interface, you will need the username and password for the account you created when you activated the license key for Metasploit Pro. If you can't remember the password you set up for the account, you'll need to reset your password.

Supported Browsers

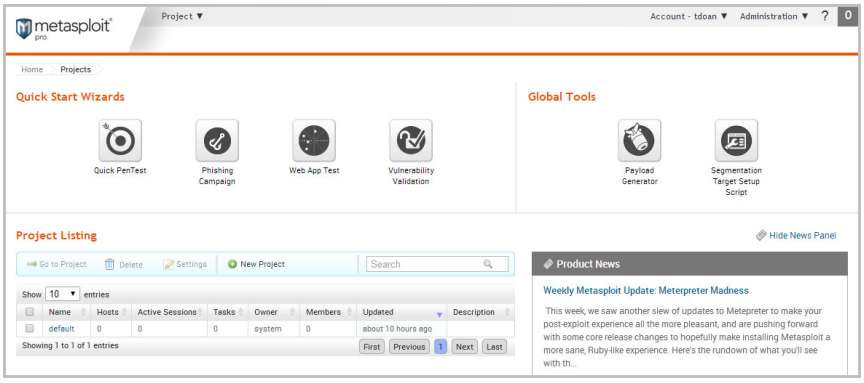
If the user interface is not displaying all of its elements properly, please make sure that you are using one of the supported browsers listed below:

- Google Chrome 10+
- Mozilla Firefox 18+
- Internet Explorer 10+
- Iceweasel 18+

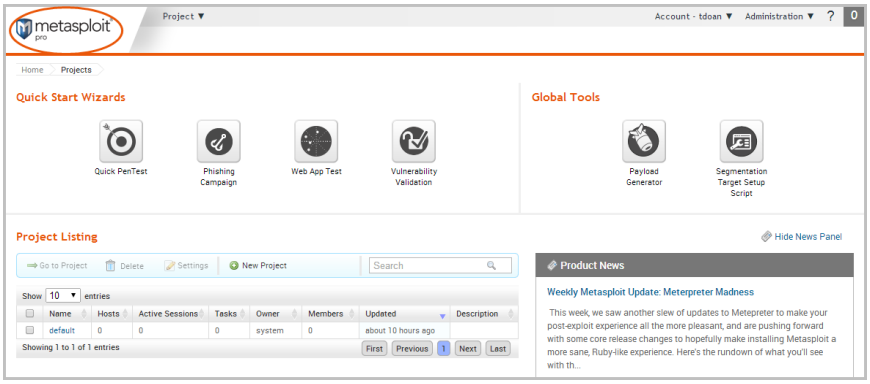
Touring the Projects Page

Now that you are familiar with some of the basics of Metasploit, let's take a more in depth look at Metasploit Pro.

After you log in to Metasploit Pro, the first screen that appears is the Projects page. The Projects page lists all of the projects that are currently stored in the Metasploit Pro instance and provides you with access to the quick start wizards, global tools, and product news.

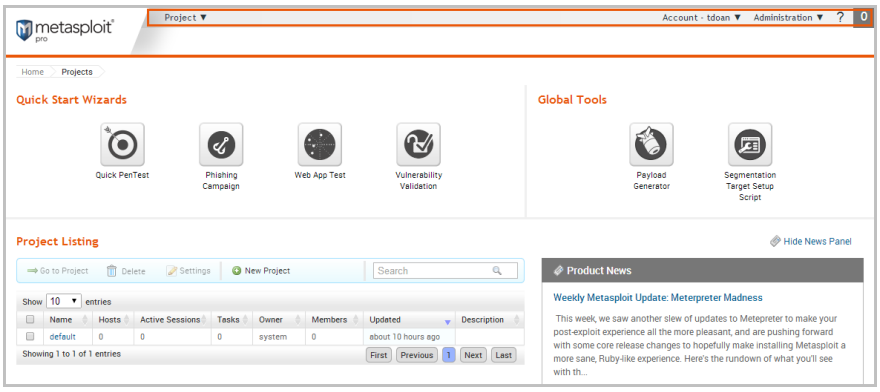


Regardless of where you are in the application, you can select **Project > Show All Projects** from the Global toolbar or click on the Metasploit Pro logo to access the Projects page, as shown below:



Global Toolbar

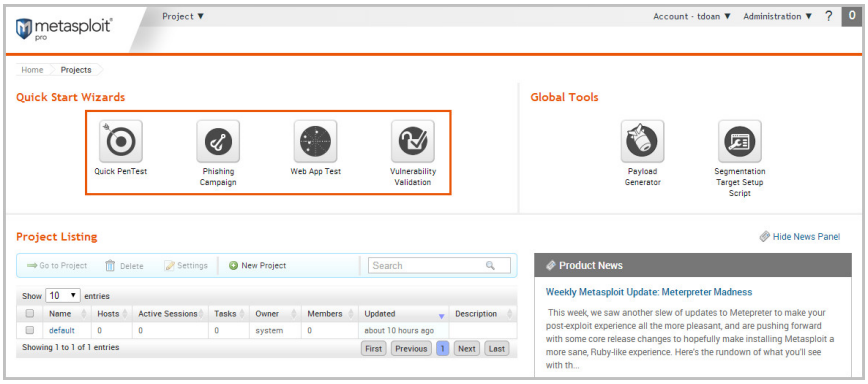
The Global toolbar is located at the top of web interface. This toolbar is available from anywhere in Metasploit Pro. You can use the Global toolbar to access the Projects menu, your account settings, and the Administration menu.



Quick Start Wizards

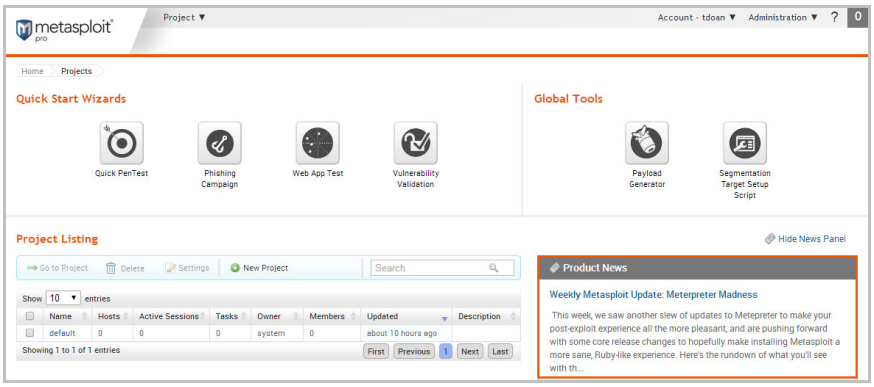
Each quick start wizard provides a guided interface that walks you through a common penetration testing task, such as scanning and exploiting a target, building social engineering campaigns, scanning and exploiting web applications, and validating vulnerabilities.

You can click on any of the quick start wizard icons to launch its guided interface.

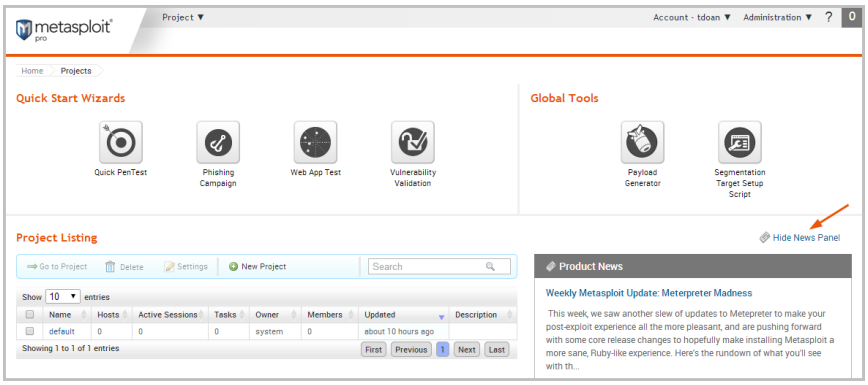


Product News

The Product News shows you the most recent blogs from Rapid7. If you want to keep up with the newest modules and security news from Rapid7 and the community, the Product News panel is a great place to check for the latest content.



If for some reason, you don't want to see the Product News panel, you can hide it so that it does not display on the Projects page.



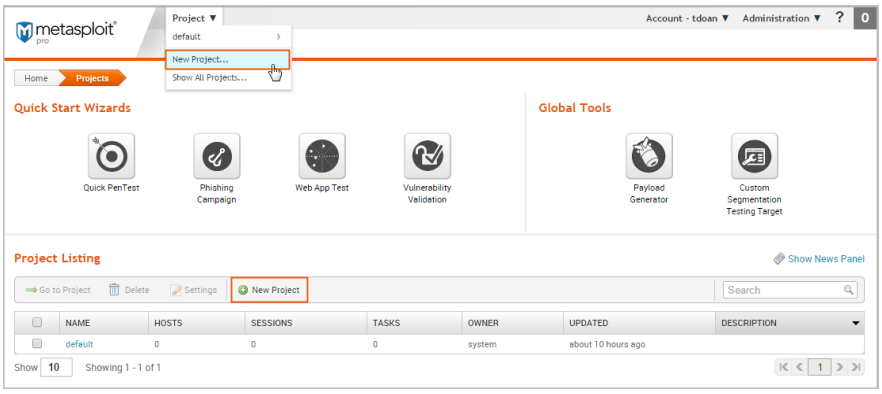
Creating a Project

Now that you're familiar with the Projects page, let's actually create a project.

A project contains the workspace, stores data, and enables you to separate an engagement into logical groupings. Oftentimes, you will have different requirements for the various subnets in an organization. Therefore, it may be efficient to have multiple 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 the organization.

Creating a project is easy. You can click on the **New Project** button on the Projects page or you can select **Project > New Project** from the global toolbar.



When the New Projects page appears, you only need to provide a project name. If you want to customize the project, you can also add a description, specify a network range, and assign user access levels.

Getting Target Data

The next thing you want to do is add data to your project. There are a couple of ways you can do this:

- Run a discovery scan
- Import data you already have

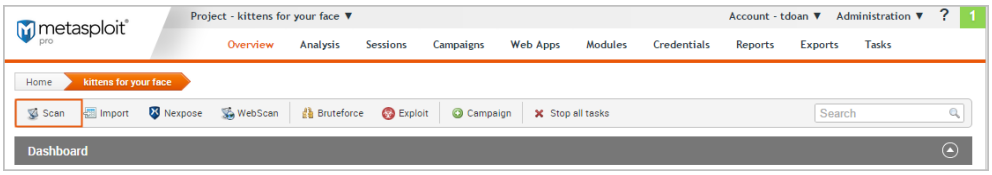
Scanning Targets

Scanning is the process of fingerprinting hosts and enumerating open ports to gain visibility into services running within a network. Scanning enables you to identify the active systems with services that you can communicate with so that you can build an effective attack plan. Metasploit has its own built-in discovery scanner that uses Nmap to perform basic TCP port scanning and gather additional information about the target hosts .

By default, the discovery scan includes a UDP scan, which sends UDP probes to the most commonly known UDP ports, such as NETBIOS, DHCP, DNS, and SNMP. The scan tests approximately 250 ports that are typically exposed for external services and are more commonly tested during a penetration test.

During a discovery scan, Metasploit Pro automatically stores the host data in the project. You can review the host data to obtain a better understanding of the topology of the network and to determine the best way to exploit each target. Oftentimes, the network topology provides insight into the types of applications and devices the target has in place. The more information that you can gather about a target, the more it will help you fine-tune a test for it.

Running a discovery scan is simple. From within a project, click the **Scan** button.



When the New Discovery Scan form appears, enter the hosts you want to scan in the **Target addresses** field. You can enter a single IP address, an IP range described with hyphens, or a standard CIDR notation. Each item needs to appear on a newline.

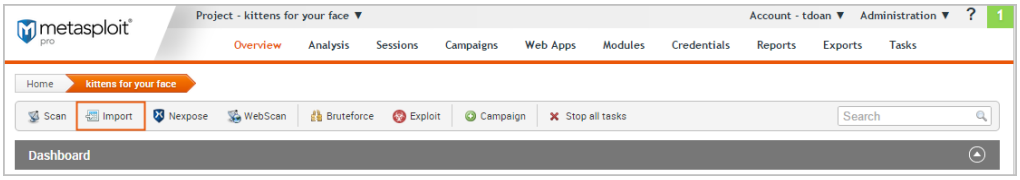
You can run the scan with just a target range; however, if you want to fine-tune the scan, you can configure the advanced options. For example, you can specify the hosts you want to exclude from the scan and set the scan speed from the advanced options. For more information on scanning, see the section on Discovery Scans.

Importing Data

If you are using a vulnerability scanner, you can import your vulnerability report into a Metasploit project for validation. The imported vulnerability data also includes the host metadata, which you can analyze to identify additional attack routes. Metasploit supports several third-party vulnerability scanners, including Nessus, Qualys, and Core Impact. For a full list of supported import types, see the section on Importing Data.

You can also export and import data from one Metasploit project into another. This enables you to share findings between projects and other team members.

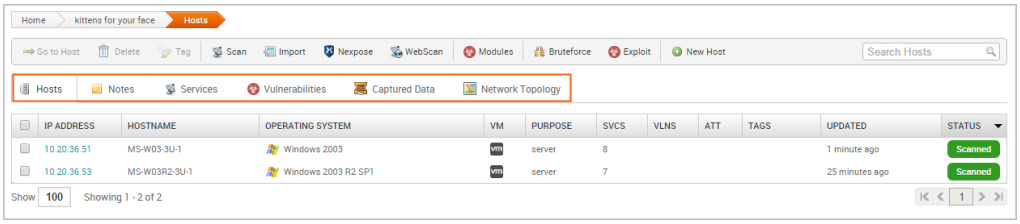
To import data into a project, click the **Import** button located in the Quick Tasks bar. When the Import Data page appears, select either the **Import from Nexpose** or **Import from File** option. Depending on the option you choose, the form displays the options you need to configure to import a file.



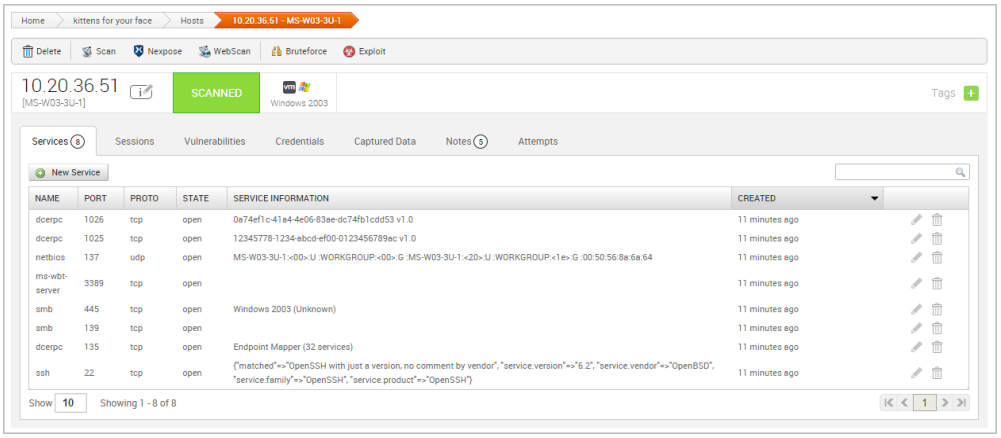
For example, if you choose to import from Nexpose, you will need to choose the console you want to use to run a scan or import a site. If you choose to import a file, you will need to browse to the location of the file.

Viewing and Managing Host Data

You can view host data at the project level or at the host level. At the project level, Metasploit provides a high-level view of all hosts that have been added to the project. To access the project view, select **Analysis > Hosts**. The project view initially shows the Hosts list, which displays the fingerprint and enumerated ports and services for each host. You can also view all the notes, services, vulnerabilities, and captured data for the project. To access these other views, click on their tabs from the project view.



To view the granular details for a host, you can click the host's IP address to access the single host view. This is a good way to drill down to see the vulnerabilities and credentials for a particular host.

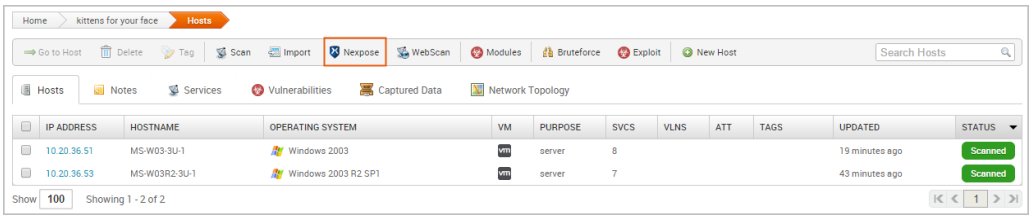


Running a Vulnerability Scan

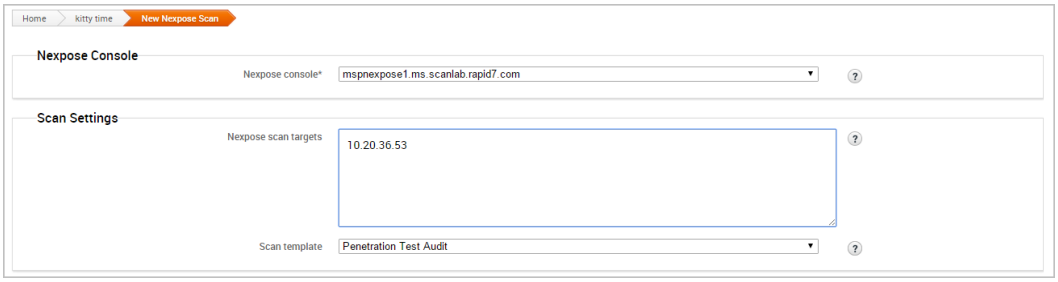
After you add target data to your project, you can run a vulnerability scan to pinpoint security flaws that can be exploited. Vulnerability scanners leverage vulnerability databases and checks to find known vulnerabilities and configuration errors that exist on the target machines. This information can help you identify potential attack vectors and build and attack plan that will enable you to compromise the targets during exploitation.

The integration with Nexpose enables you to launch a vulnerability scan directly from the Metasploit web interface. A Nexpose scan identifies the active services, open ports, and applications that run on each host and attempts to identify vulnerabilities that may exist based on the attributes of the known services and applications. Nexpose discloses the results in a scan report, which you can share with Metasploit for validation purposes.

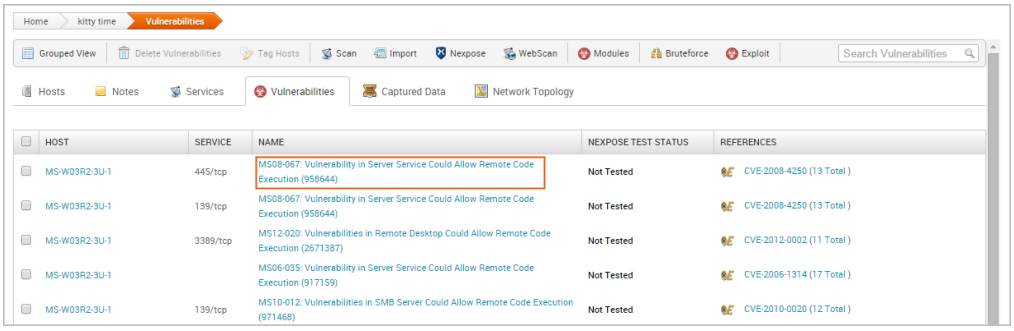
To run a Nexpose scan, click the **Nexpose** button located in the Quick Tasks bar.



When the Nexpose configuration form appears, you need to configure and select the console you want to use to perform the scan. Similarly to a discovery scan, you need to define the hosts you want to scan. You'll also need to choose one of the available scan templates, which defines the audit level that Nexpose uses. For more information on scan templates, check out the Nexpose User Guide.



To view all potential vulnerabilities that found by Nexpose, select **Analysis > Vulnerabilities**. You can click on the vulnerability name to view the modules that can be used to exploit the vulnerability.



This information becomes handy in the next phase of the pentest: exploitation.

! Vulnerability scanners are useful tools that can help you quickly find potential security flaws on a target. However, there are times when you may want to avoid detection and limit the amount of noise you create. In these cases, you may want to run some auxiliary modules, such as the FTP, SMB, and VNC login scanners, to manually identify potential vulnerabilities that can be exploited. Manual vulnerability analysis is considerably more time consuming and requires research, critical thinking, and in-depth knowledge on your part, but it can help you create an accurate and effective attack plan.

Finding and Exploiting Vulnerabilities the Easy Way

The easiest way to scan and check for vulnerabilities is through the Vulnerability Validation Wizard, which automates the validation process for Nexpose and Metasploit Pro users. The wizard provides a guided interface that walks you through each step of the validation process–from importing Nexpose data to auto-exploiting vulnerabilities to sending the validation results back to Nexpose. For more information on vulnerability validation, see the section Validating a Vulnerability.

If you don't have access to Nexpose and/or Metasploit Pro, the validation process requires manual analysis of the vulnerabilities. Manual validation requires a bit more legwork, but provides much more control over the vulnerabilities that are targeted.

Exploiting Known Vulnerabilities

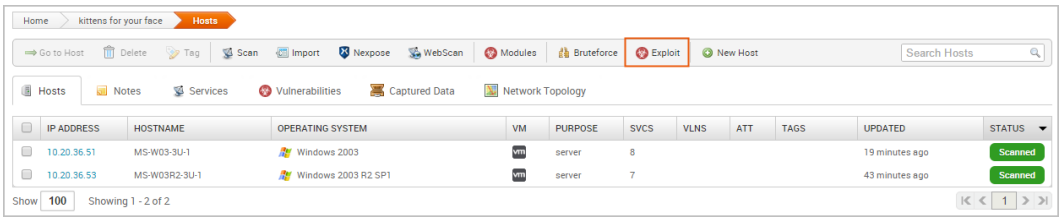
After you have gathered information about your targets and identified potential vulnerabilities, you can move to the exploitation phase. Exploitation is simply the process of running exploits against the discovered vulnerabilities. Successful exploit attempts provide access to the target systems so you can do things like steal password hashes and download configuration files. They also enable you to identify and validate the risk that a vulnerability presents.

Metasploit offers a couple different methods you can use to perform exploitation.

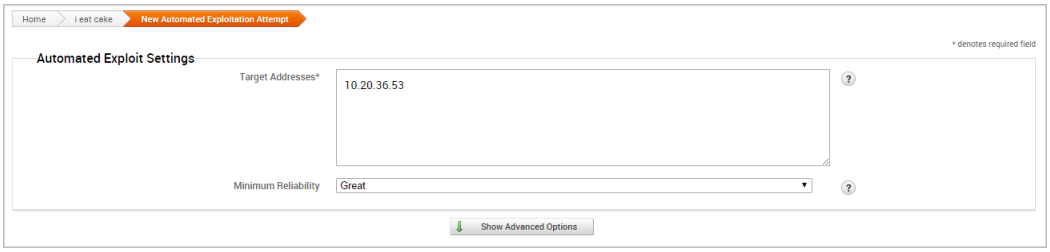
Automatic Exploitation

Metasploit automatically cross-references open services, vulnerability references, and fingerprint data to find matching exploits. All matching exploits are added to an attack plan, which basically contains all the exploits that are can be run. The simple goal of auto-exploitation is to get a session as quickly as possible by leveraging the data that Metasploit has for the target hosts.

To run auto-exploitation, click the **Exploit** button located in the Quick Tasks bar.



At a minimum, you'll need to provide the hosts you want to exploit and the minimum reliability for each exploit. The minimum reliability can be set to guarantee the safety of the exploits that are launched. The higher the reliability level, the less likely the exploits used will crash services or negatively impact a target.



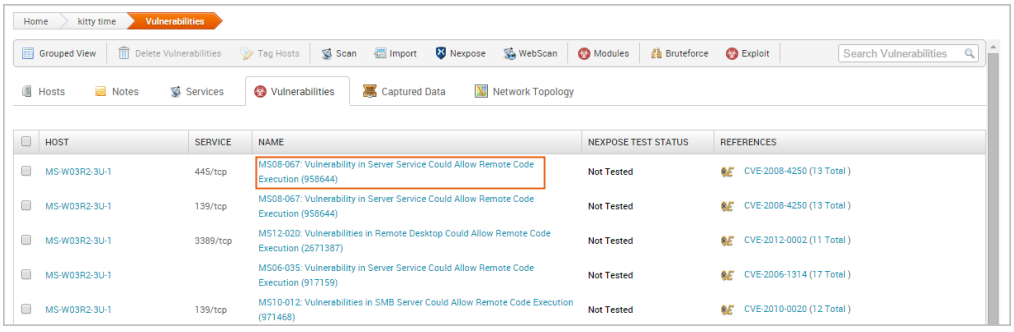
You can also configure advanced options to define payload options and exploit selection settings.

Manual Exploitation

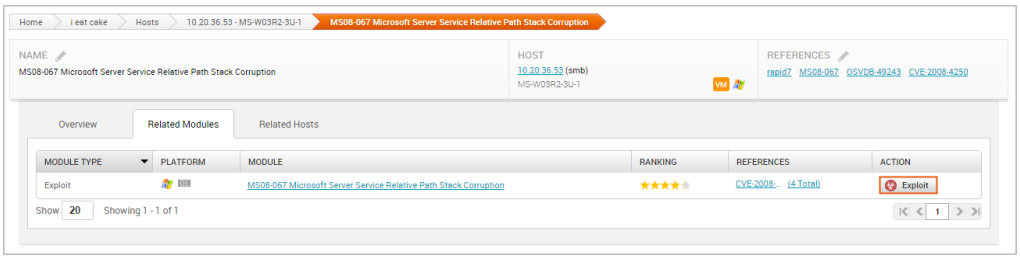
Manual exploitation provides a more targeted and methodical approach to exploiting vulnerabilities. It enables you to run individual exploits one at a time. This method is particularly useful if there is a specific vulnerability that you want to exploit. For example, if you know that the SMB server on a Windows XP target does not have the MS08-067 patch, you may want to try to run the corresponding module against it.

To search for modules, select **Modules > Search** and enter the name of the module you want to run. The best way to find an exact module match is to search by vulnerability reference. For example, if you want to search for ms08-067, you can either search for 'ms08-067'. You can also search by the module path: `exploit/windows/smb/ms08_067_netapi`.

One of the easiest ways to find an exploit for a vulnerability is directly from the vulnerability page. To view all vulnerabilities in the project, select **Analysis > Vulnerabilities**. You can click on the vulnerability name to view the related modules that can be used to exploit the vulnerability.



The single vulnerability view shows a list of the exploits that can be run against the host. You can click the **Exploit** button to open the configuration page for the module.



Configuring Common Exploit Module Settings

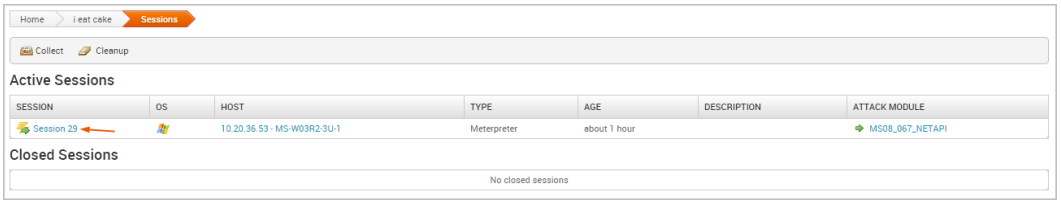
Each module has its own set of options that can be customized to your needs. Here are some options that are commonly used to configure modules:

- Payload Type: Specifies the type of payload that the exploit will deliver to the target. Choose one of the following payload types:
  - **Command**: A command execution payload that enables you to execute commands on the remote machine.
  - **Meterpreter**: An advanced payload that provides a command line that enables you to deliver commands and inject extensions on the fly.
- Connection Type: Specifies how you want your Metasploit instance to connect to the target. Choose one of the following connection types:
  - **Auto**: Automatically uses a bind connection when NAT is detected; otherwise, a reverse connection is used.
  - **Bind**: Uses a bind connection, which is useful when the targets are behind a firewall or a NAT gateway.
  - **Reverse**: Uses a reverse connection, which is useful if your system is unable to initiate connections to the targets.
- LHOST: Defines the address for the local host.
- LPORT: Defines the ports that you want to use for reverse connections.
- RHOST: Defines the target address.
- RPORT: Defines the remote port you want to attack.
- Target Settings: Specifies the target operating system and version.
- Exploit Timeout: Defines the timeout for the module in minutes.

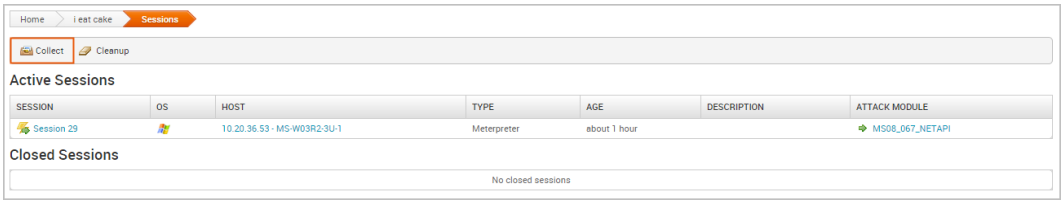
Post-Exploitation and Collecting Evidence

Any exploit that successfully takes advantage of a vulnerability results in an open session you can use to extract information from a target.The real value of the attack depends on the data that you can collect from the target, such as password hashes, system files, and screenshots and how you can leverage that data to gain access to additional systems.

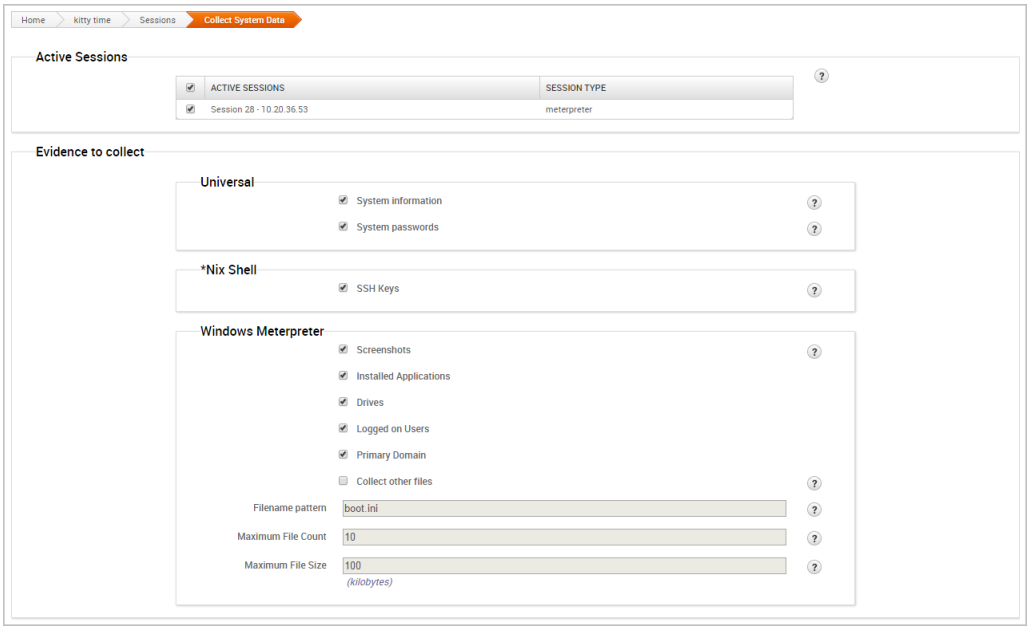
To view a list of open sessions, select the **Sessions** tab. Click on the session ID to view the post-exploitation tasks that can be run against the host.



To collect evidence from an exploited system, click the **Collect** button.



A list of all open sessions displays and shows you the type of evidence that can be collected.



Generating a Report

At the end of the pentest, you'll want to create a deliverable that contains the results of your pentest. Metasploit provides a number of reports that you can use to compile test results and consolidate data into a distributable and tangible format. Each report organizes your findings into relevant sections, displays charts and graphs for statistical data, and summarizes major findings. For more information, check out the section on reports.

Creating and Managing Projects

A project contains the workspace that you use to perform the different steps for a penetration test and stores the data that you collect from the target. You create a project to configure tasks and to run tests. You can create as many projects as you need and switch between projects while tasks are in progress.

Every project has an owner. The owner can choose the users who can access the project to edit, view, and run tasks. However, users with administrative access can view and edit any project, regardless of whether or not the project owner gives them access.

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.

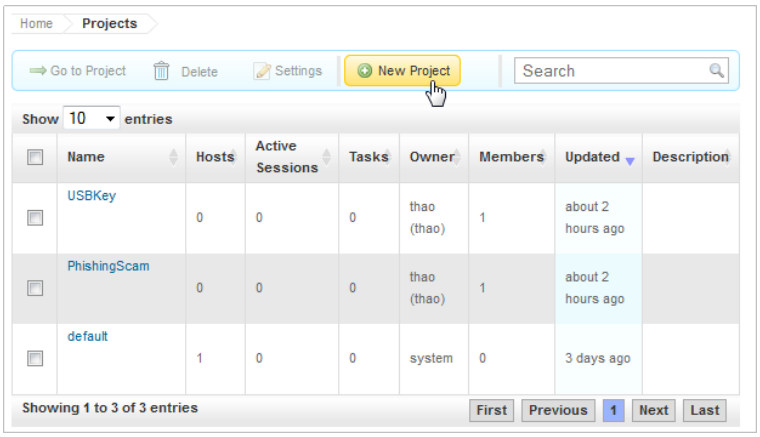
Creating a Project

A project is the workspace that you use to build a penetration test. Each project logically groups together the hosts that you want to exploit and the type of information that you want to obtain. Every project has the following information:

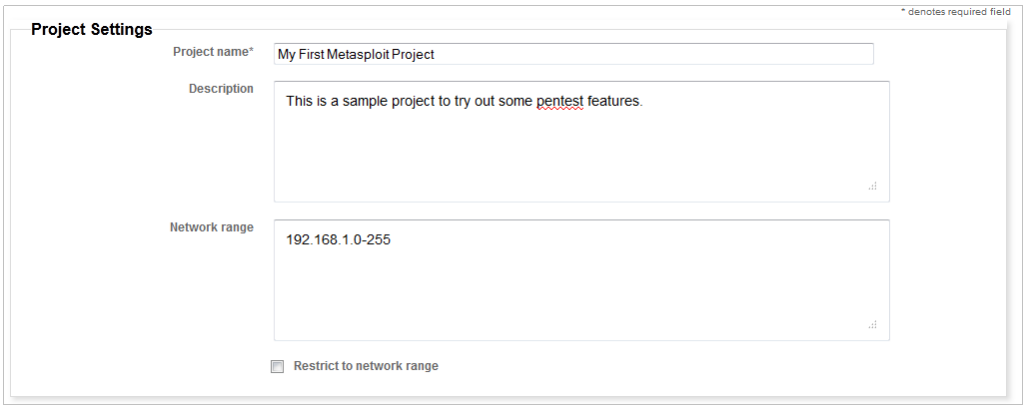
- Name: Provides a unique identifier for the project.
- Description: Describes the purpose and scope of the project.
- Network range: Defines the default network range for the project. When you create a project, Metasploit Pro automatically populates the default target range with the network range that you define for the project. Metasploit Pro 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 Pro will not run tasks against a target whose address does not fall within the network range.

To create a project:

1. From the Projects page, click the **New Project** button.



2. When the New Project page appears, find the **Project Settings** area, and enter the project name, description, and network range:



3. Select the **Restrict to network range** option if you want to enforce network boundaries on the project.
4. From the **User Access** area, select the following information:

   - **Project owner** - The person who owns the project.
   - **Project members** - The users who can access, edit, and perform tasks in the project.

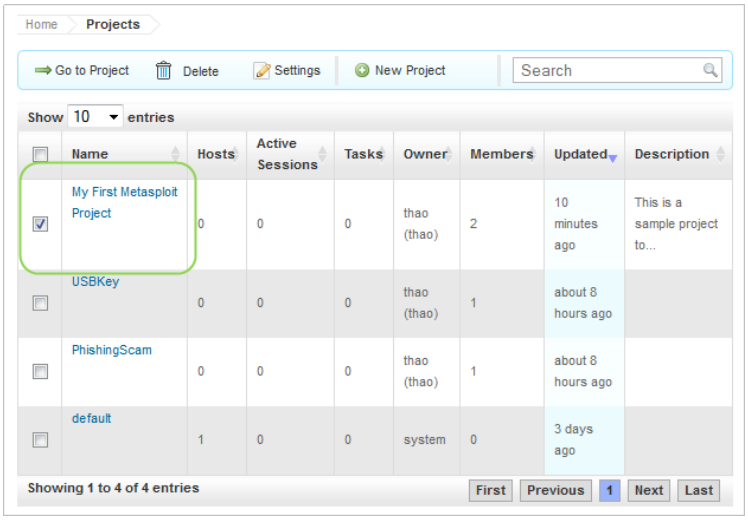
5. Create the project.

Deleting a Project

When you delete a project, you remove all the data that the project contains, including reports, host data, evidence, vulnerability data, and host tags. After you delete a project, you cannot view or access the project again.

If you want to delete the project, but save the project data, you can export the project data. When you export the project data, the system provides you with an XML or ZIP file of the project contents. You can import the XML or ZIP file to bring the project data back into Metasploit Pro.

1. Select **Project > Show All Projects** from the Main menu.
2. When the Projects page appears, select the projects that you want to delete.



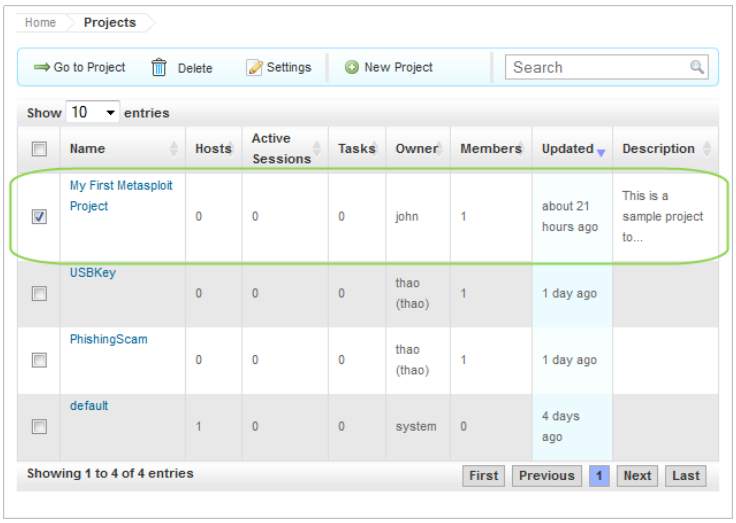
3. Click **Delete**.
4. When the confirmation window appears, click **OK** to delete the project.

Setting the Network Range

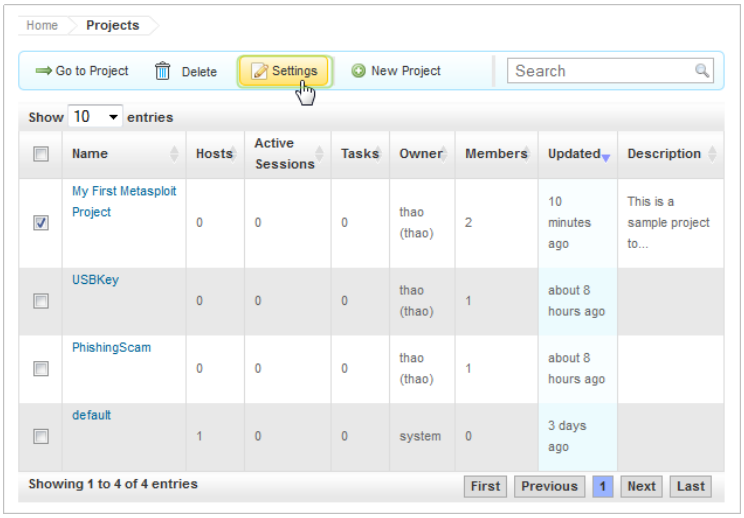
When you create a project, you can define an optional network range that sets the scope of the project. The network range defines the addresses that Metasploit Pro uses to automatically populate the target addresses for discovery scans and Nexpose scans. It also defines network boundaries that Metasploit Pro can enforce for the project.

You do not need to set the network range unless you want to enforce network boundaries. If you choose to enforce network boundaries on a project, Metasploit Pro uses the network range that you define for the project.

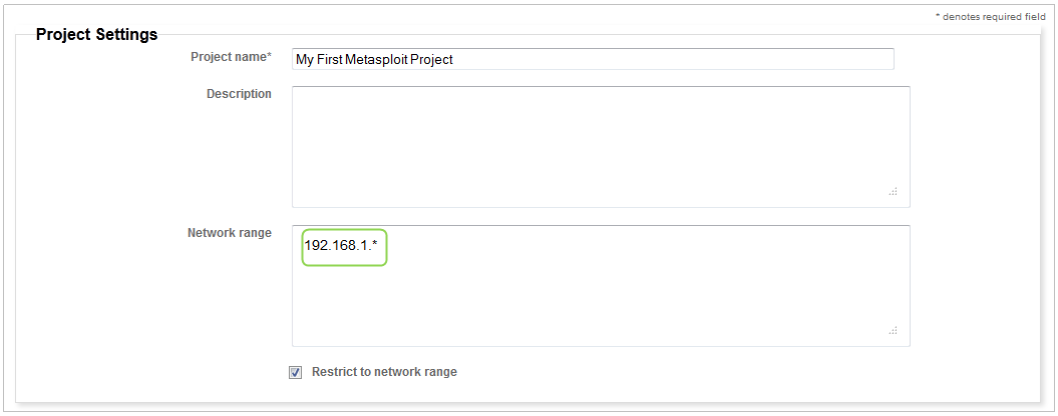
1. From within a project, select **Project >Show All Projects** from the Main menu.
2. Select the project that you want to set the network range for.



3. Click the **Settings** button.



4. In the **Network range** field, enter the network range that you want to restrict the project to. You can enter a single IP address, an IP range described with hypens, or a standard CIDR notation. If you define a CIDR notation, you can use an asterisk as a wild card. For example 192.168.1.* indicates 192.168.1.1-255.



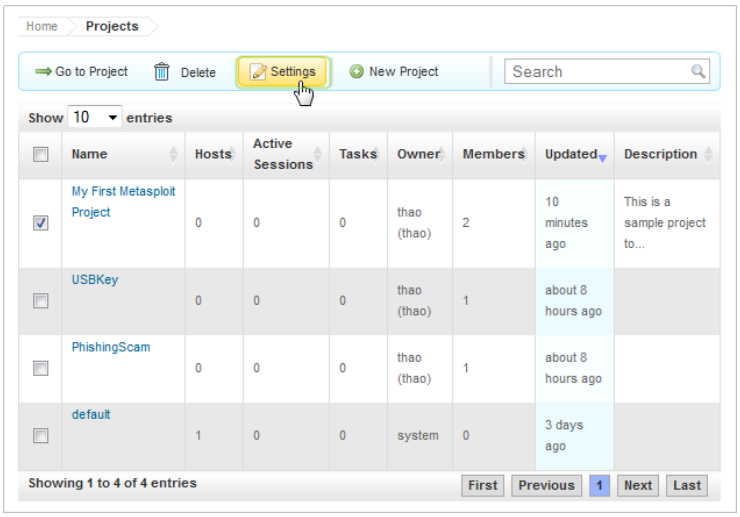
5. Click the **Update Project** button.

Restricting a Project to a Network Range

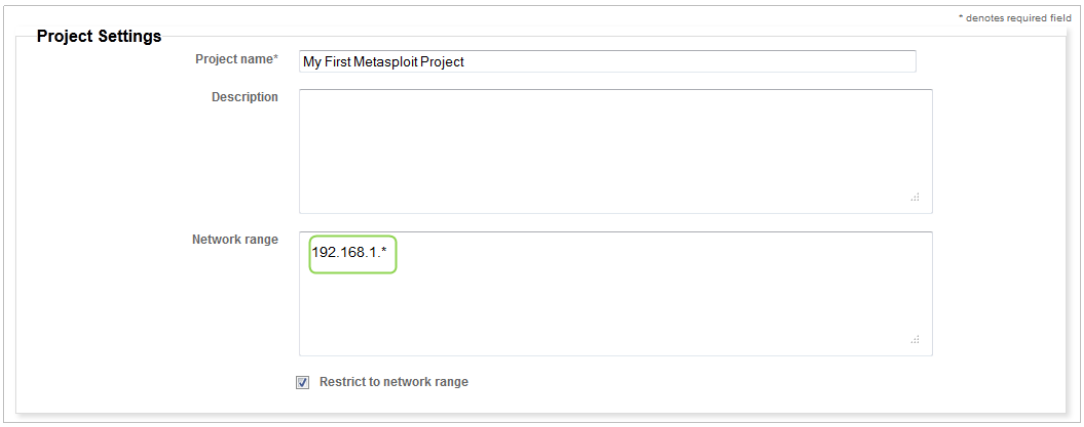
You can restrict the network range to enforce network boundaries on a project. When you restrict a project to a network range, you cannot run any tasks unless the target addresses fall within network range that you define.

For example, if you have a client who wants you to test a specific network range, you can set the network range and restrict the project to it to ensure that you do not accidentally target any devices that are outside of that range.

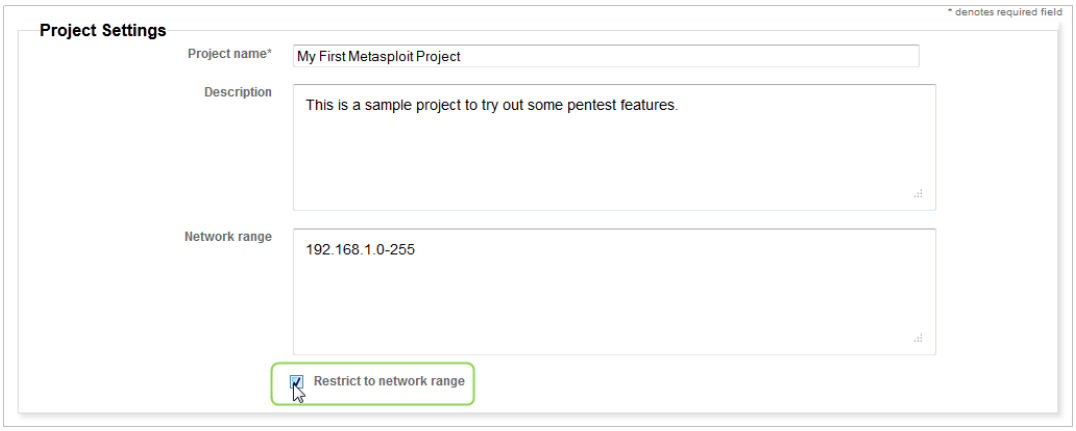
1. Select **Project > Show All Projects** from the Main menu.
2. Select a project and click the **Settings** button.



3. In the **Network range** field, enter the network range that you want to restrict the project to. You can enter a single IP address, an IP range described with hyphens, or a standard CIDR notation. If you define a CIDR notation, you can use an asterisk as a wild card. For example 192.168.184.* indicates 192.168.184.1-255.



4. Select the **Restrict to Network Range** option.



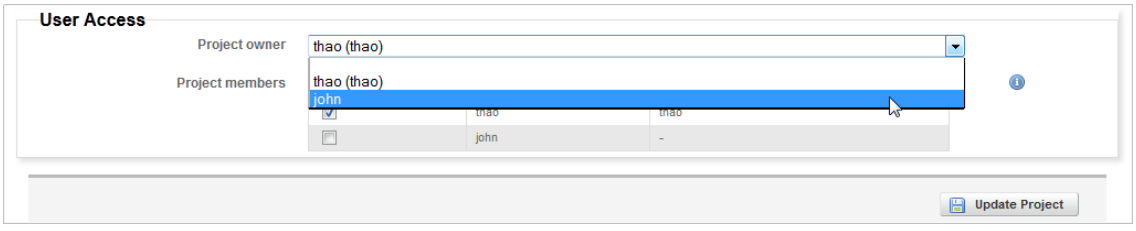
5. Click the **Update Project** button.

Changing the Project Owner

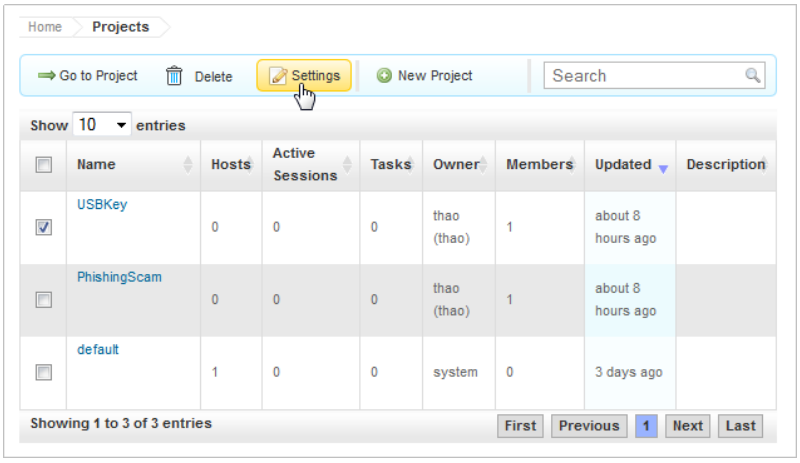
By default, the project owner is the person who initially sets up the project. You can change the project owner to transfer ownership and to assign projects to team members.

The project owner provides a way for you and your team members to easily identify the projects that each of you own. For example, if you want to see the projects that you have been assigned, you can sort the project list by owner. All of your projects will be grouped together.

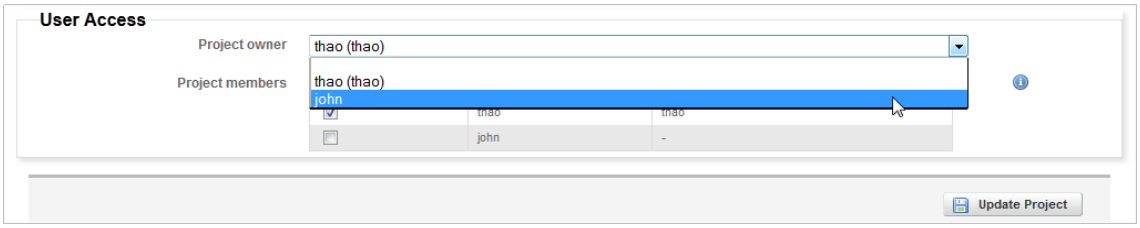
1. From the Main menu, select **Project > Show All Projects**.



2. When the Projects page appears, select the project that you want to assign an owner.
3. Click the **Settings** button.



4. When the Project Settings page appears, find the User Access area.
5. Click the **Project owner** dropdown and select the person you want to assign the project to.



6. Click the **Update Project** button.

Managing User Access

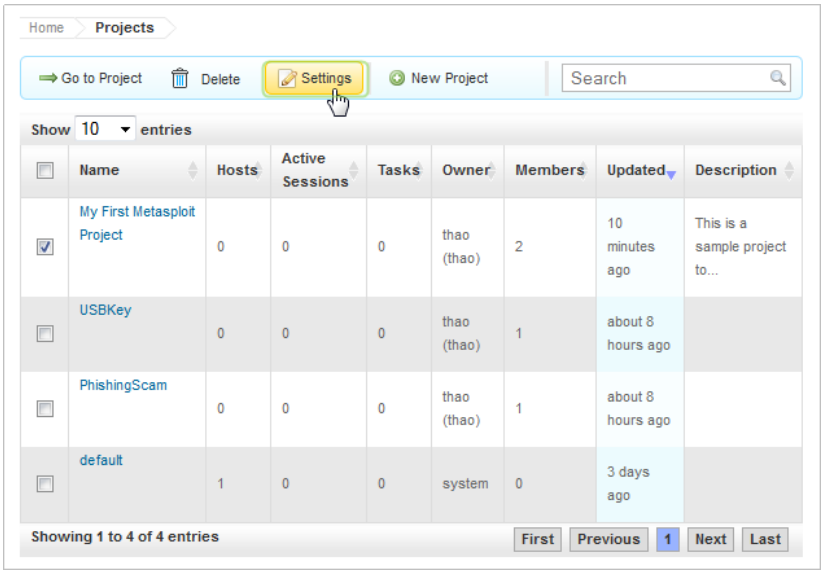
As the project owner, you may want to restrict the team members who can view and edit your project. For example, if you have data that you do not want anyone to overwrite, you can disable the access rights for other team members.

> **!** Team members that have administrative rights can view and modify all projects, regardless of the user access settings.

To manage the access that a user has to a project:

1. From the Main menu, select **Project > Show All Projects**.

2. When the Projects page appears, select the project that you want to edit.
3. Click the **Settings** button.



4. When the Project Settings page appears, find the User Access area.
5. Select project members to enable them to view and modify the project or deselect project members to prevent them from modifying the project.

6. Click the **Update Project** button.

Team Collaboration

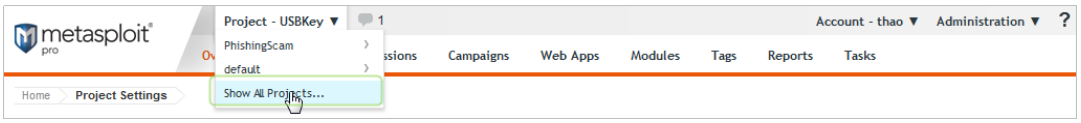
The multi-user support provides you with the ability to collaborate on an engagement or penetration test with other team members. You and your team can log into the same instance of Metasploit Pro to perform tasks, review data, and share projects. You can access Metasploit Pro through the Metasploit Web UI, which can run on the local machine or across the network.

Some features that you can implement to enhance team collaboration are network boundaries, host tags, and host comments. These features help you create separate workloads for each team member and organize an engagement into logical containers. For example, you may want to assign certain hosts to a specific team member to test.

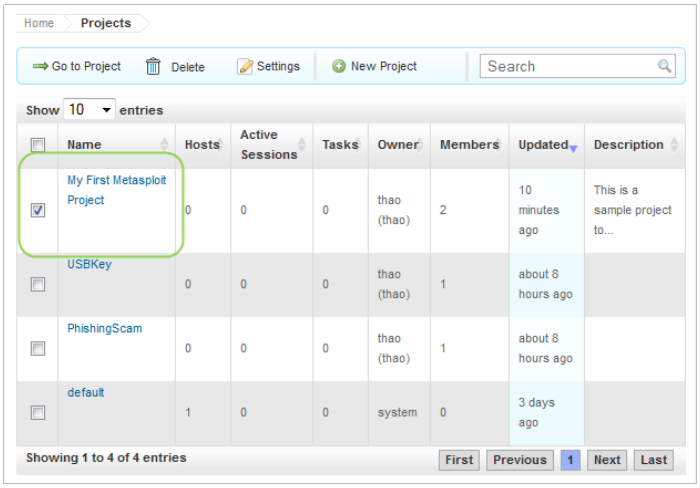
Adding Users to a Project

You can give team members access to a project so that they can view, edit, and run tasks from the project.

1. From the Main menu, select **Project > Show All Projects**.



2. Select the project that you want to add users to.



3. Find the **User Access** settings. The User Access list displays all Metasploit Pro users.
4. Click the **Settings** button.

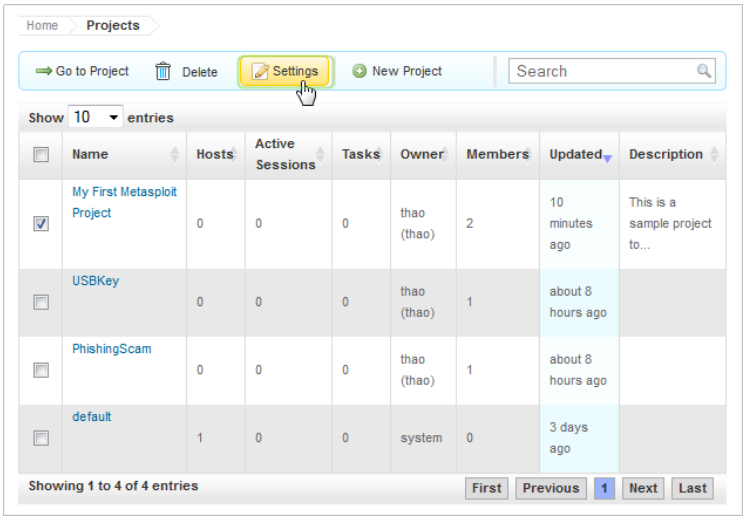
5. Select the users that you want to have access to the project.

6. Click the **Update Project** button.

Removing Users from a Project

You can remove members from a project to restrict their ability to view, change, or run tasks from the project. When you remove a user from a project, you disable their access to the project.

1. From within a project, select **Project > Project Settings**.



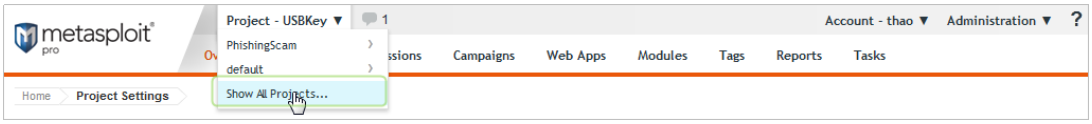
2. Find the **User Access** settings. The user access list displays all available Metasploit Pro users.
3. Deselect the users that you do not want to have access to the project.

4. Click the **Update Project** button.

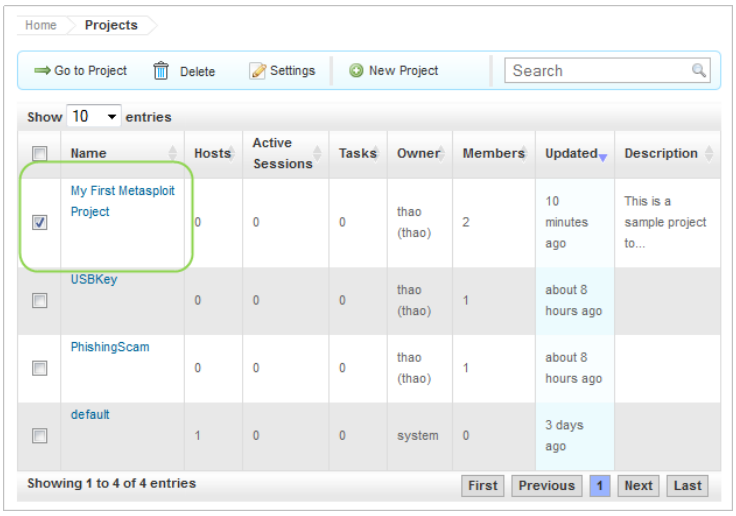
Assigning the Project to a User

The project owner is the person who sets up the project and assumes responsibility for the data and penetration test. You can use the project owner role to delegate projects or workloads to members on your team.

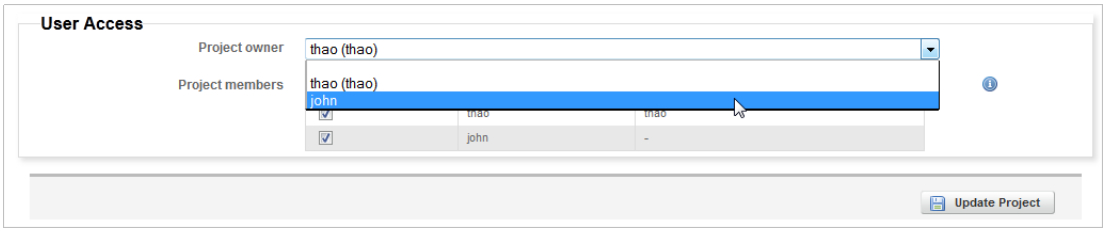
1. From the Main Menu, select **Project > Show All Projects**.



2. Select the project that you want to assign to a user.



3. Click the **Settings** button.
4. Find the User Access settings. The User Access list displays all available Metasploit Pro users.
5. From the **Project Owner** dropdown menu, choose an owner for the project.



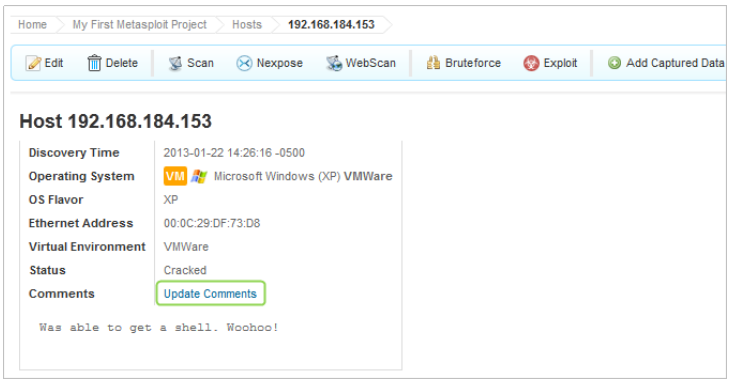
6. Click the **Update Project** button.

Host Comments

You can add a host comment to share information about a host. For example, if you identify a vulnerability on a host, and you want to share that information with other project users, you can add a host comment to that host. When you view the host details, you can see comments that other users have added to the host.

Adding Host Comments

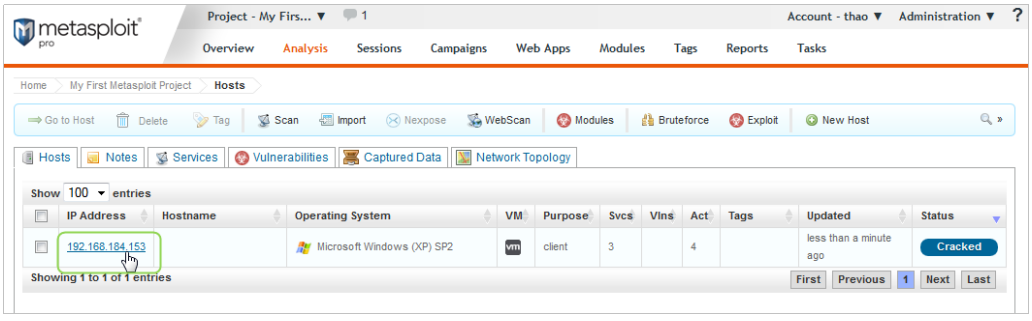
1. From within a project, select **Analysis > Hosts**.
2. Click on the name of the host to which you want to add a comment.
3. When the Host Details page appears, click the **Update Comment** button.



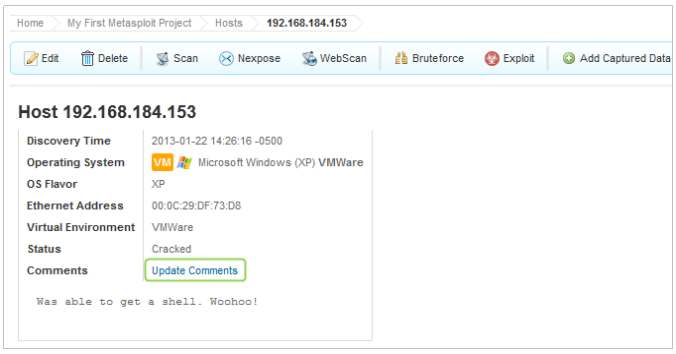
4. Enter the information you want to add to the host in the **Comments** field. For example, if you know that a host is not exploitable, you can add the information as a comment. When other team members see the note, they know that they should not attempt to exploit the host.
5. Click the **Save Comments** button.

Updating Host Comments

1. From within a project, select **Analysis > Hosts**.
2. Click on the name of the host to which you want to add a comment.



3. When the Host Details page appears, click the **Update Comment** button.



4. Edit the information in the **Comments** field.

5. Click the **Save Comments** button.

Managing Accounts

A user account provides you and your team members with access to Metasploit Pro. You use a user account to log into Metasploit Pro and to create identifies for other members on the team.

A user account consists of a login name, the user's full name, a password, and a role. Use the following components to set up a user account:

- *Login name* - The user name that the system uses to uniquely identify a person.
- *Full name* - The first and last name for the person who owns the user account.
- *Password* - An eight character string that allows access to the use account.
- *Role* - The level of access that the user has to Metasploit Pro and other projects. The role can be an administrator or basic user.

Account Types

A user account can be a non-administrator account or an administrator account. The account type determines the level of privileges that a user must have to perform certain tasks. For example, administrators have unrestricted access to the system so they can perform system updates, manage user accounts, and configure system settings. Non-administrator accounts, on the other hand, have access to Metasploit Pro, but can only perform a limited set of tasks.

Administrator Account

An administrator account has unrestricted access to all Metasploit Pro features. With an administrator account, you can do things like remove and add user accounts, update Metasploit Pro, and access all projects.

Non-Administrator Account

A non-administrator account gives a user access to Metasploit Pro, but does not provide them with unlimited control over projects and system settings.This account restricts the user to the projects that they have access to and the projects that they own.

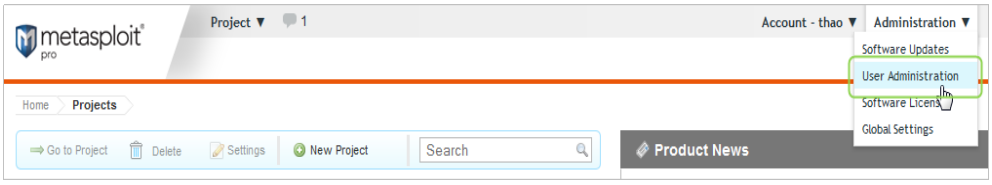
A non-administrator account cannot perform the following tasks:

- Create or manage other user accounts.
- Configure global settings for Metasploit Pro.

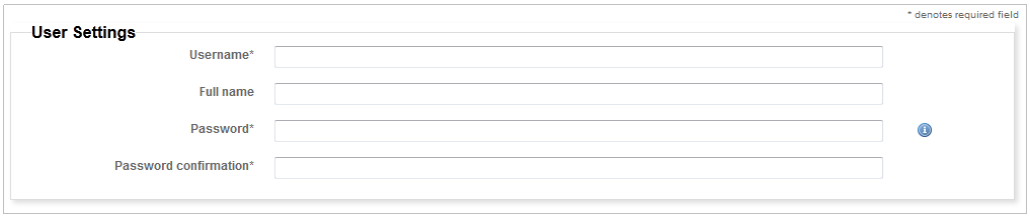
- Update Metasploit Pro.
- Update the license key.
- View projects that they do have access to.

Creating a User Account

1. Click **Administrator > User Administration** from the main menu.

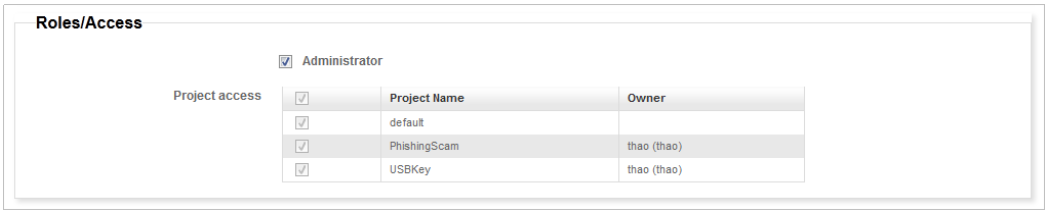


2. When the User Administration page appears, click the **New User** button.
3. When the New User page appears, fill out the following information to create a user account:



- **User name** - Enter a user ID for the account.
- **Full name** - Enter the user's first and last name.
- **Password** - Use mixed case, punctuation, numbers, and at least eight characters to create a strong password.
- **Password confirmation** - Re-enter the password.

4. Select the **Administrator** option if you want to provide the account with administrative rights. If the account has administrative privileges, the user has unrestricted access to all areas of Metasploit Pro. If the account does not have administrative rights, the user can only work with projects that they have access to and cannot update the system.
5. If the account does not have administrative rights, click the **Show Advanced Options** button to choose the projects that the user can access.



6. Save the changes to the user account.

Account Requirements

All accounts must meet the user name and password requirements. If the user name or password does not meet one of the following criteria, Metasploit Pro displays an error until you input a user name and password that complies with every requirement.

User Name Requirements

A user name can contain any combination of the following characters:

- Alphanumeric characters
- Spaces
- Non-alphanumeric characters (!@#$%^&*()+,.?/<>)

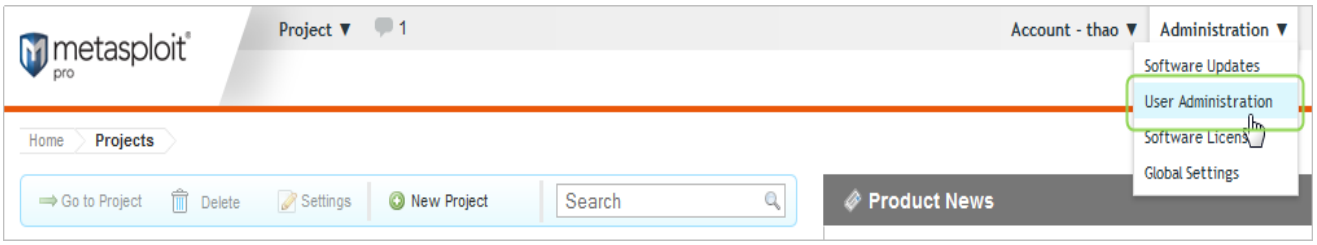
Password Requirements

A password must meet the following criteria:

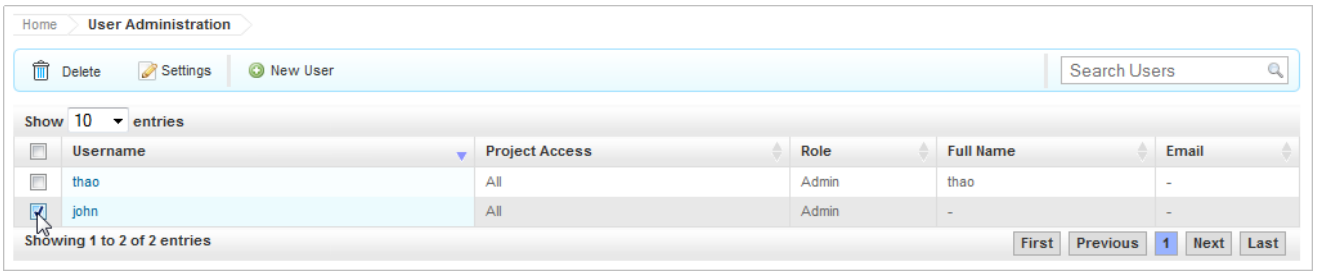
- Contains letters, numbers, and at least one special character.
- Contain at least eight characters.
- Cannot contain the user name.
- Cannot be a common password.
- Cannot use a predictable sequence of characters

Changing an Account Password

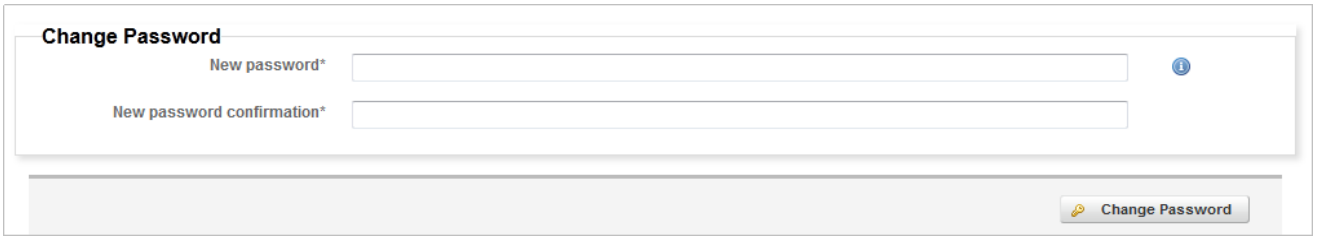
1. Choose **Administration > User Administration** from the main menu.



2. Select the user account that you want to modify.



3. Click the **Settings** button.
4. Find the Change Password area.
5. In the **New Password** field, enter a password for the account. The password must contain at least eight characters and consist of letters, numbers, and at least one special character.

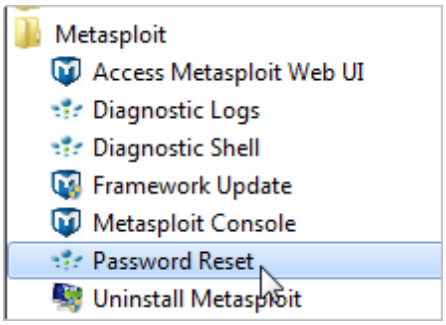


6. Reenter the password in the **Password Confirmation** field.
7. Click the **Change Password** button.

If you have forgotten your password or need reset your password, follow the instructions for your operating system.

Windows

8. From the Start menu, choose **All Programs > Metasploit > Password Reset**.
9. When the **Password Reset** window appears, wait for the environment to load.



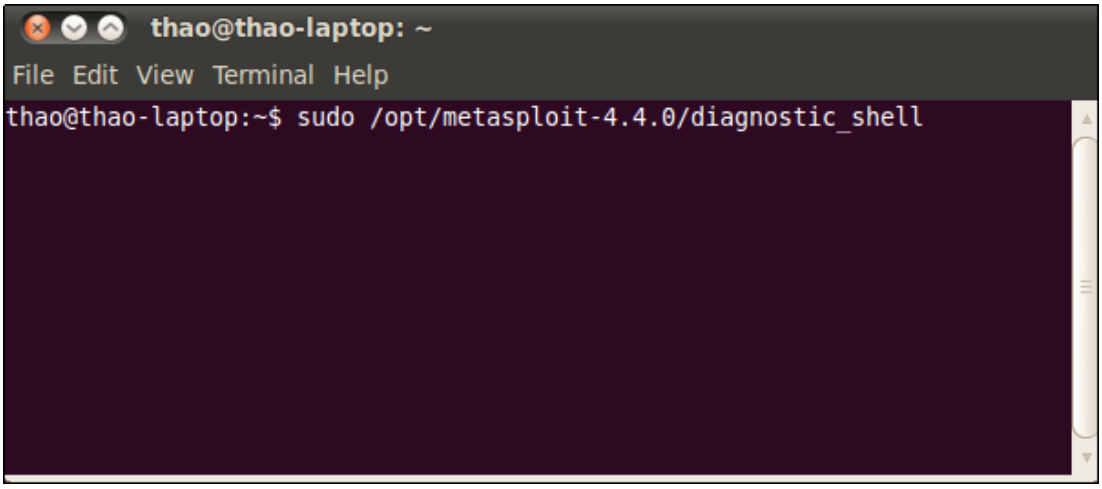
10. When the dialog prompts you to continue, enter `yes`. The system resets the password to a random value.
11. Copy the password and use the password the next time you log in to Metasploit Pro.

    You can change the password after you log in to Metasploit Pro.

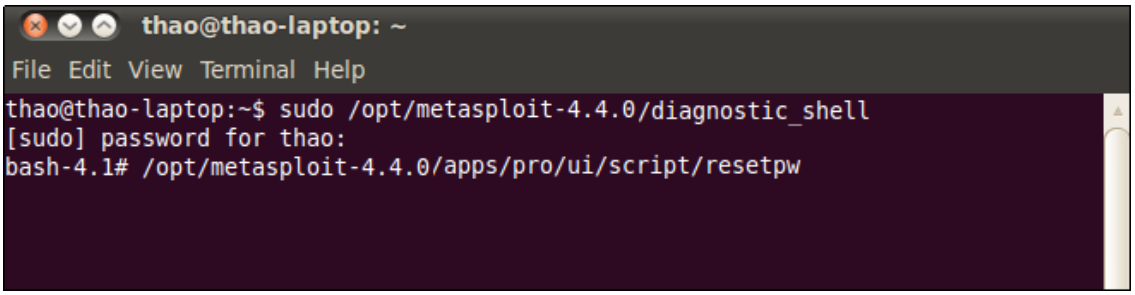
12. Exit the **Password Reset** window.

Linux

1. Open the command line terminal and execute the following command: `sudo </path/to/metasploit>/diagnostic_shell`.



2. If prompted, enter your sudo password.
3. When the system returns the `bash#` prompt, enter `</path/to/metasploit>/apps/pro/ui/script/resetpw` to run the resetpw script.



4. Copy the password and use the password the next time you log into Metasploit Pro.

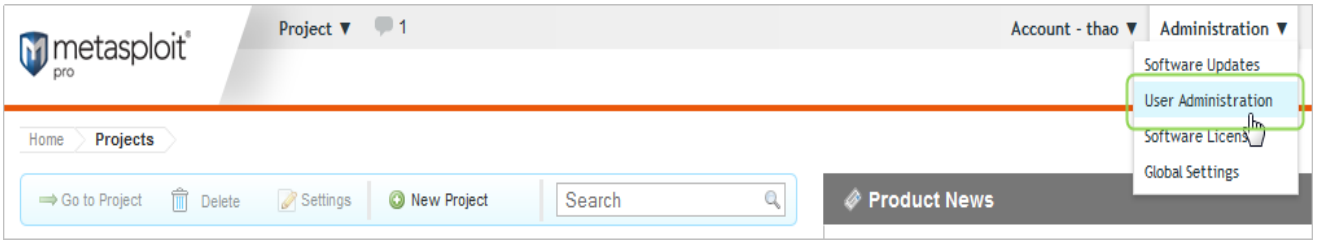
   You can change the password after you log in to Metasploit Pro.

5. Exit the console.

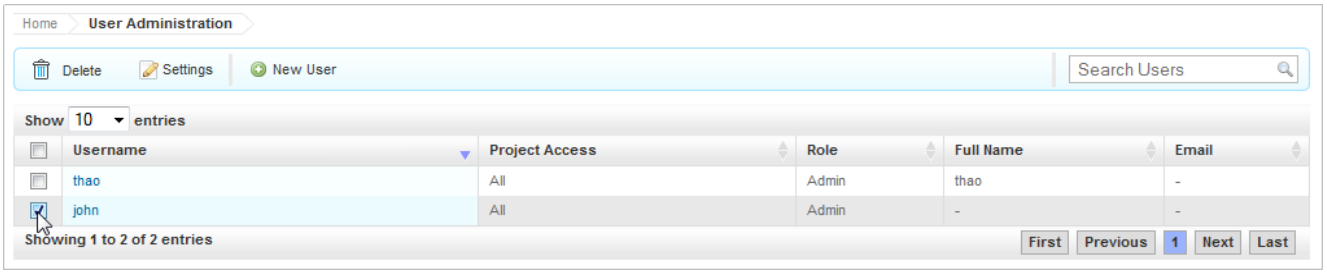
Deleting a User Account

If you have an administrator account, you can delete user accounts that you no longer need. When you delete a user account, the system reassigns the projects that belong to the account to the system. Any project that does not have a project owner will have system listed as the project owner.

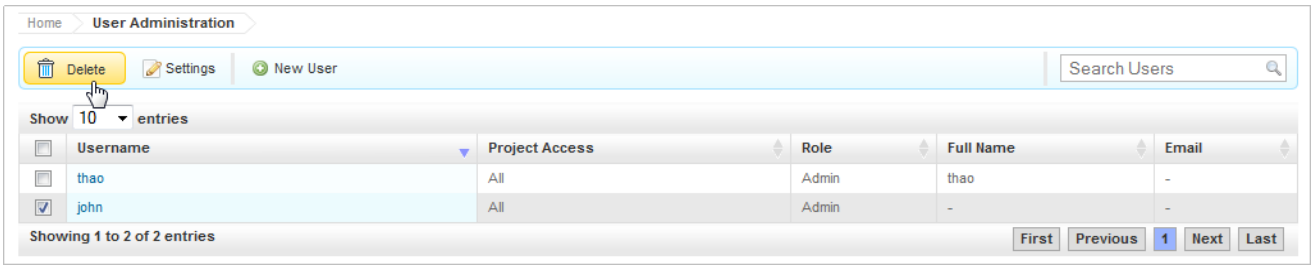
1. Choose **Administration > User Administration** from the main menu.



2. Select the user account that you want to delete.



3. Click **Delete**.



4. Click **OK** to confirm that you want to delete the account.

Discovery Scan

One of the first steps in penetration testing is reconnaissance. Reconnaissance is the process of gathering information to obtain a better understanding of a network. It enables you to create list of target IP addresses and devise a plan of attack. Once you have a list of IP addresses, you can run a discovery scan to learn more about those hosts. A discovery scan identifies the operating systems that are running on a network, maps those systems to IP addresses, and enumerates the open ports and services on those systems.

A discovery scan is the internal Metasploit scanner. It uses Nmap to perform basic TCP port scanning and runs additional scanner modules to gather more information about the target hosts. By default, the discovery scan includes a UDP scan, which sends UDP probes to the most commonly known UDP ports, such as NETBIOS, DHCP, DNS, and SNMP. The discovery scan tests approximately 250 ports that are typically exposed for external services and are more commonly tested during a penetration test.

During a discovery scan, Metasploit Pro automatically adds the host data to the project. You can review the host data to obtain a better understanding of the topology of the network and to determine the best way to exploit each target. Oftentimes, the network topology provides insight into the types of applications and devices the target has in place. The more information that you can gather about a target, the more it will help you fine-tune a test for it.

How a Discovery Scan Works

A discovery scan can be divided into four distinct phases:

- Ping scan
- Port scan
- OS and version detection
- Data import

Ping Scan

The first phase of a discovery scan, ping scanning, determines if the hosts are online. The discovery scan sets the -PI option, which tells Nmap to perform a standard ICMP ping sweep. A single ICMP echo request is sent to the target. If there is an ICMP echo reply, the host is considered ‘up’ or online. If a host is online, the discovery scan includes the host in the port scan.

Port Scan

During the second phase, port scanning, Metasploit Pro runs Nmap to identify the ports that are open and the services are available on those ports. Nmap sends probes to various ports and classifies the responses to determine the current state of the port. The scan covers a wide variety of commonly exposed ports, such as HTTP, telnet, SSH, and FTP.

The discovery scan uses the default Nmap settings, but you can add custom Nmap options to customize the Nmap scan. For example, the discovery scan runs a TCP SYN scan by default. If you want to run a TCP Connect Scan instead of a TCP SYN Scan, you can supply the -sT option. Any options that you specify override the default Nmap settings that the discovery scan uses.

OS and Version Detection

After the discovery scan identifies the open ports, the third phase begins. Nmap sends a variety of probes to the open ports and detects the service version numbers and operating system based on how the system responds to the probes. The operating system and version numbers provide valuable information about the system and help you identify a possible vulnerability and eliminate false positives.

Data Import

Finally, after Nmap collects all the data and creates a report, Metasploit Pro imports the data into the project. Metasploit Pro uses the service information to send additional modules that target the discovered services and to probe the target for more data. For example, if the discovery scan sweeps a target with telnet probes, the target system may return a login prompt. A login prompt can indicate that the service allows remote access to the system, so at this point, you may want to run a bruteforce attack to crack the credentials.

Ports Included in the Discovery Scan

In total, the discovery scan includes over 250 ports, which includes the following set of ports:

- Standard and well known ports, such as ports 20, 21, 22, 23, 25 53, 80, and 443.
- Alternative ports for a service, such as ports 8080 and 8442, which are additional ports that HTTP and web services can use.
- Ports listed as the default port in a module.

If you do not see the port that you want to scan, you can manually add the port to the discovery scan. For example, if you know that your company runs web servers with port 9998 open, you need to manually add port 9998 to the discovery scan. This ensures that the discovery scan includes every port that is potentially open.

If you want to scan all ports, you can specify 1-65535 as the port range. Keep in mind that a discovery scan that includes all ports can take several hours to complete.

If there is a port that you do not want to scan, you can exclude the port from the discovery scan. The discovery scan will not scan any ports on the excluded list. For example, if your company uses an application that runs on port 1234, and you do not want to affect the application's performance, you can add the port to the excluded list.

Discovery Scan Options

You can configure the following options for a discovery scan:

Option	Description
Target addresses	Defines the individual hosts or network range that you want to scan.
Perform initial port scan	Performs a port scan before the discovery scan performs service version verification.
Custom Nmap arguments	Sends flags and commands to the Nmap executable. Discovery scan does not support the following Nmap options: -o, -i, -resume, -script, -datadir, and -stylesheet.
Additional TCP ports	Appends additional TCP ports to port scan. By default, the port scan covers a small, but wide range of ports. Use this option if you want to add more ports to the scan.
Excluded TCP ports	Excludes certain TCP ports from service discovery. By default, the port scan covers a specific range of ports. Use this option to add a port that you want to exclude from the scan.
Custom TCP port range	Specifies a range of TCP ports for the discovery scan to use instead of the default ports. If you set a custom TCP port range, the discovery scan ignores all default ports and uses the range that you define instead.
Custom TCP source range	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	Controls the Nmap timing option. Choose from the following timing templates: **Insane (5)** - Speeds up the scan. Assumes that you are on a fast network and sacrifices accuracy for speed. The scan delay is less than 5 ms. **Aggressive (4)** - Speeds up the scan. Assumes that you are on a fast and reliable network. The scan delay is less than 10 ms. **Normal (3)** - The default port scan speed and does not affect the scan. **Polite (2)** - Uses less bandwidth and target resources to slow the scan. **Sneaky (1)** - The speed used for IDS evasion.

Option	Description
	Paranoid (0) - The speed used for IDS evasion.
Portscan timeout	Determines the amount of time Nmap spends on each host. The default value is 5 minutes.
UDP service discovery	Sets the discovery scan to find all services that are on the network. Metasploit uses custom modules instead of Nmap to perform UDP service discovery.
Scan SNMP community strings	Launches a background task that scans for devices that respond to a variety of community strings.
Scan H.323 video endpoints	Scans for H.323 devices.
Enumerate users via finger	Queries user names and attempts to bruteforce the user list if the discovery scan detects the Finger protocol.
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	If enabled, this option prepares the scan and shows all of the options that the Discovery Scan will use in the task log. However, it does not launch the scan.
Web scan: run the Pro Web Scanner	Automatically runs a web scan, web audit, and web exploit along with a discovery scan. It is generally recommended that you do not enable this option unless you are running a scan against a very small set of hosts. If you are running a discovery scan against a large number of hosts, you should run the web scanner separately from the discovery scan.
SMB user name	Defines the SMB user name that the discovery scan uses to attempt to login to SMB services.
SMB password	Defines the SMB password that the discovery scan uses to attempt to login to SMB services.
SMB domain	Defines the SMB server name and share name.

Specifying IPv6 Addresses

Metasploit Pro 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 Pro. To identify individual IPv6 addresses, you can use SNMP, Nmap, or thc-alive6, which is part of the thc-ipv6 toolkit.

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.

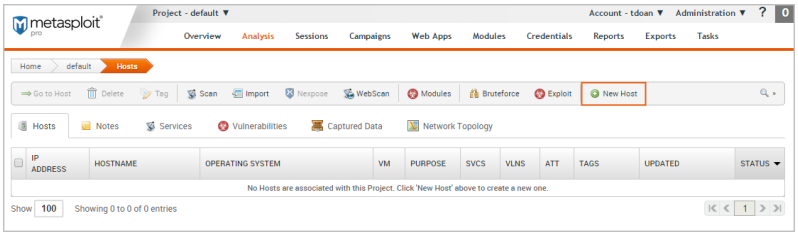
Importing a File that Contains IPv6 Addresses

To import a file, select **Analysis > Hosts** . When the Hosts page appears, click the **Import** button. When the Import Data page appears, browse to the location of the host address file and import the host address file. The file must be a text file that lists each IPv6 address on a new line, as shown below:

```
FE80:0000:0000:0000:0202:B3FF:FE1E:8329
FE80:0000:0000:0000:0202:B3FF:FE1E:8328
```

Manually Adding a Host with an IPv6 Address

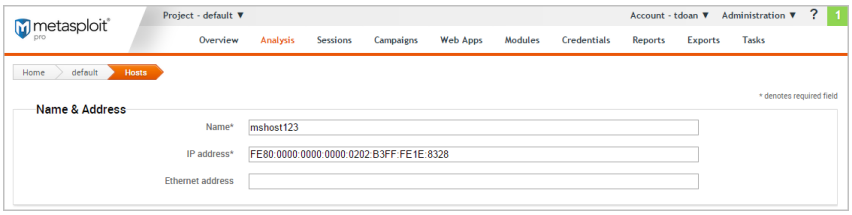
To manually add a host, select **Analysis > Hosts**. When the Hosts page appears, click the **New Host** button.



When the Hosts page appears, enter the following information:

- **Name**: A name for the host.
- **IP address**: The IPv6 address for the host.

The other fields, such as Ethernet address and OS information, are optional.



Running a Discovery Scan

A discovery scan runs Nmap along with a few service specific modules to identify the systems that are alive and to find the open ports and services. At a minimum, you need to specify the addresses of the systems that you want scan. There are also advanced options that you can configure to fine-tune the different scan phases. For example, you can bypass the port scanning phase and move onto version

detection, or you can scan each host individually to accelerate the import of hosts into the project. Additionally, these advanced settings let you choose the ports, the target services, the scan speed, and the scan mode.

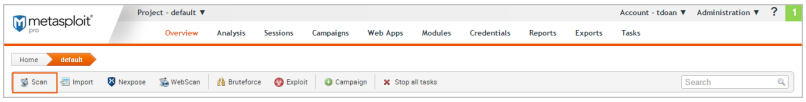
Since the discovery scan mostly leverages Nmap, you can specify additional Nmap options to customize the scan. For example, if you want to change the scanning technique, you can provide the Nmap command line option for the technique that you want to use, and the discovery scan applies those settings instead of the default ones. For more information on Nmap options, visit the Nmap documentation.

To run a discovery scan:

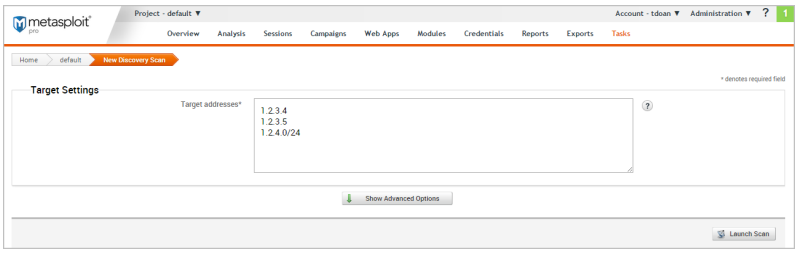
1. From within a project, click the **Overview** tab.

   Note: You can also access the **Scan** button from the Analysis page.

2. When the Overview page appears, click the **Scan** button.



3. When the New Discovery Scan page appears, enter the target addresses that you want to include in the scan in the **Target addresses** field.



   You can enter a single IP address, an address range, or a CIDR notation. If there are multiple addresses or address ranges, use a newline to separate each entry.

4. At this point, you can launch the scan. However, if you want to fine tune the scan, you can click the **Show Advanced Options** button to display additional options that you can set for the discovery scan. For example, you can specify the IP addresses that you want to explicitly include and exclude from the scan.

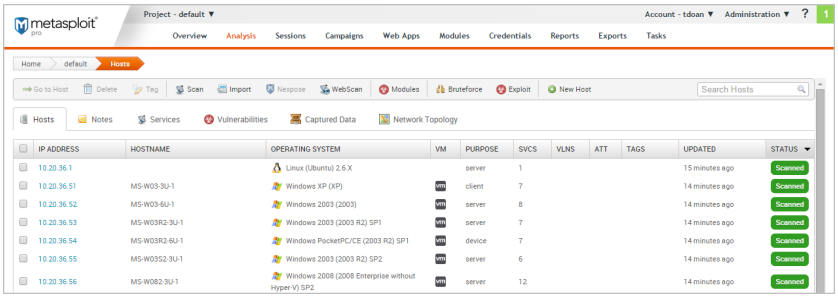
   For more information about the scan options that are available, see Discovery Scan Options.

5. When you are ready to run the scan, click the **Launch Scan** button.

   After the discovery scan launches, the task log displays and shows you the status of the progress and status of the scan. If the scan finishes without error, the status is 'complete'. Otherwise,

Viewing Scan Results

The best way to view the data collected by the Discovery Scan is from the Hosts page. To view the Hosts page, select **Hosts > Analysis**. Each host will have one of the following statuses:scanned, cracked, shelled, or looted. For recently scanned hosts, the easiest way to identify them to sort them by date and their status.



Data Gathered from a Discovery Scan

You'll notice that for each scanned or imported host, the following information is displayed, if available:

- The IP address
- The host name
- The operating system
- The active services
- The timestamp when the host was last updated
- The host status

Decoding the Host Status

The host status describes the last current event that occurred with the host. There's a hierarchical order to the statuses.

- Scanned - Indicates a discovery scan, Nexpose scan, or import was performed.
- Shelled - Indicates that a session was opened on the host.
- Looted - Indicates that
- Cracked

Vulnerability Scanning with Nexpose

Vulnerability scanning and analysis is the process that detects and assesses the vulnerabilities that exist within an network infrastructure. A vulnerability is a characteristic of an asset that an attacker can exploit to gain unauthorized access to sensitive data, inject malicious code, or generate a denial of service attack. To prevent security breaches, it is important to identify and remediate security holes and vulnerabilities that can expose an asset to an attack.

You can use Nexpose to scan a network for vulnerabilities. Nexpose identifies the active services, open ports, and running applications on each machine, and it attempts to find vulnerabilities that may exist based on the attributes of the known services and applications. Nexpose discloses the results in a scan report, which helps you to prioritize vulnerabilities based on risk factor and determine the most effective solution to implement.

Nexpose integrates with Metasploit Pro to provide a vulnerability assessment and validation tool that helps you eliminate false positives, verify vulnerabilities, and test remediation measures. There are a couple of ways that you can use Metasploit Pro with Nexpose. Metasploit Pro provides a connector that allows you to add a Nexpose Console so that you can run a vulnerability scan directly from the web interface and automatically import the scan results into a project. You can also run scans from Nexpose and import the scan reports into Metasploit Pro to perform vulnerability analysis and validation. You just need to choose the method that works best for you.

Nexpose Terminology

Some terms in Nexpose differ from those used in Metasploit. Here are some Nexpose terms you should familiarize yourself with:

- **Asset**: A host on a network.
- **Site**: A logical group of assets that has a dedicated scan engine. A site can run over a long period of time and provide you with historical, trending data and is similar to a project in Metasploit.
- **Scan Template**: A template that defines the audit level that Nexpose uses to perform a vulnerability scan. For more information on scan templates, check out the Nexpose User Guide.

Downloading and Installing Nexpose

You can download the Community edition of Nexpose from the Rapid7 site. For more information on how to install and configure Nexpose, read this handy installation guide.

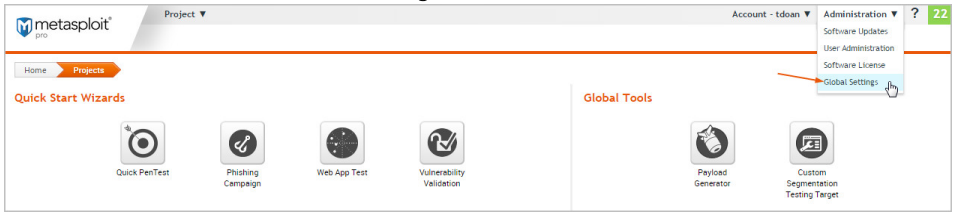
If you are interested in Nexpose Enterprise, please contact the Rapid7 sales team.

Adding a Nexpose Console

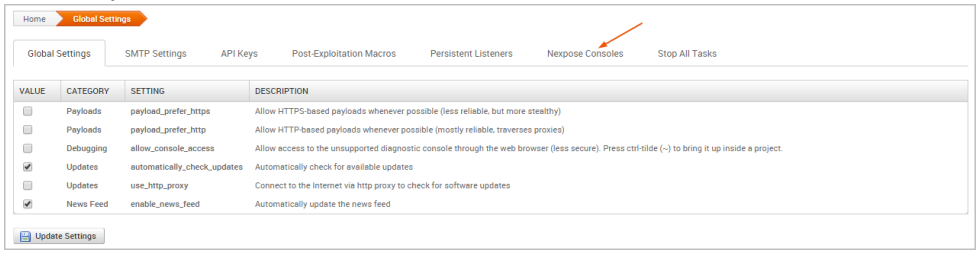
Before you can run a Nexpose scan from Metasploit Pro, you must add a Nexpose Console. You'll need to know the address and port Nexpose runs on, and you'll need the credentials for an account that can be used to log into the Nexpose console.

To add a Nexpose Console:

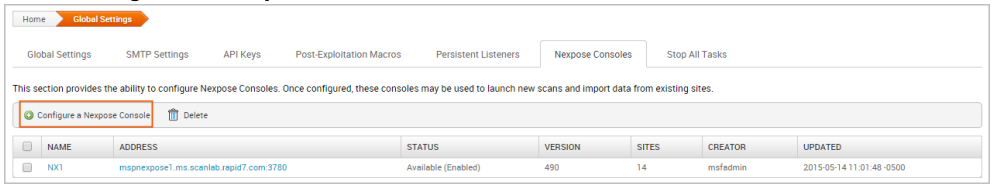
1. Choose **Administration > Global Settings** from the main menu.



2. Click the Nexpose Consoles tab.

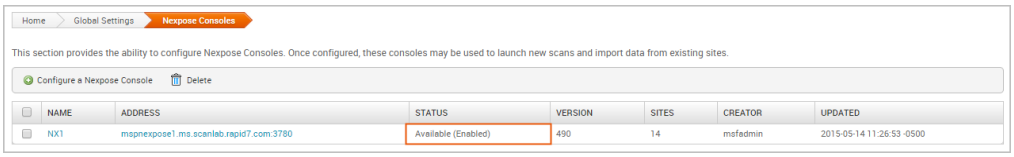


3. Click the **Configure a Nexpose Console** button.



4. When the Nexpose configuration page appears, enter the following information:
   - Console Address: The IP or server address for the Nexpose instance.
   - Console Port: The port that runs the Nexpose service. The default port is 3780.
   - Console Username: The user name that will be used to log in to the console.
   - Console Password: The password that will be used to authenticate the account.
5. Select the **Enabled** option to initialize and activate the Nexpose Console.
6. Save the configuration.

The Nexpose Consoles table is updated with the console. If Metasploit Pro is able to successfully connect and authenticate to the Nexpose console, the status is 'Available (Enabled)', as shown below:



Otherwise, an 'Error' status displays if there is an issue with the console's configuration. The following errors may appear:

- 'Error: Nexpose host is unreachable' indicates that Metasploit Pro cannot access the console. You will need to verify that you have entered the correct address and port.
- 'Error: Authentication required for API access' indicates that the credentials that you have provided cannot be used to authenticate to the Nexpose server. You will need to verify that you have entered the correct credentials.

Running a Nexpose Scan

To be able to prioritize security risks, you must know what devices are running in an environment and understand how they are vulnerable to attacks. You can run a Nexpose scan to discover the services and applications that are running on a host and identify potential vulnerabilities that may exist based on the collected data. To learn how Nexpose works, check out the Nexpose User Guide.

All scan data collected from Nexpose is stored in a Metasploit project and can be viewed from the Analysis area. The information gathered from each host includes the IP address, host name, operating system, running services, and possible vulnerabilities. Metasploit Pro maps each vulnerability to a related module, if one exists in the module database for it. These modules are viewable from the Modules tab on the single host view.

To configure a Nexpose scan:

1. From within a project, click the **Overview** or **Analysis** tab.
2. Click the **Import** button located in the Quick Tasks bar.
3. When the Import page appears, click the **Choose a Nexpose console** dropdown and select the console you want to use to run the scan.

   The list shows Nexpose consoles that you have added to Metasploit Pro. If there are not any consoles available, please add a Nexpose console before you continue.

4. Enter the addresses you want to scan in the **Scan targets** field.

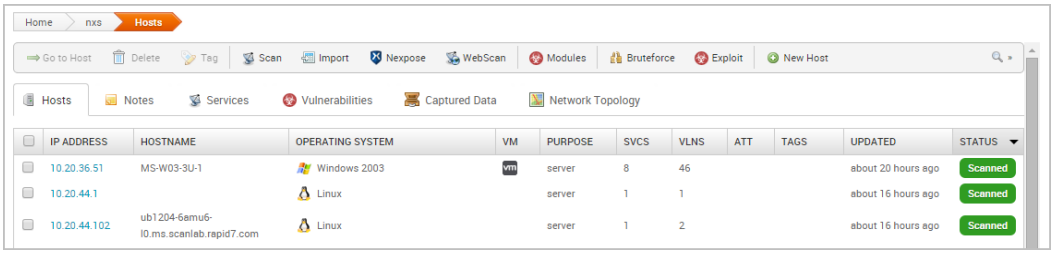
   You can specify an IP address, an IP range, or a CIDR notation. Each item must be listed on a newline.

   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.

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

5. Click the **Scan template** dropdown and select a template. For more information on scan templates, please check out the Nexpose User Guide.
6. If you do not want the scan to overwrite the data for existing hosts in the project, select the Don't change existing hosts option.
7. Click the **Import data** button to start the scan.

After the scan completes, select **Analysis > Hosts** to view the scan results.



After you run a Nexpose scan from Metasploit Pro, a temporary site is created on the Nexpose console. The naming syntax for a temporary site is `Metasploit-<project name>-<ID>`. In Nexpose, select **Assets > Sites** to view a list of sites and search for the site by project name.

Importing Nexpose Data

If you prefer to run scans directly from the Nexpose Console, you can import the scan results to share the results and validate them with Metasploit Pro. When you import data from Nexpose, Metasploit Pro automatically indexes the vulnerability data from Nexpose by using the service and vulnerability reference

ID to map each vulnerability to a matching exploit. The mapped exploits helps you to easily launch attacks against the vulnerability and to quickly determine if the vulnerability is a real risk or a false positive.

You can either import a site directly from a Nexpose Console or you can import a Nexpose Simple XML or XML export file.

Importing a Nexpose Simple XML or XML Export File

1. From within a project, click the **Overview** or **Analysis** tab.
2. Click the **Import** button located in the Quick Tasks bar.
3. When the Import Data page appears, select the **Import from file** radial button.
4. Click on the **Choose file** button to open the File Upload window.
5. When the File Upload window appears, browse to the location of the file you want to import, select it, and click the **Open** button.

   Metasploit Pro supports the following Nexpose export types: XML Export, XML Export 2.0, and Nexpose Simple XML Export.
6. Configure any of the additional settings (optional):

   - **Excluded Addresses**: Enter any hosts you do not want to include in the import. You can enter a single host, an IP range, or a CIDR notation. Each item must appear on a new line.
   - **Don't change existing hosts**: Select this option if you do not want to overwrite the data for a host that already exists in the project.
   - **Automatic tagging**: Enter any tags you want to apply to the imported hosts. You can also select the **Automatically tag by OS** option to add an OS tag, such as 'os_windows', 'os_linux' or 'os_unknown' tag, to each imported host.
7. Click the **Import Data** button to start the import.

   The task log appears and shows you the status of the import. When the import completes, the task log displays a 'Completed' status. To see the imported data, select **Analysis > Hosts** to go to the Hosts page.

Importing Existing Nexpose Sites

1. Open the project that you want to import data into.
2. From the Tasks bar, click the **Import** button. The **Import Data** page appears.
3. Select the **Import from Nexpose** option.
4. Click the **Choose a Nexpose Console** dropdown and select the console from which you want to import data.

5. Select the **Import existing data** option.
6. Select the site(s) you want to import from the Sites table.
7. Select **Do not change existing hosts** if you do not want to modify any existing hosts that are stored in the project.
8. Click the **Import Data** button to start the import.

   The task log appears and shows you the status of the import. When the import completes, the task log displays a 'Completed' status. To see the imported data, select **Analysis > Hosts** to go to the Hosts page.

Importing Data

You can perform a data import to upload vulnerability scan data or bring in data from other Metasploit projects. The import feature is useful if you have existing vulnerability data to validate or you have data that you want to share between projects.

Importing Data from Vulnerability Scanners

Metasploit allows you to import scan reports from third party vulnerability scanners, such as Nessus, Core Impact, and Qualys. When you import a scan report, host data, such as each host's operating system, services, and discovered vulnerabilities, is imported into the project.

To import a scan report from a third party vulnerability scanner:

1. From within a project, click the **Overview** tab.
2. Find and click the **Import** button.
3. When the Import Data page appears, select the **Import from file** radial button.
4. Click on the **Choose file** button to open the File Upload window.
5. When the File Upload window appears, browse to the location of the file you want to import, select it, and click the **Open** button.
6. Configure any of the additional settings (optional):
   - **Excluded Addresses**: Enter any hosts you do not want to include in the import. You can enter a single host, an IP range, or a CIDR notation. Each item must appear on a new line.
   - **Don't change existing hosts**: Select this option if you do not want to overwrite the data for a host that already exists in the project.
   - **Automatic tagging**: Enter any tags you want to apply to the imported hosts. You can also select the **Automatically tag by OS** option to add an OS tag, such as 'os_windows', 'os_linux' or 'os_unknown' tag, to each imported host.
7. Click the **Import Data** button to start the import.

   The task log appears and shows you the status of the import. When the import completes, the task log displays a 'Completed' status. To see the imported data, select **Analysis > Hosts** to go to the Hosts page.

Supported Third Party Scan Reports

Metasploit supports most of the major scanners on the market, including Rapid7's own Nexpose, and other tools like Qualys and Core Impact. The following scan reports are supported:

- 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
- Critical Watch VM XML
- IP Address List
- Libpcap Network Capture
- Spiceworks Inventory Summary CSV
- Core Impact XML

! Metasploit Pro does not import service and port information from Qualys Asset files. If you import a Qualys Asset file, you must run a discovery scan to enumerate services and ports that are active on the imported hosts.

Importing Nexpose Data

If you prefer to run scans directly from the Nexpose Console, you can import the scan results to share the results and validate them with Metasploit Pro. When you import data from Nexpose, Metasploit Pro

automatically indexes the vulnerability data from Nexpose by using the service and vulnerability reference ID to map each vulnerability to a matching exploit. The mapped exploits helps you to easily launch attacks against the vulnerability and to quickly determine if the vulnerability is a real risk or a false positive.

You can either import a site directly from a Nexpose Console or you can import a Nexpose Simple XML or XML export file.

Importing a Nexpose Simple XML or XML Export File

1. From within a project, click the **Overview** or **Analysis** tab.
2. Click the **Import** button located in the Quick Tasks bar.
3. When the Import Data page appears, select the **Import from file** radial button.
4. Click on the **Choose file** button to open the File Upload window.
5. When the File Upload window appears, browse to the location of the file you want to import, select it, and click the **Open** button.

   Metasploit Pro supports the following Nexpose export types: XML Export, XML Export 2.0, and Nexpose Simple XML Export.
6. Configure any of the additional settings (optional):

   - **Excluded Addresses**: Enter any hosts you do not want to include in the import. You can enter a single host, an IP range, or a CIDR notation. Each item must appear on a new line.
   - **Don't change existing hosts**: Select this option if you do not want to overwrite the data for a host that already exists in the project.
   - **Automatic tagging**: Enter any tags you want to apply to the imported hosts. You can also select the **Automatically tag by OS** option to add an OS tag, such as 'os_windows', 'os_linux' or 'os_unknown' tag, to each imported host.
7. Click the **Import Data** button to start the import.

   The task log appears and shows you the status of the import. When the import completes, the task log displays a 'Completed' status. To see the imported data, select **Analysis > Hosts** to go to the Hosts page.

Importing Existing Nexpose Sites

1. Open the project that you want to import data into.
2. From the Tasks bar, click the **Import** button. The **Import Data** page appears.
3. Select the **Import from Nexpose** option.
4. Click the **Choose a Nexpose Console** dropdown and select the console from which you want to import data.

5. Select the **Import existing data** option.
6. Select the site(s) you want to import from the Sites table.
7. Select **Do not change existing hosts** if you do not want to modify any existing hosts that are stored in the project.
8. Click the **Import Data** button to start the import.

   The task log appears and shows you the status of the import. When the import completes, the task log displays a 'Completed' status. To see the imported data, select **Analysis > Hosts** to go to the Hosts page.

Validating a Vulnerability

You've scanned your targets and identified potential vulnerabilities. The next step is to determine whether or not those vulnerabilities present a real risk. To validate a vulnerability, you have a couple of options:

- **The Vulnerability Validation Wizard** - The Vulnerability Validation Wizard provides an all-in-one interface that guides you through importing and exploiting vulnerabilities discovered by Nexpose. It enables you quickly determine the exploitability of those vulnerabilities and share that information with Nexpose. This feature is extremely handy if you use Nexpose to find and manage vulnerabilities.
- **Manual Validation** - Manual validation requires a bit more legwork than the wizard. This method provides you with much more control over the vulnerabilities that are targeted. It is generally used when you want to validate individual vulnerabilities or vulnerabilities discovered by other third-party scanners like Qualys or Nessus.

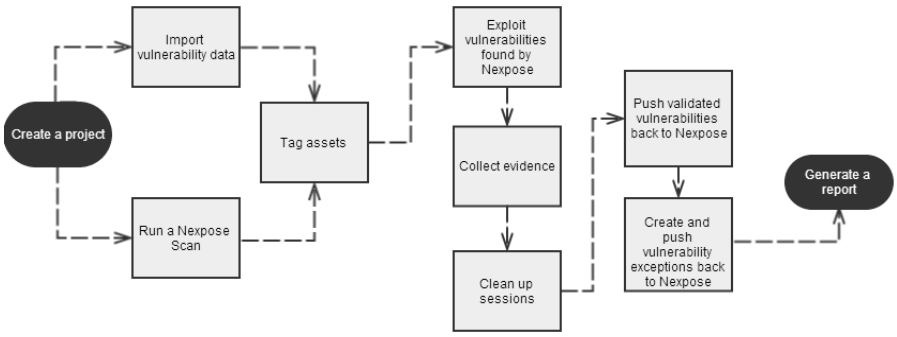
  When you perform manual validation, you will need to set up a penetration test as you normally would, which includes creating a project and adding vulnerability data via import or scan. Then, you need to try to exploit each vulnerability to determine whether or not they are valid threats. If the vulnerabilities were discovered by Nexpose, you have the option to send the results Nexpose.

Vulnerability Validation Wizard

The Vulnerability Validation Wizard simplifies and streamlines the vulnerability validation process for Nexpose users. It provides a guided interface that walks you through each step of the vulnerability validation process–from importing Nexpose data to auto-exploiting vulnerabilities to sending the validation results back to Nexpose. You can even generate a report that details the vulnerability validation test results and create exceptions for vulnerabilities that were not exploited.

Vulnerability Validation Wizard Workflow

To give you an idea of how you can configure the Vulnerability Validation Wizard, check out the workflow below:



Before You Can Use the Vulnerability Validation Wizard

Before you can run the Vulnerability Validation Wizard, you will need to make sure that you have access to a Nexpose instance.You can only validate vulnerabilities if you have Nexpose Enterprise or Nexpose Consultant version 5.7.16 or higher. Please check your Nexpose edition before attempting to use the Vulnerability Validation Wizard.

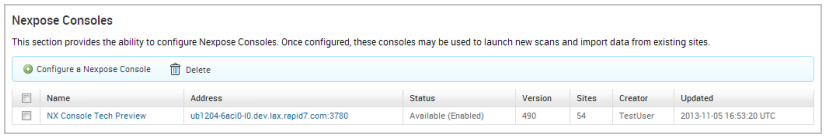
You must also have at least one site set up in Nexpose. To learn how to set up a site, please view the Nexpose Installation and Quick Start Guide.

Adding a Nexpose Console

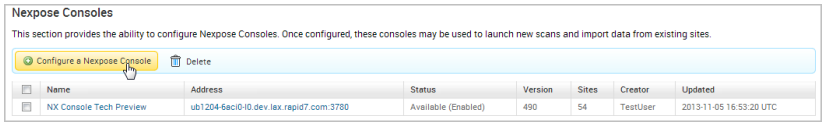
You can either add a Nexpose Console from the Global Settings or from the Vulnerability Validation Wizard. If you want to add a Nexpose Console from the Global Settings, follow the steps below.

To add a Nexpose Console:

1. Select **Administration > Global Settings** from the Administration menu.
2. Find the Nexpose Consoles area.

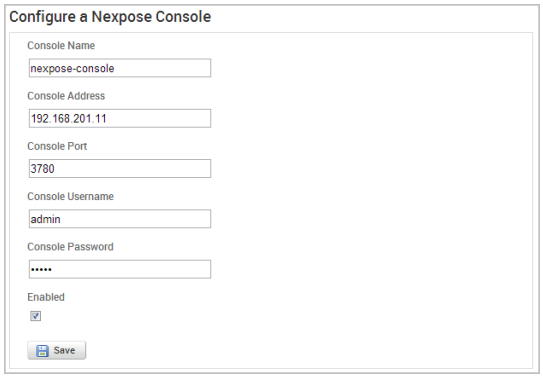


3. Click the **Configure a Nexpose Console** button.



4. When the Configure a Nexpose Console page appears, enter the following information:

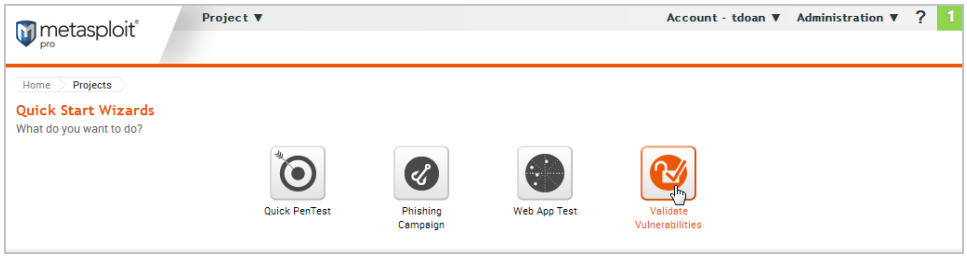
- **Console Address** - The IP address to the server that runs Nexpose. You can also specify the server name.
- **Console Port** - The port that runs the Nexpose service. The default port is 3780.
- **Console Username** - The Nexpose user name that will be used to log in to the console.
- **Console Password** - The Nexpose password that will be used to authenticate the user account.



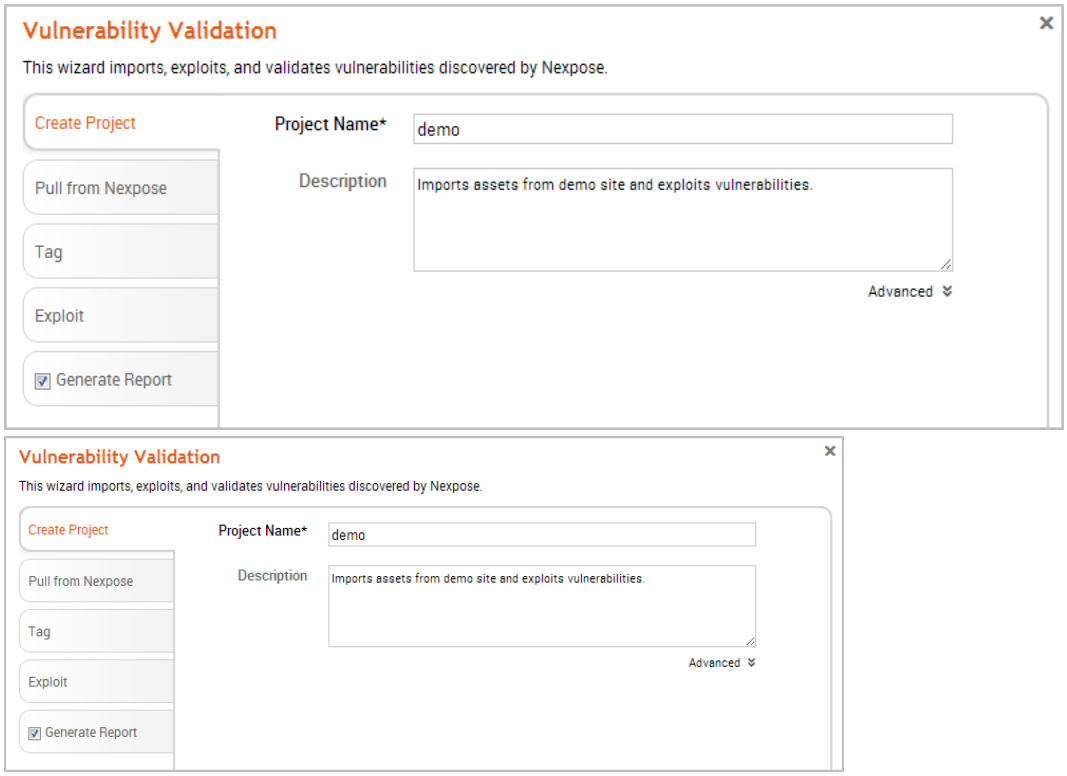
5. Save the Nexpose Console.

Configuring and Running the Vulnerability Validation Wizard

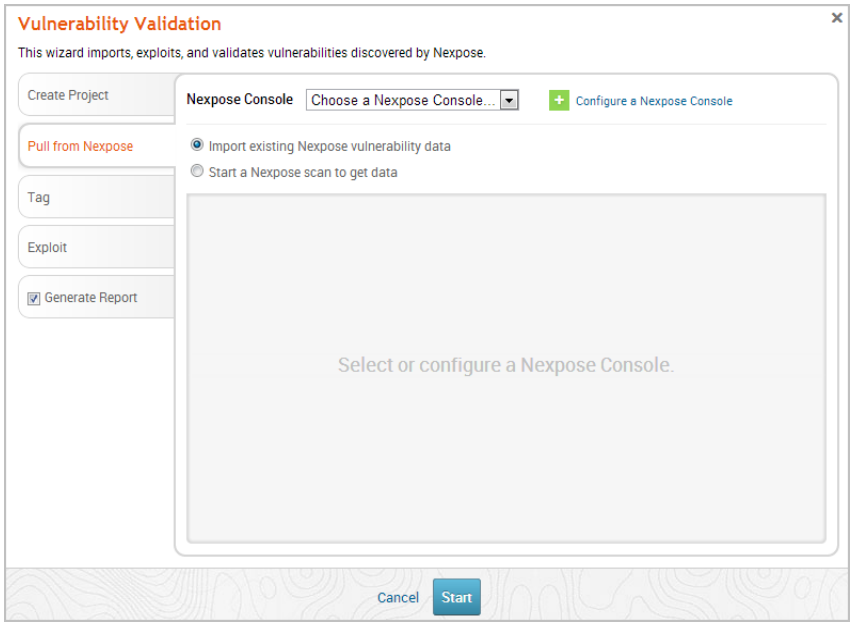
1. From the Projects page, click on the **Validate Vulnerabilities** widget located under the Quick Start Wizards area. The Validate Vulnerabilities Wizard opens and displays the **Create Project** page.

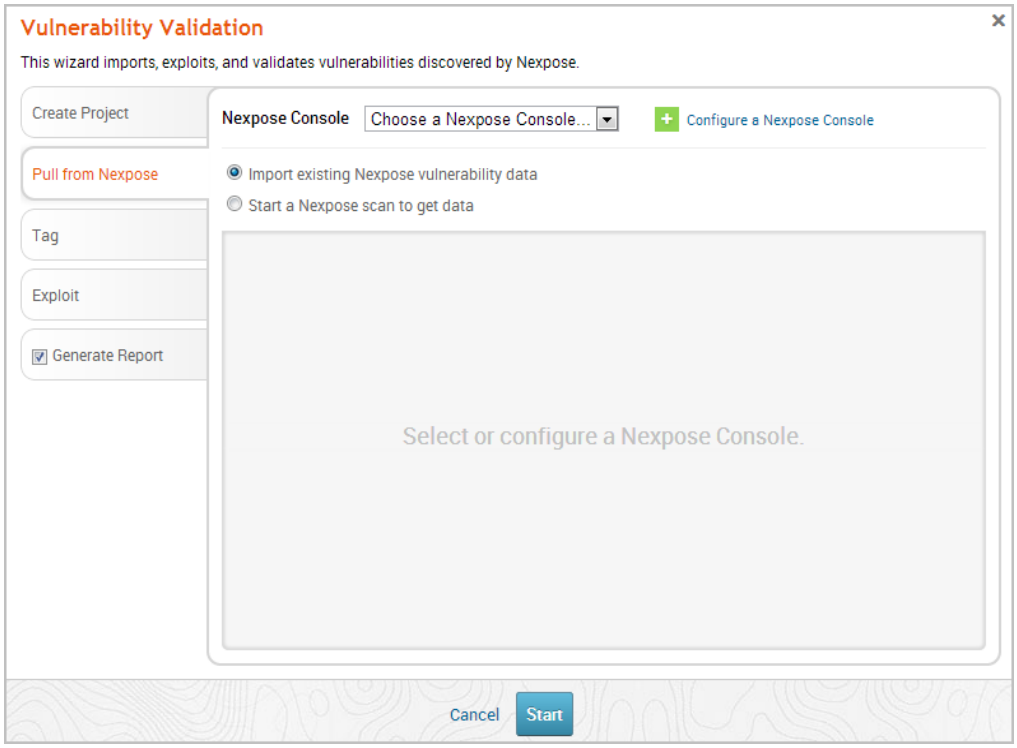


2. In the **Project Name** field, enter a name for the project. The project name can contain any combination of alphanumeric characters, special characters, and spaces. You can also provide an optional description for the project, which typically explains the purpose and scope of the test.

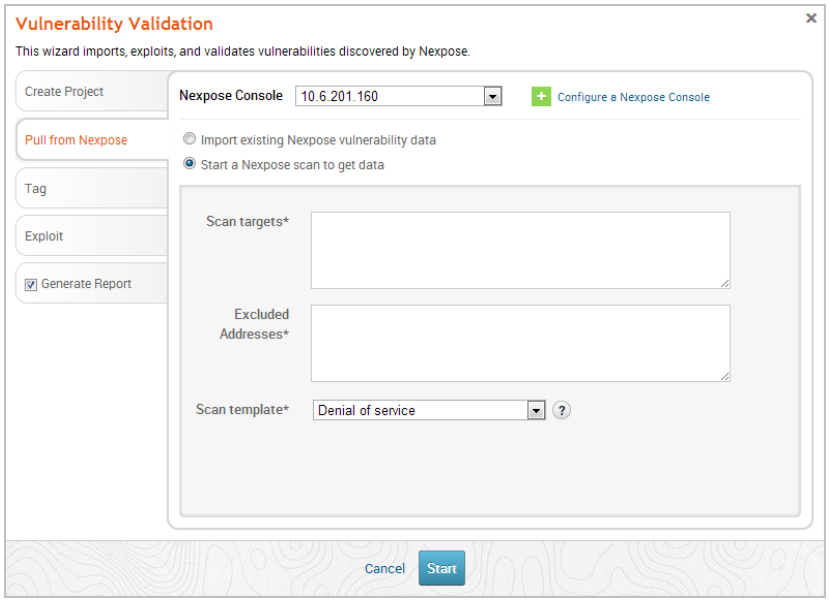


3. Click on the **Pull from Nexpose** tab. The Nexpose Consoles page appears.

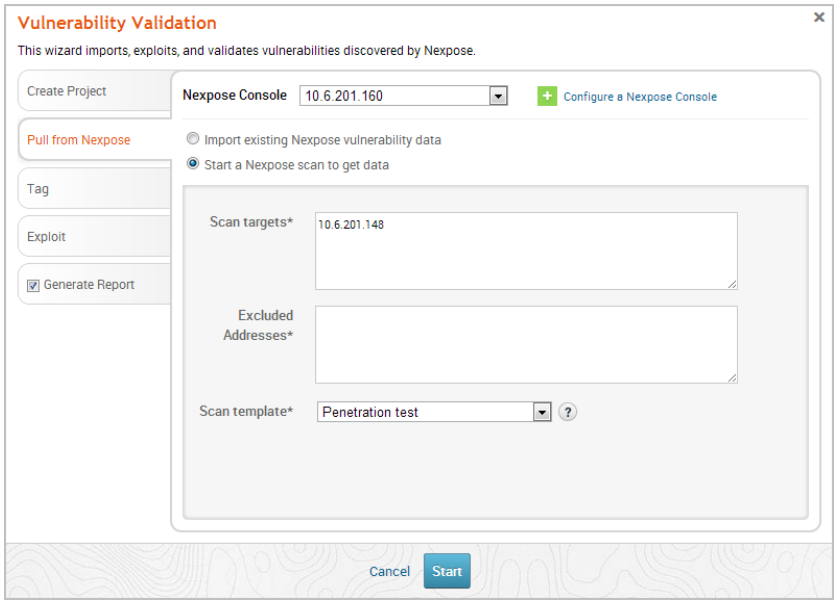




4. Click the **Nexpose Console** dropdown and select the console that you want to pull data from. If there are no consoles available, you can click the **Configure an Nexpose Console** link to add one.

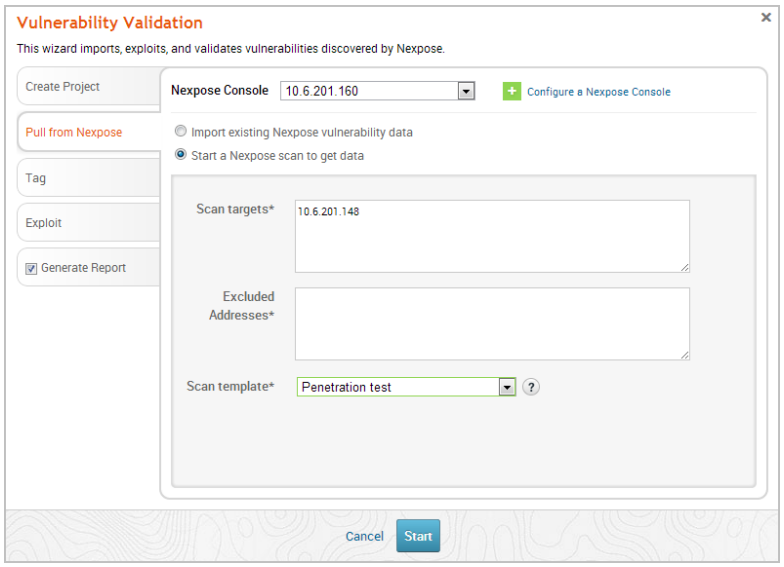


5. After you select a console, you can choose whether you want to run a Nexpose scan or import existing Nexpose data. Depending on the option you choose, the wizard will show the appropriate configuration page.
6. the **Start a Nexpose Scan to get data** option.
7. Enter the host addresses, or assets, that you want to scan in the **Scan targets** field. You can enter a single IP address, a comma separated list of IP addresses, an IP range described with hyphens, or a standard CIDR notation.



8. Click the **Scan template** dropdown and select the template you want to use.

   Note: A scan template is a predefined set of scan options. There are a few default ones that you can choose from. For more information on each scan template, please see the Nexpose User's Guide.



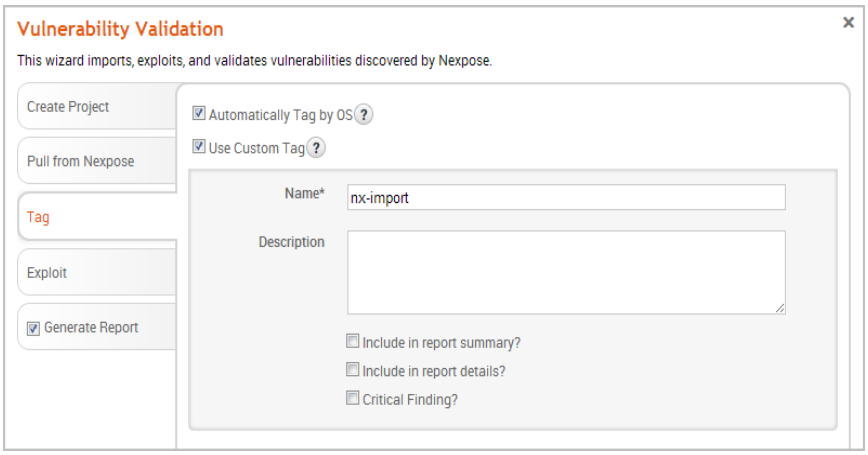
9. Click the **Tag** tab.

   Note: If you do not want to tag assets, go to Step 13.

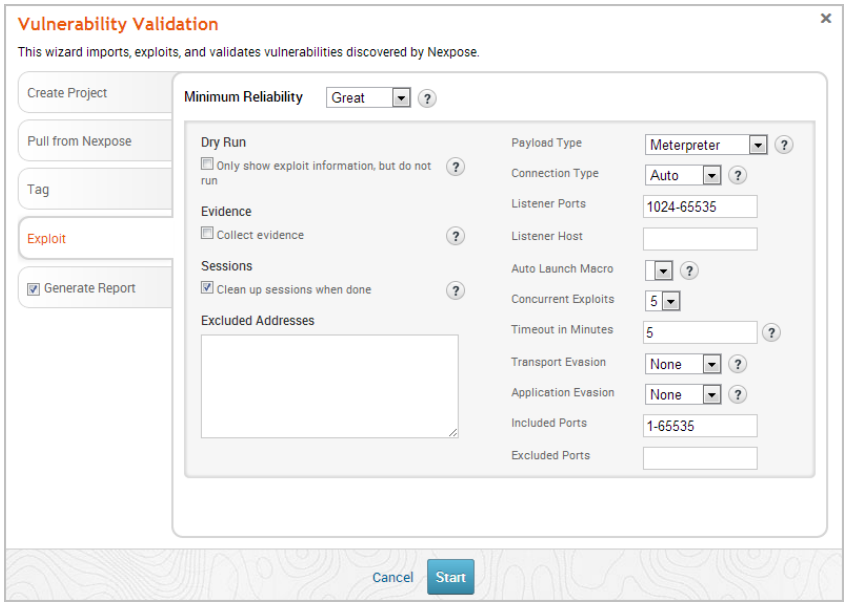
10. Select the **Automatically tag by OS** option if you want to tag each host with its operating system.

    Note: If enabled, hosts will be tagged with `os_linux` or `os_windows`.

11. Select the **Use custom tag** option if you want to tag each host with a user-defined tag. If this option is enabled, the Vulnerability Validation Wizard displays the fields and options that you can use to create a custom tag.

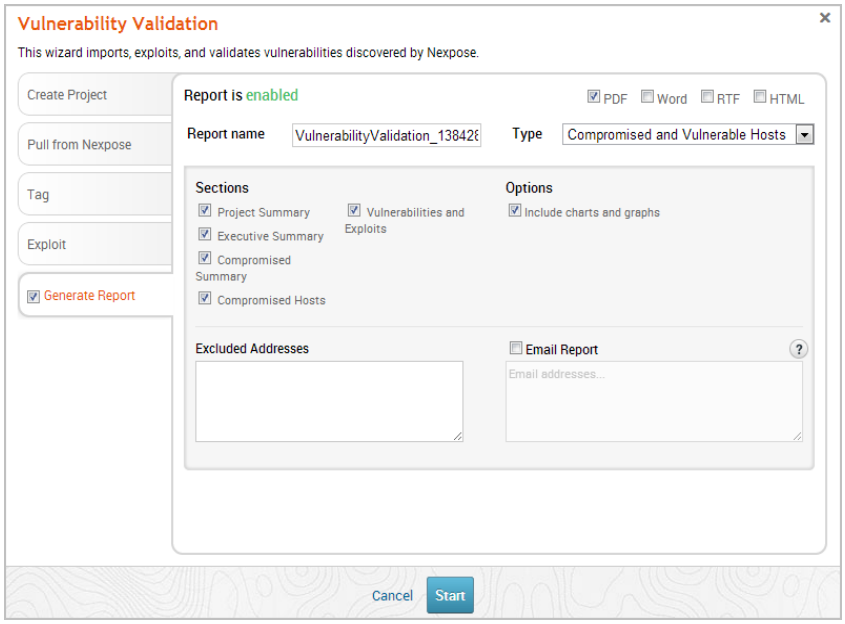


12. After you configure the tagging options, click on the **Exploit** tab. The Auto-Exploitation page appears.

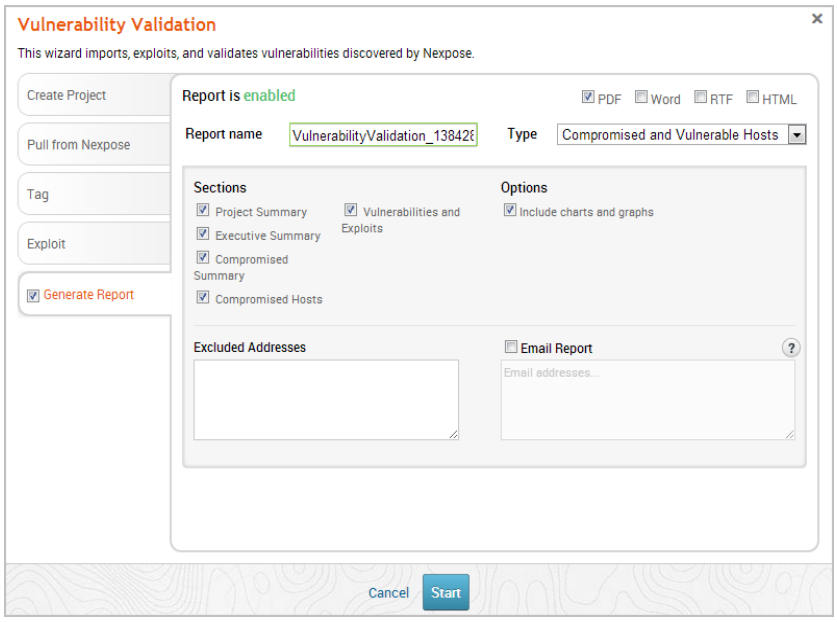


13. Click the **Minimum Reliability** dropdown and choose the module ranking you want to use. You should use Great or Excellent.

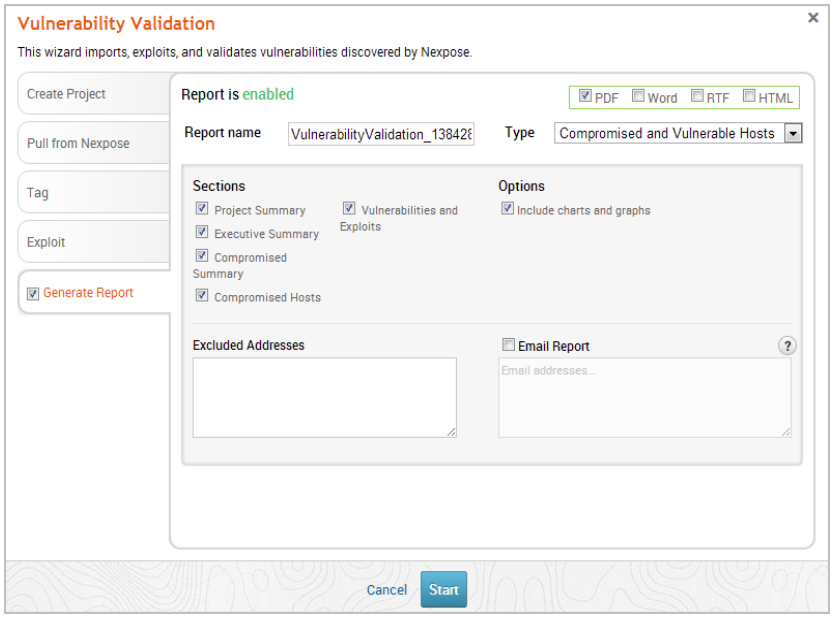
14. Click the **Generate Report** tab if you want to include an auto-generated report at the end of the vulnerability validation test. If you do not want to include a report, deselect the **Generate Report** option and skip to the last step.



15. Enter a name for the report in the **Report Name** field, if you want to use a custom report name. Otherwise, the wizard uses an auto-generated report name.

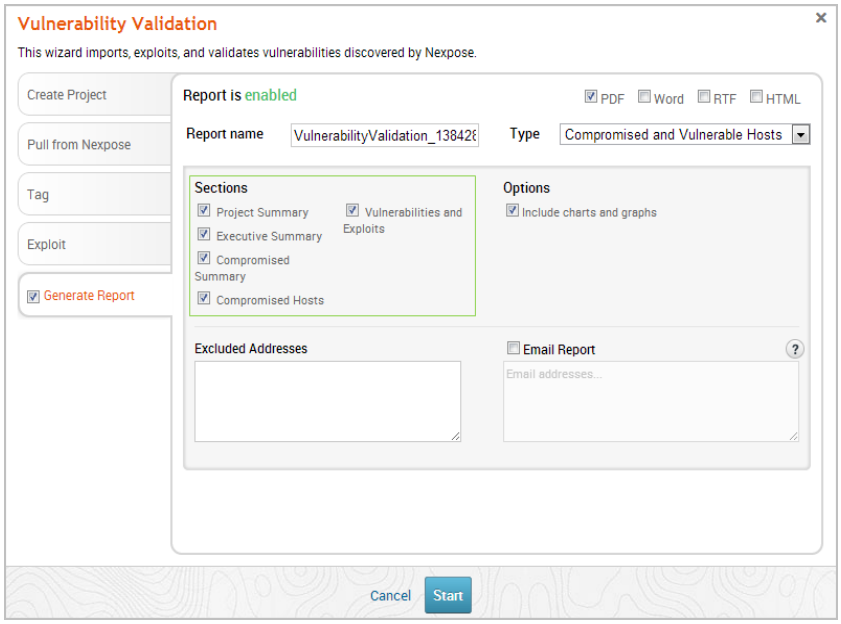


16. Select whether you want to generate the report in PDF, RTF, or HTML. PDF is the preferred and default format.

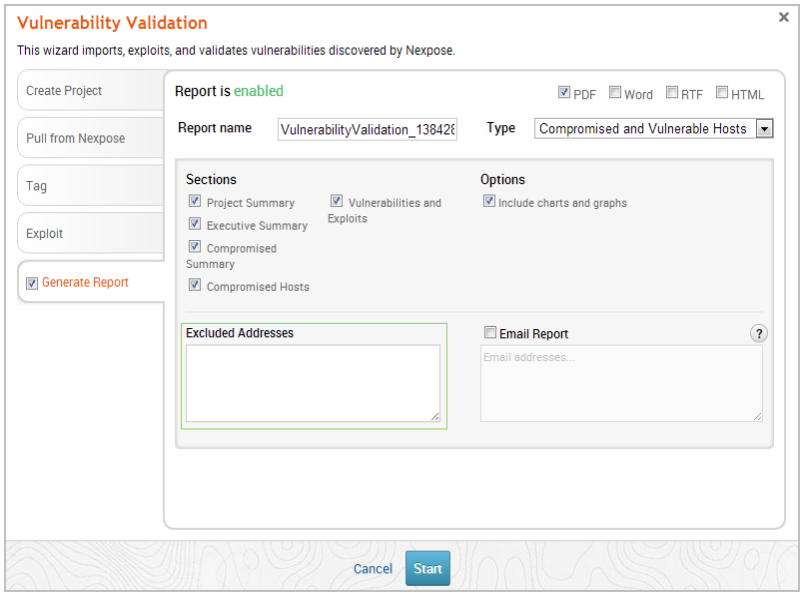


17. Click the **Type** dropdown and select the report type you want to generate. You can choose the Audit report or the Compromised and Vulnerable Hosts report.

18. From the Sections area, deselect any sections you do not want to include in the report. Skip this step if you want to generate all the report sections.

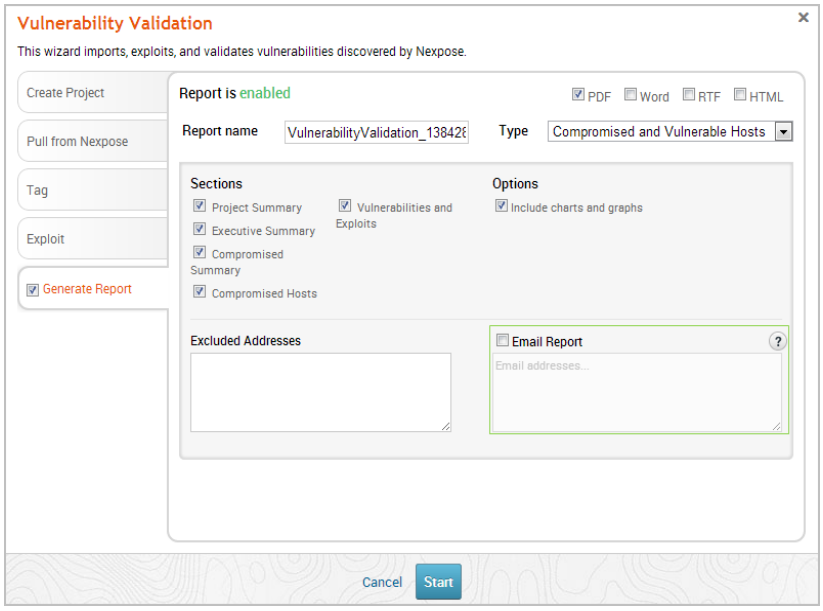


19. Enter any hosts, or assets, whose information you do not want included in the report in the **Excluded Addresses** field. You can enter a single IP address, a comma separated list of IP addresses, an IP range described with hyphens, or a standard CIDR notation.



20. Select the **Email Report** option if you want to e-mail the report after it generates. If you enable this option, you need to supply a comma separated list of e-mail addresses.

Note: If you want to e-mail a report, you must set up a local mail server or e-mail relay service for Metasploit Pro to use. To define your mail server settings, select **Administration > Global Settings > SMTP Settings**.



21. Click the **Launch** button. The Findings window appears and shows the statistics for the test.

Sharing Results with Nexpose

Now that you have finished the vulnerability validation process, you are ready to share the results with Nexpose. The process of sharing vulnerability validation results with Nexpose is called pushing. When you push the results to Nexpose, Metasploit Pro marks the validated vulnerabilities as exploited and adds the non-exploited vulnerabilities to the Vulnerability Exceptions and Policy Overrides page. This makes it much easier for you to track and prioritize the vulnerabilities that have already been tested.

There are a couple of ways that you can share results with Nexpose:

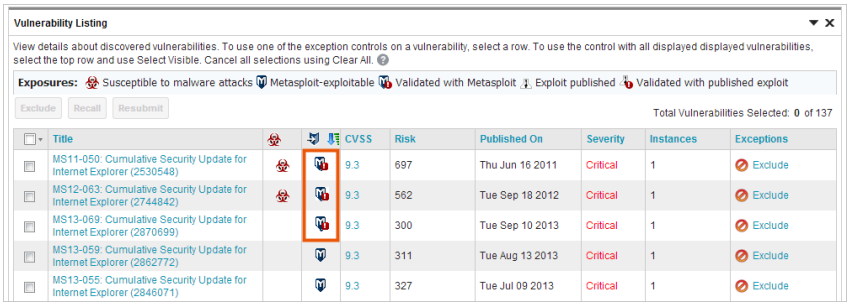
- Using the Vulnerability Validation Wizard - If you are using the Vulnerability Validation Wizard, you have the option to push validations and exceptions directly from the Findings page.
- Performing a manual push - If you are manually validating vulnerabilities, you can push validations and exceptions from either the vulnerabilities index page or the single vulnerability page.

Validation Results

There are two sets of results that you can share with Nexpose: validated vulnerabilities and vulnerability exceptions.

Validated Vulnerabilities

A validated vulnerability is a vulnerability that Metasploit Pro was able to successfully exploit to obtain a session on the target. A validated vulnerability will have a validated icon next to it on the asset page's Vulnerability Listing in Nexpose, as shown below:



This simply lets you know that the vulnerability has been tested and was successfully exploited by Metasploit Pro.

Vulnerability Exceptions

A vulnerability exception is vulnerability found by Nexpose that Metasploit Pro was unable to exploit. These vulnerabilities have a status of **Not Exploited**, which indicates that Metasploit Pro was unable to obtain a session on the target host due to compensating controls or back porting. Generally, vulnerability exceptions represent vulnerabilities that are typically low-risk or are used deliberately to mitigate bigger threats. Therefore, vulnerability exceptions are useful because they allow you to exclude certain vulnerabilities from a report so that you can manage your risk score.

Here are some reasons you may want to create a vulnerability exception:

- The vulnerability is used as compensating controls or to mitigate additional risks.
- The vulnerability exists due to an acceptable use case or deliberate practice, such as anonymous FTP access.
- The vulnerability represents an acceptable risk and may require more resources than you are willing to invest to remediate.This type of vulnerability typically poses a minimal risk.
- The vulnerability is a false positive.

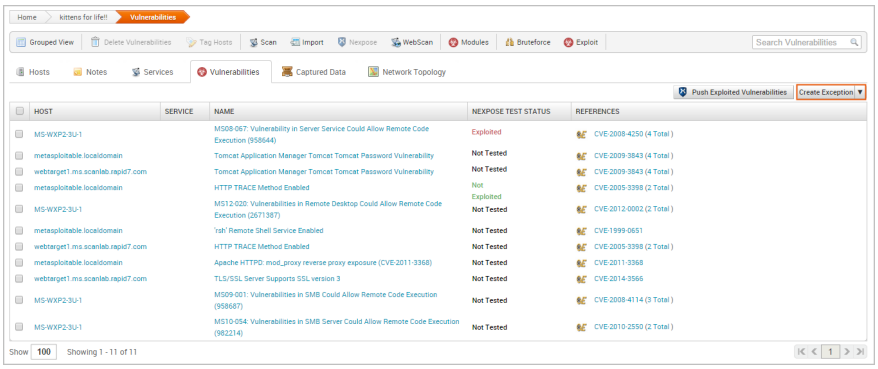
Accessing the Exceptions Page

You create and push Nexpose exceptions from the Exceptions page. As previously mentioned, there are a few places in the application from which you can create and push exceptions: the Vulnerability Validation Wizard's Findings page, the vulnerabilities index page, and the single vulnerability page.

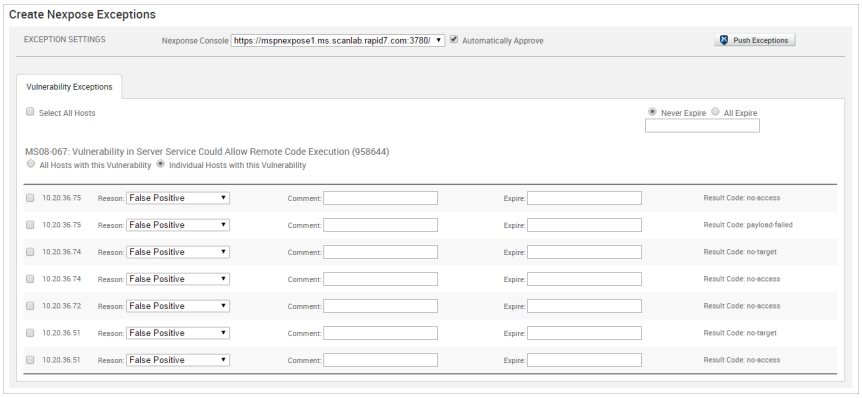
! To access the exceptions page from the vulnerabilities index page or the single vulnerability page, you must have set up a Nexpose console that Metasploit Pro can access. If Metasploit Pro does not detect a Nexpose console, it will not display the push options.

Accessing the Exceptions Page from the Vulnerabilities Index

To access the Exceptions page from the vulnerabilities index, select **Analysis > Vulnerabilities**. When the vulnerabilities list displays, you will see the **Push Exploited Vulnerabilities** button and the **Create Exception** button. Click on the **Create Exception** button to open the Exceptions page.

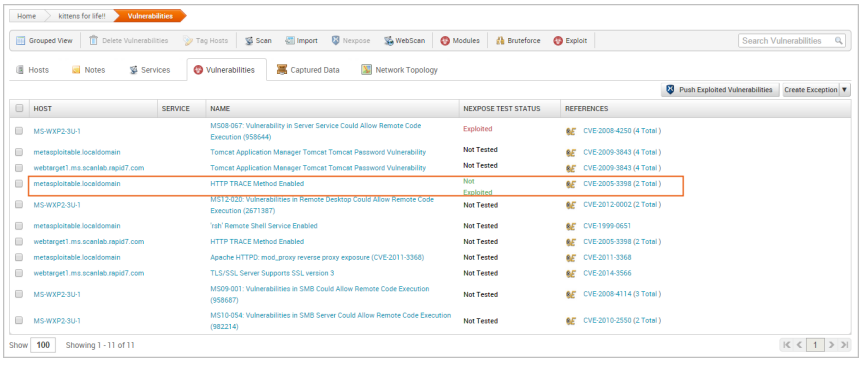


The Exceptions page is shown below. From this page, you can set the reason and expiration date for all vulnerability exceptions in the project. You can also bypass having to review them in Nexpose by automatically approving them.



Accessing the Exceptions Page from the Single Vulnerability Page

To access the Exceptions page from the single vulnerability page, select **Analysis > Vulnerabilities**. When the vulnerabilities list displays, click on a vulnerability name to open the single vulnerability page. From this page, you can configure the exception reason and expiration date for an individual vulnerability.



! Remember that you can only create exceptions for vulnerabilities that have a status of **Not Exploited**.

Understanding Statuses

All vulnerabilities imported from Nexpose have a status. The status lets you easily identify if the vulnerability has been tested and the results of the test. The status you see for a particular vulnerability will depend on whether you are viewing the vulnerability index or the single vulnerability page.

Statuses on the Vulnerability Index

The statuses on the vulnerability index indicate whether or not a vulnerability has been tested. These statuses are viewable from the **Nexpose Test Status** column.

- Not tested - A matching remote exploit module with a ranking of great or higher was not found for the vulnerability, so the Vulnerability Validation Wizard was unable to run auto-exploitation against it.However, there may be exploit modules with a lower ranking or auxiliary modules that you can run manually against the vulnerability to test for exploitability.

  To check for matching modules that can be run against the vulnerability, go to the single vulnerability page and view the **Related Modules** tab.
- Exploit attempted - A matching remote exploit module with a ranking of great or higher was found for the vulnerability, but the exploit attempt was unsuccessful. A vulnerability with a status of **Exploit Attempted** will have a **Failed** module run result.

  For any vulnerability that has an **Exploit Attempted** status, you can choose to mark it as **Not Exploitable** if you know that the vulnerability is not a valid risk. When you mark a vulnerability as **Not Exploitable**, the vulnerability is sent to Nexpose as an exception.
- Exploited - A matching remote exploit with a ranking of great or higher was found for the vulnerability. The exploit was able to successfully open a session on the target.

Statuses on the Single Vulnerability Page

The statuses on the single vulnerability page indicate the results of a module run.

- Failed - The module was unable to open a session on the target.
- Exploited - The module was able to successfully open a session on the target.
- Not Exploitable - The module failed to open a session, and you manually marked the vulnerability as **Not Exploitable**.
- No status available- The vulnerability has not been tested because there were not any matching remote exploits available.

Understanding Result Codes

A result code provides the reason why an exploit failed. If you see a **Failed** status for a module run, you can hover over the status to see the result code, which can help you troubleshoot the issue. The following result codes are available:

- None - Indicates that Metasploit could not determine if the module ran successfully or failed.
- Unknown - Indicates that Metasploit could not determine if the module ran successfully or failed.
- Unreachable - Indicates that Metasploit could not reach the network service.
- Bad-config - Indicates that the exploit settings were configured incorrectly.
- Disconnected - Indicates that the network service disconnected during a module run.
- Not-found - Indicates that Metasploit could not find the application or service.
- Unexpected-reply - Indicates that Metasploit did not receive the expected response from the application.
- Timeout-expired - Indicates that a timeout occurred.
- User-interrupt - Indicates that the user stopped the module run.
- No-access - Indicates that Metasploit could not access the application.
- No-target - Indicates that the module configuration was not compatible with the target.
- Not-vulnerable - Indicates that the application was not vulnerable.
- Payload-failed - Indicates that Metasploit delivered a payload, but was unable to open a session.

Marking a Vulnerability as Not Exploitable

You can manually assign a **Not exploitable** status for any vulnerability that has a Nexpose test status of **Exploit attempted** or **Not Tested**. The **Not exploitable** status implies that the vulnerability does not present a real risk and can be treated as an exception.

To mark an vulnerability as not exploitable, you need to select the **Mark as Not Exploitable** checkbox located on the single vulnerability page, as shown below:

! You can only assign a **Not exploitable** status to vulnerabilities that have a Nexpose test status of **Exploit attempted** or **Not Tested**.

Pushing Validated Vulnerabilities

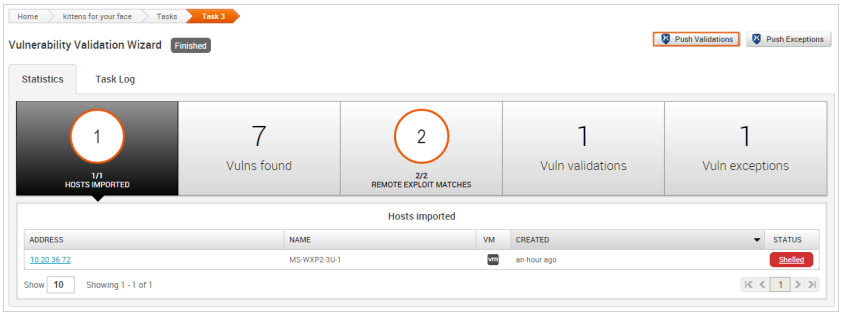
Pushing validated vulnerabilities is a one-button push process. When you are ready to push validated vulnerabilities back to Nexpose, there are a few ways that you can do it:

- From the Vulnerability Validation Wizard's Findings
- From the vulnerabilities index
- From the single vulnerability page

! To push validations to Nexpose, you must have an active Nexpose console that Metasploit Procan reach. If Metasploit Pro does not detect a Nexpose console, the push capabilities will be disabled.

Pushing Validated Vulnerabilities from the Vulnerability Validation Wizard's Findings

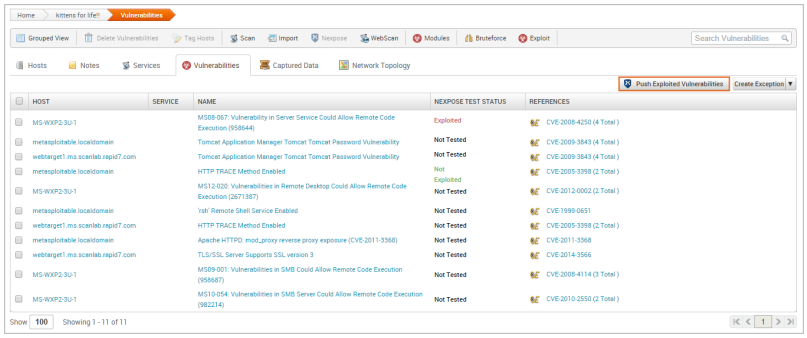
When the Vulnerability Validation Wizard finishes its run, you will be able to push validated vulnerabilities to Nexpose. The process of pushing validated vulnerabilities to Nexpose simply requires clicking the **Push Validations** button located on the Findings window, which is only active if there are valid vulnerabilities to send to Nexpose. The image below shows the active **Push Validations** button:



When you push the validations to Nexpose, any vulnerability that was successfully exploited by that have been exploited will be marked as validated in your Nexpose console, as shown below:

Pushing Validated Vulnerabilities from the Vulnerabilities Index

1. From within a project, select **Analysis > Vulnerabilities**. The vulnerabilities index appears.
2. Find the **Push Exploited Vulnerabilities** button. This button will be available if there are validations that need to be sent to Nexpose.



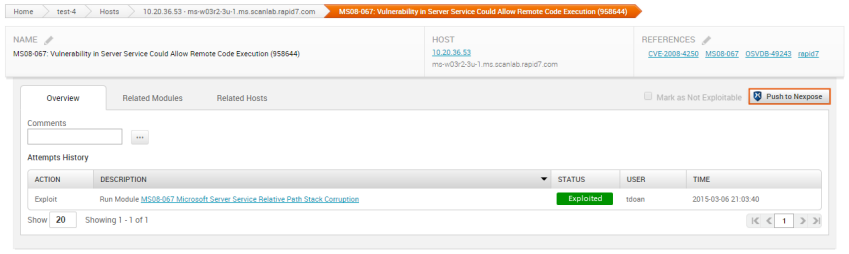
3.

4. Click the **Push Exploited Vulnerabilities** button. The Task Log appears and shows you when the push is complete.

After you push the validations to Nexpose, any vulnerability that was successfully exploited by that have been exploited will be marked as validated in your Nexpose console, as shown below:

Pushing a Single Validated Vulnerability

1. From within a project, select **Analysis > Vulnerabilities**. The vulnerabilities index appears.
2. Find the validated vulnerability you want to push to Nexpose and click on the name to open the single vulnerability page. Validated vulnerabilities will have a status of **Exploited**.



3. Click the **Push to Nexpose** button.
4. When the confirmation window appears, click **Yes** to push the validations to Nexpose.

   If Metasploit is unable to reach the Nexpose console, an error message appears and alerts you that there is an issue with the console. You can click **Yes** to try the push again. If the error continues to persist, you will need to close the modal and diagnose the console connectivity.

Creating and Pushing Vulnerability Exceptions

When you are ready to create and push vulnerability exceptions, you can do it from a few different areas in the application:

- From the Vulnerability Validation Wizard's Findings
- From the vulnerabilities index
- From the single vulnerability page

> ! To push exceptions to Nexpose, you must have an active Nexpose console that Metasploit Procan reach. If Metasploit Pro does not detect a Nexpose console, the push capabilities will be disabled.

As previously mentioned vulnerability exception is vulnerability found by Nexpose that Metasploit Pro was unable to exploit. When you create a vulnerability exception, you must set an expiration date that determines when the exception will no longer be effective and provide a reason that explains why the exception exists.

You must assign the exception one of the following reasons:

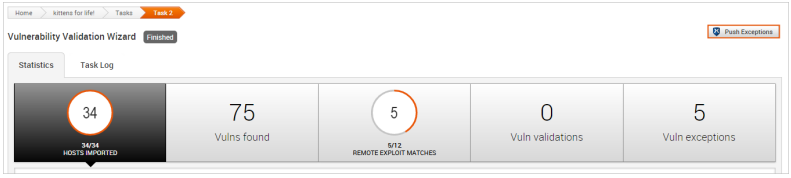
- False positive - Indicates that the vulnerability does not exist.
- Compensating control - Indicates that the vulnerability is a compensating control, or a workaround for a security requirement.
- Acceptable use - Use this exception reason for any vulnerability that is used as part of organizational practices.
- Acceptable risk - Indicates that the vulnerability is considered low risk. These vulnerabilities tend to pose minimal security risk and are likely to consume more resources than they are worth.
- Other - Indicates that the vulnerability has a custom exception reason. If you select **Other**, you can provide a custom exception reason in the **Comment** field.

Pushing Vulnerability Exceptions to Nexpose from the Vulnerability Validation Wizard's Findings

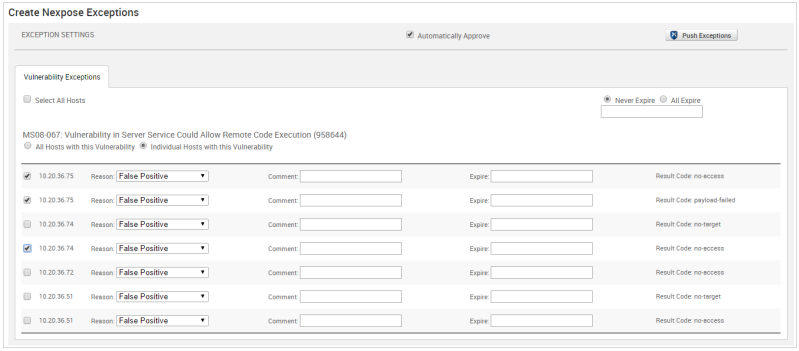
The Vulnerability Validation Wizard makes it extremely easy for you to push validations to Nexpose. When the Vulnerability Validation Wizard finishes its run, the **Push Exceptions** button appears on the Findings window if Metasploit Pro was unable to exploit any of the tested vulnerabilities. You can click the **Push Exceptions** button to open the Create Nexpose Exceptions page. From this page, you will be able to create and push vulnerability exceptions.

To push exceptions from the Vulnerability Validation Wizard's Findings:

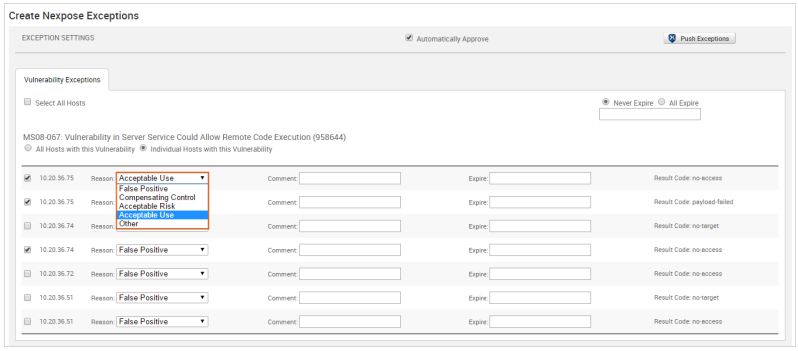
1. Click the **Push Exceptions** button located on the Findings window. The Create Nexpose Exceptions page appears.



2. Select the hosts that you want to create exceptions for. Use the **Select All Hosts** checkbox if you want to create exceptions for all hosts that have a non-exploitable vulnerability.

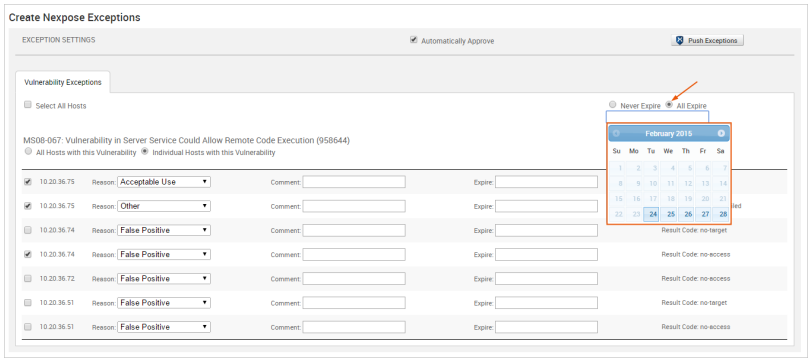


3. For each vulnerability, click the **Reason** dropdown and choose the vulnerability exception reason you want to assign to it. You can also provide additional information for the exception in the **Comment** field.



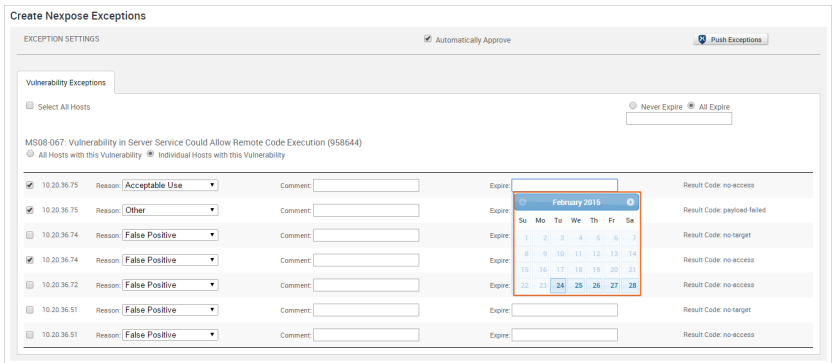
4. Choose the **All Expire** option if you want to set an expiration date for all the vulnerability exceptions. If you do not want to set an expiration date for any vulnerability exceptions, keep the default **Never Expire** option selected and go to Step 6.

   To set the same expiration date for all vulnerability exceptions, select on the **All Expire** option. A calendar appears. Find and select the date that you want to use.

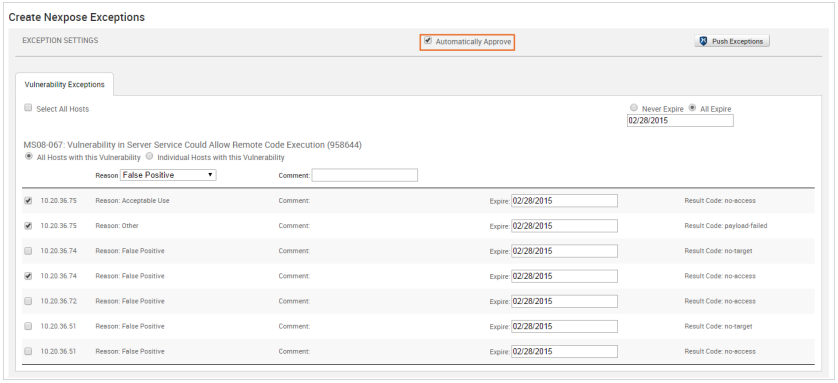


   If you want to set a unique expiration date for each host, skip this step and go to step 5.

5. To set a unique expiration date for each host, click on the **Expire** field next to each exception to display the calendar. Find the expiration date that you want to use and select it.



6. Verify that you want to approve all vulnerability exception requests from Metasploit Pro. If the **Automatically Approve** option is selected, Nexpose will automatically approve vulnerability exception requests imported from Metasploit Pro. Otherwise, the vulnerability exceptions will need to be manually reviewed and approved from the Nexpose console.



7. When you are ready to push the exceptions, click the **Push Exceptions** button.

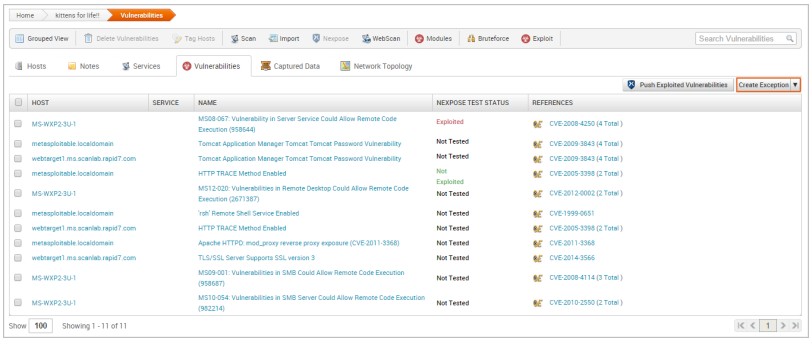
   If the push is successful, the "Push succeeded" status appears in place of the **Push** button.

Pushing Vulnerability Exceptions to Nexpose from the Vulnerabilities Index

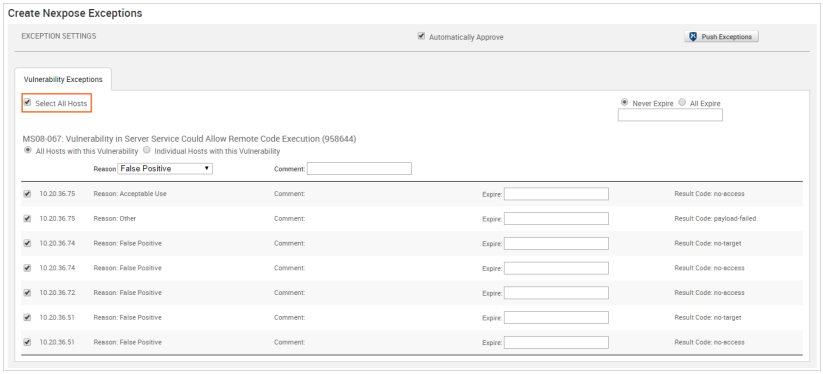
You can push from the vulnerabilities index if you want to push all exceptions or a select number of exceptions at the same time.

To push exceptions from the vulnerabilities index:

1. From within a project, select **Analysis > Vulnerabilities**. The vulnerabilities index appears.
2. Click the **Create Exceptions** dropdown and select the **All not exploited** option. The Create Nexpose Exceptions page appears.



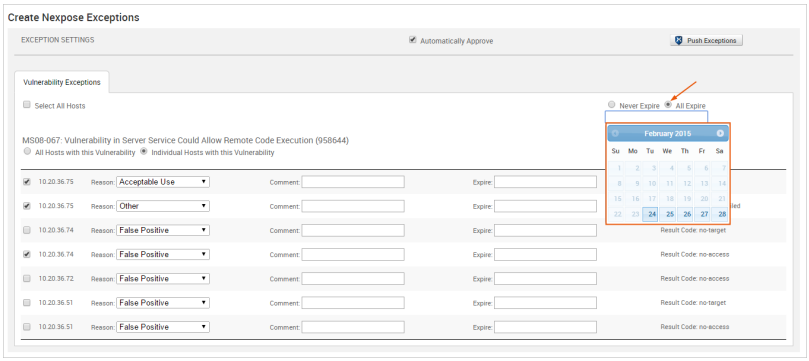
3. Select the hosts that you want to create exceptions for. Use the **Select All Hosts** checkbox if you want to create exceptions for all hosts that have a non-exploitable vulnerability.



4. For each vulnerability, click the **Reason** dropdown and choose the vulnerability exception reason you want to assign to it. You can also provide additional information for the exception in the **Comment** field.

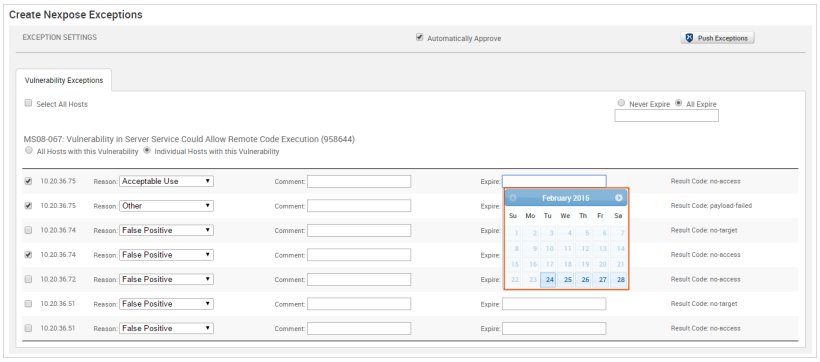
5. Choose the **All Expire** option if you want to set an expiration date for all the vulnerability exceptions. If you do not want to set an expiration date for any vulnerability exceptions, keep the default **Never Expire** option selected and go to Step 7.

   To set the same expiration date for all vulnerability exceptions, select on the **All Expire** option. A calendar appears. Find and select the date that you want to use.

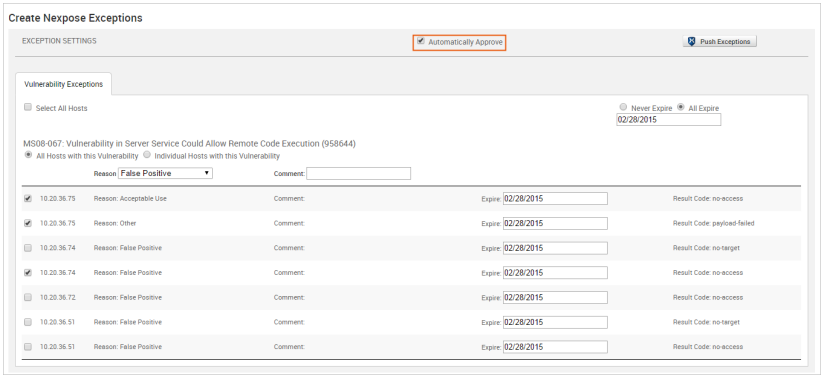


   If you want to set a unique expiration date for each host, skip this step and go to the next step.

6. To set a unique expiration date for each host, select the **Individual hosts with this vulnerability** option. Click on the **Expire** field next to each exception to display the calendar. Find the expiration date that you want to use and select it.

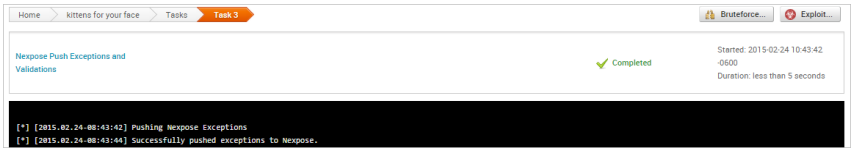


7. Verify that you want to approve all vulnerability exception requests from Metasploit Pro. If the **Automatically Approve** option is selected, Nexpose will automatically approve vulnerability exception requests imported from Metasploit Pro. Otherwise, the vulnerability exceptions will need to be manually reviewed and approved from the Nexpose console.



8. When you are ready to push the exceptions, click the **Push Exceptions** button.

   The task log appears and shows you the status of the push. If the push is successful, the message "Successfully pushed exceptions to Nexpose" appears in the task log.



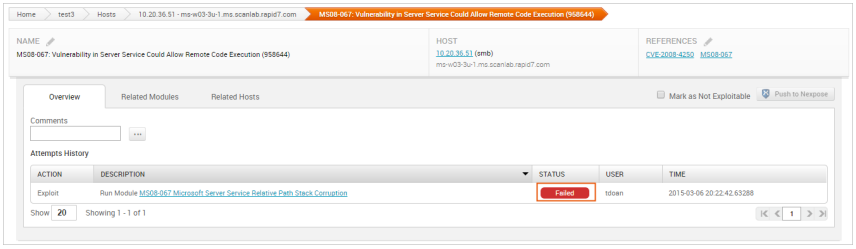
   If Metasploit is unable to reach the Nexpose console, an error message appears and alerts you that there is an issue with the console. You can click **Yes** to try the push again. If the error continues to persist, you will need to close the modal and diagnose the console connectivity.

Pushing Vulnerability Exceptions to Nexpose from the Single Vulnerability Page

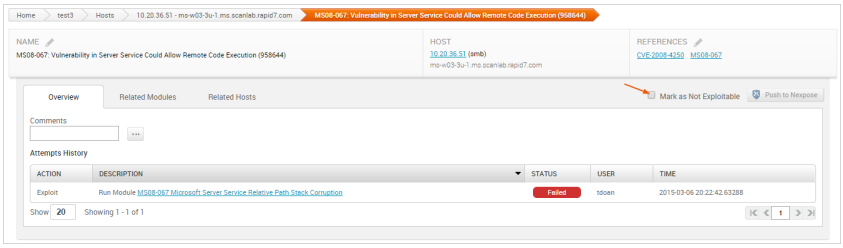
You can push from the single vulnerability page if you want to push a specific exception back to Nexpose.

1. From within a project, select **Analysis > Vulnerabilities**. The vulnerabilities index appears.
2. Find and click on the vulnerability that you want to push to Nexpose as an exception. Vulnerability exceptions can have a status of **Not Exploited** or **Not Tested**.

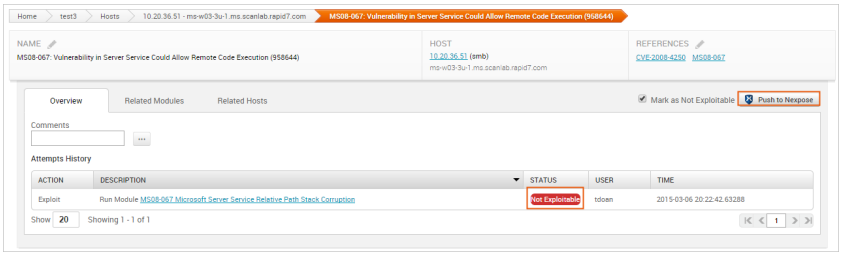
   When the single vulnerability page appears, the vulnerability should either have a status of **Failed** or it does not have a status if the vulnerability has not been tested.



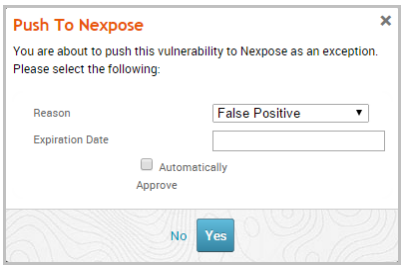
3. Select the **Mark as Not Exploitable** option.



After you select the option, the module status changes to **Not Exploitable** and the **Push** button becomes active.



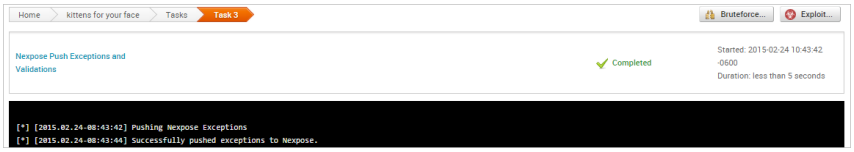
4. Click the **Push to Nexpose** button. The Push to Nexpose dialog appears.



5. Click the **Reason** dropdown and choose the vulnerability exception reason you want to assign to it.

6. Click the **Expiration Date** field and choose a date on which the exception will no longer be effective. If you do not want to specify an expiration date, leave this field empty.
7. Select the **Automatically Approve** option if you want to automatically approve vulnerability exception requests imported from Metasploit Pro. Otherwise, the vulnerability exceptions will need to be manually reviewed and approved from the Nexpose console.
8. Click the **Yes** button to push the exceptions to Nexpose.

   The task log appears and shows you the status of the push. If the push is successful, the message "Successfully pushed exceptions to Nexpose" appears in the task log.



   If Metasploit is unable to reach the Nexpose console, an error message appears and alerts you that there is an issue with the console. You can click **Yes** to try the push again. If the error continues to persist, you will need to close the modal and diagnose the console connectivity.

Updating Vulnerability Exceptions in Nexpose

At some point, you may want to update vulnerability validations and exceptions after they have been pushed from Metasploit Pro to Nexpose. In order to update vulnerability validations and exceptions after they have been pushed to Nexpose, you must log in to the Nexpose Console and manually update them. Currently, there is no way to update them from Metasploit Pro. For more information on how to manage exceptions, please take a look at the Nexpose User's Guide.

Tracking Real-Time Statistics and Events during Vulnerability Validation

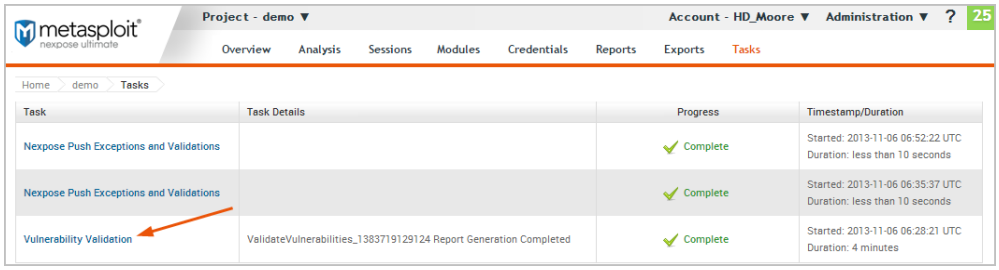
The Findings window displays the real-time statistics for the test and the task log. You can click on the tabs at the top of the Findings window to switch between the real-time statistics and the task log. You can also automatically push validated vulnerabilities and access the Vulnerabilities Exceptions configuration page.

Accessing the Findings Window

The Findings window automatically appears when you start the Vulnerability Validation Wizard. If you navigate away from the Findings window, you can go to the Tasks page to access it again.

To access the Findings Window:

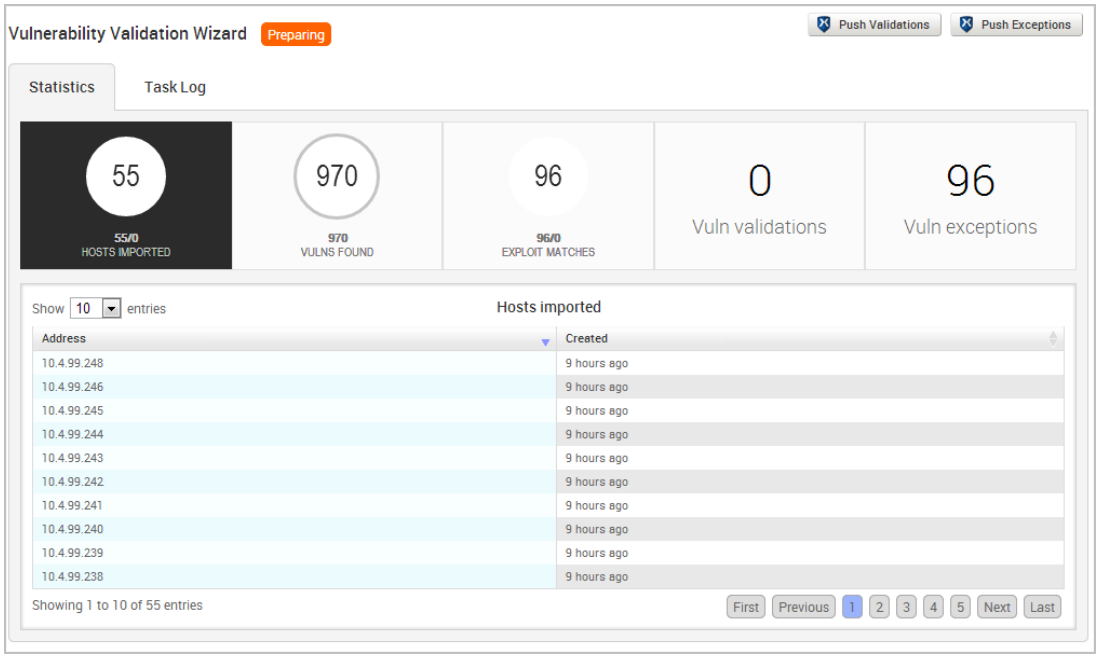
1. From within a project, select **Tasks > Show Tasks** from the Project Tab bar. The Tasks page appears.
2. Find the Vulnerability Validation task.



3. Click the **Vulnerability Validation** task name. The Findings window appears.

The Statistics Tab

The Statistics tab shows a high-level, count of hosts, vulnerabilities, and exploits. Each value is displayed in a stat bubble with an orange progress bar. The progress bar wraps around the stat bubble and only displays when there is activity occurring for a particular finding.

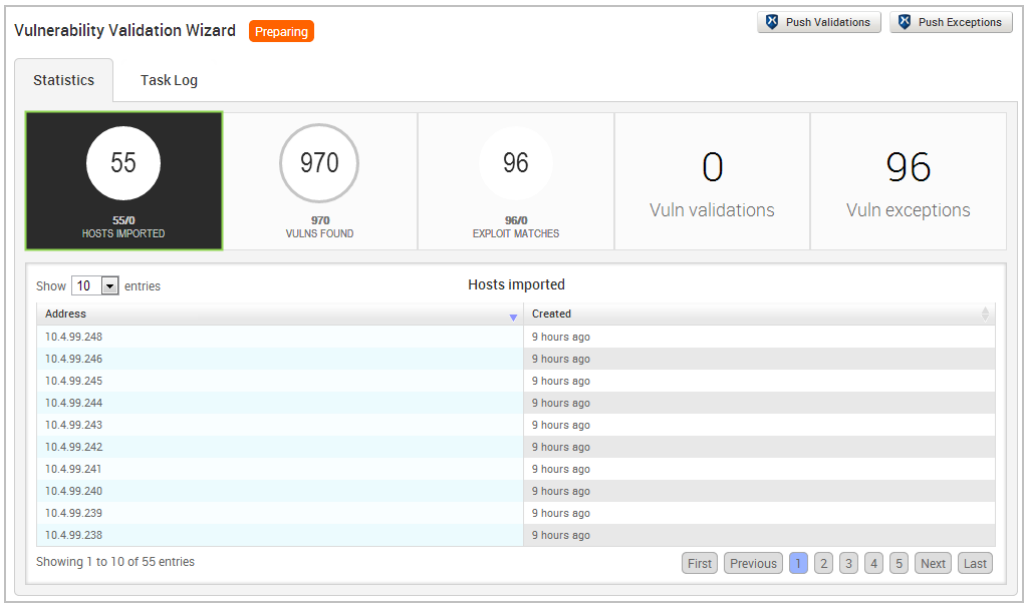


From the Statistics tab, you can track the following data:

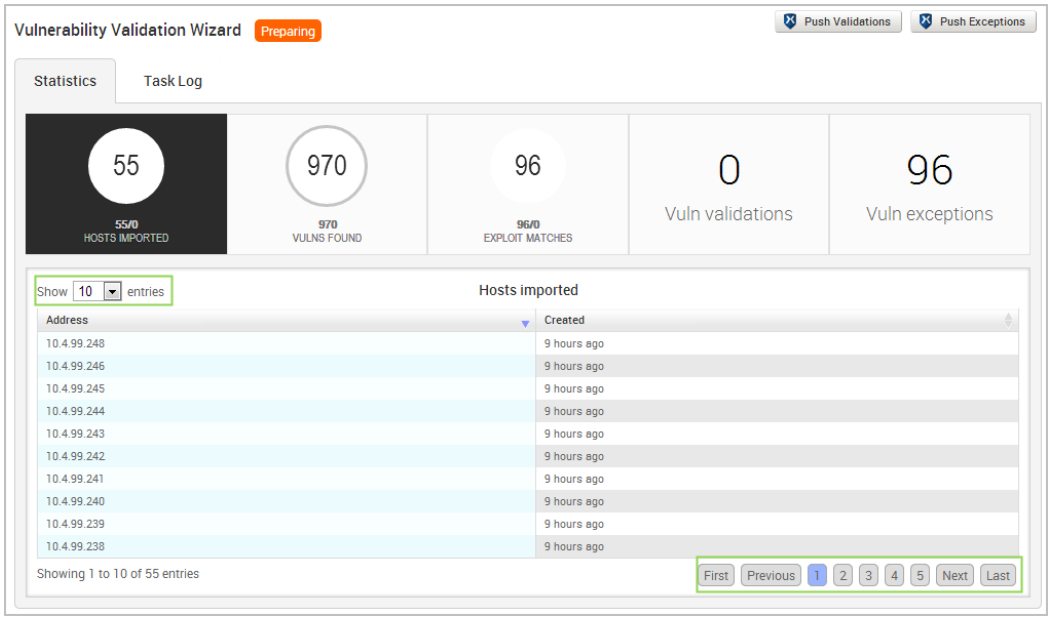
- The total number of hosts that have been scanned or imported.
- The total number of unique vulnerabilities that have been identified.
- The total number of exploit modules that match Nexpose vulnerabilities.
- The total number of vulnerabilities that Metasploit Pro was able to exploit.
- The total number of vulnerabilities that Metasploit Pro was unable to exploit.

Viewing a List of Imported Hosts from the Findings Window

1. Open the Findings window.
2. Click on the **Hosts Imported** tab. The Hosts list appears and displays the IP addresses for each host that has been imported from a Nexpose site.

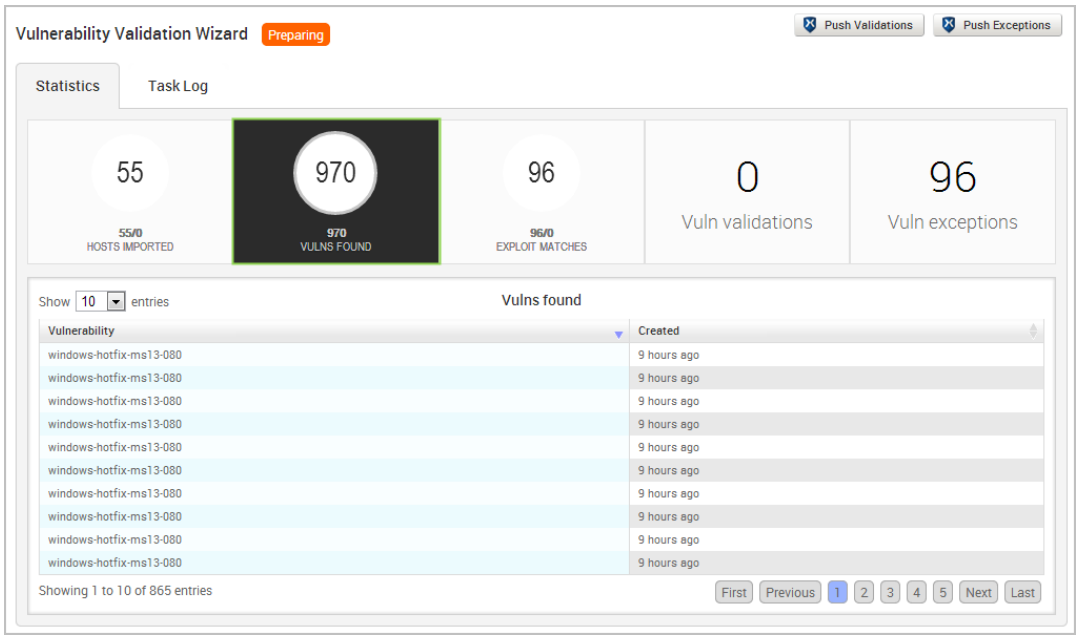


3. Use the navigational page buttons to view more hosts or click the **Show Entries** dropdown to expand the number of hosts displayed.

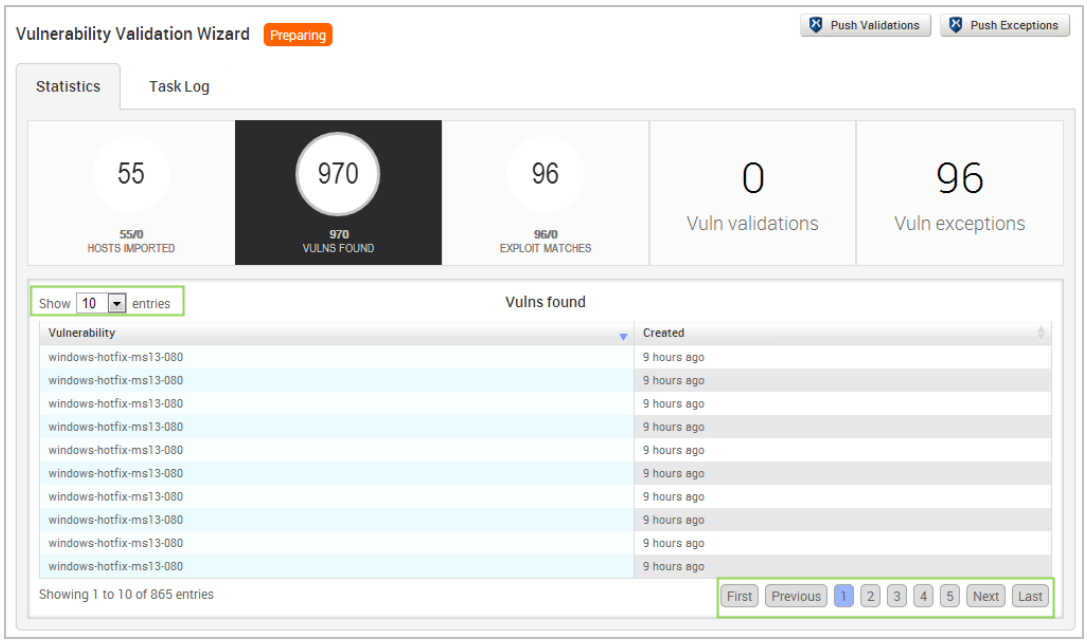


Viewing a List of Imported Vulnerabilities from the Findings Window

1. Open the Findings Window.
2. Click the **Vulns Found** tab. A list of imported vulnerabilities appears.

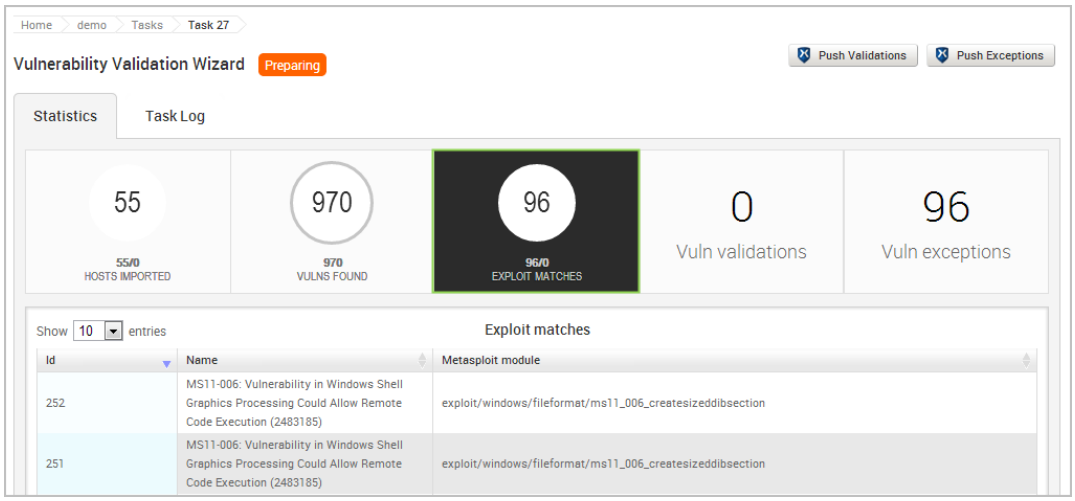


3. Use the navigational page buttons to view more hosts or click the **Show Entries** dropdown to expand the number of vulnerabilities displayed.



Viewing a List of Exploit Matches from the Findings Window

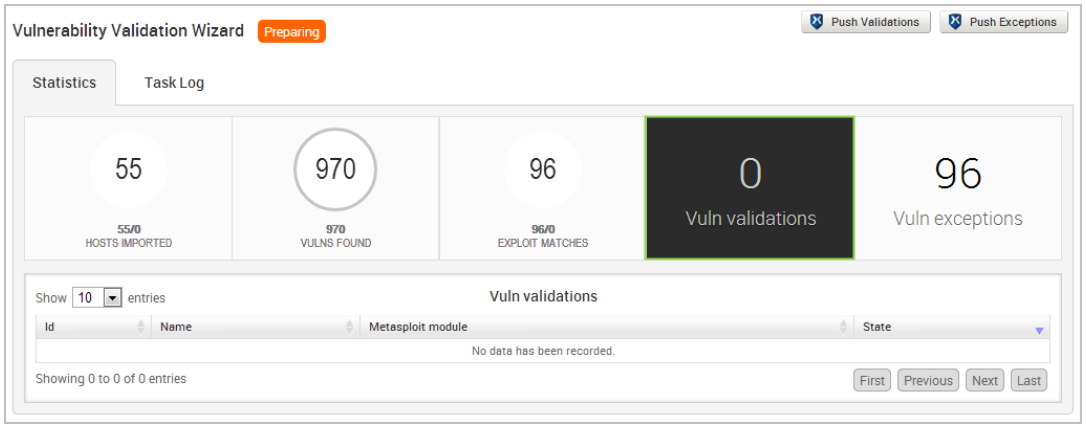
1. Open the Findings Window.
2. Click the **Exploit Matches** tab. A list of imported vulnerabilities appears.



3. Use the navigational page buttons to view more hosts or click the **Show Entries** dropdown to expand the number of exploit modules displayed.

Viewing a List of Validated Vulnerabilities from the Findings Window

1. Open the Findings Window.
2. Click the **Vulns validations** tab. A list of imported vulnerabilities appears.

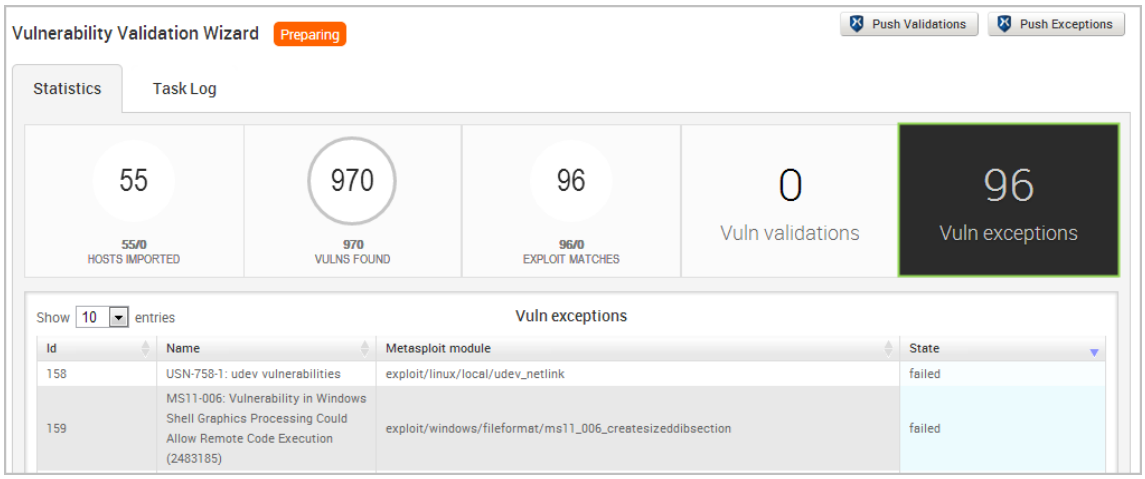


You can view the vulnerability name, the exploit module that was run against the vulnerability, and the result of the exploit. For vulnerability validations, the state will be `exploited`.

3. Use the navigational page buttons to view more hosts or click the **Show Entries** dropdown to expand the number of validations displayed.

Viewing a List of Vulnerability Exceptions from the Findings Window

1. Open the Findings Window.
2. Click the **Vulns exceptions** tab. A list of vulnerability exceptions appears.



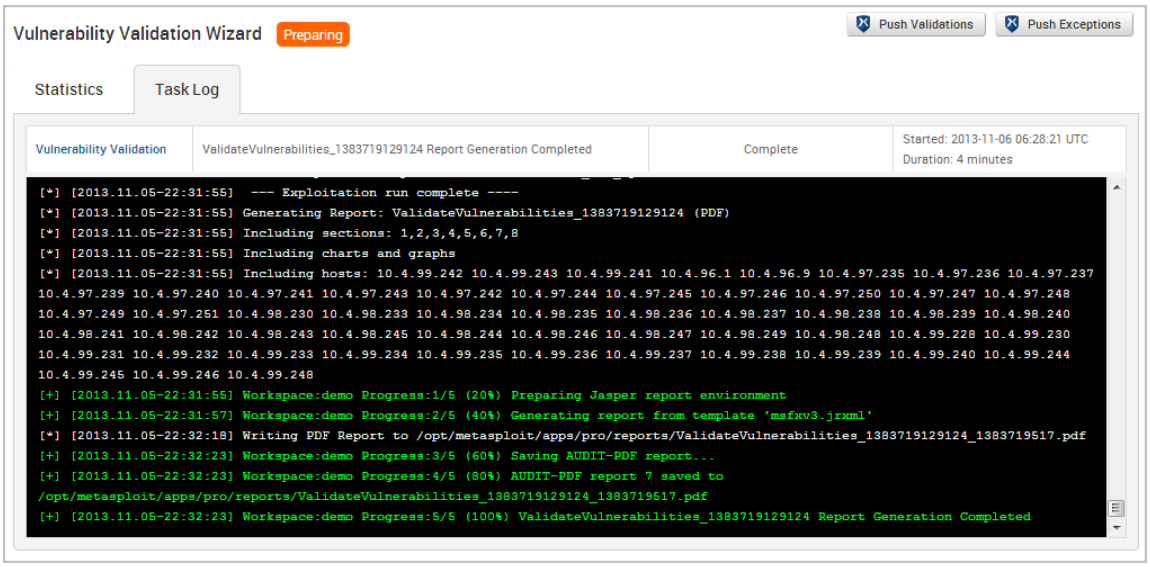
   You can view the vulnerability name, the exploit module that was run against the vulnerability, and the result of the exploit. For vulnerability exceptions, the state will be `failed`.

3. Use the navigational page buttons to view more hosts or click the **Show Entries** dropdown to expand the number of exceptions displayed.

The Tasks Log Tab

The Tasks Log tab shows a detailed activity log for the Vulnerability Validation Wizard. Each task that Metasploit Pro performs is documented in the Tasks Log. For example, you can view the assets and vulnerability definitions as they are being imported into a project or you can view the exploit modules as they are being run. If you have chosen to perform a dry run of the auto-exploitation task, you can go to the Tasks Log to view the proposed attack plan.

Additionally, the Tasks log shows you the current state of the test, the start time of the test, and the amount of time that the test has been running.


Vulnerability Validation Wizard
Preparing
Push Validations
Push Exceptions
Statistics
Task Log
Vulnerability Validation
ValidateVulnerabilities_1383719129124 Report Generation Completed
Complete
Started: 2013-11-06 06:28:21 UTC
Duration: 4 minutes
[*] [2013.11.05-22:31:55] --- Exploitation run complete ----
[*] [2013.11.05-22:31:55] Generating Report: ValidateVulnerabilities_1383719129124 (PDF)
[*] [2013.11.05-22:31:55] Including sections: 1,2,3,4,5,6,7,8
[*] [2013.11.05-22:31:55] Including charts and graphs
[*] [2013.11.05-22:31:55] Including hosts: 10.4.99.242 10.4.99.243 10.4.99.241 10.4.96.1 10.4.96.9 10.4.97.235 10.4.97.236 10.4.97.237
10.4.97.239 10.4.97.240 10.4.97.241 10.4.97.243 10.4.97.242 10.4.97.244 10.4.97.245 10.4.97.246 10.4.97.250 10.4.97.247 10.4.97.248
10.4.97.249 10.4.97.251 10.4.98.230 10.4.98.233 10.4.98.234 10.4.98.235 10.4.98.236 10.4.98.237 10.4.98.238 10.4.98.239 10.4.98.240
10.4.98.241 10.4.98.242 10.4.98.243 10.4.98.245 10.4.98.244 10.4.98.246 10.4.98.247 10.4.98.249 10.4.98.248 10.4.99.228 10.4.99.230
10.4.99.231 10.4.99.232 10.4.99.233 10.4.99.234 10.4.99.235 10.4.99.236 10.4.99.237 10.4.99.238 10.4.99.239 10.4.99.240 10.4.99.244
10.4.99.245 10.4.99.246 10.4.99.248
[+] [2013.11.05-22:31:55] Workspace:demo Progress:1/5 (20%) Preparing Jasper report environment
[+] [2013.11.05-22:31:57] Workspace:demo Progress:2/5 (40%) Generating report from template 'msfxv3.jrxml'
[*] [2013.11.05-22:32:18] Writing PDF Report to /opt/metasploit/apps/pro/reports/ValidateVulnerabilities_1383719129124_1383719517.pdf
[+] [2013.11.05-22:32:23] Workspace:demo Progress:3/5 (60%) Saving AUDIT-PDF report...
[+] [2013.11.05-22:32:23] Workspace:demo Progress:4/5 (80%) AUDIT-PDF report 7 saved to
/opt/metasploit/apps/pro/reports/ValidateVulnerabilities_1383719129124_1383719517.pdf
[+] [2013.11.05-22:32:23] Workspace:demo Progress:5/5 (100%) ValidateVulnerabilities_1383719129124 Report Generation Completed

Managing Nexpose Exceptions

An exception defines the reason why a vulnerability exists. You apply exceptions to vulnerabilities that are typically low-risk or are used deliberately to mitigate bigger threats. Vulnerability exceptions help you exclude certain vulnerabilities from a report so that you can manage your risk score.

You can apply exceptions to vulnerabilities that Metasploit Pro was unable to exploit. These vulnerabilities have a status of Not Exploitable, which indicates that Metasploit Pro was unable to obtain a session on the target host due to some compensating control or back porting.

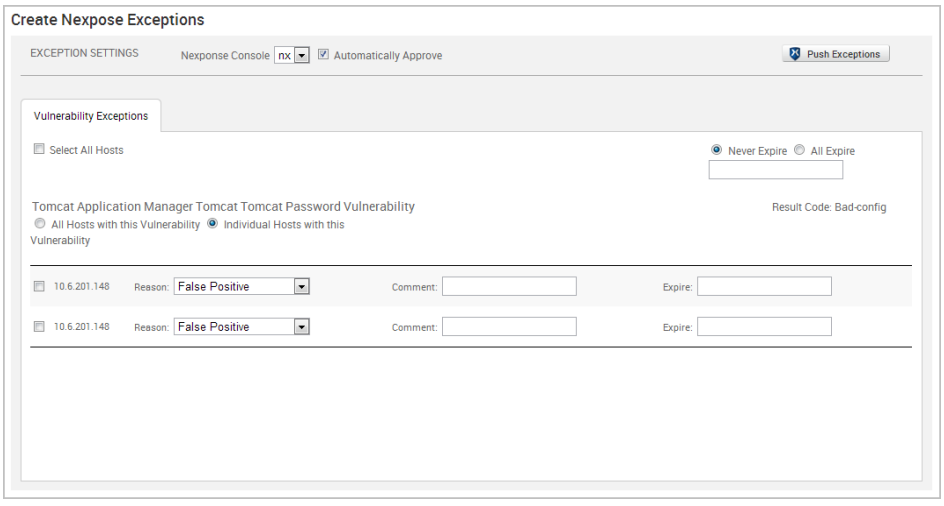
Typically, exceptions can be defined for vulnerabilities for the following reasons:

- They are used as compensating controls or to mitigate additional risks.
- They represent an acceptable use case or deliberate practice, such as anonymous FTP access.
- They represent an acceptable risk and may require more resources than you are willing to invest to remediate.These vulnerabilities typically pose a minimal risk.
- They are false positives.

The Exceptions Page

You create and push Nexpose exceptions from the Exceptions page. The Exceptions page is accessible from the Findings window or from the Vulnerabilities page.

From the Exceptions page, you can define the exception settings for a group of hosts that have a specific vulnerability or you can define them individually for each host.



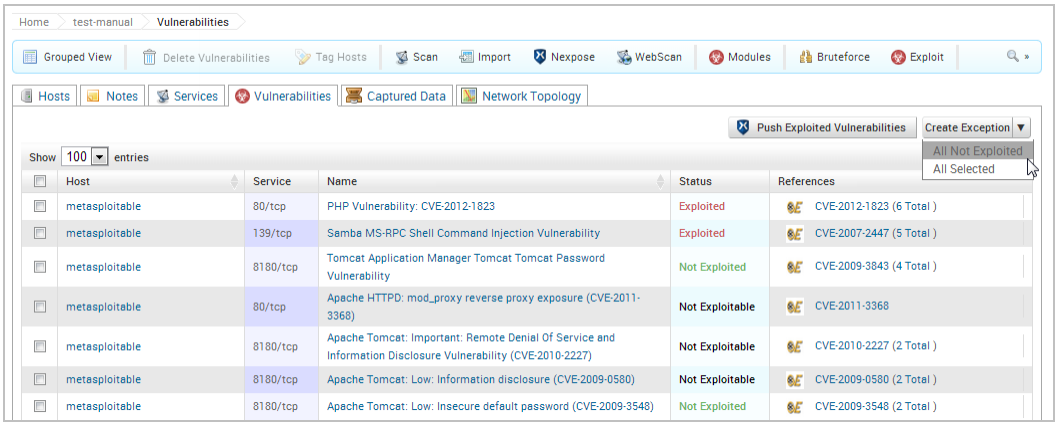
From the Exceptions page, you can perform the following tasks:

- View all the vulnerabilities that Metasploit Pro was unable to exploit.
- Assign the exception reason for each vulnerability.
- Assign expiration dates for vulnerability exceptions.
- Add comments to the vulnerability exception.
- Automatically approve vulnerability exception requests.
- Push exceptions back to Nexpose.

Accessing the Exceptions Page

1. From within a project, select **Analysis > Vulnerabilities** from the Project Tab bar. The Vulnerabilities page appears.
2. Click the **Create Exception** dropdown and select **All Not Exploited**. The Exceptions page appears.

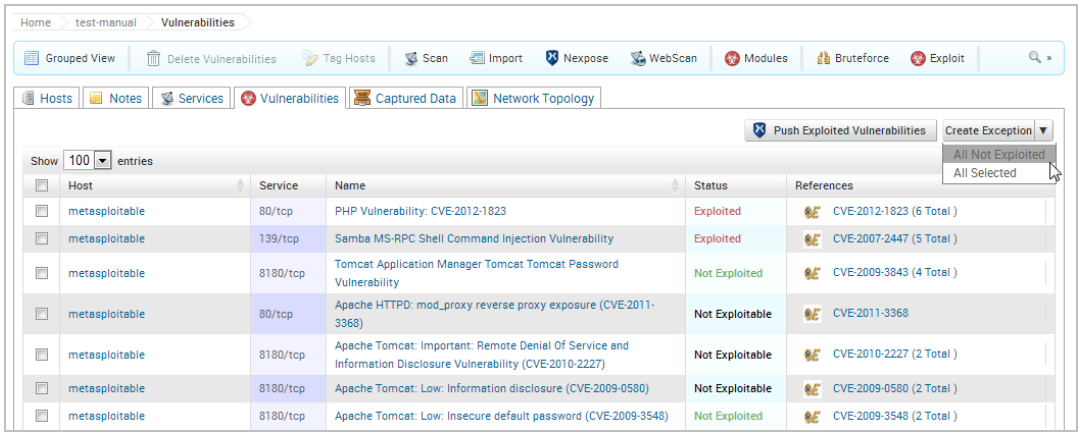
   Note: The **All Not Exploited** option selects all vulnerabilities that have an **Not Exploited** status and displays them on the Exceptions page. If you only want to create exceptions for a few specific vulnerabilities, you can manually select them from the Vulnerabilities list and choose the **All Selected** option instead.



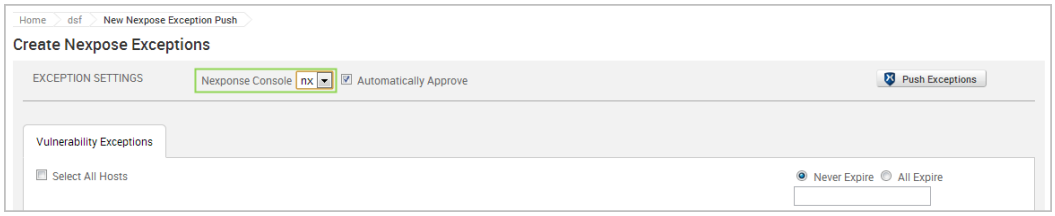
Creating and Pushing Nexpose Exceptions

1. From within a project, select **Analysis > Vulnerabilities** from the Project Tab bar. The Vulnerabilities page appears.
2. Click the **Create Exception** dropdown and select **All Not Exploited**. The Exceptions page appears.

Note: The **Create Exceptions** button is available on the Findings window when there are vulnerabilities ready for you to create exceptions for; otherwise, it will be grayed out.

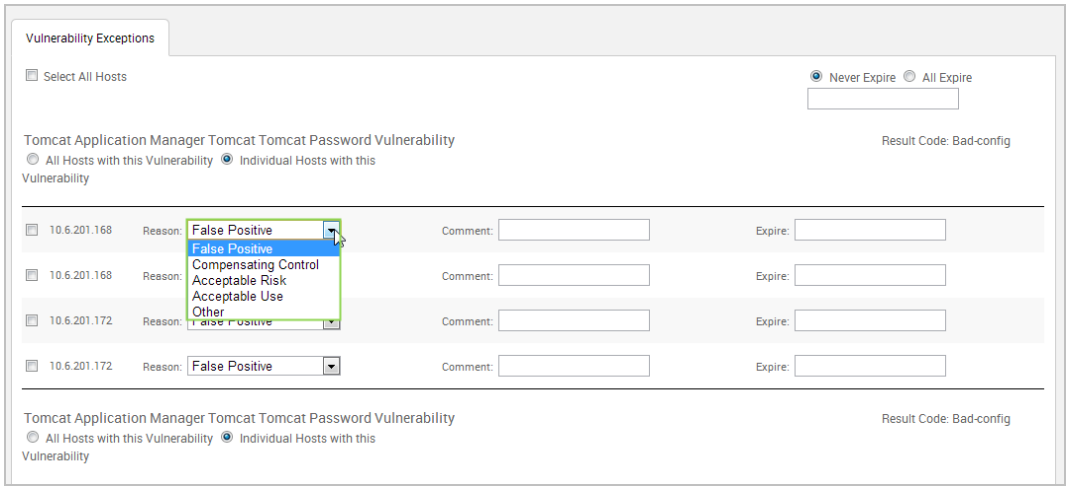


3. Click the **Nexpose Console** dropdown and select the console you want to push the exceptions to.

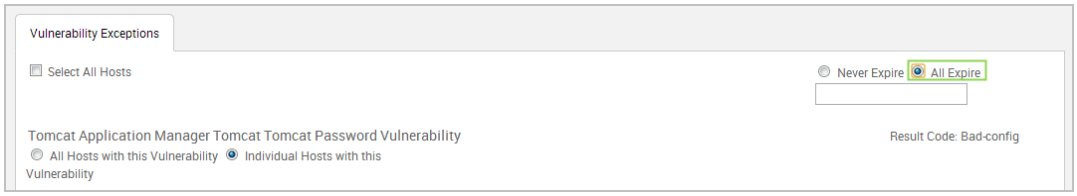


4. For each vulnerability, click the **Reason** dropdown and choose the vulnerability exception reason you want to assign to it. You can also provide additional information for the exception in the **Comment** field. For more information on exception reasons, see *Vulnerability Exception Reasons* on page 95.

Note: If you want to define bulk exception settings for all hosts in a vulnerability group, select the **All Hosts with this Vulnerability** option. The Reason and Comment fields become available under to the vulnerability name.The reason you select applies to all hosts in that vulnerability group.



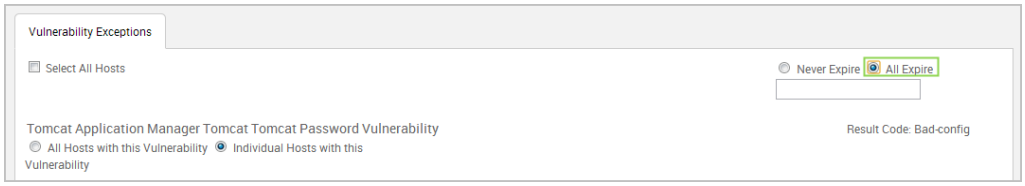
5. Choose the **All Expire** option if you want to set an expiration date for the vulnerability exceptions. If you do not want to set an expiration date for any vulnerability exceptions, keep the default **Never Expire** option selected and go to Step 6.



6. To set the same expiration date for all vulnerability exceptions, select on the **All Expire** option. A calendar appears. Find and select the date that you want to use. If you want to set a unique expiration date for each host, skip this step and go to the next step.

7. To set a unique expiration date for each host:

   a. Select the **All Expire** option.

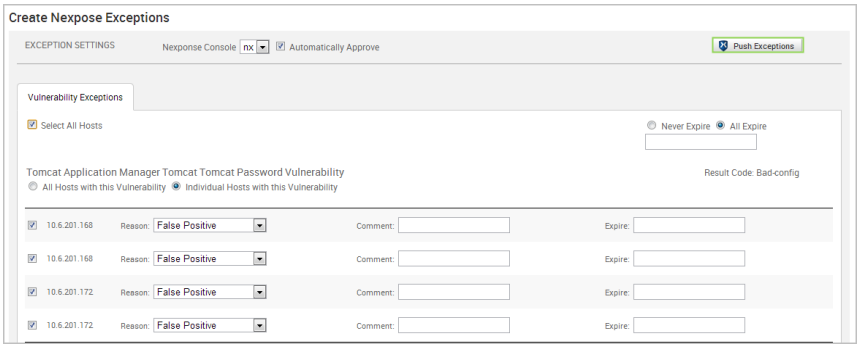


   b. Click on the **Expire** field next to each host to display the calendar.

   c. Find the expiration date that you want to use and select it.

8. Deselect the **Automatically Approve** option if you do not want to approve any of the vulnerability exception requests from Metasploit Pro. Instead, you will manually approve the exception requests through the Nexpose Console.

9. Select the hosts that you want to push exceptions for. Use the **Select All Hosts** checkbox if you want to push exceptions for all hosts.
10. When you are ready to push the exceptions, click the **Push Exceptions** button.



Vulnerability Exception Reasons

The following vulnerability exception reasons are available:

- **False positive** - Use this exception reason for a vulnerability that does not exist.
- **Compensating control** - Use this exception reason to indicate that a vulnerability is a compensating control, or a workaround for a security requirement.
- **Acceptable use** - Use this exception reason for any vulnerability that is used as part of organizational practices.
- **Acceptable risk** - Use this exception reason for any vulnerability that are considered low risk. These vulnerabilities tend to pose minimal security risk and are likely to consume more resources than they are worth.
- **Other** - Use this exception reason if the reason does not match any of the . You can provide a reason in the **Comment** field.

Module Result Codes

A result code identifies the reason an exploit failed. You can view the result code for a vulnerability on the Vulnerability Exceptions page.

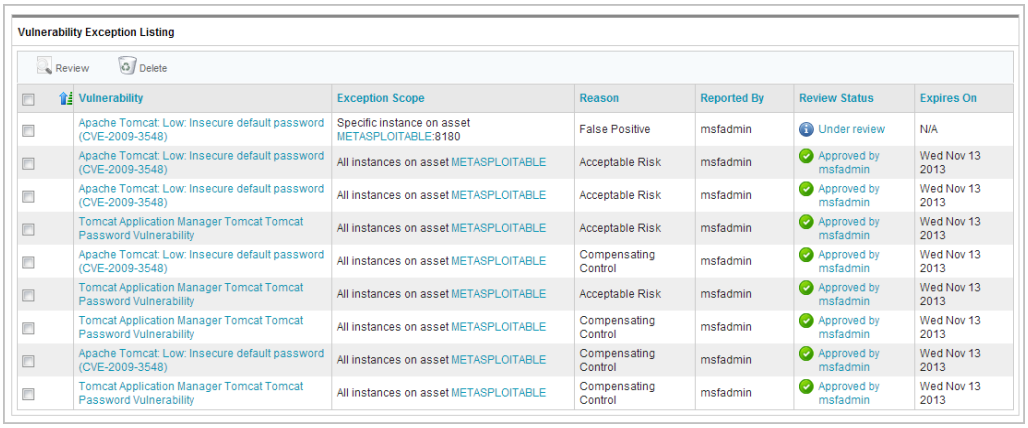
The following result codes are available:

None - Indicates that Metasploit Pro could not determine if the module ran successfully or failed.

- **Unknown** - Indicates that Metasploit Pro could not determine if the module ran successfully or failed.
- **Unreachable** - Indicates that Metasploit Pro could not reach the network service.
- **Bad-config** - Indicates that the exploit settings were configured incorrectly.
- **Disconnected** - Indicates that the network service disconnected during a module run.
- **Not-found** - Indicates that Metasploit Pro could not find the application or service.
- **Unexpected-reply** - Indicates that Metasploit Pro did not receive the expected response from the application.
- **Timeout-expired** - Indicates that a timeout occurred.
- **User-interrupt** - Indicates that the user stopped the module run.
- **No-access** - Indicates that Metasploit Pro could not access the application.
- **No-target** - Indicates that the module configuration was not compatible with the target.
- **Not-vulnerable** - Indicates that the application was not vulnerable.
- **Payload-failed** - Indicates that Metasploit Pro delivered a payload, but was unable to open a session.

Viewing Vulnerability Exceptions in Nexpose

After you push the exceptions, you can go to the Vulnerability Exception Listing (**Administration > Exceptions and Overrides > Manage**) in the Nexpose Console to view the exception requests that have been approved or are awaiting review. The Vulnerability Exception Listing shows the exceptions that are active across all sites. For more information on how to manage vulnerability exceptions, please see the Nexpose User's Guide.



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.

Metasploit Pro offers automated exploits and manual exploits. The type of exploit that you use depends on the level of granular control you want over the exploits.

When you run an automated exploit, Metasploit Pro builds an attack plan based on the service, operating system, and vulnerability information that it has for the target system. Automated exploits cross reference open ports, imported vulnerabilities, and fingerprint information with exploit modules. The attack plan defines the exploit modules that Metasploit Pro will use to attack the target systems.

An automated exploit uses reverse connect or bind listener payloads and does not abuse normal authenticated control mechanisms.

To run an automated exploit, you must specify the hosts that you want to exploit and the minimum reliability setting that Metasploit Pro should use. The minimum reliability setting indicates the potential impact that the exploits have on the target system. If you use a high ranking, such as excellent or great, Metasploit Pro uses exploits that will be unlikely to crash the service or system. Exploits that typically have a high reliability ranking include SQL injection exploits, web application exploits, and command execution exploits. Exploits that corrupt memory will most likely not have a high reliability ranking.

You can also specify the payload type that you want the exploit to use. By default, automated exploits use Meterpreter, but you can choose to use a command shell instead.

Automated Exploits

When you run an automated exploit, Metasploit Pro builds an attack plan based on the service, operating system, and vulnerability information that it has for the target system. Automated exploits cross reference open ports, imported vulnerabilities, and fingerprint information with exploit modules. The attack plan defines the exploit modules that Metasploit Pro will use to attack the target systems.

An automated exploit uses reverse connect or bind listener payloads and does not abuse normal authenticated control mechanisms.

To run an automated exploit, you must specify the hosts that you want to exploit and the minimum reliability setting that Metasploit Pro should use. The minimum reliability setting indicates the potential impact that the exploits have on the target system. If you use a high ranking, such as excellent or great, Metasploit Pro uses exploits that will be unlikely to crash the service or system. Exploits that typically

have a high reliability ranking include SQL injection exploits, web application exploits, and command execution exploits. Exploits that corrupt memory will most likely not have a high reliability ranking.

You can also specify the payload type that you want the exploit to use. By default, automated exploits use Meterpreter, but you can choose to use a command shell instead.

Running Automated Exploits

1. From within a project, click the **Analysis** tab.
2. When the Hosts window appears, select the hosts that you want to exploit and click the **Exploit** button.
3. When the New Automated Exploitation Attempt window appears, verify that target address field contains the addresses that you want to exploit.
4. Select the minimum reliability for the exploit.
5. Define the hosts that you want to exclude from the exploit.
6. Define the payload options. This determines the type of payload the exploit uses, the type of connection the payload creates, and the listener ports that the exploit uses.
7. Define the exploit selection options. This determines the ports that the exploit includes and excludes from the attack.
8. Define the advanced options. The advanced options lets you define the number of exploits you can run concurrently, the time out for each exploit, and evasion options.
9. Run the exploit.

Configuring Auto-Exploitation Options

- Dry Run : Prints a transcript of the exploits in the attack plan without running them.
- Collect Evidence: Collects loot, such as screenshots, system files, passwords, and configuration settings from open sessions.
- Clean Up Sessions: Closes all sessions after all tasks have run.
- Payload Type: Specifies the type of payload that the exploit will deliver to the target. Choose one of the following payload types:
  - **Command**: A command execution payload that enables you to execute commands on the remote machine.
  - **Meterpreter**: An advanced payload that provides a command line that enables you to deliver commands and inject extensions on the fly.

- Connection Type: Specifies how you want your Metasploit instance to connect to the target. Choose one of the following connection types:
  - **Auto**: Automatically uses a bind connection when NAT is detected; otherwise, a reverse connection is used.
  - **Bind**: Uses a bind connection, which is useful when the targets are behind a firewall or a NAT gateway.
  - **Reverse**: Uses a reverse connection, which is useful if your system is unable to initiate connections to the targets.
- Listener Ports: Defines the ports that you want to use for reverse connections.
- Listener Host: Defines the IP address you want to connect back to.
- Auto Launch Macro: Specifies the macro that you want to run during post-exploitation.
- Concurrent Exploits: Specifies the number of exploit attempts you want to launch at one time.
- Timeout in Minutes: Defines the number of minutes an exploit waits before it times out.
- Transport Evasion: Choose from the following transport evasion levels:
  - **Low**: Inserts delays between TCP packets.
  - **Medium**: Sends small TCP packets.
  - **High**: Sends small TCP packets and inserts delays between them.
- Application Evasion: Adjusts application-specific evasion options for exploits involving DCERPC, SMB and HTTP. The higher the application evasion level, the more evasion techniques are applied.
- Included Ports: Defines the specific ports you want to target for exploitation.
- Excluded Ports: Defines the specific ports you want to exclude from exploitation.

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.

Manual exploitation provides granular control over the module and evasion options that an exploit uses. Whereas automated exploits enable you to run simultaneously multiple exploits, manual exploits enable you to run one exploit at a time.

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.

Searching for Exploits

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.

1. From within a project, click the **Modules** tab.
2. In the **Search Modules** field, enter a keyword expression to search for a specific exploit.
3. Use the keyword tags to define the keyword expression.
4. Press **Enter** to perform the search.

Running Automated Exploits

1. From within a project, click the **Analysis** tab.
2. When the Hosts window appears, select the hosts that you want to exploit.
3. Click **Exploit**.
4. When the New Automated Exploitation Attempt window appears, verify that target address field contains the addresses that you want to exploit.
5. Select the minimum reliability for the exploit.
6. Click **Show Advanced Options**.
7. Define the target hosts that you want to include or exclude from the exploit.
8. Define the payload options. This determines the type of payload the exploit uses, the type of connection the payload creates, and the listener ports that the exploit uses.
9. Define the exploit selection options. This determines the ports that the exploit includes and excludes from the attack.
10. Define the advanced options. The advanced options lets you define the number of exploits you can run concurrently, the time out for each exploit, and evasion options.
11. Run the exploit.

Setting Up a Listener

1. Select **Administration > Global Settings** from the main menu.
2. Click **New Listener**, which is located under Persistent Listeners.
3. When the Create a Listener window appears, choose an associated project for the listener.
4. Define the listener payload type.
5. Enter an IP address for the listener.
6. Enter a port for the listener.
7. Choose a post-exploitation macro to deploy after the listener connects to the target system. Enable the listener.
8. Save the listener.

Credentials

These days, more and more organizations are becoming vulnerable to outside threats due to weak password policies and insecure password management systems. Credentials provide a gateway into various accounts and systems, which can potentially give access to additional targets on the network and lead to the extraction of confidential data from these targets. Therefore, as part of a penetration test, it is important to discover and present credential data that compels organizations to strengthen and enforce complex password policies to prevent vulnerabilities like password reuse and weak passwords.

As part of your credentials audit, you want to identify weak passwords, the most commonly used passwords, and top base passwords. You will also want to reuse valid credentials, so that you can identify the impact of the stolen credentials across a network.This will help an organization understand their current posture, identify how they can strengthen password policies, and enforce passwords requirements that meet industry best practices.

To help you understand how credentials are obtained, stored, and managed by Metasploit Pro, the following section will provide an overview of the key concepts and terms you must know before working with credentials.

Understanding Credential Terminology

Typically, when you think of a credential, you think of a username and password. In Metasploit Pro, a username is referred to as a public, and the password is known as a private; therefore, a credential can be a private, public, or a credential pair.

To summarize the key credential terms:

- A **public** is the username that is used to log in to a target.
- A **private** is essentially the password that is used to authenticate to a target. It is usually a plaintext password, an SSH key, NTLM hash, or nonreplayable hash. Since the private can be an SSH key or hash, the term password is not broad enough to include these private types.
- A **credential pair** is a public and private combination that can be used to authenticate to a target.

In addition to the key terms, here are some additional ones that you should familiarize yourself with:

Private Type

A **private type** refers to whether the private is a plaintext password, an SSH key, NTLM hash, or nonreplayable hash.

Nonreplayable Hash

A **nonreplayable hash** is a hash that cannot be replayed to authenticate to services. For example, any hash that was looted from `/etc/passwd` or `/etc/shadow` are nonreplayable hashes.

NTLM Hash

An **NTLM hash** is a hash that can be replayed to authenticate to SMB.

Realm

A **realm** refers to the functional grouping of database schemas to which the credential belongs. A realm type can be an Active Domain Directory, a Postgres database, a DB2 database, or an Oracle System Identifier (SID). A public, private, or credential pair can have a realm, but it is not mandatory.

Incomplete Public

An **incomplete public** refers a public that does not have a private. It can have a realm, but it is not required.

Incomplete Private

An **incomplete private** refers to a private that does not have a public. It can have a realm, but it is not required.

Login

A login refers to a username and private that is associated with a particular service. A login indicates that you can theoretically authenticate to a service using the credential pair. Metasploit Pro creates logins when it collects evidence from an exploited target and when it successfully bruteforces a target. During exploitation, if a host is successfully looted, Metasploit Pro will attempt to create logins based on the type of credential that was captured. For example, if NTLM hashes were looted, then a login for SMB will be added for each hash.

For example, a credential pair, such as admin/admin, that can be used to authenticate to a service, like telnet, is a login.

Origin

The **origin** refers to how the credential was obtained or added to the project, such as through Bruteforce Guess, manual entry, or an imported credentials list. A origin can be manual, import, session, service, or cracked password.

Validated Credential

A **validated credential** refers to a credential that has successfully authenticated to a target.

To obtain credentials, you must loot them from a compromised target. Looted credentials can be used to attempt additional logins to other services and hosts.

Metasploit Pro stores all looted, imported, and manually added credentials in a project.

Obtaining Credentials

There are a few ways that you can obtain credentials. The main methods of acquiring credentials include exploiting a vulnerability and dumping the credentials from the compromised target; bruteforcing targets using weak and common default credentials; and searching publicly available resources for stolen credentials. The method you use depends on the level of access that you have to a target.

Metasploit Pro enables you to leverage multiple attack methods to acquire credentials, such as exploiting unpatched vulnerabilities. For example, if you are able to discover a Windows system that is vulnerable to MS08-067, you may be able to exploit that target and log in to the system to gather information from it. With access to the system, you can extract data such as password hashes, plaintext passwords, and domain tokens.

Many information systems are configured to use passwords as the first, and sometimes only, line of defense. And oftentimes, the passwords that are used are easy to guess passwords, or even blank passwords. This means that if you have the username, you can try to guess the password to log in to the target. For example, a Windows domain account that uses a weak or blank password can be easily guessed via bruteforce.

Additionally, many systems are configured with the default account settings. These accounts usually share the same password across multiple instances, which means that if you know the default account settings for one account, you will be able to leverage those credentials to compromise other targets across the network as well. In this case, you can manually add common default credentials and use the Quick Validation feature to validate the account credentials. If any credentials successfully authenticate to a target, you can run Credential Reuse to find additional targets on which the credentials are valid.

So, to summarize the methods that you can use to obtain credentials with Metasploit:

- You can find vulnerabilities and exploit them to obtain access to the target. Once you have access to a target, you can dump credentials and other confidential data from the exploited target.
- You can run Bruteforce to guess commonly used, weak, and default credentials on services like AFP, DB2, FTP, HTTP, HTTPS, MSSQL, MySQL, POP3, PostgreSQL, SMB, SNMP, SSH, telnet, VNC, and WinRM.

- You can manually add or import credentials to a project and run Quick Validation or Credential Reuse to find targets that can be authenticated. This method is useful when you have a set of commonly used credentials or known credentials you want to try on a set of targets.

Credential Origins

Every credential that is added to a project has an origin, which refers to the source of the credential. A credential will always have one of the following origins:

- A **manual** origin indicates that you manually added the credential to the project using the Add button on the Manage Credentials page.
- An **import** origin indicates that you imported the credential by uploading a CSV file or PWDump to the project.
- A **service** origin indicates that the credential was obtained with Bruteforce.
- A **session** origin indicates that the credential was collected from a session on an exploited target.
- A **cracked password** origin indicates that Metasploit Pro was able to crack the hash during evidence collection and decipher the plaintext password.

Managing Credentials

During a credentials audit, you will be collecting sensitive data from your targets and managing it from the Manage Credentials page. The Manage Credentials page displays all the credentials that are available in a particular project and provides access to features that let you add, delete, and export credential data. The following sections will show you how you can manage credential data within a project.

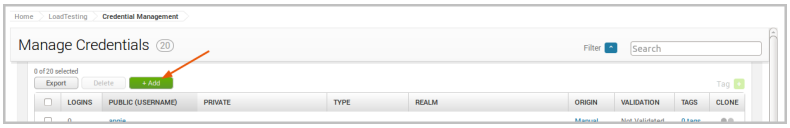
Adding Credentials

To add credentials to a project, you can either manually input each credential individually or you can import a PWDump or CSV file. The following sections show you how to manually add a plaintext password, SSH key, NTLM hash, and nonreplayable hash.

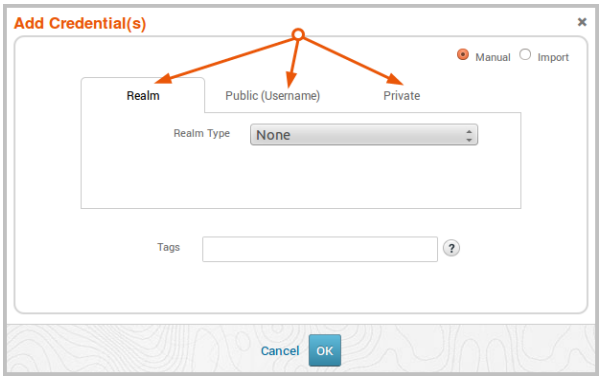
Manually Entering a Password

You can manually add a password when you have a single plaintext password that you want to add to a project, such as a common default like admin/admin. If you have multiple credentials that you want to add, you should create a CSV file for them and import them into the project. Importing credentials, in that particular case, will be much more efficient.

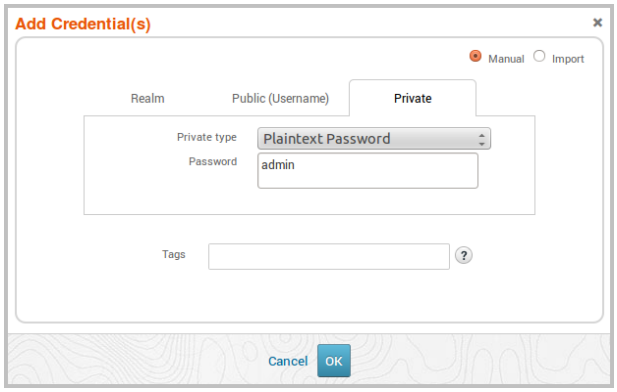
1. From within a project, go to **Credentials > Manage** to access the Manage Credentials area.
2. When the Manage Credentials page appears, click the **Add** button.



The **Add Credentials** window appears and displays tabs for the three different parts of a credential: the realm, the public , and the private. You can click on any of the tabs to configure their options.



3. Click the **Private (Passwords)** tab.
4. Click the **Credential Type** dropdown and select **Plaintext Password**.



5. Enter the password in the **Password** field.
6. Click the **Public (Username)** tab and enter the username. The username will be *BLANK* if you do not specify one. (Optional)

7. Click the **Realm** tab and select one of the following realm types: None, Active Directory Domain, Postgres DB, DB2, or Oracle SID.(Optional) If you do not know the realm, you can use the default value of none.

8. If you specified a realm type, enter its name in the **Realm Name** field. (Optional)
9. Enter tags for the credential pair. (Optional)

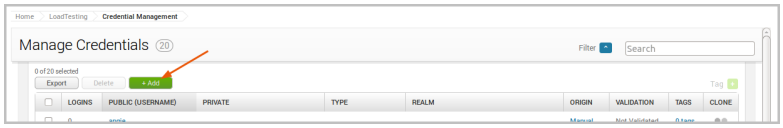
   To enter a tag, start typing the name of the tag you want to use in the **Tag** field. As you type in the search box, Metasploit Pro automatically predicts the tags that may be similar to the ones you are searching for. If the tag does not exist, Metasploit Pro creates it.

10. Click **OK**.

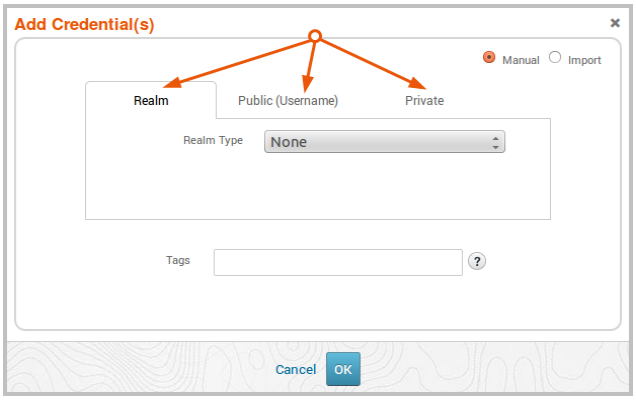
    The password is added to the project and is viewable from the Manage Credentials page.

Manually Adding a Private SSH Key

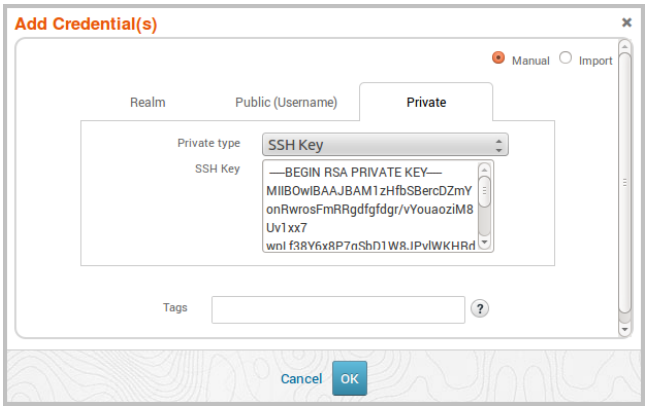
1. From within a project, go to **Credentials > Manage** to access the Manage Credentials area.
2. When the Manage Credentials page appears, click the **Add** button.



The **Add Credentials** window appears and displays tabs for the three different parts of a credential: the realm, the public , and the private. You can click on the tabs to configure their options.



3. Click the **Private (Passwords)** tab.
4. Click the **Credential Type** dropdown and select **SSH Key**.



5. Copy the contents of the private SSH key and paste it into the **SSH key** field. The key must start with `-----BEGIN RSA PRIVATE KEY-----` and end with `-----END RSA PRIVATE KEY-----`.
6. Enter tags for the SSH key. (Optional)

   To enter a tag, start typing the name of the tag you want to use in the **Tag** field. As you type in the search box, Metasploit Pro automatically predicts the tags that may be similar to the ones you are searching for. If the tag does not exist, Metasploit Pro creates it.

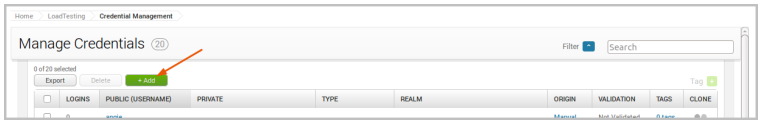
7. Click the **Public (Username)** tab and enter the username. All SSH keys must have a username.

8. Click **OK**.

   The SSH key is added to the project and is viewable from the Manage Credentials page.

Manually Adding an NTLM Hash

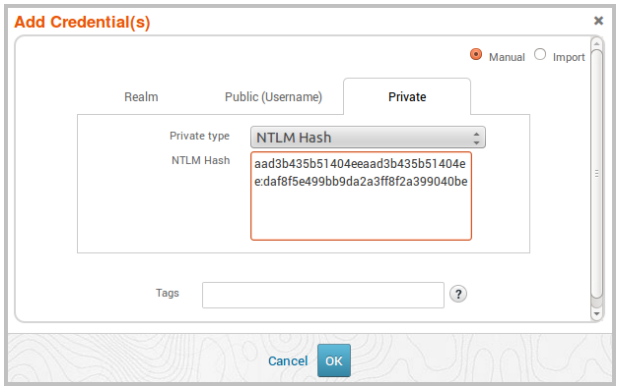
1. From within a project, go to **Credentials > Manage** to access the Manage Credentials area.
2. When the Manage Credentials page appears, click the **Add** button.



   The **Add Credentials** window appears and displays tabs for the three different parts of a credential: the realm, the public , and the private. You can click on the tabs to configure their options.

3. Click the **Private (Passwords)** tab.

4. Click the **Credential Type** dropdown and select **NTLM Hash**.



5. Copy the hash and paste it into the **NTLM Hash** field.

   A valid NTLM hash uses the following format: `<LAN Manager hex digest>:<NT LAN Manager hex digest>`, where each hex digest is 32 lowercase hexadecimal characters. For example, the following is a valid input for an NTLM hash: `aad3b435b51404eeaad3b435b51404ee:daf8f5e499bb9da2a3ff8f2a399040be`.

6. Click the **Public (Username)** tab and enter the username. The username will be *BLANK* if you do not specify one. (Optional)

7. Click the **Realm** tab and select one of the following realm types: None, Domain Name, Postgres DB, DB2, or Oracle SID.(Optional)
8. If you specified a realm type and know its name, enter its name in the **Realm Name** field.
9. Enter tags for the NTLM hash. (Optional)

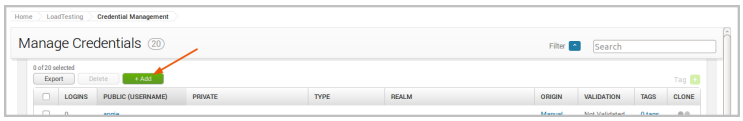
   To enter a tag, start typing the name of the tag you want to use in the **Tag** field. As you type in the search box, Metasploit Pro automatically predicts the tags that may be similar to the ones you are searching for. If the tag does not exist, Metasploit Pro creates it.

10. Click **OK**.

    The NTLM hash is added to the project and is viewable from the Manage Credentials page.

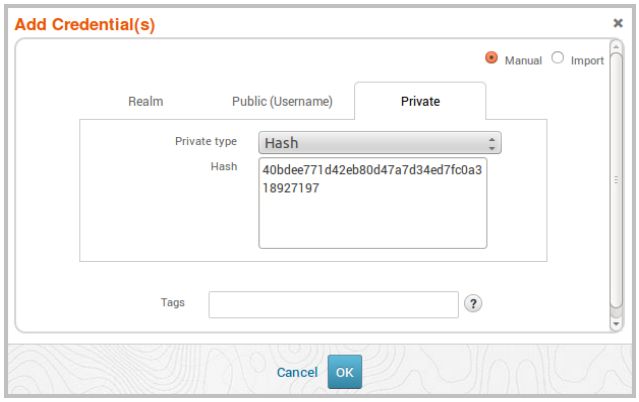
Manually Adding a Nonreplayable Hash

1. From within a project, go to **Credentials > Manage** to access the Manage Credentials area.
2. When the Manage Credentials page appears, click the **Add** button.



The **Add Credentials** window appears and displays tabs for the three different parts of a credential: the realm, the public , and the private. You can click on the tabs to configure their options.

3. Click the **Private (Passwords)** tab.
4. Click the **Credential Type** dropdown and select **Hash**.



5. Copy the hash and paste it into the **Hash** field.

6. Click the **Public (Username)** tab and enter the username. (Optional)

   The username will be *BLANK* if you do not specify one.

7. Click the **Realm** tab and select one of the following realm types: None, Domain Name, Postgres DB, DB2, or Oracle SID.(Optional)
8. If you specified a realm type, enter its name in the **Realm Name** field.
9. Enter tags for the hash. (Optional)

   To enter a tag, start typing the name of the tag you want to use in the **Tag** field. As you type in the search box, Metasploit Pro automatically predicts the tags that may be similar to the ones you are searching for. If the tag does not exist, Metasploit Pro creates it.

10. Click **OK**.

    The SSH key is added to the project and is viewable from the Manage Credentials page.

Importing and Exporting Credentials

You can import and export credentials to easily share them between projects or with other members of the organization. It is vital that this confidential information be shared responsibly, as it may contain plaintext credentials and extremely sensitive data.

Credentials can be imported in a couple of ways. You can import them as part of a workspace ZIP, which will automatically include all credentials contained in the export, as well as any other data that was part of the project. Workspace ZIP files can be imported from the Hosts or Overview page. If you only want to import credentials, you will need to do so from the Manage Credentials page. You can import credentials that have been exported from a project or you can import a credentials list that you have manually created.

When you export credentials from a project, Metasploit Pro creates a manifest file that contains the credential data and compresses it into a ZIP file. The manifest file is a CSV file that lists every credential in the project and includes the following information for each credential: username, private type, private,

realm type, realm name, host address, service port, service, and service protocol. If the project contains SSH keys, they will be included in the exported file. Each SSH key will be mapped to its corresponding username in the manifest file.

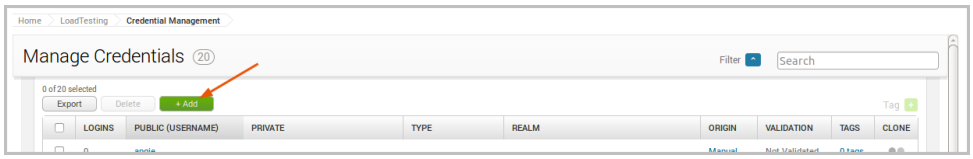
To help you understand how you can share credentials, the following sections walk you through importing and exporting credentials.

Importing Credentials Exported from Other Projects

When you export credentials from a project, Metasploit Pro creates a manifest file, which is a CSV file that contains all of the project's credential data, and compresses it into a ZIP file. If you want to import credentials that have been exported from another project, you must import the workspace ZIP file. This ensures that the file contains the required header row that Metasploit Pro needs to properly import the credentials and any additional data, such as SSH keys, that are associated with the manifest file. You cannot simply import the `manifest.csv` file; the import will fail if you attempt to import a manifest file that was created by Metasploit Pro.

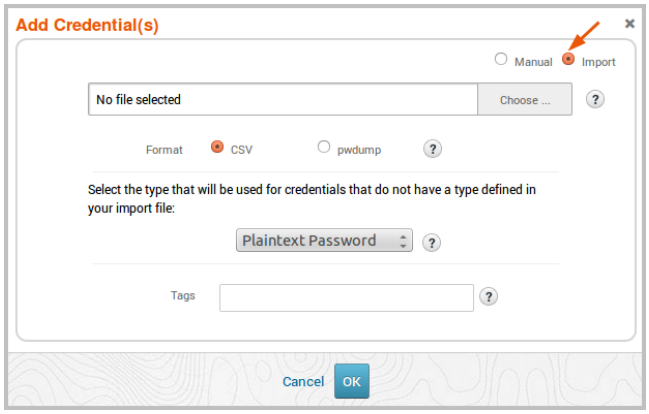
To import exported credentials:

1. From within a project, go to **Credentials > Manage** to access the Manage Credentials area.
2. Click the **Add** button.

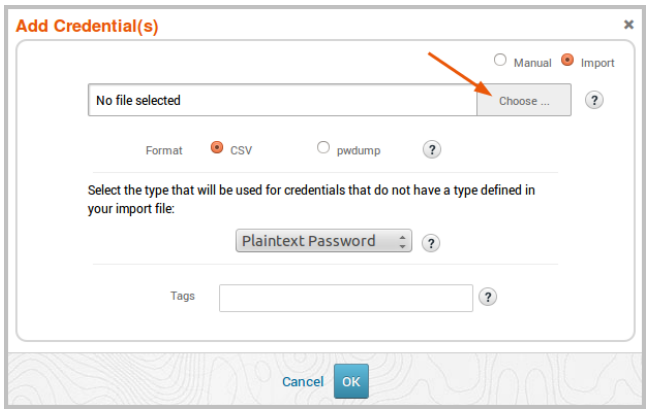


The **Add Credentials** window appears.

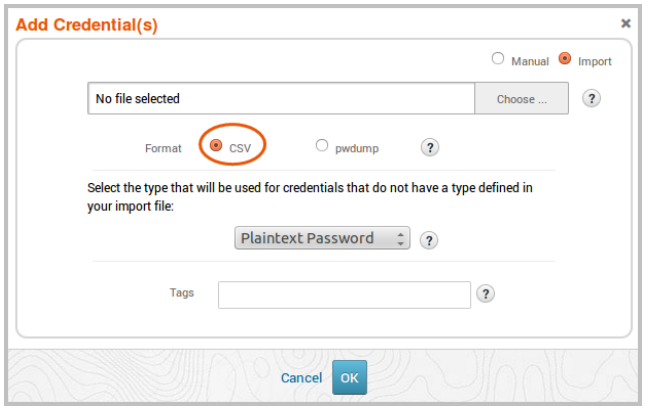
3. Select the **Import** option.



4. Click the **Choose** button and navigate to the location of the ZIP file you want to import.



5. Select the file and click **Open**.
6. From the Add Credentials window, select **CSV** as the format.



7. Click the **Password Type** dropdown and select the type you want to assign to credentials that do not have a type defined in your import file.

   Any credential that has a private must have a type defined for it. This option lets you set the default type for any credential that has an empty type field in the import file.

8. Enter tags for the credential pair. (Optional)

   To enter a tag, start typing the name of the tag you want to use in the **Tag** field. As you type in the search box, Metasploit Pro automatically predicts the tags that may be similar to the ones you are searching for. If the tag does not exist, Metasploit Pro creates it.

9. Click **OK**.

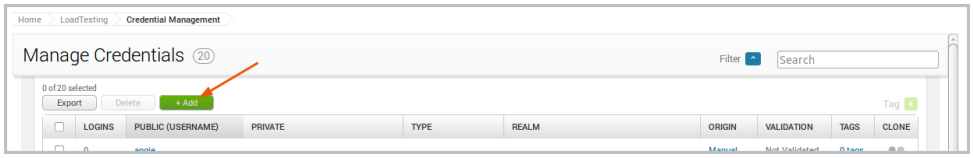
   The CSV is imported into the project. You can go to the Manage Credentials page to view the imported credentials.

Importing a Manually Created Credentials File

If you have a credentials list that you manually created, you can import it from the Manage Credentials page. The credentials file that you upload must be a CSV file that contains the following header row: `username,private_data`.

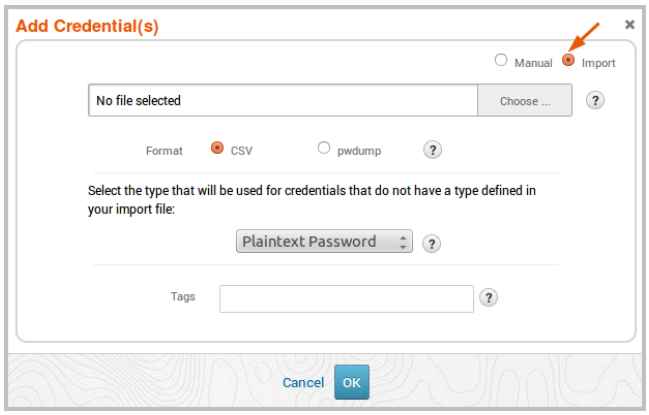
To import a manually created credentials file:

1. From within a project, go to **Credentials > Manage** to access the Manage Credentials area.
2. Click the **Add** button.

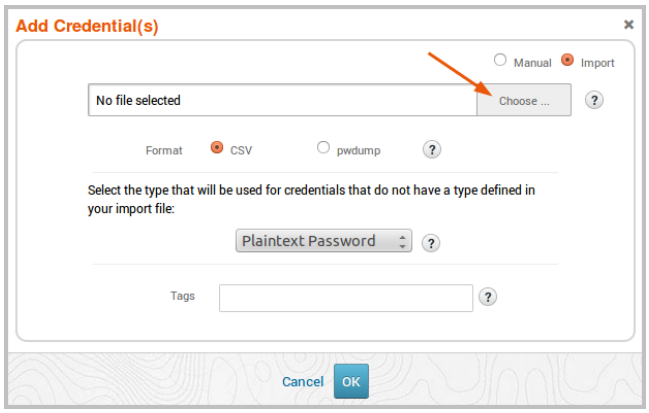


The **Add Credentials** window appears.

3. Select the **Import** option.

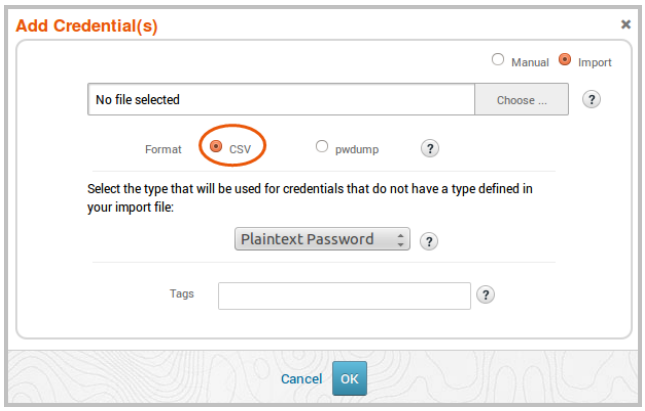


4. Click the **Choose** button and navigate to the location of the CSV file you want to import.



   The CSV file must contain the following header row: `username,private_data`.

5. Select the file and click **Open**.
6. From the Add Credentials window, select **CSV** as the format.



7. Enter tags for the credential pair. (Optional)

   To enter a tag, start typing the name of the tag you want to use in the **Tag** field. As you type in the search box, Metasploit Pro automatically predicts the tags that may be similar to the ones you are searching for. If the tag does not exist, Metasploit Pro creates it.

8. Click **OK**.

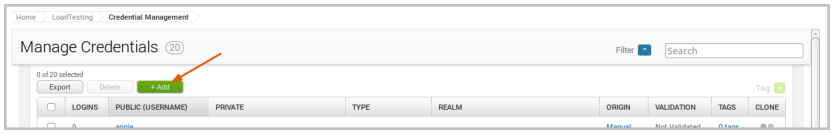
   The CSV is imported into the project. You can go to the Manage Credentials page to view the imported credentials.

Importing a PWDump

A PWDump is a text file that contains credentials that have logins associated with them. The PWDump that you upload must be one that was exported from Metasploit Pro. Other types of password dumps are not supported.

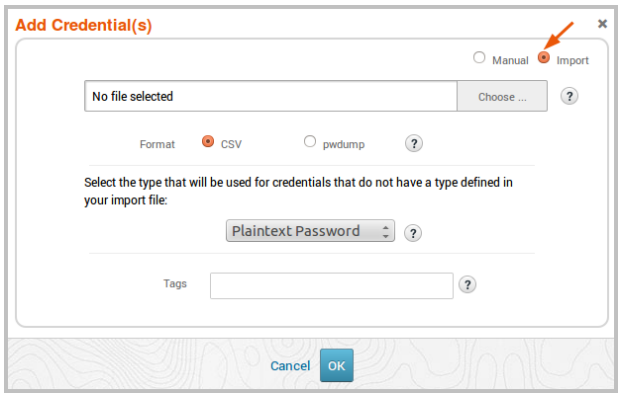
To import a PWDump:

1. From within a project, go to **Credentials > Manage** to access the Manage Credentials area.
2. Click the **Add** button.

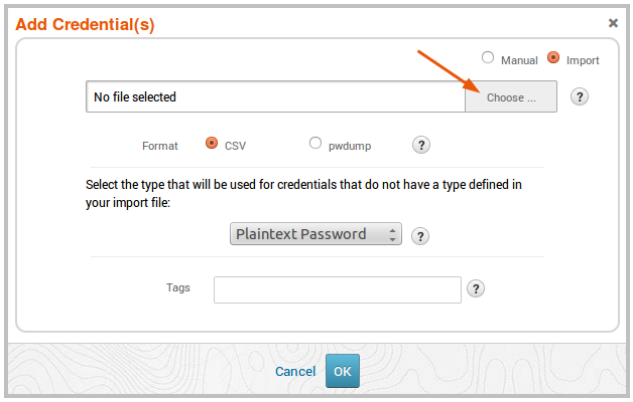


The **Add Credentials** window appears.

3. Select the **Import** option.

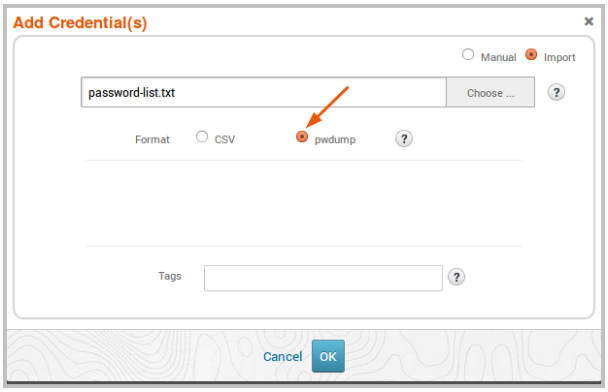


4. Click the **Browse** button and navigate to the location of the PWDump you want to import.



5. Select the file and click **Open**.

6. Select the **pwdump** format option.



7. Enter tags for the credential pair. (Optional)

   To enter a tag, start typing the name of the tag you want to use in the **Tag** field. As you type in the search box, Metasploit Pro automatically predicts the tags that may be similar to the ones you are searching for. If the tag does not exist, Metasploit Pro creates it.

8. Click **OK**.

   The PWDump is imported into the project. You can go to the Manage Credentials page to view the imported credentials.

Exporting All Credentials

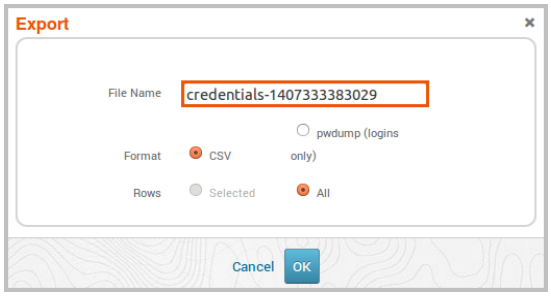
There are a couple of ways that you can export all credential data from a project. You can do it at the project level by exporting a workspace ZIP, will contain all of the information stored in the project, such as host data, collected evidence, and reports, as well as a credentials folder that contains the credential data.

If you only want to export credential data from the project, you can export them from the Manage Credentials page. When you export credential data, Metasploit Pro creates the manifest file and automatically compresses it into a ZIP file for you, which enables you to import the file with no additional changes.

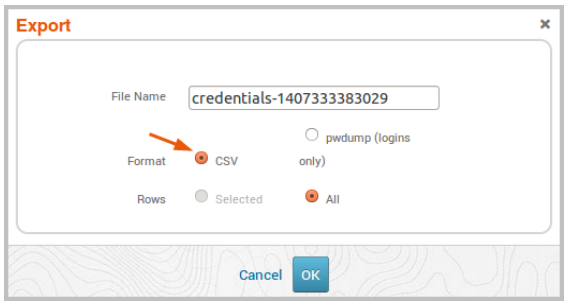
To export all credentials:

1. From within a project, go to **Credentials > Manage** to access the Manage Credentials area.
2. Click the **Export** button.

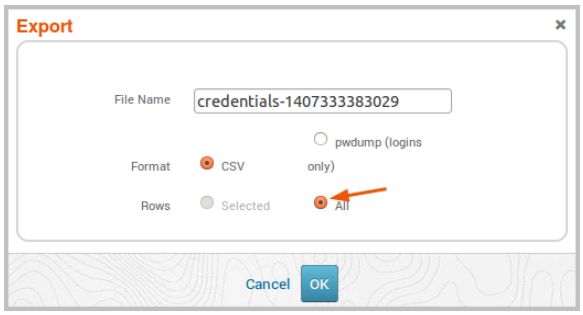
3. Enter a name for the file in the **File Name** field if you want to provide a custom name. Otherwise, you can use the auto-generated file name.



4. For the **Format** option, select CSV.



5. For the **Row** option, select All.



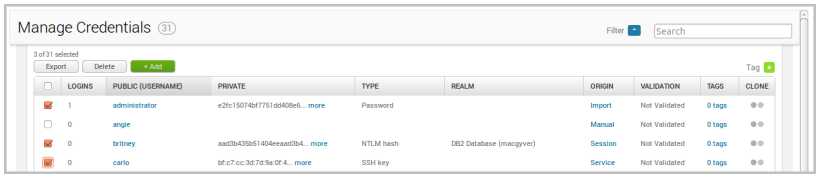
6. Click **OK** to begin the export.

   The export will automatically begin. Your system may prompt you to save the file if it is not configured for automatic downloads.

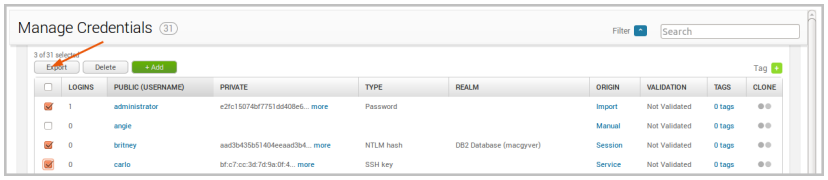
Exporting Selected Credentials

You can export specific credentials from a project from the Manage Credentials page.

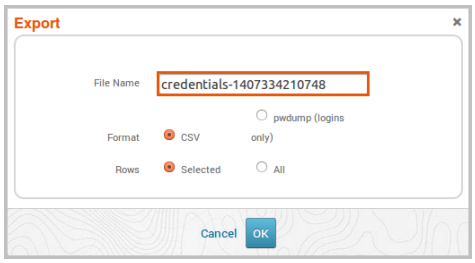
1. From within a project, go to **Credentials > Manage** to access the Manage Credentials area.
2. Select the credentials that you want to export.



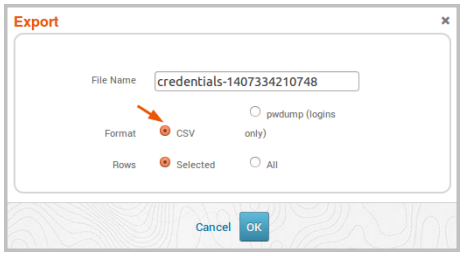
3. Click the **Export** button.



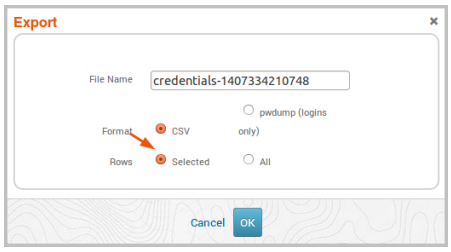
4. Enter a name for the file in the **File Name** field if you want to provide a custom name. Otherwise, you can use the auto-generated file name.



5. For the **Format** option, choose CSV.



6. For the **Row** option, choose Selected.



7. Click **OK** to begin the export.

   The export will automatically begin. Your system may prompt you to save the file if it is not configured for automatic downloads.

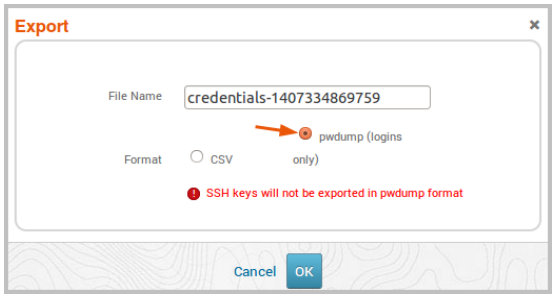
Exporting a PWDump

A PWDump is a Metasploit Pro export type that only exports credentials that have logins. Metasploit Pro exports the PWDump as a text file that can be imported into other projects.

To export a PWDump:

1. From within a project, go to **Credentials > Manage** to access the Manage Credentials area.
2. Click the **Export** button.

3. For the **Format** option, select **pwdump (logins only)**.



4. Click **OK** to begin the export.

   The export will automatically begin. Your system may prompt you to save the file if it is not configured for automatic downloads.

Creating a Credentials CSV File to Import

You can create CSV files to import credentials into a project. For example, if you have a word list or password list that you want to add to a project, you will need to create a CSV file for those credentials. You can use a spreadsheet program, like Microsoft Excel or Google Docs, to create a CSV file.

The CSV file must include the header row `username,private_data`, which defines the fields in the table. The following image shows an example of a credentials list that was created in Microsoft Excel:

	A	B
1	username	private_data
2	administrator	administrator
3	guest	guest
4	username	password
5	admin	admin
6	root	
7		password

As you can see, the first row contains the required header row. The subsequent rows contain the data specified by each header. If you want to leave the username or private blank, you can leave the field empty.

When you are done creating the file, you will need to save with a .csv file extension so that you can import it into a project. For information on how to import a CSV file into a project, see *Importing and Exporting Credentials* on page 113.

Cloning and Editing Credentials

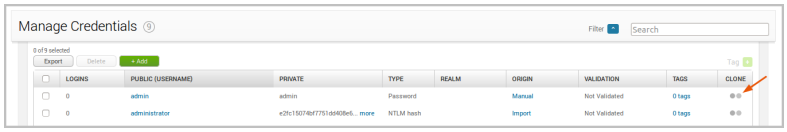
Cloning is a useful feature if you want to make a copy of a credential with some minor changes. For example, if you have a credential pair, such as `admin:admin`, you might want to add a variation, like `admin1:admin1`. The fastest way to do this would be to clone the original credential pair and tweak the public and private data for it.

Note: A credential cannot be modified after you save it to a project. The only way to edit a credential is to clone and modify it. If there are changes that you need to make to a credential, you will need to clone the credential, edit the clone, save the clone, an delete the original credential.

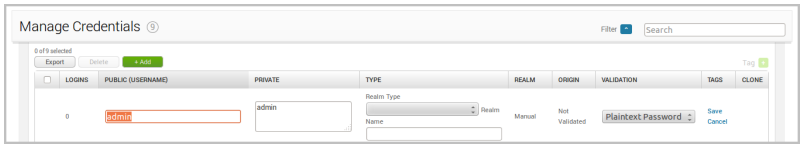
To clone a credential:

1. From within a project, go to **Credentials > Manage** to access the Manage Credentials area.
2. When the Manage Credentials page appears, find the row that contains the credential you want to clone.

3. Click the **Clone** button.



A duplicate of the credential is created and added to the table. Any data that you can modify is displayed in an editable field.



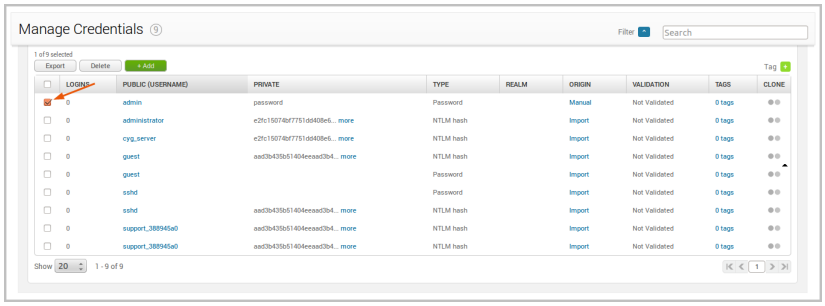
4. Edit any of the following fields: public (username), private (password), realm, origin, or password type. You must modify one of the fields because a project cannot contain duplicates of the exact credential entry. You will not be able to save the credential unless the credential is unique. For example, you can have an two entries for admin/admin as long as either the realm or private type is different.
5. Click **Save** when you are done.

   The credential is added to the project and viewable from the Manage Credentials page.

Deleting Credentials

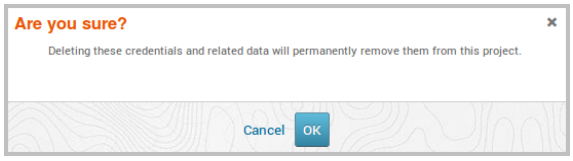
If there are credentials that you no longer need to store in a project, you can delete them. All deleted credentials are permanently removed from the project and cannot be recovered.

1. From within a project, go to **Credentials > Manage** to access the Manage Credentials area.
2. When the Manage Credentials page appears, select the credentials you want to delete.



3. Click the **Delete** button.

When the confirmation window appears, click **OK** to delete the credentials from the workspace. The credential, including all of its logins, will be removed from the project. You must be absolutely sure that you want to delete the credential. You will not be able to restore deleted credentials.



Reusing Credentials

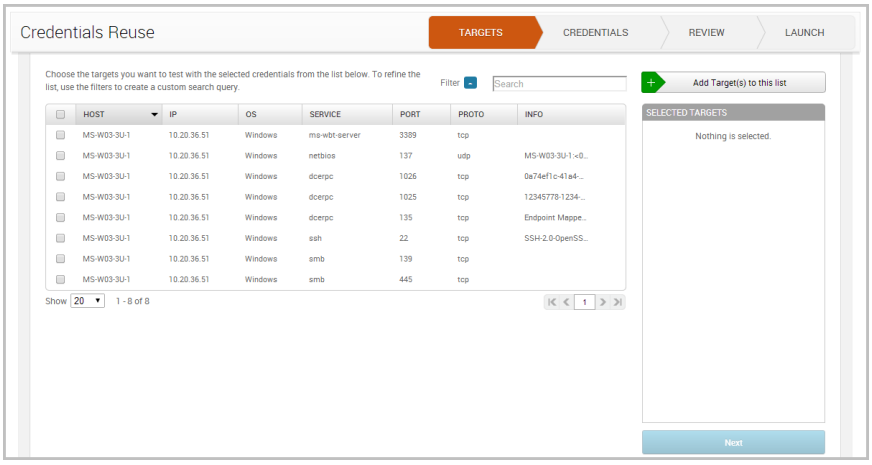
With exceedingly more and more stringent password requirements, credential reuse is becoming a common issue within many organizations. Users inundated with complex password policies may resort to reusing the same password across multiple accounts so that they can easily manage their credentials. This can cause major security issues when those credentials are compromised. For example, if an attacker is able to obtain valid credentials on one target, they can try those credentials on other targets to further compromise the network.

To help an organization audit their passwords, you can reuse credentials to identify additional targets that will be vulnerable if a particular credential is compromised. Credentials Reuse is a Metasploit Pro feature reuses validated credentials to attempt to authenticate to additional targets. This feature is useful when you have validated or known credentials that you want to try on a set of targets. For example, if you were able to obtain an NTLM hash on a target, you should try to reuse that hash on other SMB targets. If a system administrator commonly deploys the default configuration for a system, the likelihood that the credential will work is high.

Credentials Reuse Workflow

Credentials Reuse provides guided workflow for the required tasks that need to be configured. Each task in the workflow is displayed on its own tab. You can click on any of the tabs to switch between the different tasks in the workflow; however, you must complete each task before you can move to the next.

The Credentials workflow is shown below:



The workflow is quite simple and comprises of four steps:

1. Select the targets that you want to try the credentials on.
2. Select the credentials that you want to use to authenticate to the selected targets.
3. Configure the reuse settings, such as the timeout and validation limits, and review the target that you want to use.
4. Launch the task.

Credentials Reuse Targets

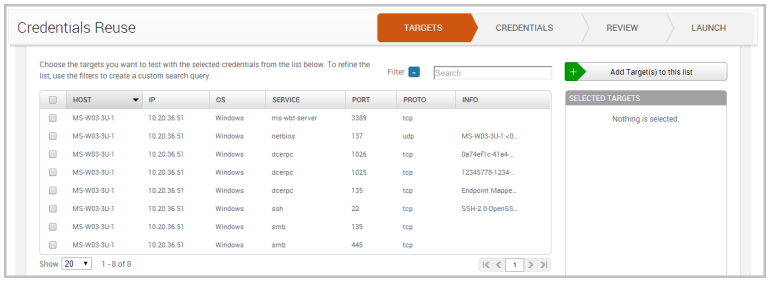
Credentials Reuse utilizes several login scanners from the Metasploit Framework, which enable Credentials Reuse to target the following services: AFP, DB2, FTP, HTTP, HTTPS, MSSQL, MySQL, POP3, PostgreSQL, SMB, SNMP, SSH, telnet, VNC, and WinRM.

Targets that do not have a login scanner, such as DCERPC, will be skipped.

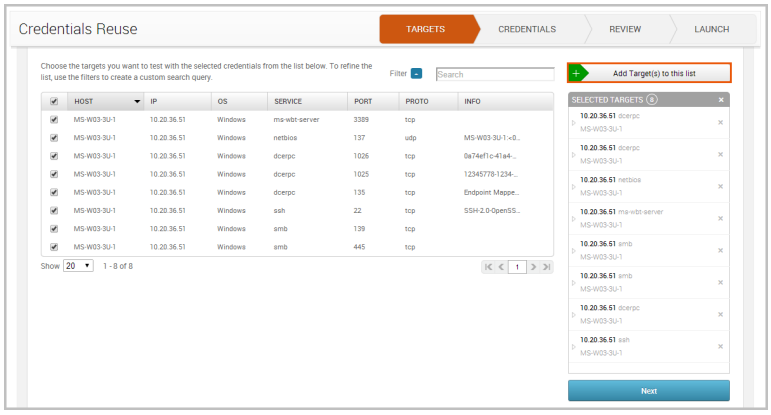
Configuring and Running Credentials Reuse

1. From within a project, select **Credentials > Reuse**.

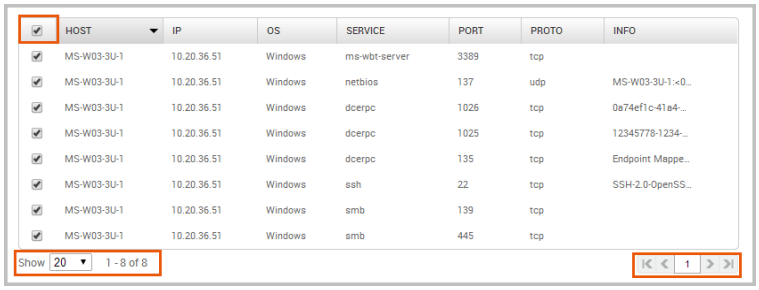
   The Credentials Reuse workflow appears. The Targets tab displays first and shows you the targets that are available in the project. Targets, in this instance, are services. Metasploit Pro pulls this data from the host and service data that is stored in the project.



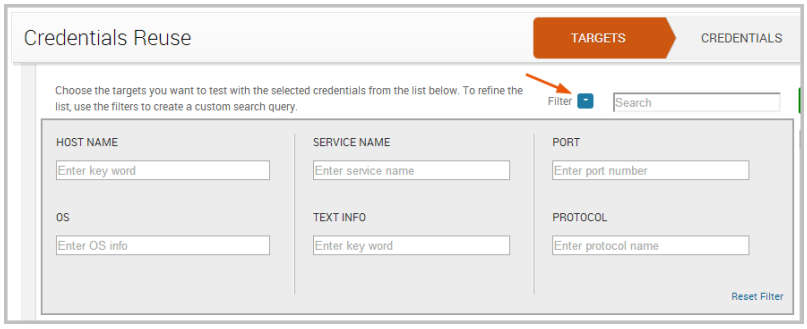
2. Select the targets that you want to try the credentials on. You can select as many targets as you want. When you are done selecting targets, click the **Add Target(s) to this list** button. The targets will be added to the Selected Targets list.



You can use the **Select All** checkbox to choose all the targets in the project and the page navigation arrows to look through the targets list. If you want to view all targets in the project, select the **All** option from the **Show** dropdown menu.



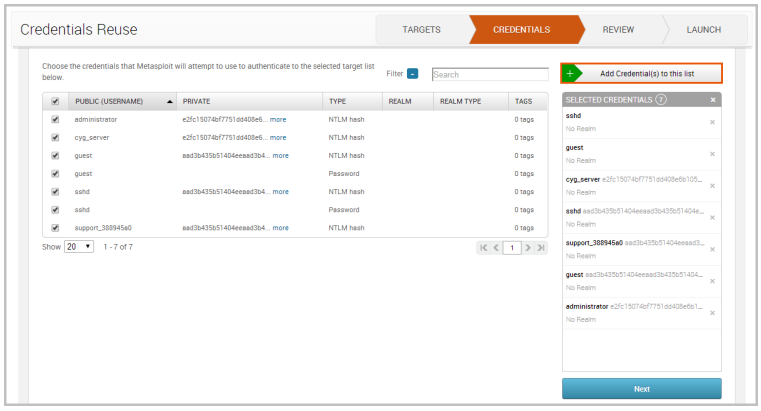
You can also click on the **Filter** button to find targets based on host, service, port, operating system, protocol, and keyword.



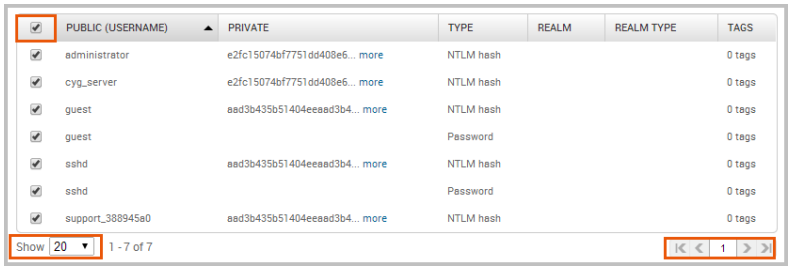
3. Click the **Next** button when you are done selecting targets.

The Credentials tab displays and shows you the credentials that are available in the project.

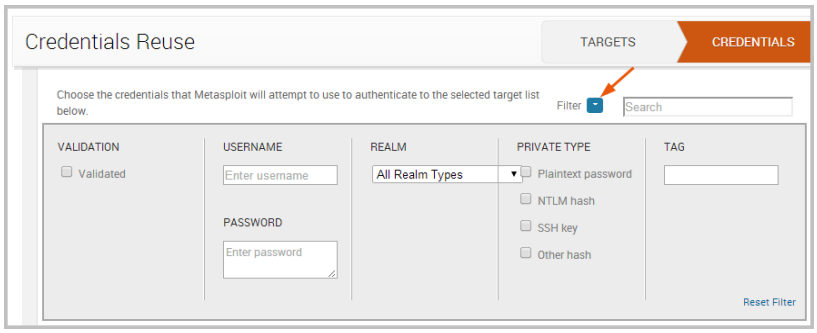
4. Select the credentials that you want to reuse. You can select as many credentials as you want. When you are done selecting credentials, click the **Add Credential(s) to this list** button. The targets will be added to the Selected Credentials list.



You can use the **Select All** checkbox to choose all the targets in the project and the page navigation arrows to look through the credentials list. If you want to view all credentials in the project, select the **All** option from the **Show** dropdown menu.

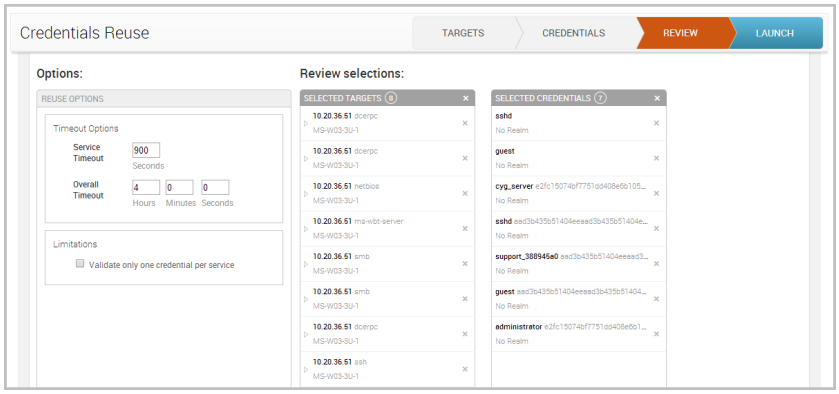


You can also click on the **Filter** button to find credentials based on validation status, username, password, private type, realm, and tag.



5. Select the **Next** button when you are done selecting credentials.

   The Review tab appears and shows you the options that you can set for Credentials Reuse and the targets and credentials that you have selected.

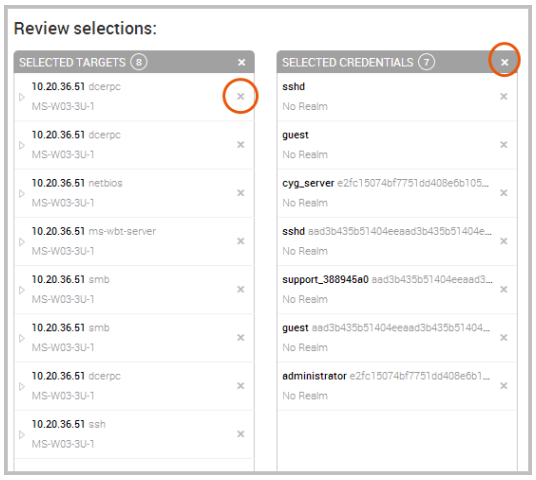


6. If you want to control the timeout settings for Credentials Reuse, you can configure any of the following options:

   - **Service Timeout** - Sets the timeout, in seconds, for each target.
   - **Overall Timeout** - Sets the timeout for the entire Credentials Reuse task.
   - **Validate one credential per service** - Limits the number of credentials that are validated for a service on a host to one. Once a credential has been validated for a service, Credentials Reuse will stop testing with other credentials.

7. Review the targets and credentials that you have selected for Credentials Reuse.

   If there are any that you want to remove, you can click on the **Remove 'X'** button located next to the target or credential or you can click on the **Remove 'X'** button located at the top of each list to remove all targets or credentials.

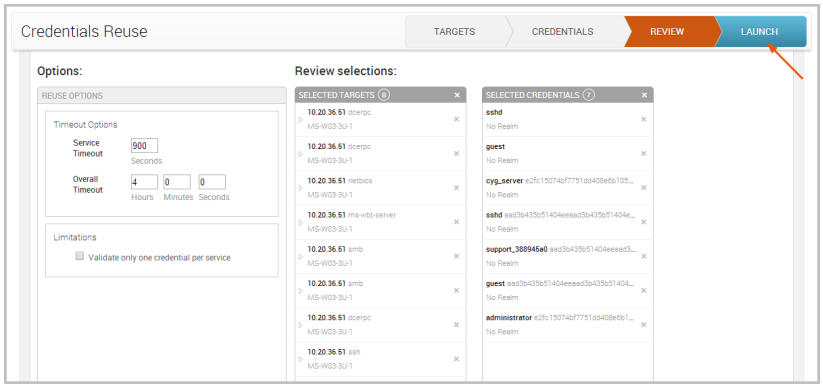


   If there are targets or credentials you want to add, you can either click on the tab for the item you want

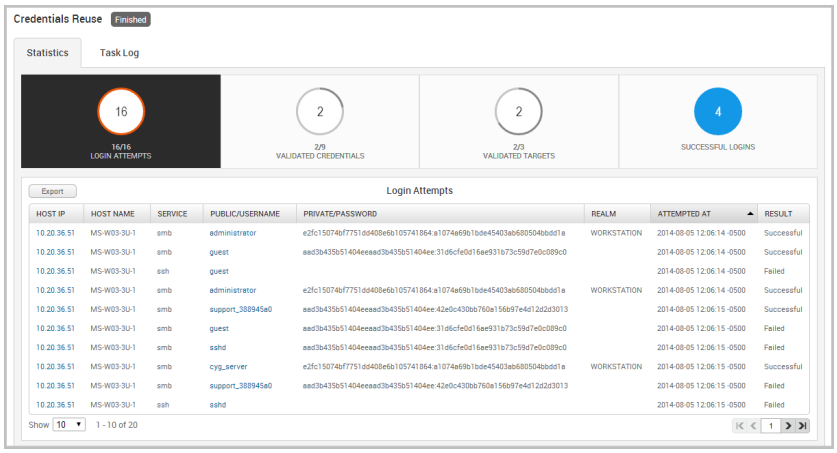
to add or you can click on the **Go back and edit** link located at the bottom of the Selected Targets and Selected Credentials lists.

If you want to see additional information for a particular target, you can click on the dropdown arrow located next to each target to display the operating system, port, and protocol.

8. Click the **Launch** button when you are ready to run Credentials Reuse. There is a launch button located at the top and bottom of the workflow.



When you launch Credentials Reuse, the Findings window appears and shows you the statistics for the task run. You can click on any of the statistic bubbles to view the details for that particular statistic.



The Findings window shows you the following statistics:

- **Login attempts** - The total number of login attempts that were made. Credentials Reuse will only attempt logins for the following services: AFP, DB2, FTP, HTTP, HTTPS, MSSQL, MySQL, POP3, PostgreSQL, SMB, SNMP, SSH, telnet, VNC, and WinRM. Therefore, the Login Attempts may not include all of the targets that you have selected. The Login Attempts also shows you results for the login and when the login was last attempted.
- **Validated credentials** - The total number of credentials that successfully authenticated.
- **Validated targets** - The total number of targets that were validated.

- **Successful logins** - The total number of logins that were successful. This number is derived from the number of validated credentials and validated targets.

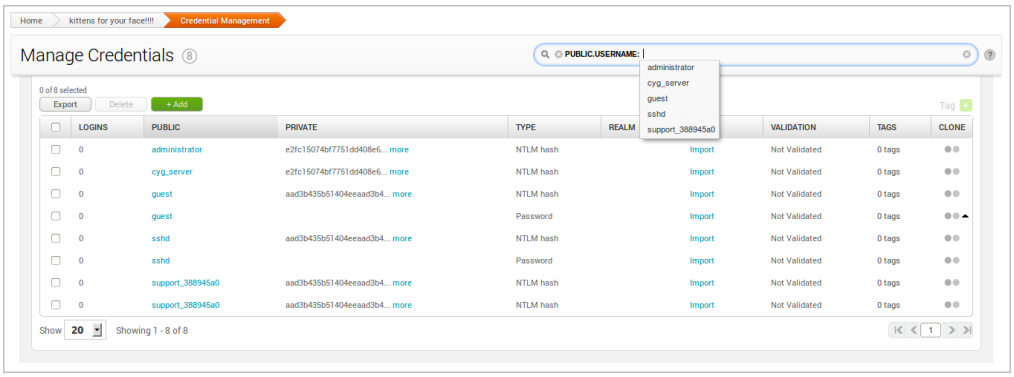
Note: If Metasploit Pro is able to identify a realm for a credential, it will update the credential with the realm information. You will be able to view the updated credential from the Manage Credentials page.

Searching for Credentials

You can create advanced search queries to find exact credential matches on the Credential Management and Credentials Reuse pages. This capability can help you narrow the search results down to a subset of data, such as a specific public and private.

Creating a Search Query

To search for credentials, click on the **Search** field located on the Credential Management or Credentials Reuse page. A dropdown displays and shows you the search operators that are available. You will need to select a search operator from the list to continue. The search field displays the possible keywords that are available for the selected operator. You can choose a keyword from the list or you can start typing to refine the list.



To create a search query for credentials, you need to specify at least one search operator and keyword. A search operator indicates the type of data you want to query and a keyword refers to the term that the search uses to find matching records. You can use as many search operators as you need. As you add search operators to the query, the table automatically updates the credentials that are listed.

By default, the search query uses the AND connector between each search operator. For example, the query "PUBLIC.USERNAME: John, PRIVATE.DATA: abc123" returns any username that contains "John" that has a password of "abc123". However, if a query contains multiple operators of the same type, the query uses the OR connector between those operators instead. For example, the query "PUBLIC.USERNAME: John, PUBLIC.USERNAME: Mike, PRIVATE.DATA: abc123" returns any username that contains "John" or "Mike" that has a password of "abc123".

! The search query automatically adds an AND connector between each search operator. However, if the search query uses more than one search operator of the same type, the query uses the OR connector between those operators instead.

Search Operators

The following search operators are available for credentials:

- public.username - This search operator matches a username.
- private.data - This search operator matches a private.
- private.type - This search operator matches a private type.
- realm.key - This search operator matches a realm type.
- realm.value - This search operator matches a realm name.
- tags.name - This search operator matches a a tag.

Credential Search Syntax

You must use the following syntax when searching for credentials: `<search operator>:<keyword>`. For example, if you want to find all publics that have a value of `admin`, you need to create the following query: `public.username:admin`. This query ensures that the search only looks in the **Public** column in the credentials table for the `administrator` keyword and only returns credentials that have `administrator` as its public value.

Searching for a Public

To search for a public, your query must use the **public.username** operator. For example, the query `public.username:admin` searches for any credential that has a public of 'admin'.

Searching for a Private

To search for a private, your query must use the **private.data** operator. For example, the `private.data:abc123` query searches for any credential that has a private of 'abc123'.

Searching for a Private Type

To search for a private type, your query must use the **private.type** operator. For example, the `private.type:hash` query searches for any credential that has a private type of 'hash'.

Searching for a Realm Type

To search for a realm type, your query must use the **realm.key** operator. For example, the `realm.type:DB2 Database` query searches for any credential that has a realm type of 'DB2 Database'.

Searching for a Realm Name

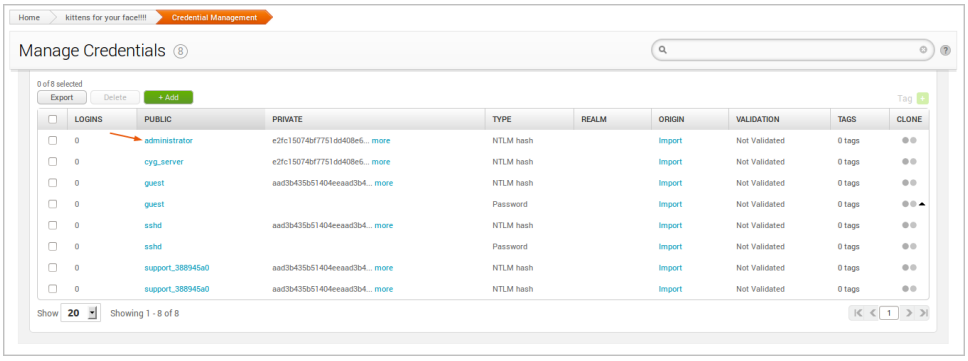
To search for a realm, your query must use the **realm.value** operator. For example, the `realm.value:DC` query searches for any credential that has a realm name of 'DC'.

Searching for Credentials with a Specific Tag

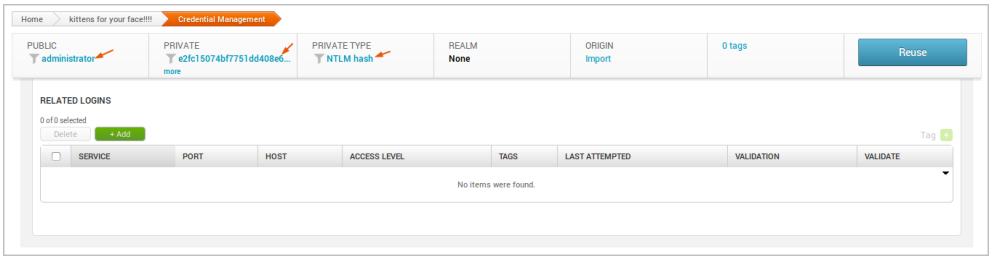
To search for a credential with a specific tag, your query must use the **tags.name** operator. For example, the `tags.name:window` query searches for any credential that has the 'windows' tag.

Filtering by Credential Metadata

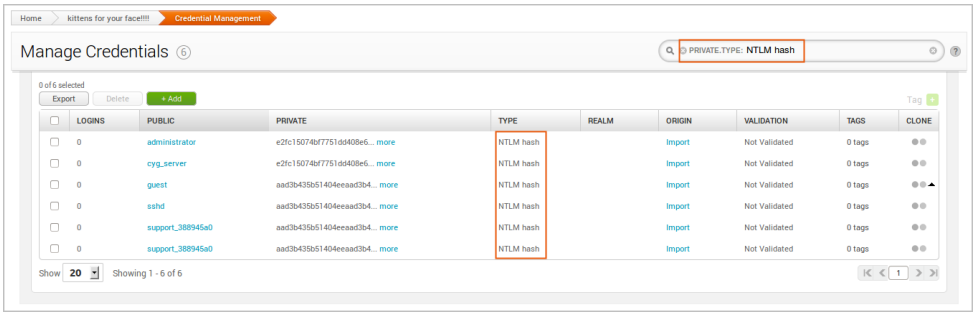
The single credential page shows you details for a particular credential, such as its metadata and related logins. To access the single credential page, you need to click on the private on the Manage Credentials page, as shown below:



When you click on the private, the single credential page slides into view. Note that the public, private, and private type values display as links. You can click on these links to query other credentials in the project that share the same public, private, or private type.



The Manage Credentials page appears and shows the results of the query. The filters will be preselected based on the type of data you queried. For example, if you are viewing the data for a public, such as administrator, you may want to see other credentials in the project that share the same public. To do this, you can simply click on the public. When the Manage Credentials page appears, you can choose additional filters to further narrow down the credentials list.

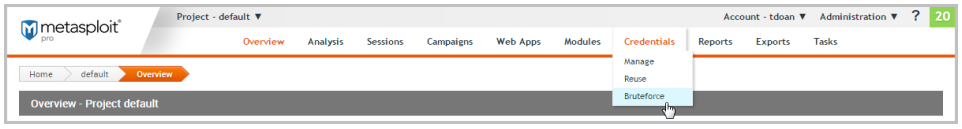


Bruteforce Attacks

A bruteforce attack automatically and systematically attempts to guess the correct username and private combination for a service. Its goal is to find valid logins and leverage them to gain access to a network to extract sensitive data, such as password hashes and tokens. As part of a penetration test, it is important that you assess the effectiveness of a bruteforce attack against a network so that you can identify audit password policies and identify potential attack vectors. This knowledge enables you to create a refined list of technical recommendations and provide real business risk analysis. To help you perform a bruteforce attack, you can use the Bruteforce Workflow, which provides a guided interface that helps you configure an automated password attack against a set of targets.

Accessing the Bruteforce Workflow

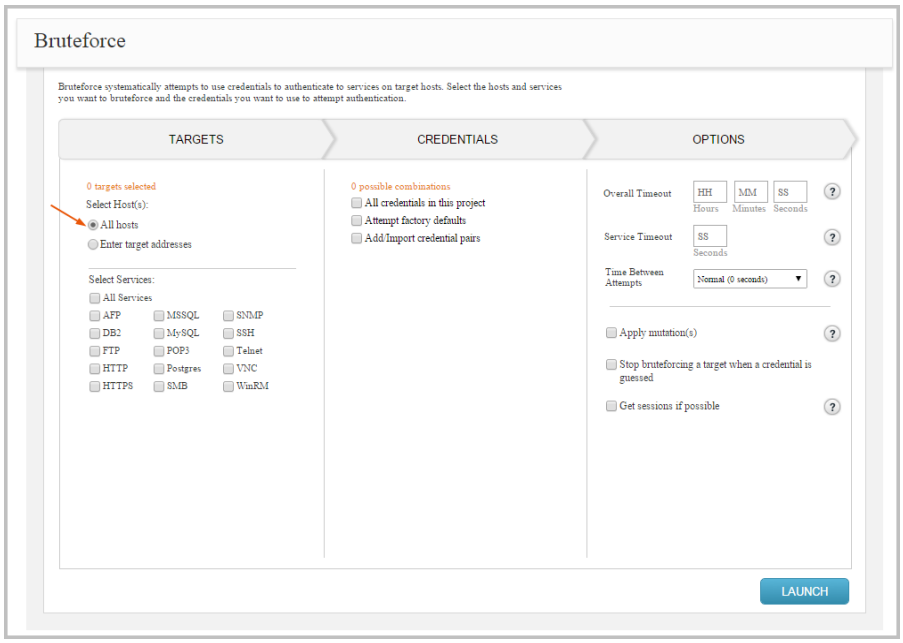
To access the Bruteforce Workflow, select **Credentials > Bruteforce** from the project tab bar, as shown below.



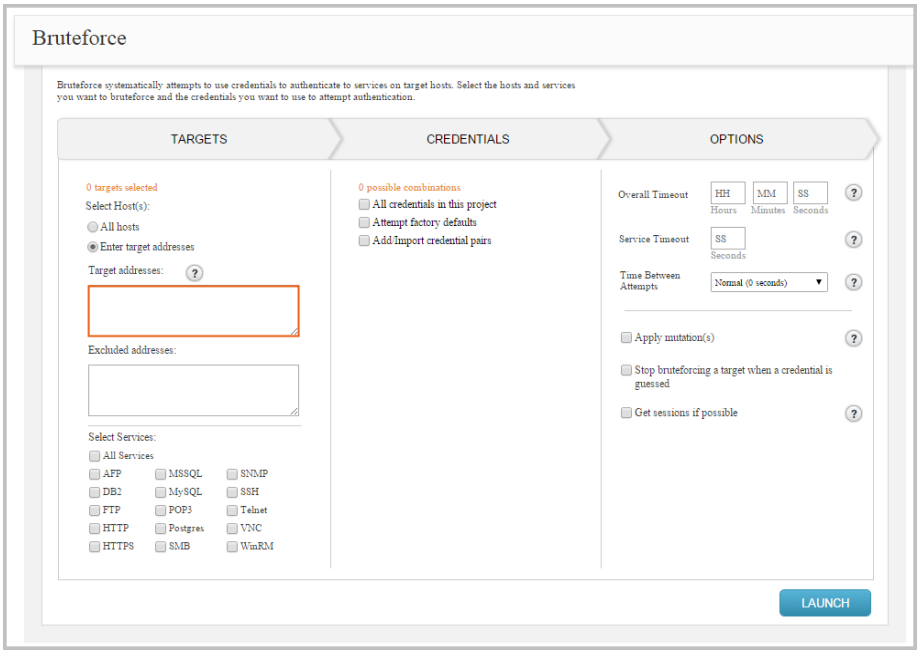
Defining Hosts for a Bruteforce Attack

The first thing you need to do in the Bruteforce Workflow is define the scope for the attack. The scope determines the hosts in the project that you want to target during the attack. You can choose to attack all hosts in the project or you can manually define them if you want granular control over the scope of the attack.

To attack all hosts in a project, select the **All hosts** option from the **Targets** section, as shown below.



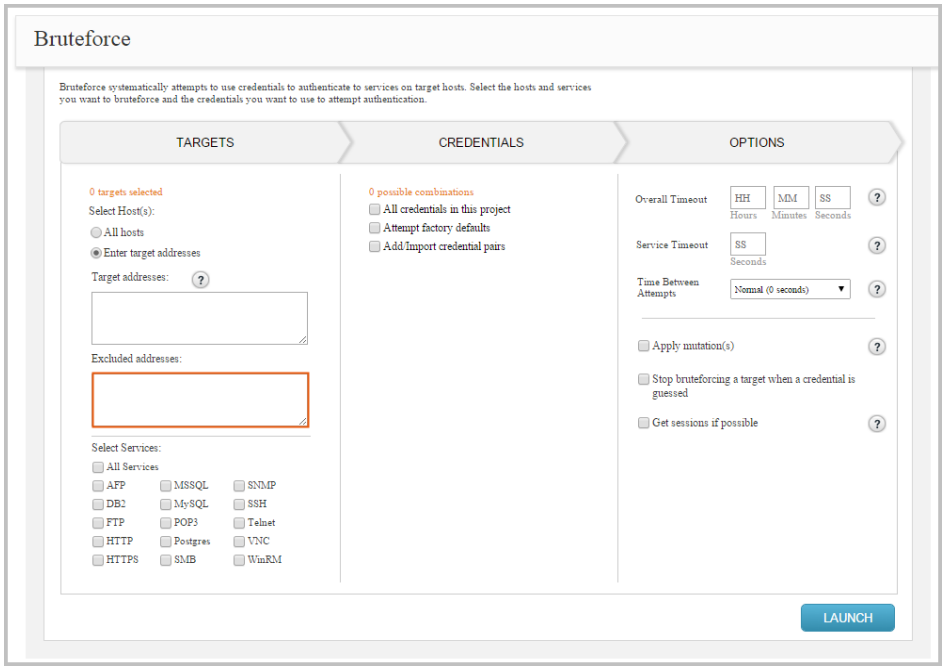
To attack specific hosts in a project, select the **Enter target addresses** option from the **Targets** section, as shown below. You can enter a single address (192.168.1.1), a range (192.168.1.1-192.168.1.100), a CIDR notation (192.168.1.0/24), or a wildcard (192.168.1.*). You must use a newline to separate each entry. If you want to include all hosts in the project, you can leave this field empty.



Excluding Hosts from a Bruteforce Attack

An exclusion list defines the hosts that you do not want to attack. An exclusion list is particularly useful if you want to define a range for the target hosts and want to exclude a few hosts from the range. For example, if you have defined 192.168.0.0/24 as the target address range, but you know that you cannot test 192.168.0.1 and 198.168.0.2 due to lockout risks, you can add them to the exclusion list.

To exclude hosts from a bruteforce attack, select the **Enter target addresses** option from the Targets section. Enter the hosts you want to blacklist in the **Excluded addresses** field, as shown below.



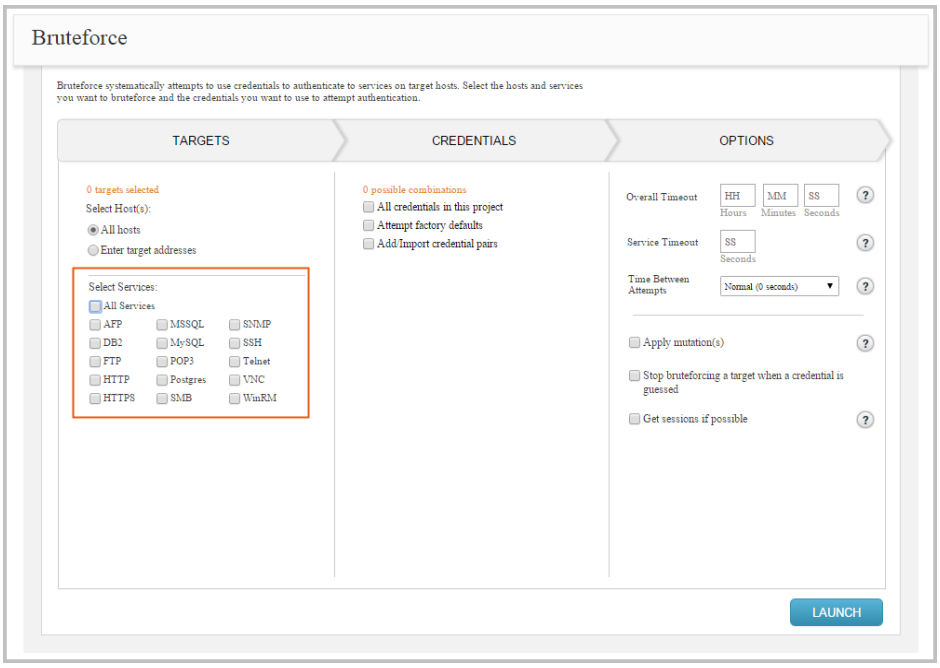
You can enter a single IP address, an address range, or a CIDR notation. If there are multiple addresses or address ranges, use a newline to separate each entry.

You can leave the **Target addresses** field empty to include all hosts in the project, except for the ones listed in the **Excluded addresses** field.

Selecting Services for a Bruteforce Attack

After you select the hosts that you want to attack, you need to choose the service logins you want to bruteforce. The services that bruteforce targets are limited to the following: AFP, DB2, FTP, HTTP, HTTPS, MSSQL, MySQL, POP3, PostgreSQL, SMB, SNMP, SSH, Telnet, VNC, and WinRM. You can choose to target all services, or you can choose any combination of them. A login attempt only occurs if the service is open on the host. Otherwise, it is skipped.

To specify the services for a bruteforce attack, select them from the **Services** list, as shown below:



After you select services for the bruteforce attack, the total targets count is updated under the Targets section. The total number of targets that are selected is calculated based on the number of hosts and services you have selected.

Building a Password List for a Bruteforce Attack

A bruteforce attack uses a password list, which contains the credentials that can be used to bruteforce service logins. You can obtain password lists online that contain commonly used credentials, such as admin/admin, or you can custom build password list using the data you have gathered about the target. For example, if you were able to obtain and crack NTLM hashes from a target, you should add them to the password list so that the bruteforce attack can try them against additional targets.

With the Bruteforce Workflow, you can use any combination of the following methods to build a password list for the bruteforce attack:

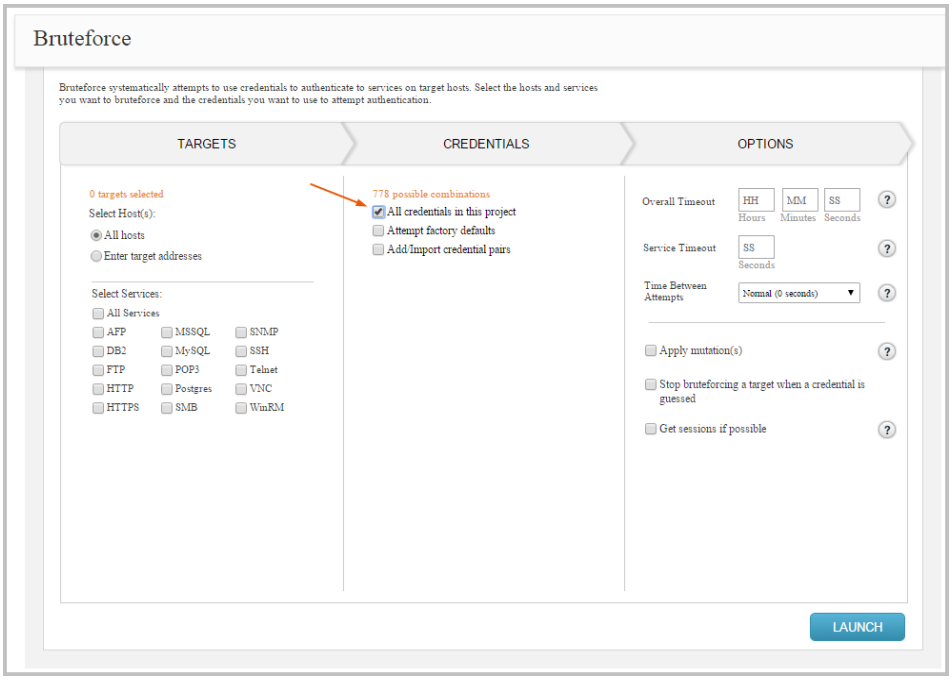
- You can choose all credentials stored in the project.
- You can try common account default settings.
- You can import a password list.
- You can manually enter a password list.

Bruteforce tries each credential pair in the password list to attempt to authenticate to a service. If it is able to authenticate to a service with a particular credential, the credential is saved to the project and a login for the service is created. Bruteforce continues to iterate through the password list until all credentials have been tried or until it reaches a limit that you have defined.

The total number of credentials that are selected is calculated based on the Cartesian product of the credentials you have selected and the number of mutations you have applied.

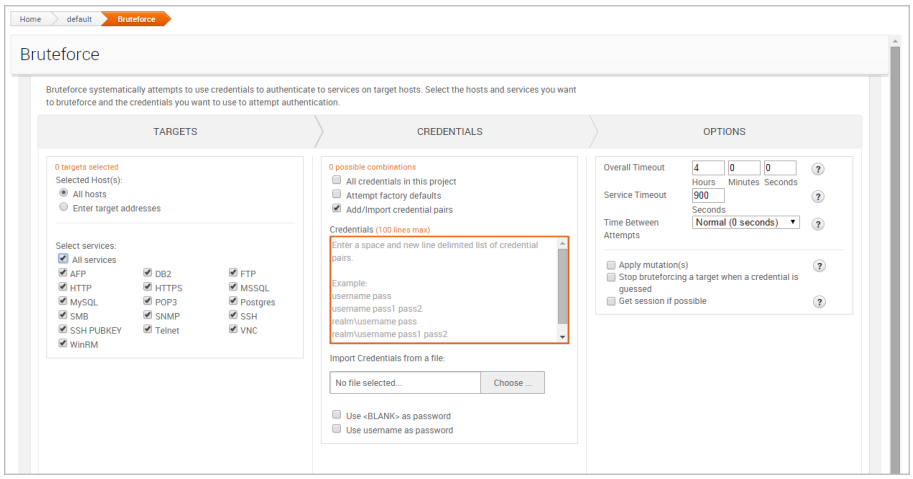
Using All Credentials in a Project for a Bruteforce Attack

To configure a bruteforce attack to use all the credentials in a project, select the **All credentials in this project** option from the Credentials section of the Bruteforce Workflow, as shown below.



Manually Entering Credentials for a Bruteforce Attack

You can manually create the password list for a bruteforce attack. To manually add credential pairs for the bruteforce attack to use, select the **Add/Import** credential pairs option from the Credentials section. The **Manually Add Credentials** text box appears, as shown below.



You can provide a space and newline delimited list of credential pairs. The first word on each line is treated as the username. Each word that follows the username is the password. You can enter up to 100 credential pairs in the text box. If you need to add more than 100 credential pairs, you will need to create a credentials file and import the file. For more information on importing a credentials file, see *Importing a Password List for a Bruteforce Attack* on page 142.

You must follow these syntax rules when you manually enter a password list:

- To define a credential pair, use the following format: `username password`.
- To specify multiple passwords for a username, enter the username followed by the passwords. Each password must be separated by a space.
- Each credential entry must be on a newline.
- Each item must be space delimited.
- To specify a blank username, use `<BLANK>` for the username.
- To specify a username with a blank password, enter the username only.

Password List Example

```
username

<BLANK> pass

username pass

username pass1 pass2
```

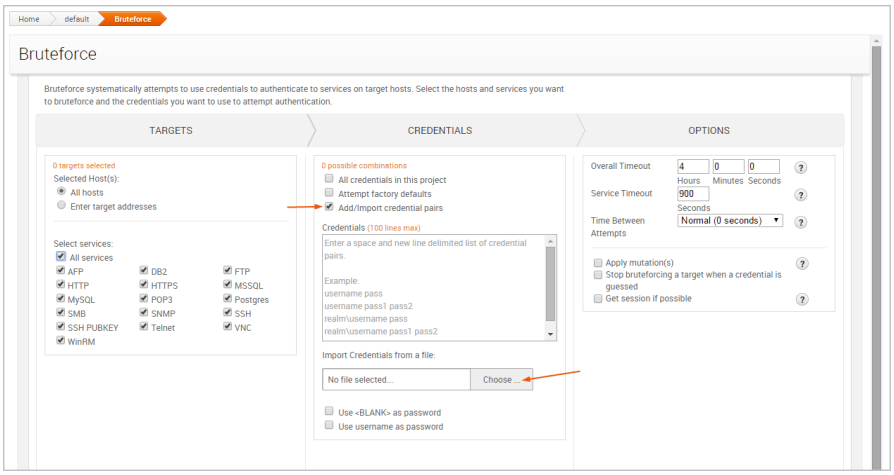
Importing a Password List for a Bruteforce Attack

A password list is a text file that contains credential pairs. You can manually create a password list using a basic text editor, like Notepad, or you can download a password list online.

The password list must follow these rules:

- Each credential pair must use the following format: `username password`.
- Each credential pair must be on a newline.
- Each item must be space delimited.
- A blank username must be defined as `<BLANK>`.
- A blank password does not have to be
- A username with no password indicates a blank password.

To import a password list, select the **Add/Import** credential pairs option from the Credentials section. Click the **Choose File** button, as shown below.

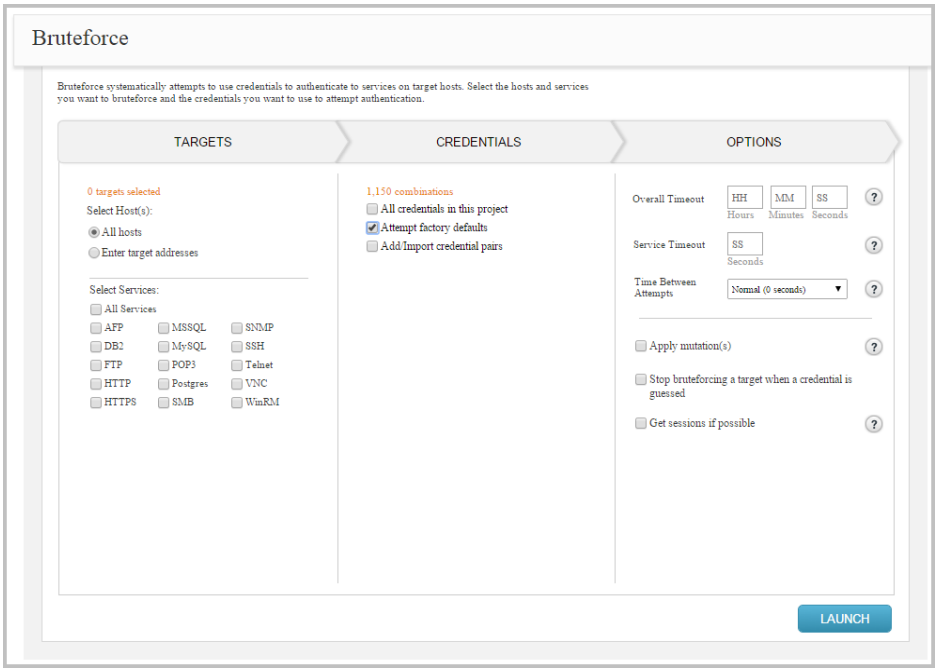


When the directory window appears, navigate to the location of the file that you want to import. Select the file and click the **Import** button.

Using Factory Defaults for a Bruteforce Attack

Default credentials are username and password pairs that are shipped with an operating system, database, or software. Oftentimes, these factory defaults are the same for all versions of a software, are publicly documented, and oftentimes left unchanged. Therefore, as a best practice, vendors always recommend that the default password be changed before the system is deployed to a production environment. However, this security practice is not always followed, and systems are often deployed with the default configuration settings, which make them prime targets for bruteforce attacks.

To help you identify systems that use the default configuration, Bruteforce includes an option called **Attempt factory defaults**, which enables you to bruteforce services using common default credentials. The following section lists the credentials that will be tried for each service if you have this option enabled.



Default Credentials for Axis2

The following usernames and passwords are common defaults for Axis2:

- Usernames - 'admin'
- Passwords - 'axis2'

Default Credentials for DB2

The following usernames and passwords are common defaults for DB2:

- Usernames - 'admin', 'dasusr1', 'db2admin', 'db2fenc1', and 'db2inst1'
- Passwords - 'admin', 'dasusr1', 'db2admin', 'db2fenc1', 'db2inst1', 'db2pass', 'db2password', and 'db2pw'

Default Credentials for FTP

The following usernames and passwords are common defaults for FTP:

- Usernames - 'admin', 'anonymous', 'ftp', 'ftp_admi', 'ftp_inst', 'ftp_nmc', 'ftp_oper', 'ftpuser', 'login', 'rapport', 'root', 'user', and 'xbox'
- Passwords - '1234', 'access', 'chrome@example.com', 'Exabyte', 'ftp', 'help1954', 'IEUser@', 'kilo1987', 'mozilla@example.com ', 'pass', 'password', 'pbxk1064', 'r@p8p0r', 'tuxalize', and 'xbox'

Default Credentials for HTTP

The following usernames and passwords are common defaults for HTTP:

- Usernames - 'admin', 'apc', 'axis2', 'cisco', 'connect', 'manager', 'newuser', 'pass', 'private', 'root', 'security', 'sitecom', 'sys', 'system', 'tomcat', 'user', 'wampp', 'xampp', and 'xampp-dav-unsecure'
- Passwords - '1234', 'admin', 'apc', 'cisco', 'connect', 'default', 'letmein', 'manager', 'none', 'pass', 'password', 'ppmax2011', 'root', 'sanfran', 'security', 'sitecom', 'sys', 'system', 'tomcat', 'turnkey', 'user', 'wampp', and 'xampp'

Default Credentials for MSSQL

The following usernames and passwords are common defaults for MSSQL:

- Usernames - 'Administrator', 'ARAdmin', 'entldbdbo', 'entldbreader', 'mon_user', 'probe', 'repl_publisher', 'repl_subscriber', 'sa', and 'WinCCConnect'
- Passwords - '2WSXcder', 'AR#Admin#', 'blank', 'dbopswd', 'pass', 'pass1', 'password', 'rdrpswd'

Default Credentials for MySQL

The following usernames and passwords are common defaults for MySQL:

- Usernames - 'admin', 'mysql', and 'root'
- Passwords - 'blank', 'pass', 'pass1', 'password', and 'vicidia1now'

Default Credentials for PostgreSQL

The following usernames and passwords are common defaults for PostgreSQL:

- Usernames - 'admin', 'postgres', 'scott', and 'tom'
- Passwords - 'admin', 'password', 'postgres', and 'tiger'

Default Credentials for SMB

The following usernames and passwords are common defaults for SMB:

- Usernames - 'backup' and 'helpdesk'
- Passwords - 'backup' and 'hpinvent'

Default Credentials for SNMP

The following usernames and passwords are common defaults for SNMP:

- Usernames - <BLANK>
- Passwords - '0392a0', '1234', '2read', '4changes', 'access'. 'adm', 'Admin', 'admin', 'agent', 'agent_steal', 'all', 'all private', 'all public', 'ANYCOM', 'apc', 'bintec', 'blue', 'c', 'C0de', 'cable-d', 'canon_admin', 'cc', 'CISCO', 'cisco', 'community', 'core', 'CR52401', 'debug', 'default', 'dilbert', 'enable', 'field', 'field-service', 'freekevin', 'fubar', 'guest', 'hello', 'hp_admin', 'IBM', 'ibm', 'ILMI', 'ilmi', 'Intermec', 'intermec', 'internal', 'l2', 'l3', 'manager', 'mngt', 'monitor', 'netman', 'network', 'NoGaH$@!', 'none', 'openview', 'OrigEquipMfr', 'pass', 'password', 'pr1v4t3', 'private', 'PRIVATE', 'Private', 'proxy', 'publ1c', 'public', 'PUBLIC', 'Public', 'read', 'read-only', 'read-write', 'readwrite', 'red', 'regional', 'rmon', 'rmon_admin', 'ro', 'root', 'router', 'rw', 'rwa', 's!a@m#n$p%c', 'san-fran', 'sanfran', 'scotty', 'SECRET', 'Secret', 'secret', 'SECURITY', 'Security', 'security', 'seri', 'SNMP', 'snmp', 'SNMP_trap', 'snmpd', 'snmptrap', 'solaris', 'SUN', 'sun', 'superuser', 'SWITCH', 'Switch', 'switch', 'SYSTEM', 'System', 'system', 'tech', 'TENmanUFactOryPOWER', 'TEST', 'test', 'test2', 'tiv0li', 'tivoli', 'trap', 'world', 'write', 'xyzzy', and 'yellow'

Default Credentials for SSH

The following usernames and passwords are common defaults for SSH:

- Usernames - 'admin', 'administrator', and 'root'
- Passwords - '1234', 'admin', 'changeme123', 'password', 'password1', 'password123', 'password123!', and 'toor'

Default Credentials for Telnet

The following usernames and passwords are common defaults for telnet:

- Usernames - 'admin', 'administrator', 'Alphanetworks', 'cisco', 'helpdesk', 'pix', and 'root'
- Passwords - '100', ‘admin’, ‘changeme123’, 'cisco', ‘password’, ‘password1’, ‘password123’, ‘password123!’, 'sanfran', 'root', 'wrgg15_di524', 'wrgg19_c_dlwbr_dir300 ', and 'wrgn22_dlwbr_dir615'

Default Credentials for VNC

The following usernames and passwords are common defaults for VNC:

- Usernames - 'admin', 'administrator', and 'root'
- Passwords - '100', '1234', 'admin', 'changeme123', 'password', 'password1', 'password123', 'password123!', and 'toor'

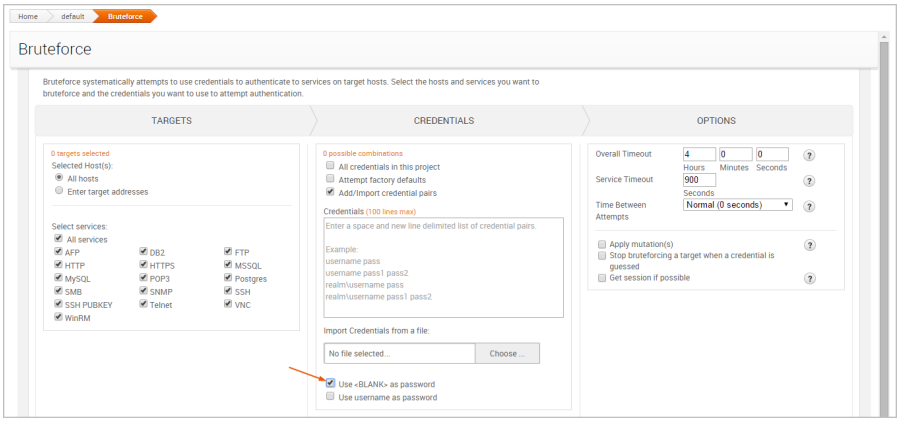
Default Credentials for WinRM

The following usernames and passwords are common defaults for WinRM:

- Usernames - 'admin', 'administrator', and 'root'
- Passwords - '1234', 'admin', 'changeme123', 'password', 'password1', 'password123', 'password123!', and 'toor'

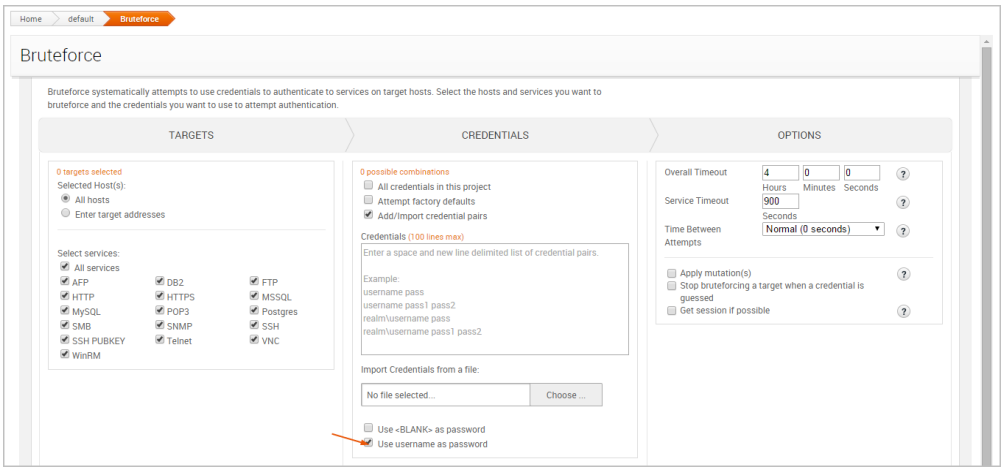
Using Blank Passwords in a Bruteforce Attack

To generate blank passwords for each username in a password list, you can enable the **Use <BLANK> as password option**, as shown below. For example, if the password list contains a credential pair like 'admin'/'admin', Bruteforce will also try admin/'<BLANK>'.



Using a Username as a Password

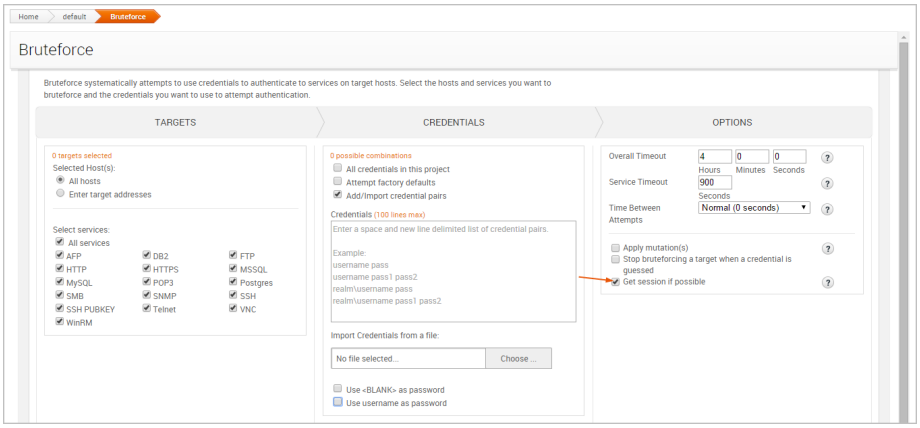
To use a username as a password, you can enable the **Use username as password** option, as shown below. For example, if the password list contains a credential pair like 'user'/'pass', the bruteforce attack will also try 'user'/'user'.



Getting Sessions on Guessed Credentials

In addition to guessing credentials, Bruteforce has the ability to open a session when a credential is guessed for specific services, such as MSSQL, MySQL, PostgreSQL, SMB, SSH, Telnet, WinRM, and some HTTP services, such as Tomcat, Axis2, or GlassFish. Open sessions can be used to perform post-exploitation tasks, such as gathering additional information from the host and leveraging that data to compromise additional hosts.

To open services when Bruteforce successfully cracks a credential on a service, you need to enable the **Get sessions if possible** option on and specify the payload options that you want to use, as shown below. The session will remain open after the attack finishes, which can be used to perform additional post-exploitation tasks.



Configuring Payload Settings for a Bruteforce Attack

The following options can be used to configure the payload settings:

- Payload type: This option determines the type of payload gets delivered to the target. You can choose one of the following options:
  - **Meterpreter**: This payload provides an advanced interactive shell that provides extensive post-exploitation capabilities that enable you to do things like escalate privileges, dump password hashes, take screenshots, launch and migrate processes, and upload files to the target. Meterpreter also includes command shell capabilities for basic tasks like adding a user account or running a script.

    Meterpreter also dynamically loads itself into an existing process on the target host using a technique called reflective DLL injection, which enables it to reside entirely in memory and remain undetected by intrusion prevention and intrusion detection systems.

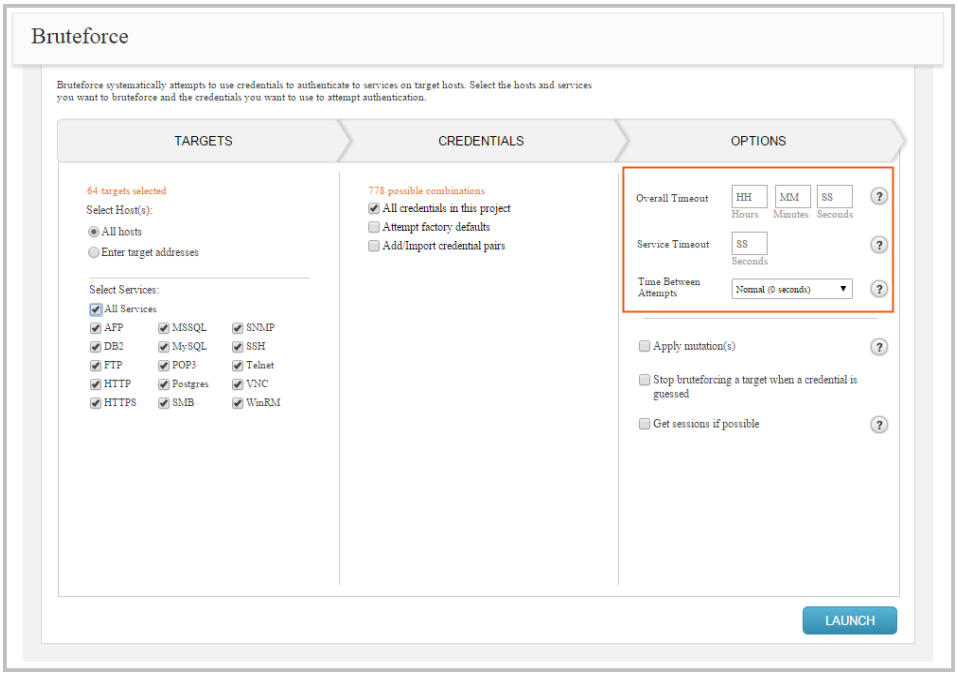
- **Command**: This payload provides a command shell that you can use to run single commands on a host to perform simple tasks like adding a user account or changing a password. A command shell provides limited capabilities, but can be later upgraded to a Meterpreter shell for more options.

  Unlike Meterpreter, a command shell can start a new process that can be easily detected by intrusion prevention and intrusion detection systems.

- Connection: This option determines how your Metasploit instance connects to the host. You can choose one of the following options:
  - **Auto**: This connection type uses a reverse connection when NAT or a firewall is detected; otherwise, it uses bind connection.
  - **Bind** - This connection type uses a bind connection. You should use this connection type if there is a direct, unrestricted connection to the target host.
  - **Reverse**: This connection type uses a reverse connection. You should select this connection type if the hosts are behind a firewall or a NAT gateway that will prevent requests from your Metasploit instance to the target.
- Listener ports: This option defines the ports that the listener uses to wait for incoming connections. You can specify a specific port, a comma separated list of ports, or a port range. If you enter a port range, the first available open port is chosen from the range.
- Listener host: This option defines the IP address the target host connects back to. This is typically going to be the external IP address of your local machine. If you do not specify a listener host, the external IP address of your local machine is used.

Setting the Timeout for a Bruteforce Attack

You can control the amount of time that is allocated to the overall bruteforce task and for each individual service. You can set timeout limits from the options area of the Bruteforce Workflow, as shown below:



The following timeout options are available:

- **Service timeout** - Sets the timeout, in minutes, for each service.
- **Overall timeout** - Sets the maximum amount of time, in minutes, that will be allocated for the bruteforce run. If an overall timeout is enforced, Bruteforce will attempt as many guesses as it can during that timeframe. Bruteforce may not be able to attempt all credentials if a timeout is set.
- **Timeout between attempts** - Sets the time that elapses, in seconds, between each login attempt.You can choose between any of the predefined time limits: Normal (0 seconds), Stealthy (5 seconds), Slow (10 seconds), and Glacial (60 seconds).

If no timeout options are set, the Bruteforce Workflow defaults to 0 and does not enforce a timeout limit.

Applying Mutation Rules for a Bruteforce Attack

Oftentimes, organizations use variations of a base word to configure default account settings, or they use leetspeak to substitute characters. To cover these particular scenarios, you can to apply mutation rules to create different permutations of a private.

A mutation rule appends, prepends, and substitutes characters in a private. You can use them to effectively build a larger list of passwords based on a set of base words. For example, if you have identified that an organization commonly uses passwords that contain the company's name, you can add the company's name to the word list and apply mutations to automatically generate multiple variations of it. Therefore, depending on the mutation rules that are applied, a private, like "mycompany" can have several variations, such as "mycompany2014", "mycompany1", "mycomp@ny", and so on.

There are several different types of mutation rules that you can apply, such as appending and prepending digits to a private, applying leetspeak substitutions to a private, and appending and prepending the current year to a private. The mutation rules are disabled by default, so you will need to enable the mutation option and select the rules you want to use. If enabled, the mutation rules will be applied to the credentials you have selected for the bruteforce attack.

Applying Leetspeak Substitutions

Leetspeak is an alternative alphabet that can be used to substitute letters with special characters and numbers.

You can enable the **1337 speak** option to perform individual leetspeak substitutions on a private. If you enable the 1337 speak option, the following rules are applied to each private:

- The mutation rule changes all instances of the letter "a" to "@".
- The mutation rule changes all instances of the letter "a" to "4".
- The mutation rule changes all instances of the letter "e" to "3".
- The mutation rule changes all instances of the letter "l" to "1".
- The mutation rule changes all instances of the letter "o" to "0".
- The mutation rule changes all instances of the letter "s" to "5".
- The mutation rule changes all instances of the letter "s" to "$".
- The mutation rule changes all instances of the letter "t" to "7".

Each leetspeak rule is applied individually. For example, if the private is "mycompany", the leetspeak mutation rule creates two permutations: "myc0mpany" and "mycomp@ny". It does not combine leetspeak rules to create "myc0mp@ny".

Prepending Special Characters (!#*)

You can enable the **Prepend special characters** option to add a special character to the beginning of a private. If enabled, the rule prepends an exclamation point (!), a hash symbol (#), an ampersand (&), and an asterisk (*) to a private. For example, if the private is "mycompany", the following permutations are created: "!mycompany", "#mycompany", "&mycompany", and "*mycompany".

Appending Special Characters (!#*)

You can enable the **Append special characters** option to add a special character to the end of a private. If enabled, the rule appends an exclamation point (!), a hash symbol (#), an ampersand (&), and an asterisk (*) to a private. For example, if the private is "mycompany", the following permutations are created: the following permutations are created: "mycompany!", "mycompany#", "mycompany&", and "mycompany*".

Prepending a Single Digit

You can enable the **Prepend single digit** option to add a single digit to the beginning of a private. If enabled, the rule prepends the digits 0-9 to a private. For example, if the private is "mycompany", the following permutations are created: "mycompany0", "mycompany1", "mycompany2", "mycompany3", and so on.

Appending Single Digit

You can enable the **Append single digit** option to add a single digit to the end of a private. If enabled, the rule appends the digits 0-9 to a private. For example, if the private is "mycompany", the following permutations are created: "0mycompany", "1mycompany", "2mycompany", "3mycompany", and so on.

Prepending Digits

You can enable the **Prepend digits** option to add three digits to the beginning of a private. For example, if the private is "mycompany", the following permutations are created: "mycompany000", "mycompany001", "mycompany002", "mycompany003", and so on.

Note: If enabled, this rule can generate up to 1,000 permutations of a single private.

Appending Digits

You can enable the **Append digits** option to add three digits to the end of a private. For example, if the private is "mycompany", the following permutations are created: "000mycompany", "001mycompany", "002mycompany", "003mycompany", and so on.

Note: If enabled, this rule can generate up to 1,000 permutations of a single private.

Prepending the Current Year

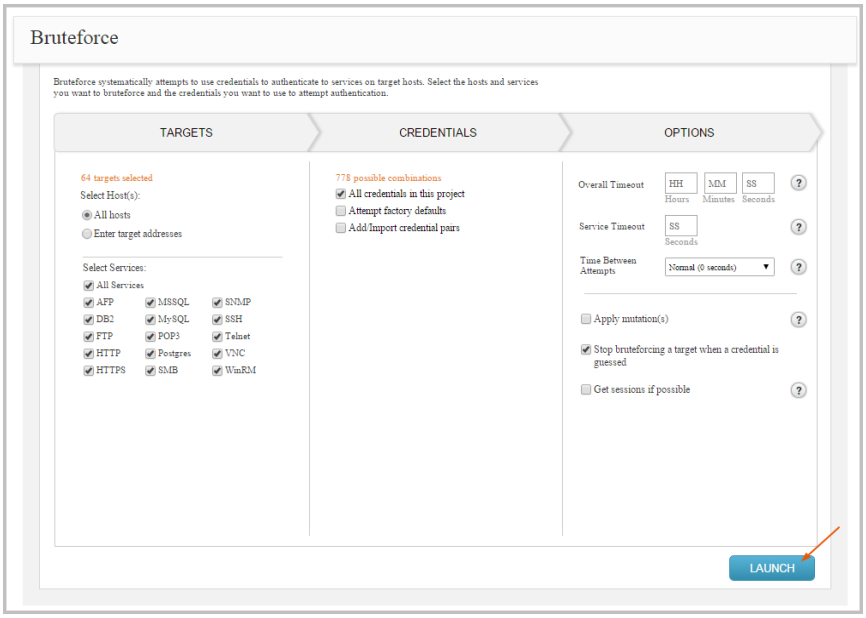
You can enable the **Prepend current year** option to add the current year to the beginning of a private. For example, if the private is "mycompany", the following permutations are created: "2014mycompany", "2014mycompany", "2014mycompany", "2014mycompany", and so on.

Appending the Current Year

You can enable the **Append current year** option to add the current year to the end of a private. . For example, if the private is "mycompany", the following permutations will be created: "mycompany2014", "mycompany2014", "mycompany2014", "mycompany2014", and so on.

Launching the Bruteforce Attack

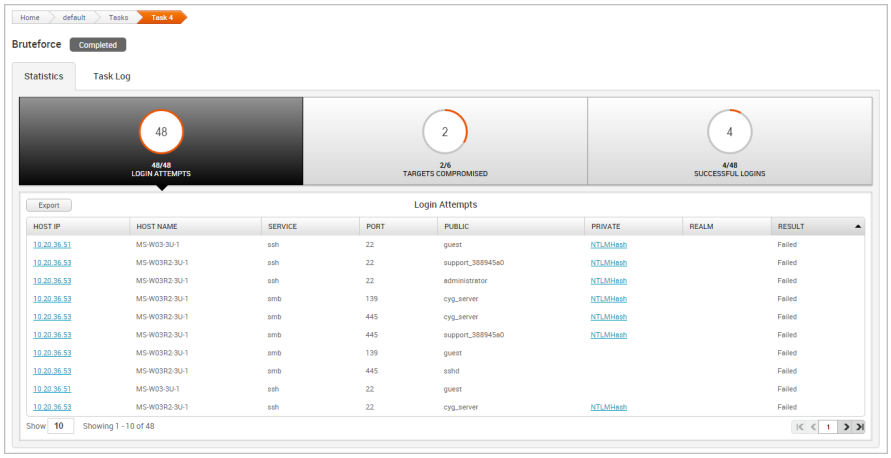
The Launch button on the Bruteforce configuration page becomes active when all required fields have been filled out. When you are ready to run the bruteforce attack, click the **Launch** button.



If there are any issues with the attack configuration, a warning will appear next to the misconfigured setting. You must fix the issue before you can launch the bruteforce attack.

Understanding Bruteforce Findings

After you launch the bruteforce attack, the findings window appears and displays the real-time results and events for the attack. To help you navigate the data, the findings window is organized into two major tabs: the Statistics tab and the Task Log tab.

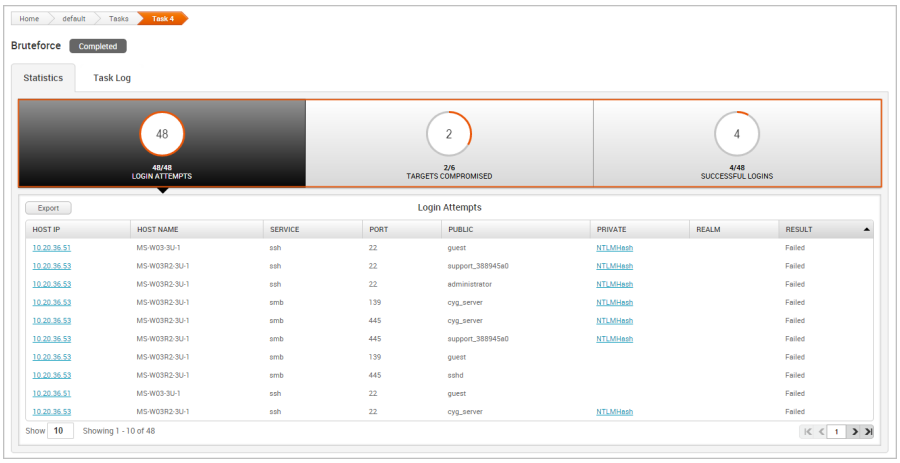


The Statistics Tab

The Statistics tab tracks the real-time results for the following:

- The total number of login attempts
- The total number of targets compromised
- The total number of successful logins

Each statistic is displayed on its own tab. You can click on a tab to view the corresponding table for each statistic.



Viewing the Login Attempts Table

The Login Attempts table displays every login that the bruteforce attack has tried. The following information is listed for each login:

- The IP for each host
- Host name
- Service name
- Service port
- Public
- Private
- Realm type
- Login result

Viewing the Targets Compromised Table

The Targets Compromised table displays every target to which the bruteforce attack was able to successfully authenticate. The following information is listed for each compromised target:

- Host IP
- Host name
- Operating system
- Service name
- Service port
- Number of captured credentials
- Sessions*

* If you configured the bruteforce attack to get sessions, the Targets Compromised table includes a Sessions column that lists the total number of sessions that were opened on each target. You can hover over the session count to view a list of links that you can use to access the details page for each session.

Viewing the Successful Logins Table

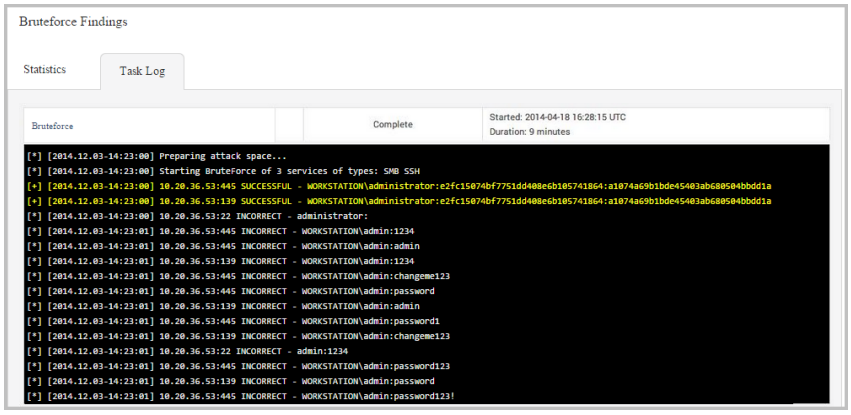
The Successful Logins table displays all the logins that the bruteforce attack was able to validate. The following information is listed for each login:

- Host IP
- Host name
- Operating system
- Service name
- Service port
- Public
- Private
- Realm type
- Sessions details*

* If you configured the bruteforce attack to get sessions, the Targets Compromised table has a Go to Sessions column that displays a link for each open session. You can click on the link to access the details page for the session. If you did not configure the bruteforce attack to get sessions, the Targets Compromised table has an Attempt Session column, which enables you to try to validate the login from the findings window.

The Task Log Tab

The task log tracks the activity for the bruteforce attack. It lists the target that is being bruteforced and the result for each guess attempt. A successful login will be highlighted in yellow, as shown below:



The task log also documents any errors or failures that occurred during the attack and can be used to troubleshoot any issues related to the bruteforce run. This is especially helpful if, for example, the attack has been running for a long time, appears to be hanging, or does not complete.

Custom Credential Mutations

Credential mutations mangle the passwords in a wordlist. You can use them to create numerous permutations of a password, such as switch out letters for numbers (password becomes “passw0rd”), thus enabling you to build a large wordlist based on a small set of passwords.Credential mutations can also add numbers and special characters to a password, toggle the casing of letters, and control the length of a password.

Metasploit Pro offers several canned mutations that you can use. However, if you want to generate a custom mutated credentials list,you will need to run your wordlist through a password cracking tool like John the Ripper. John the Ripper will apply a set of mangling rules to the wordlist and output a list of mutated credentials that you can import into a project.

John the Ripper

John the Ripper is a free password cracker that is available for many flavors of Unix, Windows, DOS, BeOS, and OpenVMS. John the Ripper is typically used to detect weak passwords and hashes, but you can also use it to generate a mutated wordlist that you can import into Metasploit Pro to use with Bruteforce Guess.

Downloading and Installing John the Ripper

The Metasploit installer includes the binaries for John the Ripper 1.7.8, which are located in `path/to/metasploit/apps/pro/msf3/data/john`. To run John the Ripper, you'll need to simply invoke the binary from the run directory for your target. If you want the latest version of John the Ripper, you will need to install it. John the Ripper has extensive installation instructions located at http://www.openwall.com/john/doc/INSTALL.shtml. However, if you'd rather learn from us, read on.

To start, you will need to download John the Ripper at http://www.openwall.com/john/. A free, community enhanced, and commercial version of John the Ripper are available. Download the version that works for your system.

John the Ripper is distributed primarily in source code form and downloaded as an archive file. After you download the archive file, you will need to extract its contents. To run John the Ripper, you will simply need to invoke it from the extraction location.

Installing John the Ripper on Linux

Open a terminal and type the following:

```
sudo -sH
cd /opt
wget http://www.openwall.com/john/j/john-1.8.0.tar.gz
tar -xvzf john-1.8.0.tar.gz
mv john-1.8.0 john
rm john-1.8.0.tar.gz
```

Note: If you are using a different version of John the Ripper, replace `1.8.0` with the version you have.

```
cd /opt/john/src
make
```

Make displays a list of targets. Choose the best one for your architecture and rerun the `make` command with the target you have chosen. For example, for a Linux x86-64 system, type the following:

```
make clean linux-x86-64
```

John the Ripper is now installed on your Linux system and is ready to use.

Installing John the Ripper on Windows

1. Go to http://www.openwall.com/john and download the latest Windows installer.

   The binaries for Windows will be available in a ZIP file.

2. Locate the downloaded file and extract it to your `C` drive. The resulting directory will be something like `C:\john179`.

   If you are using a different version of John the Ripper, replace `john179` with the version you have.

3. Rename the directory to `john`. The resulting directory will be `C:\john`.

   John the Ripper is now installed on your Windows system and is ready to use.

Running John the Ripper on Linux

Open the command line terminal and type the following:

```
sudo -sH
cd /path/to/john/run
./john
```

Note: Replace `/path/to/` with the location where you have John the Ripper installed.

John the Ripper prints a list of options that are available. For more information on John the Ripper options, see http://www.openwall.com/john/doc/OPTIONS.shtml.

Running John the Ripper on Windows

Open the command line terminal (**Start > Run > cmd**) and type the following:

```
cd \path\to\john\run
john
```

Note: Replace `\path\to\` with the location where you have John the Ripper installed.

John the Ripper prints a list of options that are available. For more information on John the Ripper options, see http://www.openwall.com/john/doc/OPTIONS.shtml.

Custom Mutation Rules

Now that you have John the Ripper installed and have verified that it runs, you will need to add mutation rules to the John configuration file. A rule enables you to perform a specific type of mutation on a word. For example, you can create a rule that appends specific special characters, such as `[!@#$%^&*()+=.?]`, to every word in the wordlist.

To define custom mutation rules, you will need to create rule sets in the John configuration file. A rule set is a logical grouping of related mutation rules. All mutation rules must be contained within a rule set. For example, you should create separate rule sets for rules that append characters, rules that prepend characters, and rules that control character lengths.You will need to supply the rule sets as part of the rules option when you generate the wordlist, like the following:

```
./john --wordlist=[path to word list] --stdout --rules:[ruleset name] >
[path to output list]
```

Custom Rule Set Requirements

Each rule set section must start with its type and name enclosed in brackets. Rule sets must use the following format: `[List.Rules:<name>]`, where `<name>` is replaced with the name of the rule set.

For example, if you have a rule set named `AppendLowercaseNumbers`, the section must be written as `[List.Rules:AppendNumbers]` in the John configuration file.

So, your rule set will look something like this:

```
[List.Rules:AppendLowercaseNumbers]
#lowercase and add numbers to the end of a password
l$[0-9]
l$[0-9]$[0-9]
l$[0-9]$[0-9]$[0-9]
l$[0-9]$[0-9]$[0-9]$[0-9]
```

Also, here are a couple of other formatting rules that you should remember:

- Section names are not case sensitive.
- Comment lines start with a hash ("#") or a semicolon (";").
- Empty lines are ignored.
- Rule sets can be placed anywhere in the John configuration file, but as a best practice, you should add them to the bottom of the file.

Accessing the John Configuration File

To access the John configuration file, go to `/path/to/john/run`. The configuration file will be named either `john.conf` or `john.ini` depending on your operating system. You will need to open the configuration file using a text editor, like Vim (Linux) or Notepad (Windows).

```
sudo -sH
cd /opt/john/run
vim john.conf
```

If you do not have Vim, you can install it with the following command: `apt-get install vim`.

The configuration file consists of several sections. Each section has a unique function and is assigned a section name that is enclosed in square brackets, as shown below:

```
# This file is part of John the Ripper password cracker,
# Copyright (c) 1996-2006,2008-2013 by Solar Designer
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted.
#
# There's ABSOLUTELY NO WARRANTY, express or implied.
#
# Please note that although this configuration file is under the cut-down BSD
# license above, many source files in John the Ripper are under GPLv2.
# For licensing terms for John the Ripper as a whole, see doc/LICENSE.
#

[Options]
# Wordlist file name, to be used in batch mode
Wordlist = $JOHN/password.lst
# Use idle cycles only
Idle = Y
# Crash recovery file saving delay in seconds
Save = 600
# Beep when a password is found (who needs this anyway?)
Beep = N

# "Single crack" mode rules
[List.Rules:Single]
# Simple rules come first...
:
-s x**
-c (?a c Q
-c l Q
-s-c x** /?u l
```

Basic John the Ripper Rules Syntax

John the Ripper has great documentation that explains the syntax for building rules. You can read them at http://www.openwall.com/john/doc/RULES.shtml. It is highly recommended that you review their documentation before building your own custom rules.

To help you get started, here is a quick overview of some of the commands you might be interested in:

- **$** appends a character or number to a word. You can define a single character, such as `$1`, which will append the number 1 to a password, or you can define a group of characters, such as $[0-9], which will append 0, 1, 2, 3 ,4, 5, 6, 7, 8, and 9 to a password.
- **^** prepends a character or number to a word. You can define a single character, such as `^1`, which will prepend the number 1 to a password, or you can define a group of characters, such as ^[0-9], which will prepend 0, 1, 2, 3 ,4, 5, 6, 7, 8, and 9 to a password.
- **l** converts all the letters in the word to lowercase.
- **c** converts all of the letters in the word to uppercase.
- **C** lowercases the first character in the word and uppercases the rest.
- **t** toggles the case of all characters in the word.

You can use any combination of commands within a rule. For example, you can create rules that prepends and appends numbers to a password, such as `^[0-9]$[0-9]`. Each rule must appear on a newline.

Creating Custom Mutation Rules

Now, let's go ahead and add some mutation rules to the John configuration file.

We will cover the following mutations:

- Appending a number to a password.
- Prepending a number to a password.
- Appending special characters to a password.
- Prepending special characters to a password.
- Lowercasing the password and adding special characters and numbers to it.
- Uppercasing the password and adding special characters and numbers to it.

To add these rules, open the John configuration file with a text editor and scroll to the bottom of the file. You can copy and paste any of these rules to the John configuration file, or you can use these as examples to build your own rules. After you have added the rules to the John configuration file, remember to save your changes.

Note: Before you make any modifications to the John configuration file, you should make a copy of the original file in case you need to revert back to it.

Appending Numbers to a Password

```
[List.Rules:AppendNum]

#Appends one, two, and three numbers to a password

$[0-9]
$[0-9]$[0-9]
$[0-9]$[0-9]$[0-9]
```

Prepending Numbers to a Password

```
[List.Rules:PrependNum]

#Prepends one, two, and three numbers to a password

^[0-9]
```

```
^[0-9]^[0-9]
^[0-9]^[0-9]^[0-9]
```

Appending Special Characters to Passwords

```
[List.Rules:AppendSpecialChar]

#Appends one, two, and three special character to a password

$[!@#$%^&*()+=.?]
$[!@#$%^&*()+=.?]$[!@#$%^&*()+=.?]
$[!@#$%^&*()+=.?]$[!@#$%^&*()+=.?]$[!@#$%^&*()+=.?]
```

Prepending Special Characters to a Password

```
[List.Rules:PrependSpecialChar]

#Prepends one, two, and three special character to a password

^[!@#$%^&*()+=.?]
^[!@#$%^&*()+=.?]^[!@#$%^&*()+=.?]
^[!@#$%^&*()+=.?]^[!@#$%^&*()+=.?]^[!@#$%^&*()+=.?]
```

Lowercasing the Password and Appending Numbers and Special Characters

```
[List.Rules:LowercaseNumChar]

#Lowercases all of the letters of the password and adds a number and/or
special character to it

l
l$[0-9]
l$[0-9]$[0-9]
l$[!@#$%^&*()+=.?]
l$[!@#$%^&*()+=.?]$[!@#$%^&*()+=.?]
l$[0-9]$[!@#$%^&*()+=.?]
l$[0-9]$[0-9]$[!@#$%^&*()+=.?]$[!@#$%^&*()+=.?]
```

Uppercasing the Password and Appending Numbers and Special Characters

```
[List.Rules:UppercaseNumChar]

#Uppercases all of the letters of the password and adds a number and/or
special character to it

u
u$[0-9]
u$[0-9]$[0-9]
u$[!@#$%^&*()+=.?]
u$[!@#$%^&*()+=.?]$[!@#$%^&*()+=.?]
u$[0-9]$[!@#$%^&*()+=.?]
u$[0-9]$[0-9]$[!@#$%^&*()+=.?]$[!@#$%^&*()+=.?]
```

Generating the Mutated Wordlist

When you are ready to generate the mutated wordlist, you will need to run the following: `--wordlist=[path to the pre-mutated wordlist] --stdout --rules:[rule set name] > [path to the generated list]`. You will need to invoke John the Ripper using the appropriate method for your operating system.

John the Ripper will generate the wordlist using the rules that you have specified. If you want to apply all the rules in the configuration file to the wordlist, you can just specify the --rules option.

Generating the Wordlist on Linux

```
./john --wordlist=[path to the wordlist] --stdout --rules:[rule set name]
> [path to the generated list]
```

You must replace the brackets with the appropriate information. For example:

```
./john --wordlist=password.lst --stdout --rules:PrependSpecialChar >
/home/usr/Desktop/mutatedpswds.lst
```

Generating the Wordlist on Windows

```
john --wordlist=[path to the wordlist] --stdout --rules:[rule set name] >
[path to the generated list]
```

You must replace the brackets with the appropriate information. For example:

```
john --wordlist=password.lst --stdout --rules:PrependSpecialChar >
/Desktop/mutatedpswds.lst
```

Importing John the Ripper Wordlists in to a Project

Now that you have a mutated wordlist, you can import it into a project to use with Bruteforce.

1. Go the **Manage Credentials** screen.
2. Click the **Add** button.
3. When the **Add Credentials** window appears, select the **Import** option.
4. Click the **Browse** button and navigate to the location of the wordlist you want to import.

   The file must be a .txt or .lst file.
5. Select the file and click **Open**.
6. From the Add Credentials window, select **Wordlist** as the format.
7. Click the **File Type** dropdown and select **Passwords**.
8. Enter tags for the passwords in the wordlist.(Optional)

   Tags will help you easily search for and identify certain passwords.To enter a tag, start typing the name of the tag you want to use in the **Tag** field. As you type in the search box, Metasploit Pro automatically predicts the tags that may be similar to the ones you are searching for. If the tag does not exist, Metasploit Pro creates it.
9. Click **OK**.

   The wordlist is imported into the project. You can go to the Manage Credentials page to view the imported credentials.

Credentials Domino MetaModule

The Credentials Domino MetaModule enables you to determine how far an attacker can get in a network if they are able to obtain a particular credential. It is performs an iterative credentials-based attack to identify the attack routes that are possible if a session is obtained or a credential is captured from a particular host. Its purpose is to help you gauge the impact of a credentials-based attack by reporting the number of hosts that can be compromised and the number of unique credentials can be captured by leveraging a particular credential.

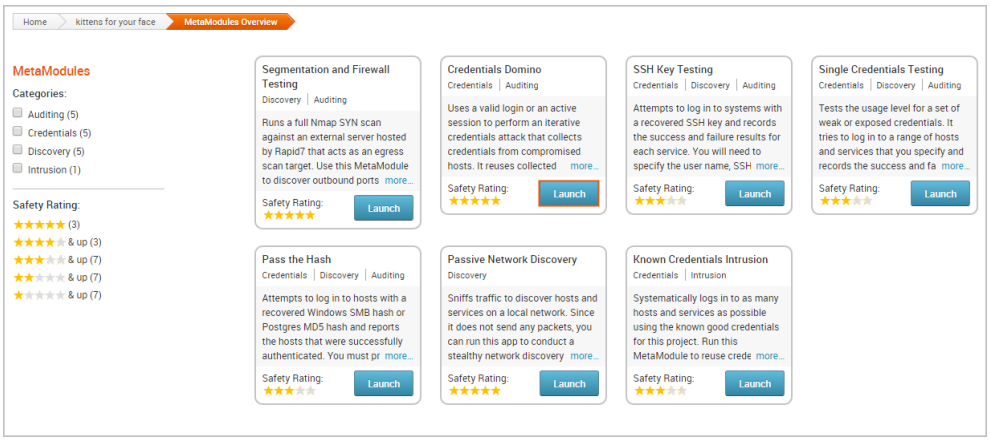
In order to run the Credentials Domino MetaModule, the project must have at least one valid login or one open session that you can use as the starting point. As previously mentioned, the Credentials Domino MetaModule performs an iterative attack, which means that it repeatedly cycles through the network and attempts to authenticate to each host with a particular credential. Each iteration represents a single cycle through a network with a different credential. The Credentials Domino MetaModule continues to run until it opens a session on every target host or it reaches a termination condition.

During the first iteration, if you choose to start with a valid login, the Credentials Domino MetaModule immediately tries to use it to authenticate to and open a session on each target host. If the Credentials Domino MetaModule is able successfully authenticate and open a session, it captures the credentials from the host and stores them in the project. Once the MetaModule has successfully opened a session on a host, it will not try the host again. However, if you choose to start with an open session, the MetaModule begins by collecting credentials from the session. Then, it tries each looted credential until it is able to find a successful login. The MetaModule uses the successful credential to start the next iteration.

To help you see how the network is impacted, the MetaModule includes a specialized report that documents the technical findings and results from the attack. It also presents real-time results for compromised hosts and captured credentials through the findings window and presents a visualization of the attack patterns that it was able to establish from the target network.

Accessing the Credentials Domino MetaModule

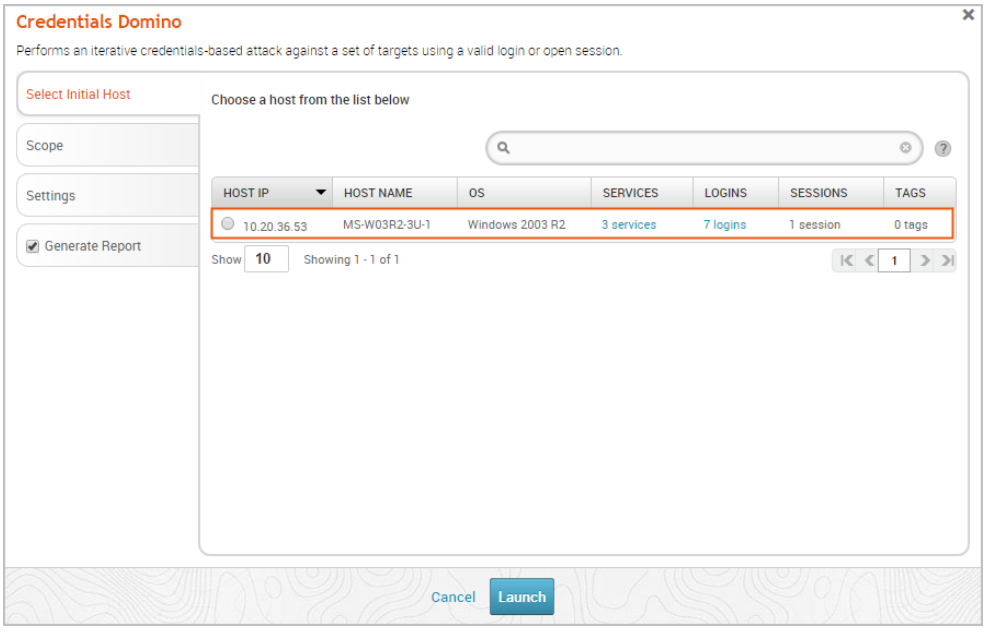
You can access the Credentials Domino MetaModule from the MetaModules page. To access the MetaModules page, select **Modules > MetaModules** from the Project tab bar. Find the Credentials Domino MetaModule and click the **Launch** button.



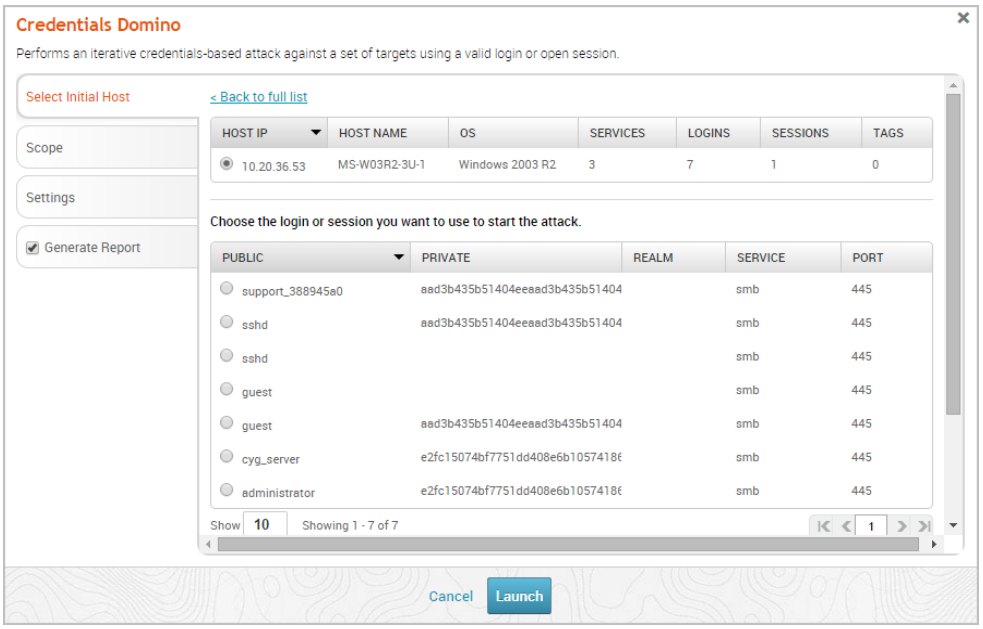
The Credentials Domino MetaModule configuration window appears with the **Select Initial Host** configuration form displayed. Each configuration step is divided into separate tabs, so you can click on any of the tabs to switch between the different configuration forms.

Selecting the Initial Host for the Credentials Domino MetaModule

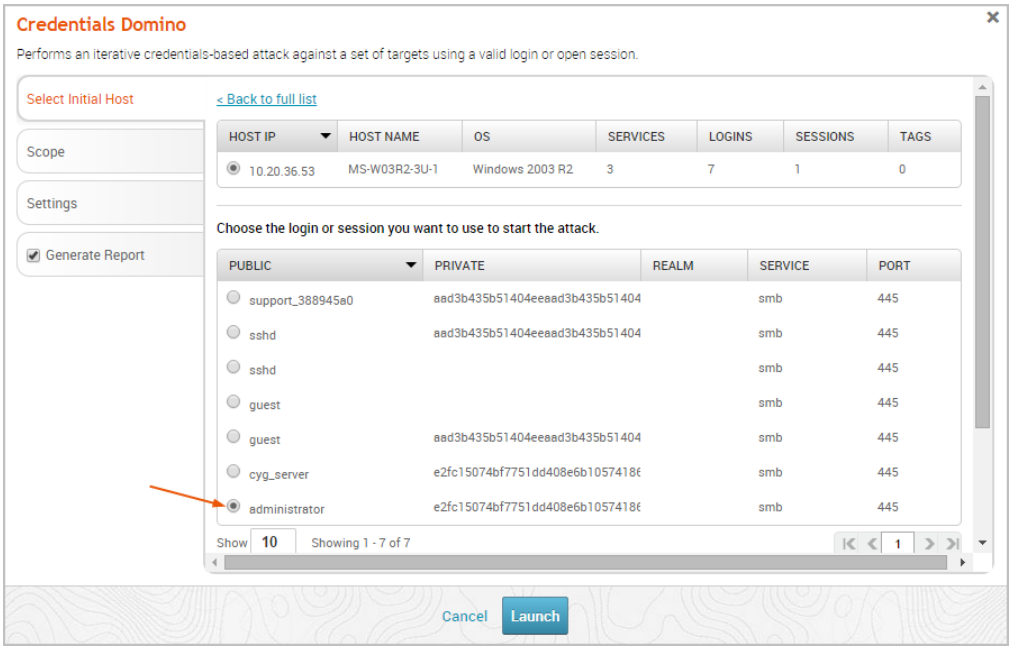
The first thing you need to do when you configure the Credentials Domino MetaModule is select the host that has the login or session you want to use. From the **Select Initial Host** tab, you can see a list of all hosts in the project that either have valid logins or open sessions.



Select the host that you want the MetaModule to use. The login and session details for the host appears after you select a host.

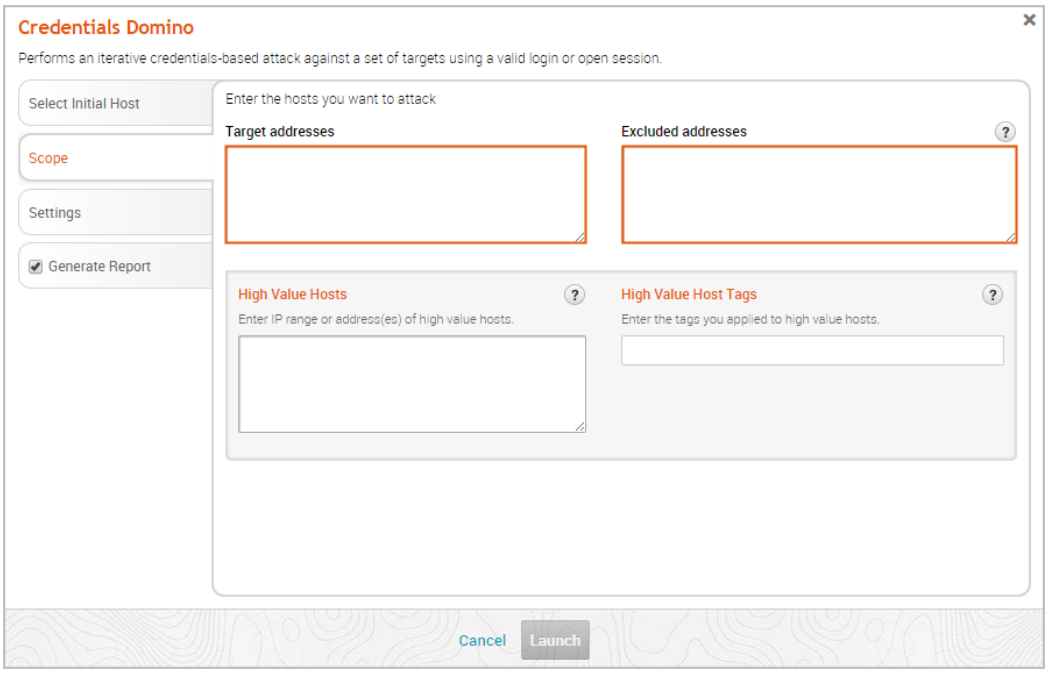


Choose the login or session that you want the MetaModule to use to start the attack.

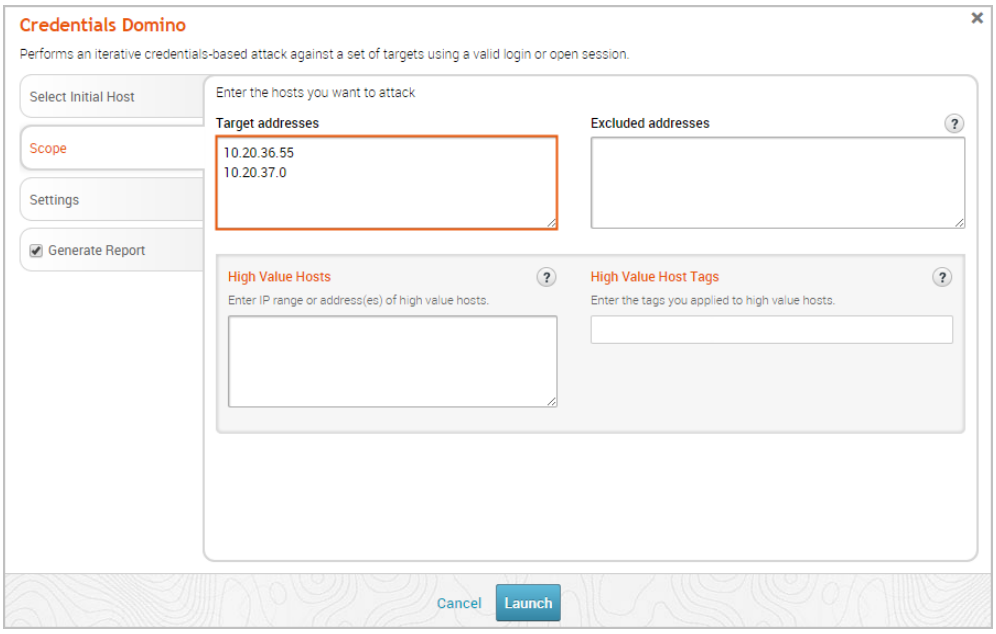


Defining the Scope for the Credentials Domino MetaModule

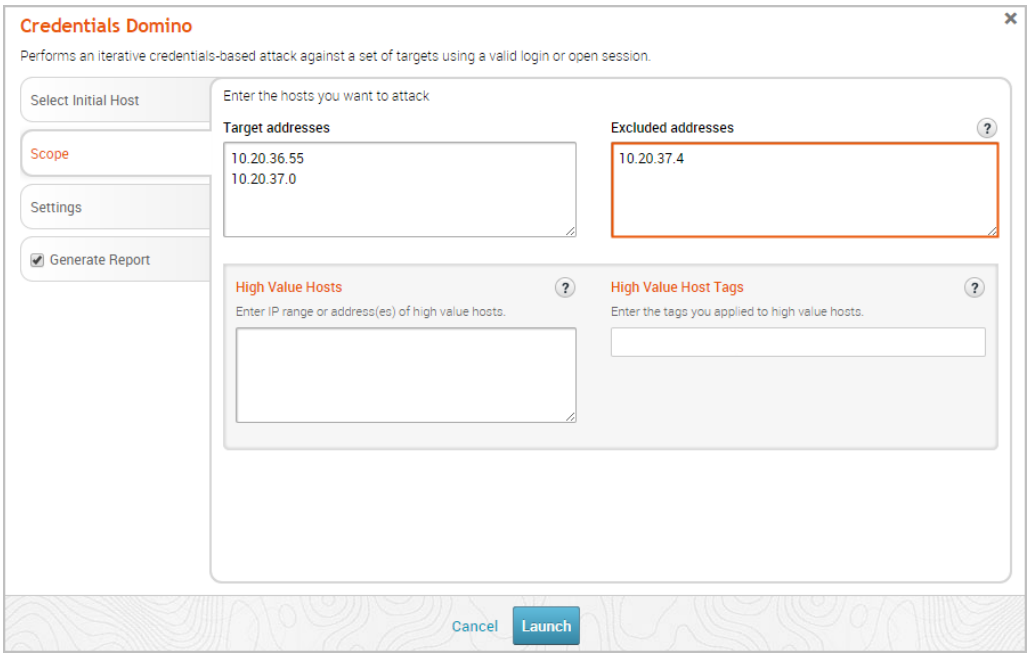
The scope identifies the hosts that you want the MetaModule to target during the attack. To define the scope, you use the **Target addresses** and **Excluded addresses** fields.



If there are specific hosts that you want to attack, you can enter them in the **Target addresses** field. You can enter a single address (192.168.1.1), a range (192.168.1.1-192.168.1.100), a CIDR notation (192.168.1.0/24), or a wildcard (192.168.1.*). You must use a newline to separate each entry. If you want to include all hosts in the project, you can leave this field empty.



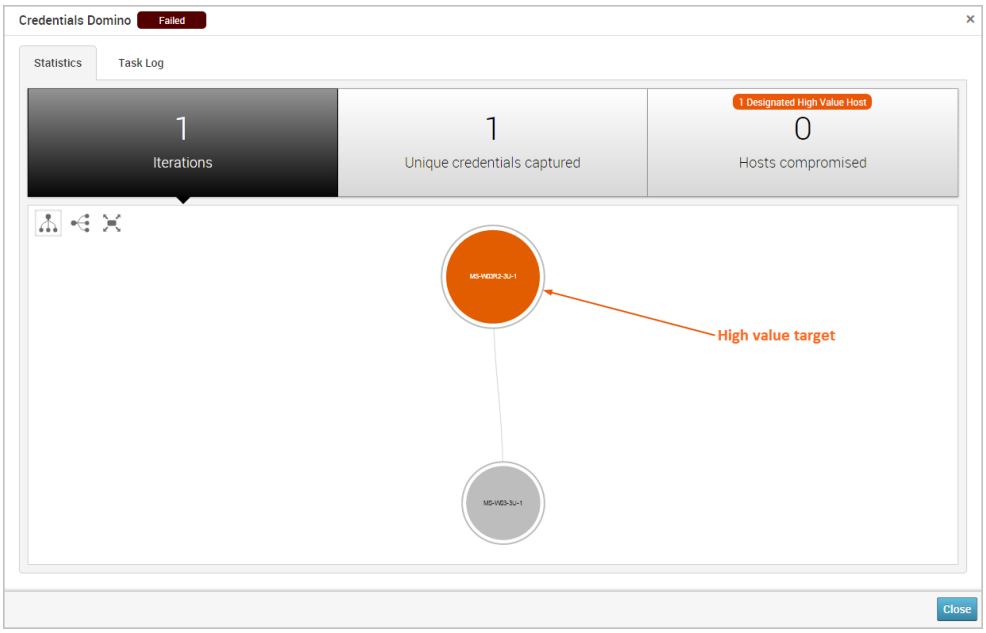
If there are specific hosts that you want to exclude from the attack, you can enter them in the **Excluded addresses** field. Again, you can specify an address range, a CIDR notation, a comma separated list of host addresses, or a single host address.



Designating High Value Hosts for the Credentials Domino MetaModule

A High Value Host is a designation that you assign to a host that you want to highlight on the Credentials Domino Findings window and in the Credentials Domino MetaModule Report. The High Value Host designation helps you quickly identify the impact of a particular stolen credential pair against critical hosts. A High Value Host designation indicates that the host is of significant importance to an organization, and any attack against the host could negatively impact business operations. For example, domain controllers and servers that contain sensitive financial information may be considered as High Value Hosts.

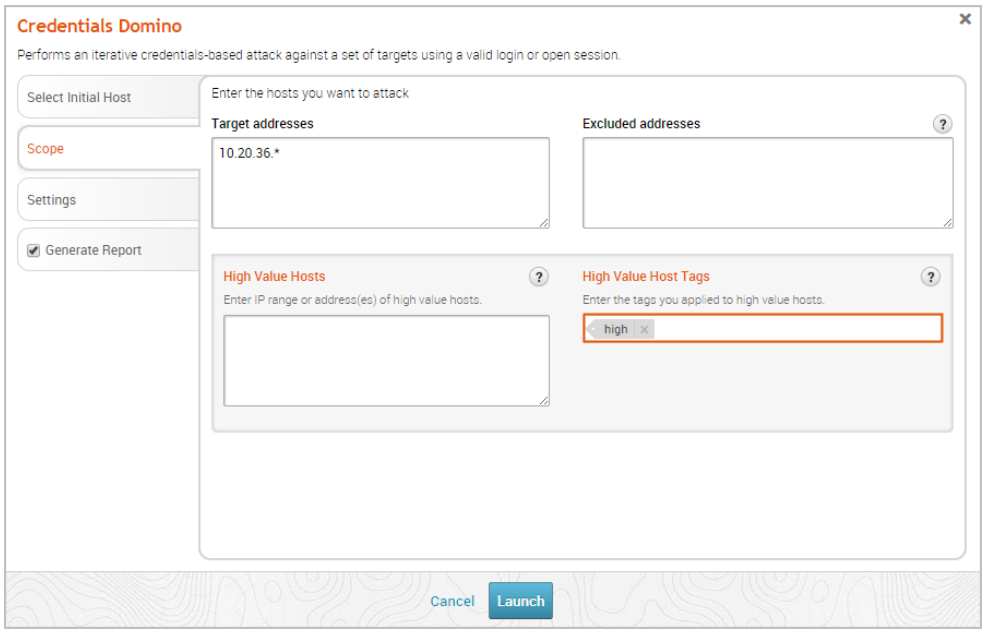
All High Value Hosts will be highlighted in orange on the Findings window, as shown below in the visualization graph:



There are a couple of ways you can designate a host as a high value host:

- **You can apply host tags** - You can apply tags to the hosts that are critical to an organization, which enables you to track, group, and report on hosts according to how they impact an organization. Tags enable you to view and filter hosts at the project level.

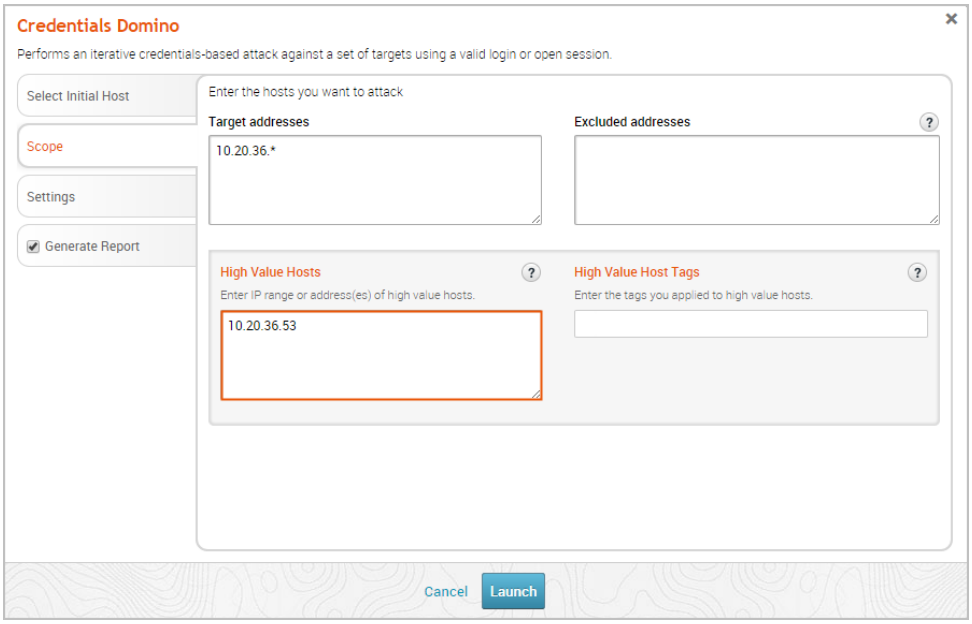
  For example, you may want to tag all accounting servers with a label like, `accounting`. You may also want to want to create and apply criticality tags, such as `High`, `Medium`, and `Low`, to isolate hosts based on their criticality levels.



- **You can specify high value hosts from the Credentials Domino MetaModule** - You can also designate high value hosts by their addresses. You can enter an address range, a single address, or a new line separated list of addresses.

  For example, if you know the IP address for a particular host that is of special interest to you, you can specify it when configuring the scope for the Credentials Domino MetaModule. Use this method if you want to manually define the hosts you want to designate as High Value Hosts and do not want to use tags.

  "



Configuring Payload Settings

You can specify the payload that you want the Credentials Domino MetaModule to deliver during the attack. To configure the payload settings, you can use the following options, which are located on the **Settings** tab:

- Payload type: This option determines the type of payload that the MetaModule delivers to the target. You can choose one of the following options:
  - **Meterpreter**: This payload provides an advanced interactive shell that provides extensive post-exploitation capabilities that enable you to do things like escalate privileges, dump password hashes, take screenshots, launch and migrate processes, and upload files to the target. Meterpreter also includes command shell capabilities for basic tasks like adding a user account or running a script.

    Meterpreter also dynamically loads itself into an existing process on the target host using a technique called reflective DLL injection, which enables it to reside entirely in memory and remain undetected by intrusion prevention and intrusion detection systems.

- **Command**: This payload provides a command shell that you can use to run single commands on a host to perform simple tasks like adding a user account or changing a password. A command shell provides limited capabilities, but can be later upgraded to a Meterpreter shell for more options.

  Unlike Meterpreter, a command shell can start a new process that can be easily detected by intrusion prevention and intrusion detection systems.

- Connection: This option determines how your Metasploit instance connects to the host. You can choose one of the following options:
  - **Auto**: This connection type uses a reverse connection when NAT or a firewall is detected; otherwise, it uses bind connection.
  - **Bind**: This connection type uses a bind connection. You should use this connection type if there is a direct, unrestricted connection to the target host.
  - **Reverse**: This connection type uses a reverse connection. You should select this connection type if the hosts are behind a firewall or a NAT gateway that will prevent requests from your Metasploit instance to the target.
- Listener ports: This option defines the ports that the listener uses to wait for incoming connections. You can specify a specific port, a comma separated list of ports, or a port range. If you enter a port range, the first available open port is chosen from the range.
- Listener host: This option defines the IP address the target host connects back to. This is typically going to be the external IP address of your local machine. If you do not specify a listener host, the MetaModule automatically uses the external IP address of your local machine.
- Clean up sessions: This option enables you to close all open sessions after the MetaModule finishes. By default, this option is enabled. If you want to keep the sessions open, deselect this option.

Setting Termination Conditions for the Credentials Domino MetaModule

You can control the number of times that the Credentials Domino MetaModule cycles through a target network by setting iteration and timeout controls. If you do not set iteration or timeout conditions, the Credentials Domino MetaModule will run until it exhausts all credential and host combinations. Depending on the scope of the attack and the number of credentials captured, the attack can go through a large number of iterations.

To set termination conditions, you use the following options, which are located on the Settings tab:

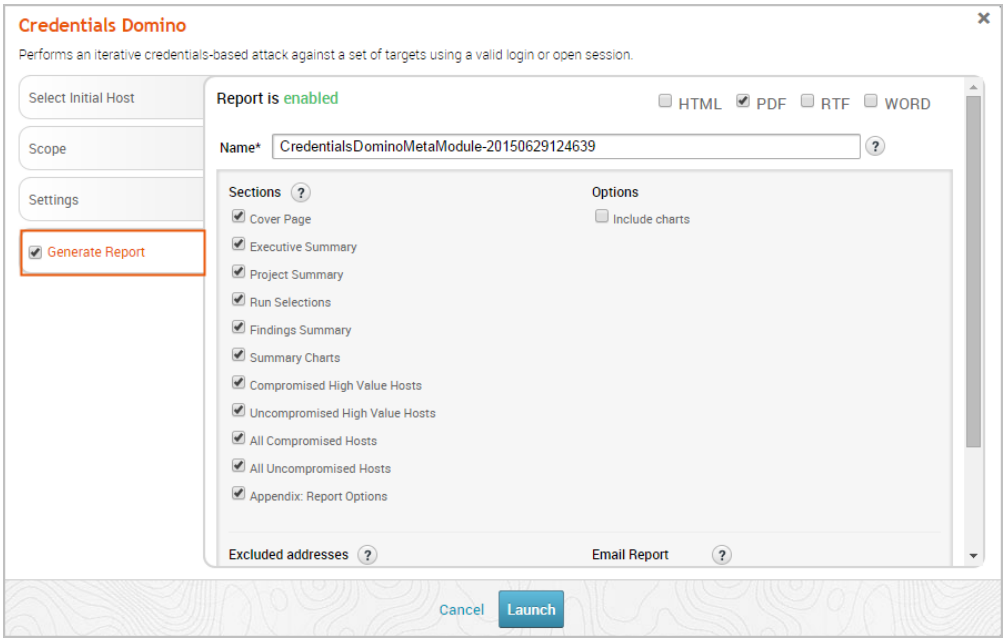
- Number of iterations: This option sets a limit on the number of iterations the MetaModule attempts. You can leave the fields blank if you want the MetaModule to continue until it runs out of credentials to try.
- Overall timeout: This option sets a timeout limit for how long the MetaModule can run in its entirety. You can specify the timeout in the following format: HH:MM:SS. You can leave the fields blank to set no timeout limit.

- Service timeout: This option sets a timeout, in seconds, for each target. You can leave the fields blank to set no timeout limit.

Including a Generated Credentials Domino MetaModule Report

The Credentials Domino MetaModule Report documents the results and technical findings from a credentials-based attack. You can view the report to determine how a validated login or opened session impacted the target network.

To include a Credentials Domino MetaModule Report, you just need to enable the report option.



If you choose to automatically generate a report, you can customize the report using any of the following options:

- Format: The report can be generated as an HTML, PDF, or RTF file. You must select at least one format for the report.
- Name: The report has a default name based on the MetaModule and current date. You can use the default name or provide a custom report name.
- Sections: The report includes the following sections: Cover Page, Project Summary, Findings Summary, Authenticated Services and Hosts Summary Charts, Authenticated Services and Host Details, and Appendix. By default, the report includes all sections. You can deselect any sections that you do not want to include in the report.
- Mask discovered credentials: The report displays all captured credentials in plaintext. If you want to mask the credentials from the report, you must enable this option.

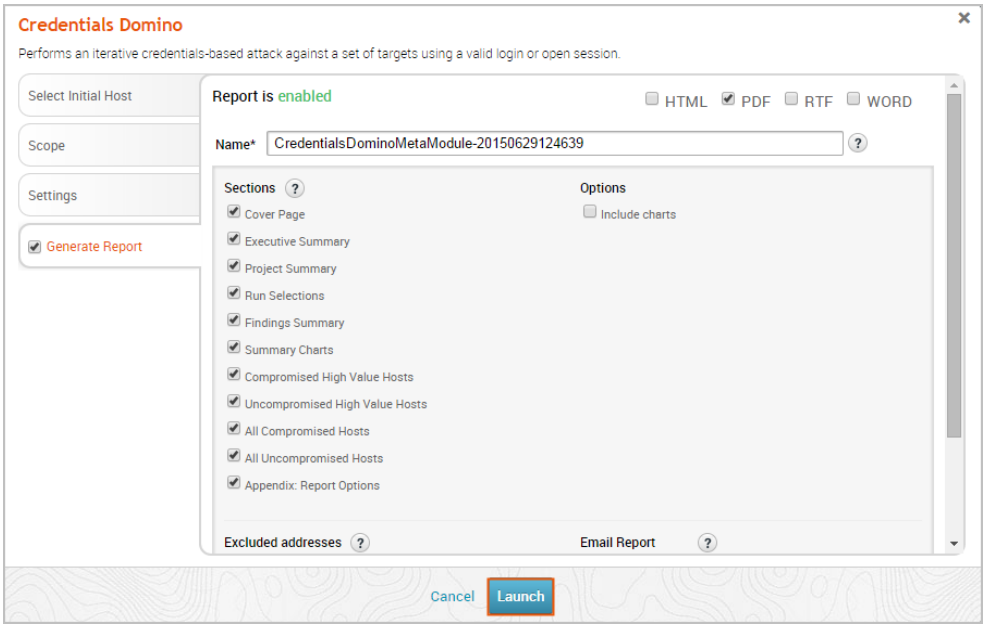
- Include charts: The report includes several charts and graphs that visualize the results from the attack. If you do not want to include visualizations in the report, you must enable this option.
- Excluded addresses: The report includes data for all hosts that you included in the scope. If there are specific hosts whose data you do not want to include in the report, you can list them in this field.
- E-mail report: You can e-mail the report after it is generated. To e-mail the report, you must enter a comma separated list of e-mail addresses in this field.

  Please note that you must already have a local mail server or e-mail relay service set up for Metasploit Pro to use. To define your mail server settings, go to **Administration > Global Settings > SMTP Settings**.

! You can only generate a MetaModule report from the MetaModule. If you do not include a generated report when you configure the MetaModule, you will not be able to do so later.

Launching the Credentials Domino MetaModule

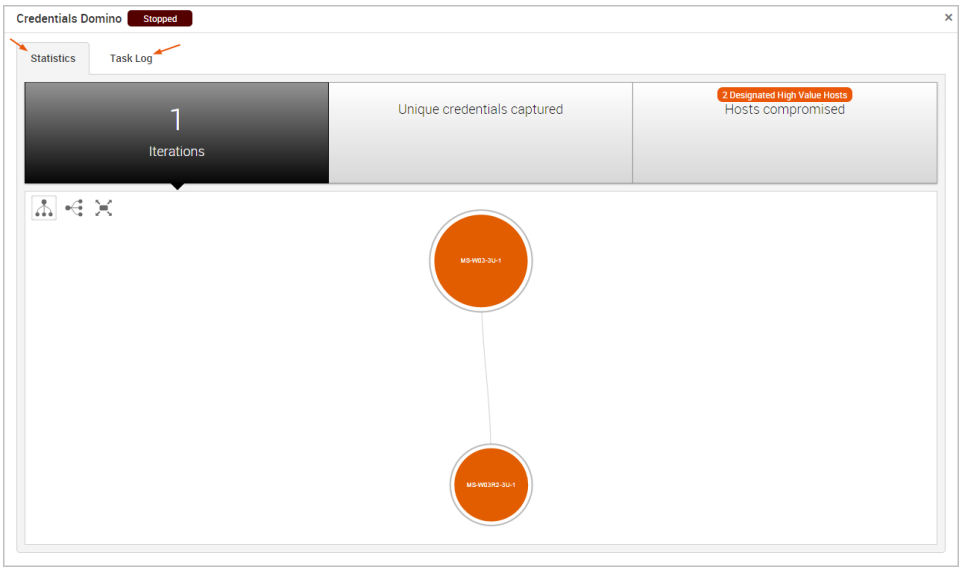
When you are ready to run the Credentials Domino MetaModule, you can click the **Launch** button.



Understanding the Credentials Domino MetaModule Findings

After you launch the Credentials Domino MetaModule, the findings window appears and displays the real-time results and events for the attack. You can quickly see the impact of the attack on the target network and identify the possible attack routes.

To help you navigate the data, the findings window is organized into two major tabs: the Statistics tab and the Task Log tab.

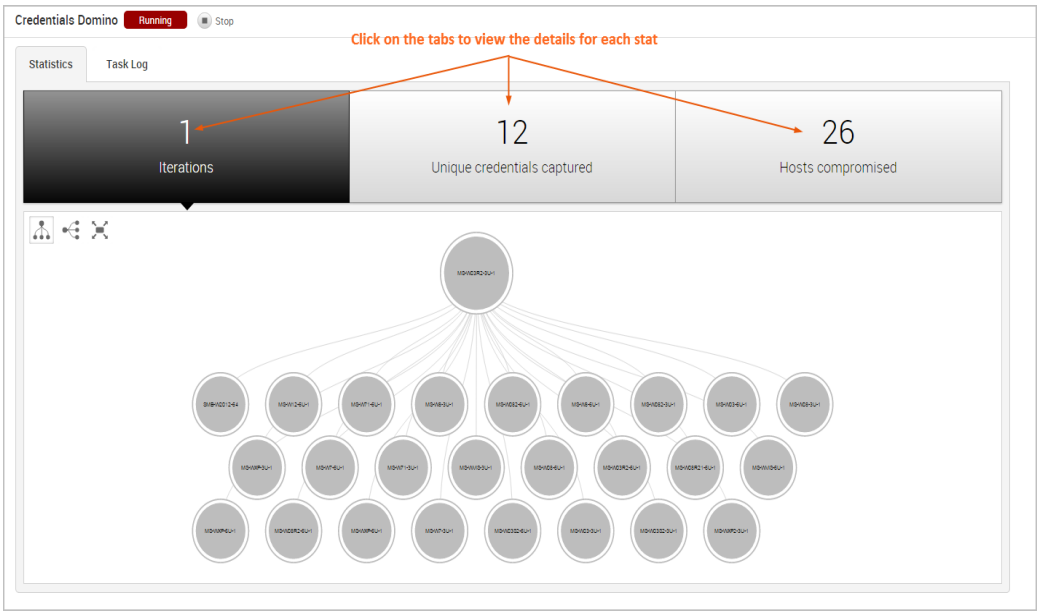


The Statistics Tab

The Statistics tab tracks the real-time results for the attack. It displays statistics for the following:

- The total number of iterations that the MetaModule performed
- The total number of captured credentials
- The total number of compromised hosts

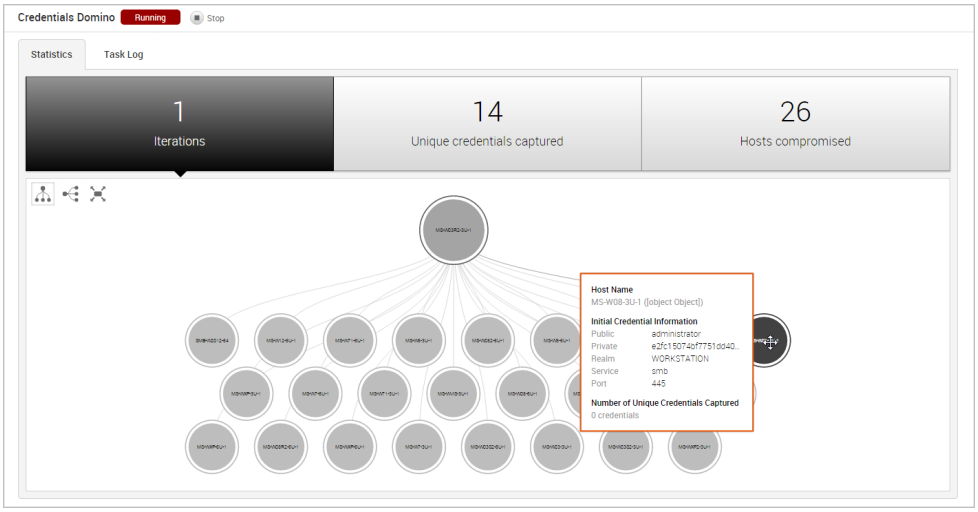
Each statistic is displayed on its own tab. You can click on any of the tabs to view the corresponding details for the statistic.



Viewing the Attack Visualization

To help you visualize the attack routes that the MetaModule was able to establish, the findings window includes an attack visualization that graphically shows the relationship and hierarchy between each compromised host on the network. The visualization uses a tree structure and presents each host as a node on the tree. The starting node is the initial host that the MetaModule used to start the attack.

The attack visualization uses a hierarchical format to connect hosts that are linked to other hosts. The connection between two nodes is created when a credential from one host is used to compromise another host. You can mouseover a node in the attack visualization to highlight the attack route that was used compromise the host.

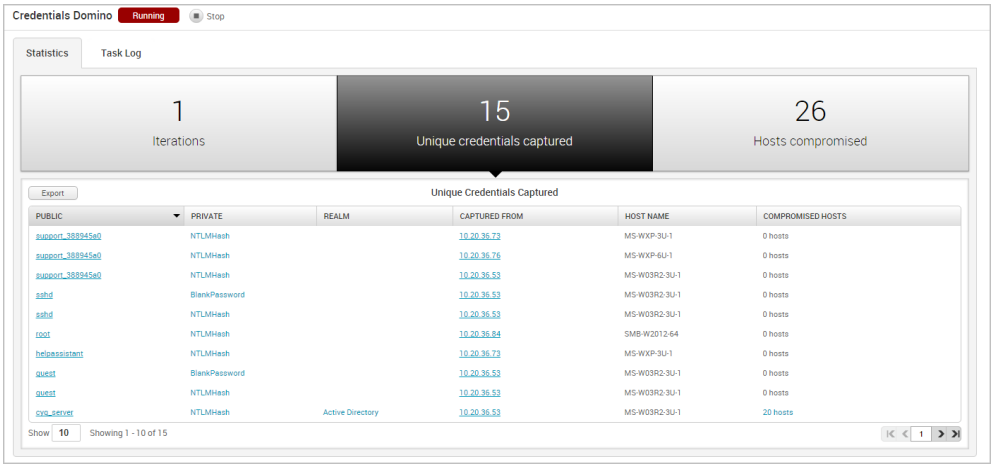


Each iteration represents a level on the tree, so the attack visualization can become quite large if the MetaModule performed multiple iterations of the attack. For example, if the MetaModule performed 20 iterations of an attack, there will be 20 levels represented on the attack visualization. If this happens, you can scroll or double-click the tree to zoom in to view a smaller set of nodes, or you can hover over a specific node to see the details for that host, such as the host name, operating system, credential information, and captured credentials count.

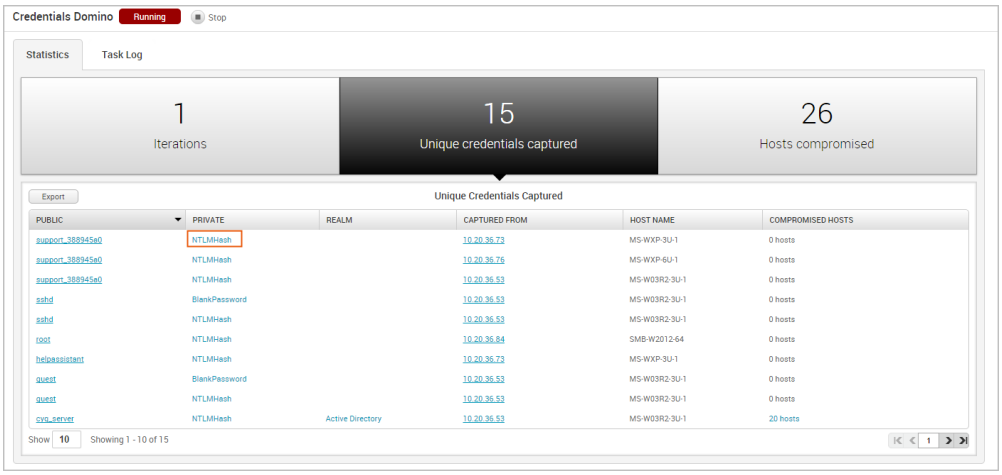
Viewing the Unique Credentials Captured

The Credentials Domino MetaModule tracks credentials that do not share the same public, private, and realm with other captured credentials. It displays this count on the **Unique Credentials Captured** tab and continuously updates the count in real-time.

For each unique credential captured, the **Unique Credentials Captured** table shows the public value, private value, realm type, source host IP, source host name, and the number of hosts from which the credential was captured.

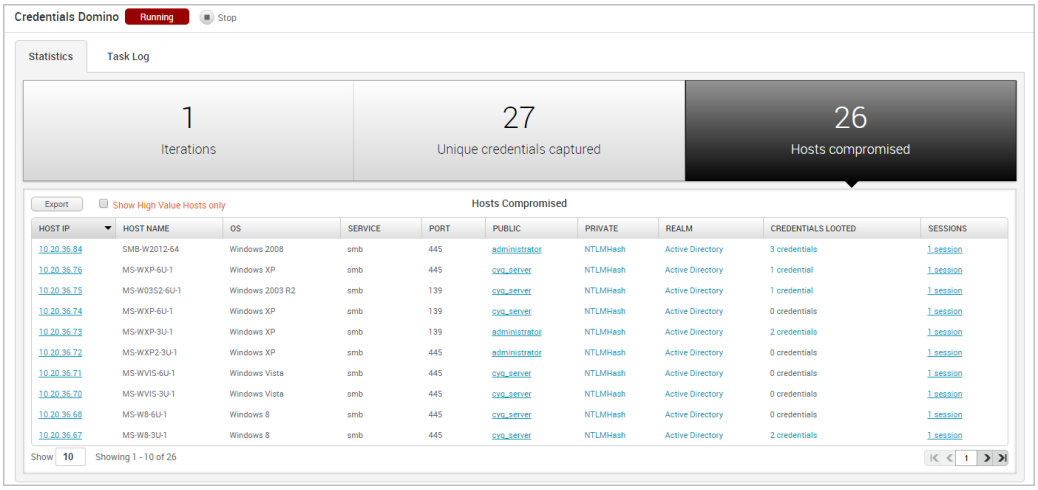


If the credential is a hash or an SSH key, the **Private** field displays the credential type instead of the private value. You can click on the private type to view the private value.



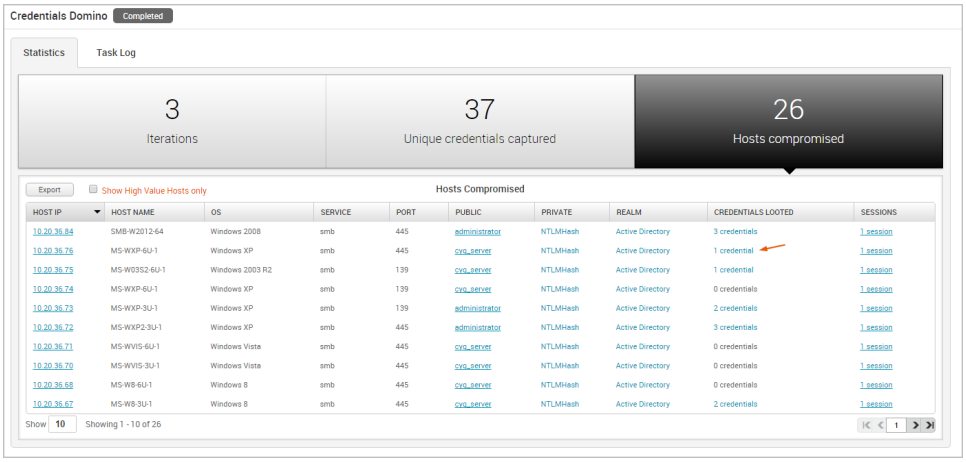
Viewing Hosts Compromised

The Credentials Domino MetaModule tracks hosts on which it was able to open sessions. It displays this count on the Hosts Compromised tab and updates the count in real-time.

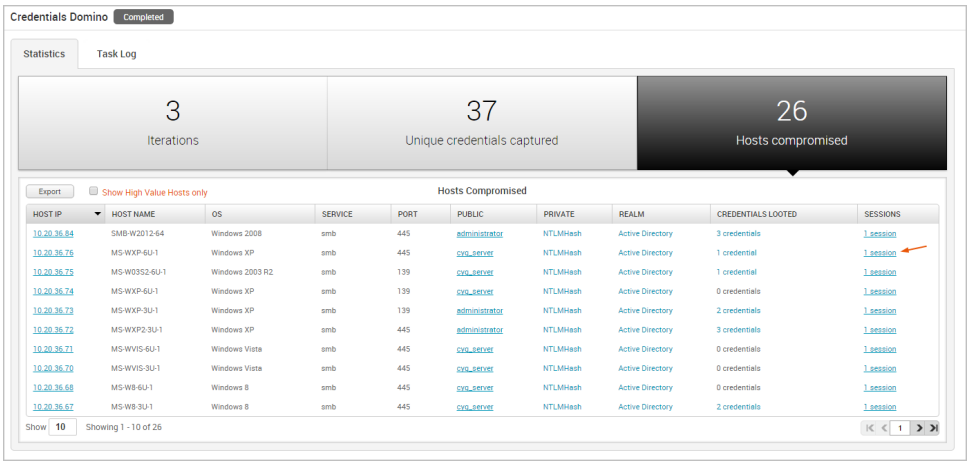


The **Hosts Compromised** table lists the host IP, host name, operating system, service name, port, public, private, realm type, number of looted credentials, and a link to the session for each compromised host.

To view a list of the credentials that were captured from the host, you can hover over the credentials count in the Credentials column.

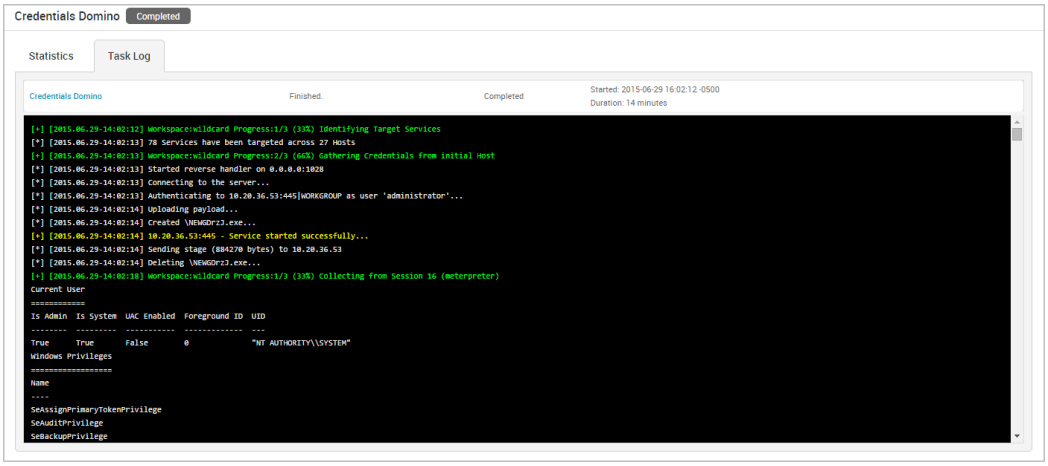


To access the details page for an open session, click on the session link in the Session column.



The Task Log Tab

The task log tracks the activity for the MetaModule, such as the host and credential pair combinations that have been attempted and the results of those attempts. The task log also documents any errors or failures that occurred during the attack and can be used to troubleshoot any issues related to the MetaModule run. This is especially helpful if, for example, the MetaModule has been running for a long time, appears to be hanging, or does not complete.



Single Credential Testing MetaModule

The Single Credential Testing MetaModule recycles a known credential pair to identify additional systems that can be authenticated. You can run this MetaModule to demonstrate how password reuse could expose major weaknesses in an enterprise's security posture. A single cracked password can enable you to easily compromise other systems that share the same password.

To use the Single Credential Testing MetaModule, you need to provide it with a known credential pair that you've uncovered through a scan, bruteforce attack, or phishing attack. When you configure this MetaModule, you need to define the target hosts and the services that you want to attempt to authenticate. After the MetaModule completes its run, it generates a report that details the hosts on which it was able to authenticate the credentials.

Lockout Risks

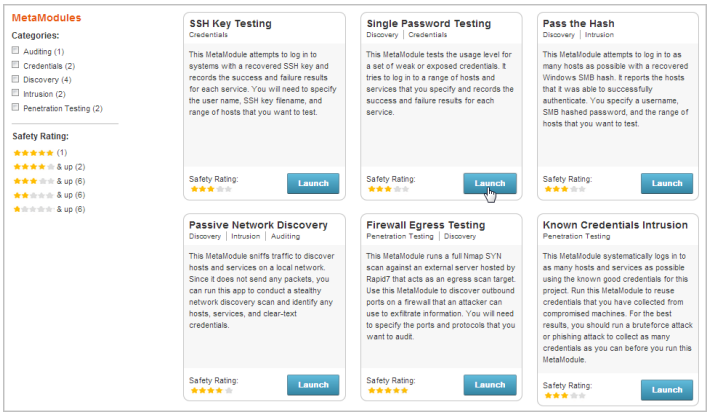
An account lockout disables an account and prevents you from accessing the account for the duration of the lockout period. When you configure the Single Credential Testing MetaModule, you should factor in the lockout risk for the services that you choose.

Each service is categorized into the following lockout risks:

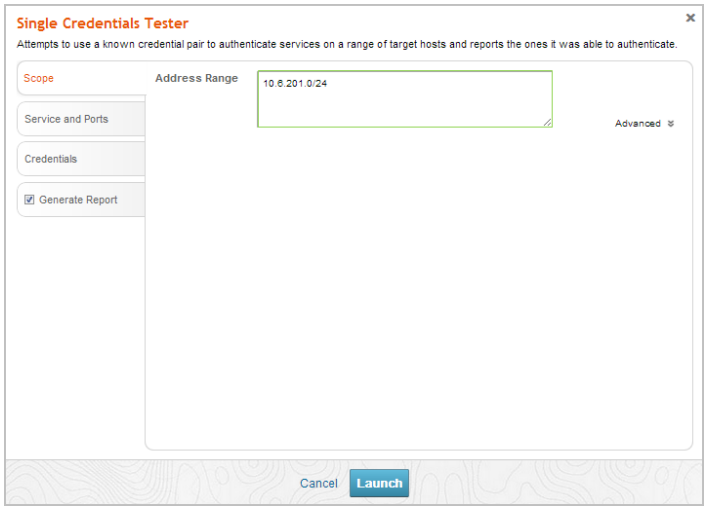
- Low Risk: Any service that typically does not enforce account lockouts, such as AFP, DB2, EXEC, FTP, HTTP, HTTPS, LOGIN, Oracle, Postgres, SHELL, SNMP, SSH_PUBKEY, Telnet, and VNC.
- Medium Risk: Any service that typically enforces account lockouts, such as MSSQL, MySQL, POP3, and SSH.
- High Risk: Any service that uses Windows authentication, such as PC Anywhere, SMB, vmauthd, and WinRM.

Running the Single Credential Testing MetaModule

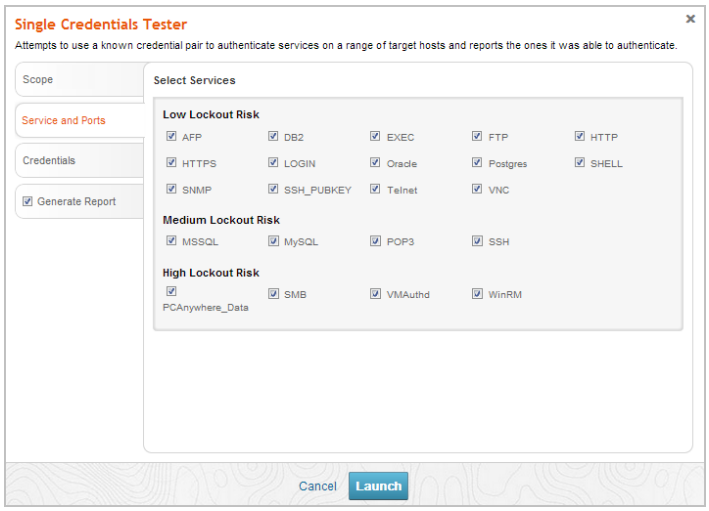
1. From within a project, select **Modules > MetaModules**.
2. Find the **Single Credential Testing** MetaModule and click the **Launch** button. The Single Credential Testing window appears.



3. From the **Scope** tab, enter the target address range you want to use for the test. The target address range must match the hosts in the workspace.



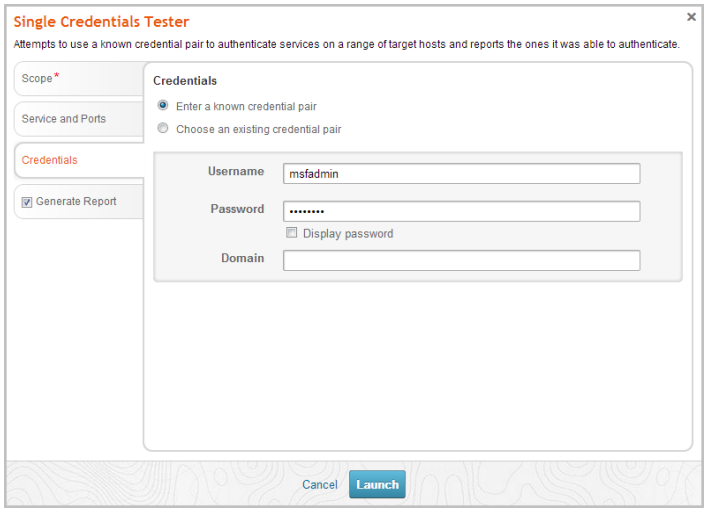
4. Click on the **Services and Ports** tab. The Services form appears.



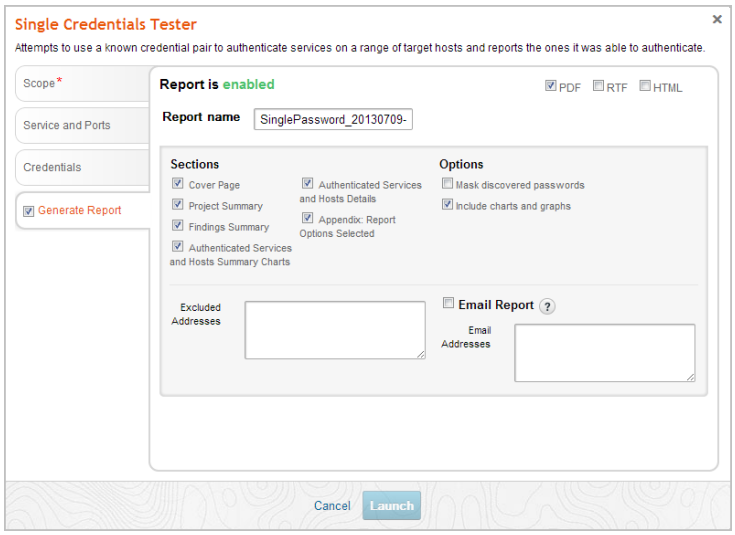
5. Select the services that you want to attempt to authenticate. All services are categorized based on their lockout risk, which is the likelihood that the service enforces account lockouts.
6. Click on the **Credentials** tab. The Credentials form appears.
7. You can choose one of the following options to supply the MetaModule with credentials:

   - Enter a known credential pair: You need to manually enter the user name and password combination that you want the MetaModule to use. Use this method for credentials obtained from phishing attacks.

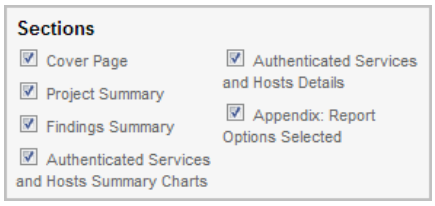
   - Choose an existing credential pair: You can select the user name and password combination from a list of known credentials. These credentials were obtained from a bruteforce attack, discovery scan, or data import.



8. Click the **Report** tab. The Report configuration form appears.



9. Enter a name for the report in the **Report Name** field, if you want to use a custom report name. Otherwise, the MetaModule uses the default report name.
10. Select PDF, Word, RTF, or HTML for the report format.
11. From the **Sections** area, deselect any sections you do not want to include in the report. Skip this step if you want to generate all the report sections.



12. Select the **Email Report** option if you want to e-mail the report after it generates. If you enable this option, you need to supply a comma separated list of e-mail addresses.

    Note: If you want to e-mail a report, you must set up a local mail server or e-mail relay service for Metasploit Pro to use. To define your mail server settings, select **Administration > Global Settings > SMTP Settings**.

13. Click the **Launch** button.

When the MetaModule launches, the Findings window appears and displays the real-time statistics and tasks log for the MetaModule run. You can track the total number of hosts that the MetaModule attempted to authenticate, the total number of login attempts, and the total number of successful logins. If you want to view all the event details, you can click on the Task Log tab.

After the MetaModule completes its run, you should go the Reports area to view the Single Credential Testing Report. The first few pages of the report show graphs and tables that provide a high-level breakdown of authenticated services and hosts. For a more detailed look at the compromised hosts, you

can look at the Authenticated Services and Hosts Details section, which shows the services that were authenticated and the sessions that were opened on each host.

SSH Key Testing MetaModule

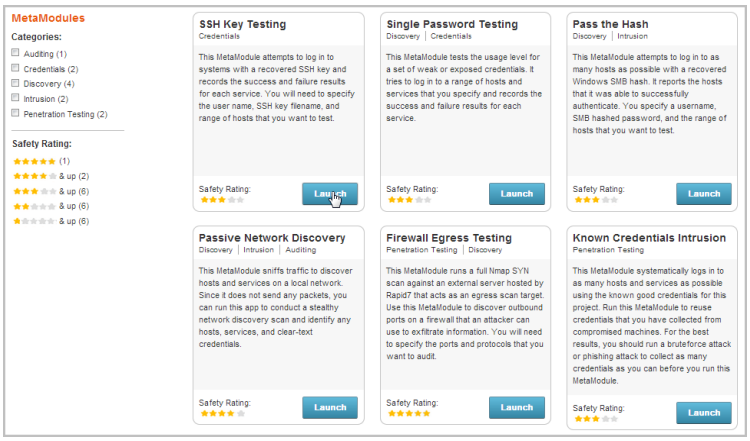
SSH public key authentication provides a secure method of logging in to a remote host. It uses an SSH key pair to authenticate a login instead of the traditional user name and password combination. The SSH key pair consists of a private and public SSH key. The private SSH key is stored on the local machine and enables you to log in to remote systems on which the corresponding public key is installed.

If you obtain an unencrypted SSH private key from a compromised target machine, you can run the SSH Key Testing MetaModule. This MetaModule enables you to bruteforce logins on a range of hosts to identify remote machines that can be authenticated with the private key. During the MetaModule run, Metasploit Pro displays real-time statistics for the number of hosts targeted, the number of login attempts made, and the number of successful logins. After the MetaModule completes its run, it generates a complete report that provides the details for the hosts it was able to successfully authenticate.

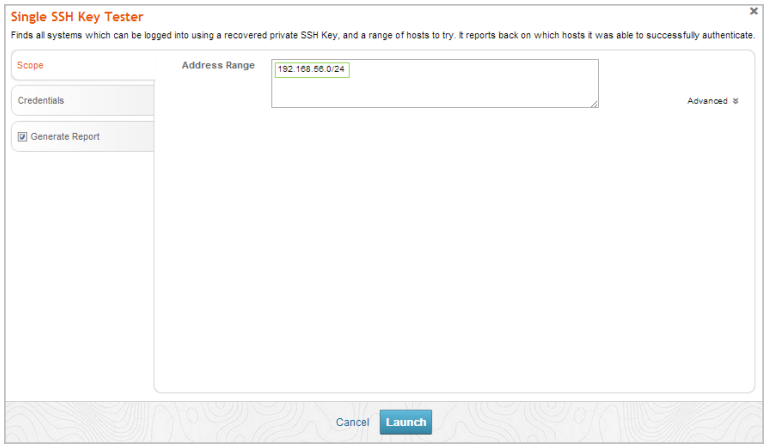
Running the SSH Key Testing MetaModule

Before you can run the SSH Key Testing MetaModule, you must either have a SSH private key available that you can upload to your project or your project must contain a looted SSH private key obtained from a scan, a bruteforce attack, or some other exploit method.

1. From within a project, select **Modules > MetaModules**.
2. Find the **SSH Key Testing** MetaModule and click the **Launch** button. The SSH Key Testing window appears.



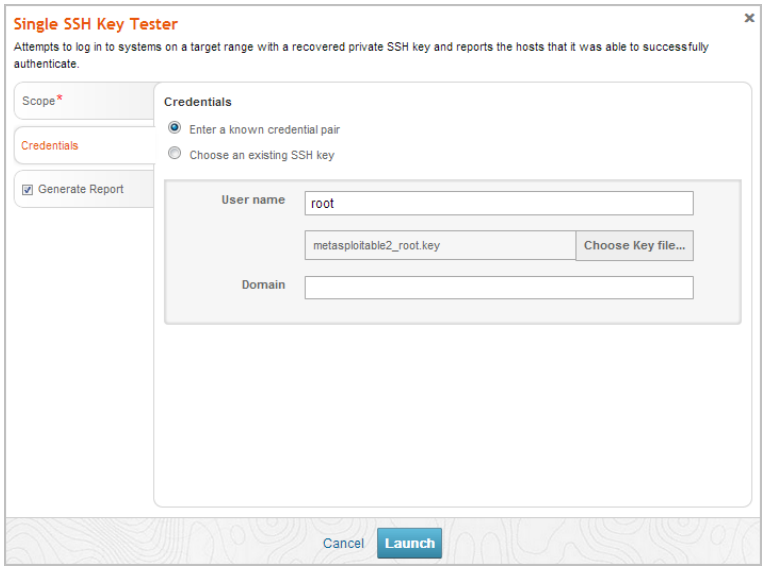
3. From the **Scope** tab, enter the target address range you want to use for the test.



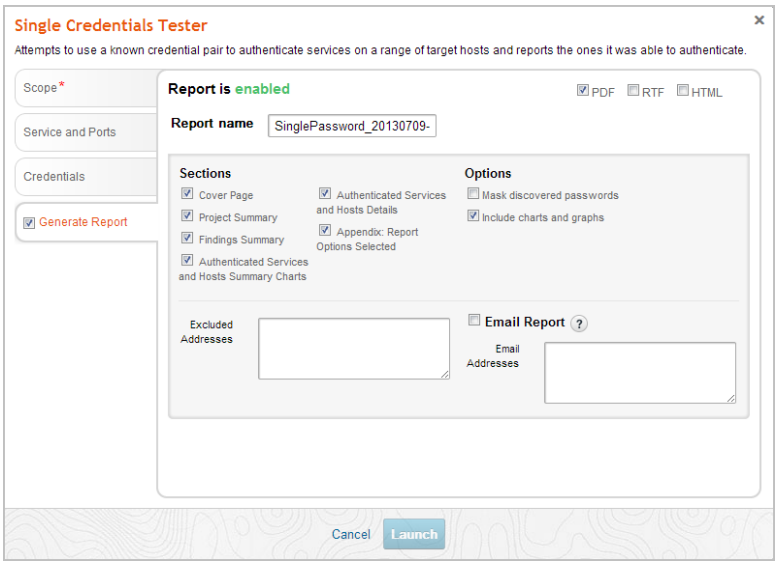
4. Click on the **Credentials** tab. The Credentials form appears.
5. Choose one of the following options to supply the MetaModule with an SSH private key:

   - **Enter a known credential pair**- You need to manually enter the user name, and then browse to the location of the private key that you want the MetaModule to use.

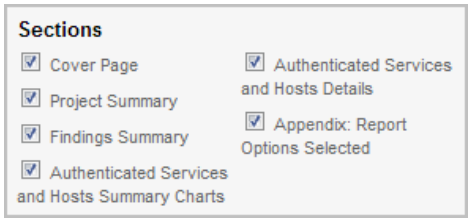
   - **Choose an existing SSH key** - You can select a user name and SSH key from a list of looted keys. These keys were obtained from a bruteforce attack, discovery scan, data import, or exploited system.



6. Click the **Report** tab. The Report configuration form appears.
7. Enter a name for the report in the **Report Name** field, if you want to use a custom report name. Otherwise, the MetaModule uses the default report name.



8. Choose whether you want to generate the report as a PDF, HTML, or RTF file.
9. From the **Sections** area, deselect any sections you do not want to include in the report. Skip this step if you want to generate all the report sections.

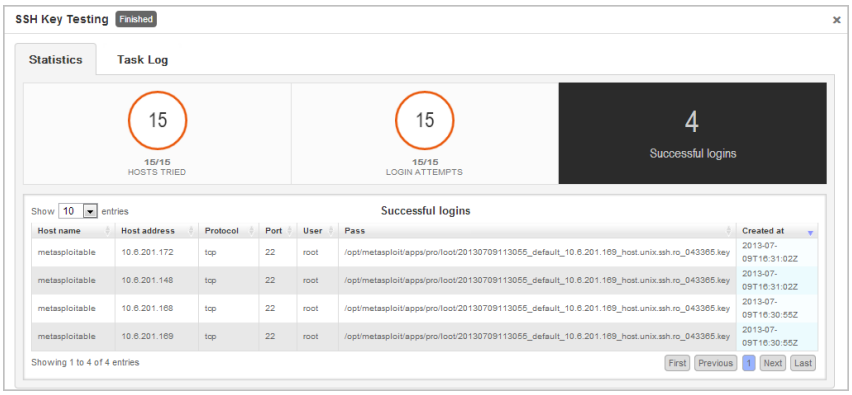


10. Select the **Email Report** option if you want to e-mail the report after it generates. If you enable this option, you need to supply a comma separated list of e-mail addresses.

    Note: If you want to e-mail a report, you must set up a local mail server or e-mail relay service for Metasploit Pro to use. To define your mail server settings, select **Administration > Global Settings > SMTP Settings**.

11. Click the **Launch** button.

When the MetaModule launches, the Findings window appears and displays the real-time statistics and tasks log for the MetaModule run. You can track the total number of hosts that the MetaModule attempted to authenticate, the total number of login attempts, and the total number of successful logins. If you want to view all the event details, you can click on the Task Log tab.



After the MetaModule completes its run, you should go the Reports area to view the SSH Key Testing Report. The first few pages of the report show graphs and tables that provide a high-level breakdown of cracked hosts and services. For a more detailed look at the hosts, you can look at the Authenticated Services and Hosts Details section, which shows the services that were authenticated and the sessions that were opened on each host.

Known Credentials Intrusion MetaModule

The Known Credentials Intrusion MetaModule logs in to a list of specified services and attempts to open sessions on a range of hosts with the known credentials in the project. You can run this MetaModule if you want to quickly get shells on the hosts in your project.

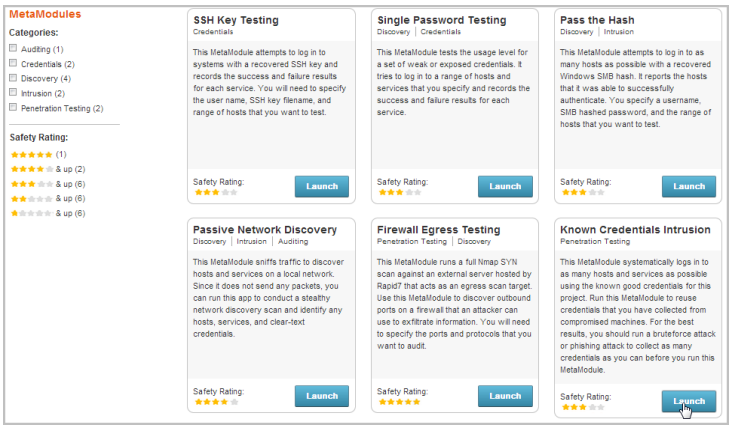
In order to run the Known Credentials Intrusion MetaModule, the project must already contain credentials that you have either collected from a Discovery Scan, bruteforce attack, or data import. The Known Credentials Intrusion MetaModule will attempt to authenticate to each service that has been enumerated for each host. If the MetaModule is able to successfully log in to the service, it attempts to open a session on the target, which you can use to do things like set up a VPN pivot, collect system data, or launch a shell to interact with the target system. It opens one session per target, and it will move onto the next host in the test if a session has already been established for a host.

During the MetaModule run, Metasploit Pro displays real-time statistics for the number of hosts targeted and the number of sessions opened. When the MetaModule completes its run, it generates a complete report that provides the details for the hosts on which it was able to open a session. You can share this report with your organization to expose weak passwords and to help mitigate vulnerabilities in its security infrastructure.

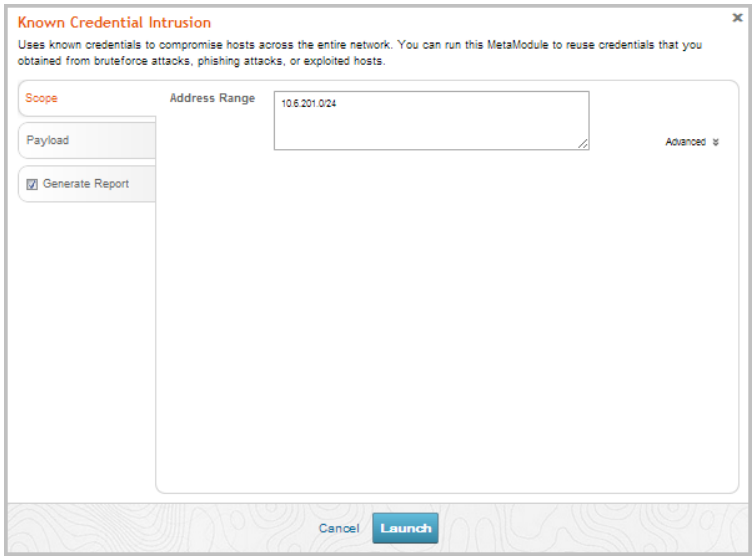
Running the Known Credentials Intrusion MetaModule

Before you can run the Known Credentials Intrusion MetaModule, you must run a Discovery Scan on the target network range or import existing host data. This populates the project with the necessary host information, such as open ports and services, that the MetaModule needs to run.

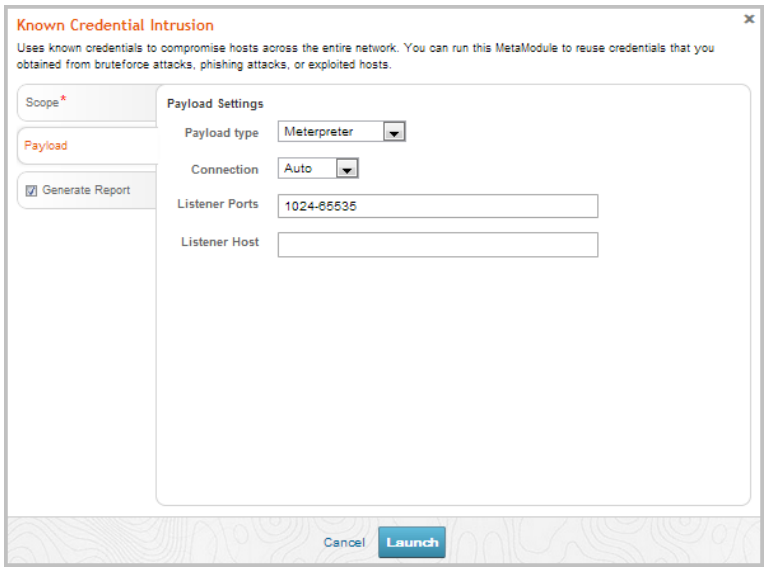
1. From within a project, select **Modules > MetaModules**.
2. Find the Known Credentials Intrusion MetaModule and click the **Launch** button. The Known Credentials Intrusion window appears.



3. From the **Scope** tab, enter the target address range you want to use for the test.



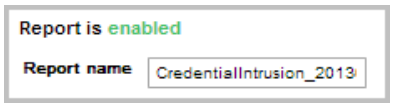
4. Click on the **Payload** tab to configure the payload settings.



5. Specify the following settings that you want to use for the payload:

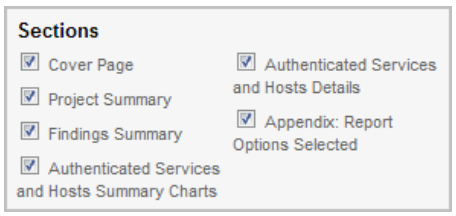
- Payload type - Choose Meterpreter for Windows or Command shell for Linux systems.
- Connection - Choose one of the following connection types:
  - **Auto** - Automatically selects the payload type. In most cases, the Auto option selects the reverse shell payload because it is more likely to establish a connection between a target machine and the attacking machine.
  - **Reverse** - Select this option if the targets are behind a firewall or use NAT. Typically, a reverse shell payload will work for most situations.
  - **Bind** - Select this option if the target devices are unable to initiate a connection.
- Listener Ports - The port that you want the listener to listen on for incoming connections. By default, ports 1024-65535 are selected; however, you can define a specific port that you want the listener to use, such as 4444.
- Listener Host - The IP address that you want the target machine to connect back to. This is typically going to be the IP address of your local machine. If you do not specify a listener host, the MetaModule automatically uses the IP address of your local machine.

6. Click the **Generate Report** tab. The Report configuration form appears.
7. Enter a name for the report in the **Report Name** field, if you want to use a custom report name. Otherwise, the MetaModule uses the default report name.

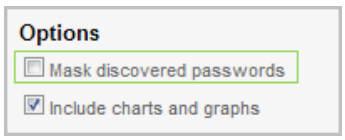


8. Choose whether you want to generate the report as a PDF, HTML, or RTF file.

9. From the **Sections** area, deselect any sections you do not want to include in the report. Skip this step if you want to generate all the report sections.



10. From the **Options** area, select the **Mask discovered passwords** option if you want to obscure any passwords that the report contains. The report replaces the password with **MASKED** . By default, this option is disabled. You should enable this option if you plan to distribute the report.



11. Select the **Email Report** option if you want to e-mail the report after it generates. If you enable this option, you need to supply a comma separated list of e-mail addresses.

    If you want to e-mail a report, you must set up a local mail server or e-mail relay service for Metasploit Pro to use. To define your mail server settings, select **Administration > Global Settings > SMTP Settings**.

12. Click the **Launch** button.

Segmentation and Firewall Testing MetaModule

When firewalls have badly configured or lax egress traffic filtering policies, they open the network up to attacks from reverse shells, data-exfiltration, and other forms of exploitation. In order to identify the open ports that allow outbound traffic and to verify that your egress filtering policies properly block traffic, you can run the Segmentation and Firewall Testing MetaModule.

The MetaModule runs an Nmap SYN scan against an egress target to reveal the outbound ports that are open from an internal host. It identifies the state of the ports in your firewall based on the traffic received by the server. If the server receives the traffic, then the MetaModule flags the port as open. If the firewall blocks the traffic, the MetaModule flags the port as filtered. The MetaModule tags the remaining ports as unfiltered or closed depending on the their response to connections. After the MetaModule completes its run, it generates a report that provides you with a comprehensive look at port state distribution and unfiltered ports.

Egress Scan Target

The egress target, egadz.metasploit.com, is a server hosted by Rapid7 and has been set up to have all 65, 535 ports open. Each port is configured to respond with a single SYN-ACK packet. In its default configuration, the MetaModule initiates a port scan using Nmap's default 1000 most common ports; however, if you need to include additional ports, you can define a custom port range.

Port States

The Segmentation and Firewall Testing MetaModule uses the following states to categorize ports.

- Open: A port is assigned an open state if it allows traffic out of the network and the EGADZ server receives it. An open state indicates that there is an application that is actively accepting TCP connections, UDP datagrams or SCTP associations.
- Filtered: A port is assigned a filtered state if it drops the traffic before it reaches the desired port on the EGADZ server. It will not receive a response from the EGADZ server. Typically, a port has a filtered state if a dedicated firewall device, router rules, or host-based firewall software has successfully blocked the port from sending traffic.
- Closed: A port is assigned a closed state if it allows traffic through the port, but there is not an application or service bound to the port. A closed port can be used to determine if t a host is up on an IP address.
- Unfiltered: A port is assigned an unfiltered traffic if it allows traffic through to the port, but it cannot be determined whether the port is open or closed.

Setting Up an Egress Testing Server

The Firewall Egress Testing MetaModule uses an external server hosted by Rapid7 to identify open outbound ports from an internal host. In some cases, you may want to set up and your own egress testing server. For example, if you want to test egress between different endpoints or if you do not want to send data to a server on the Internet, you can set up a custom egress testing server.

To help you set up a custom egress testing server, Metasploit Pro provides you with a script that you can run on an Ubuntu 12.04 LTS server. The script is downloadable from the Projects page in Metasploit Pro.

Egress Testing Server Requirements

To set up an egress testing server, you need to perform the following tasks:

1. Set up a Linux machine with your favorite distribution.
2. Add two network interfaces, or network adaptors, to the Linux machine. Each network interface should have an IP address.

   For more information on setting up a network interface on a virtual machine, please visit the documentation for your virtualization software.
3. Assign one IP address as the administrative interface. This interface will be used to control the egress testing server. It should be assigned to the eth0 interface.
4. Assign the second IP address as the egress testing server. This interface should be assigned to the eth1 interface.
5. Download and run the egress testing server script.

Set Up a Custom Egress Target

To set up a custom egress target, you will need an Ubuntu 12.04 box that is configured with two IP addresses. The two IP addresses are needed for the following interfaces:

- The admin interface: This is usually found on the eth0 interface and will be used for controlling the egress server.
- The egress server: This is usually found on eth1, or a virtual interface such as eth0:1. This is the IP address you will scan from the Firewall and Segmentation Testing MetaModule.

After you set up the box with two addresses, perform the following steps:

1. Log in to the Metasploit Pro web interface.
2. Click the Segmentation **Target Setup Script** button located under the Global Tools.

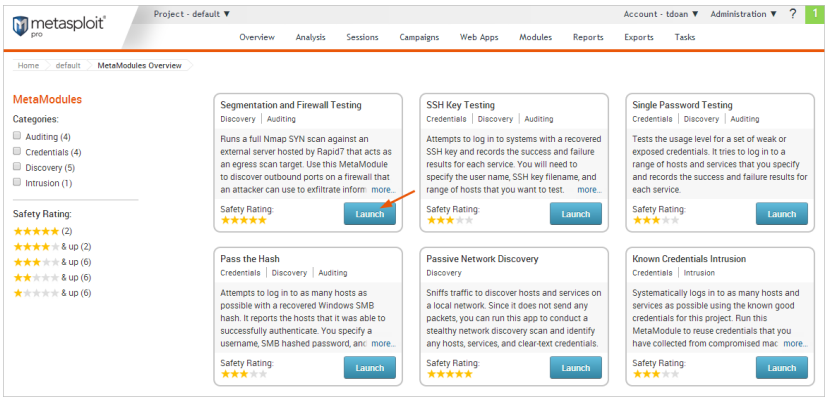
   The download process will automatically start.

   If your browser is not configured to automatically download files, a dialog window will appear and prompt you to save the file. You will need to save the file to your computer.
3. Follow the instructions provided in the `create-egadz.sh` script to set up the egress target.
4. Verify that you are able to set up an egress target using the instructions.

After you have set up the egress target, you can run the Segmentation and Firewall Testing MetaModule.

Running the Segmentation and Firewall Testing MetaModule

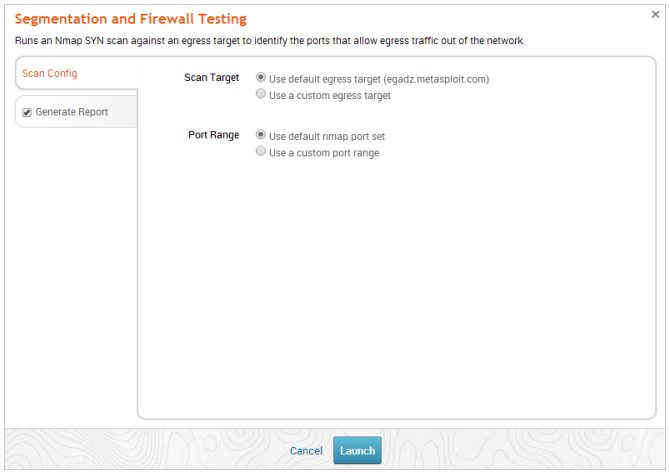
1. From within a project, select **Modules > MetaModules**.
2. Find the **Segmentation and Firewall Testing MetaModule** and click the **Launch** button.



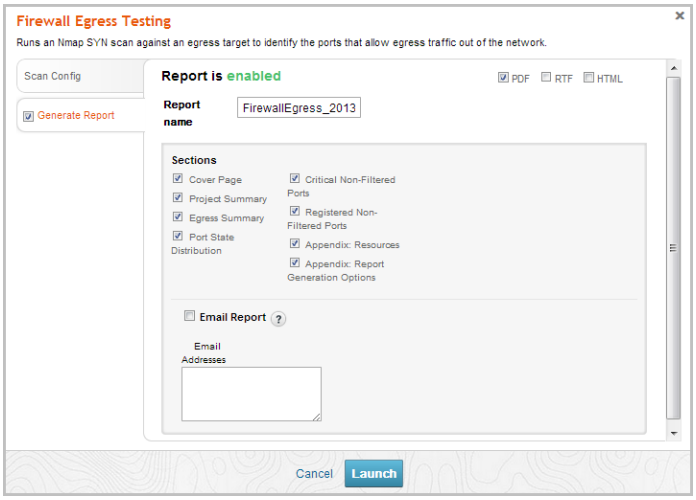
   The Segmentation and Firewall Testing configuration window appears.
3. From the Scan Config tab, choose one of the following scan target options:

   - Use default egress target - The MetaModule runs against the egress server that Metasploit has set up for testing outbound traffic.
   - Use a custom egress target - The MetaModule runs against a server that you have set up for testing outbound traffic. You can specify an IP or a fully qualified domain name. To learn how to set up a custom egress target, go to the Global Tools area located on the Projects page and download the Segmentation Target Setup Script. You can follow the instructions provided in the script to create a custom egress server.
4. From the **Scan Config** tab, choose one of the following port range options:

- Use default nmap port set: Scans Nmap's 1000 most common ports.
- Use a custom port range option: Scans the range of ports that you define.



5. Click the **Generate Report** tab.The Report configuration form appears.
6. Enter a name for the report in the **Report Name** field, if you want to use a custom report name. Otherwise, the MetaModule uses the default report name.



7. From the **Sections** area, deselect any sections you do not want to include in the report. Skip this step if you want to generate all the report sections.
8. Select the **Email Report** option if you want to e-mail the report after it generates. If you enable this option, you need to supply a comma separated list of e-mail addresses.

   If you want to e-mail a report, you must set up a local mail server or e-mail relay service for Metasploit Pro to use. To define mail server settings, select **Administration > Global Settings > SMTP Settings**.

9. Click the **Launch** button.

Exporting Data

A data export enables you to routinely back up project data and create an archive of your tests. When you export data from a project, its contents are copied and saved to a file that can be imported into other projects or shared with other instances of Metasploit Pro. All exports can be downloaded from the Exports area of the web interface or from the exports directory.

Exports Directory

When Metasploit Pro generates an export, it stores a copy of the file in `/path/to/Metasploit/apps/pro/exports`. The files that are stored in this directory will match the list of exports displayed in the web interface.

You can go to the exports directory to download or view exported data; however, you should not make any changes directly to the default exports directory. If you need to modify the export files, you should make a copy the exports directory and make your changes from the new directory. Any changes that you make directly to the export files can cause disparities between the metadata that displays for the file in the web interface and the file itself.

If you need to remove exports from a project, you should do it from within the web interface. Do not delete them directly from the exports directory.

Viewing Exports Generated with Metasploit Pro 4.8 and Earlier

All exports generated with 4.8 and earlier are stored in `/path/to/Metasploit/apps/pro/reports`. These exports were created with an older version of Metasploit Pro and were not migrated to the exports directory that was added in Metasploit Pro 4.9. These files will not be listed or accessible from the web interface.

Export Logs

The export log maintains a historical record of all export-related events. Metasploit Pro automatically updates the export log each time you export data from a project. If you experience any issues with an export, you can view the export log to find stack trace errors and troubleshoot them.

Viewing the Export Log

You can find and view the export log in the following directory: `/path/to/Metasploit/apps/pro/ui/log`. The export log is named `exports.log`.

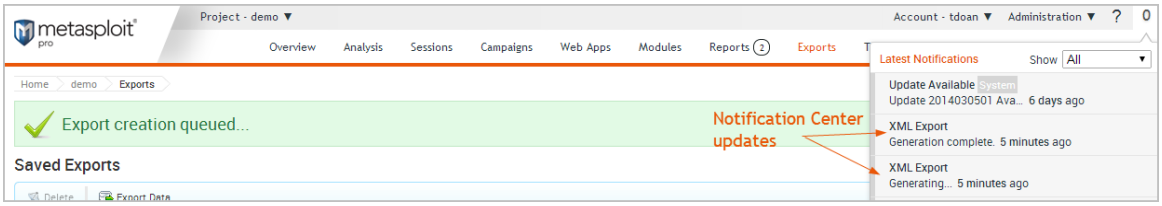
Clearing the Export Log

To clear the export log, you will need remove it from the log directory, which is located at `/path/to/Metasploit/apps/pro/ui/log`. Metasploit Pro will generate a new export log if it detects that one does not exist.

Note: Before you delete the export log, you should make a copy of it in case you need it for reference later.

Notification Center Statuses for Exports

The Notification Center alerts you when an export has started, finished, or encountered an error. The Notification Center appears as an icon in the upper-right corner of the global toolbar and turns green when there is an alert is available for you to review. You can click on the Notification Center icon to view a list of notifications for all projects.



The Notification Center displays the following statuses for exports:

- **Export started** - This status indicates that the export has started.
- **Export finished** -This status indicates that the export has completed without errors and is ready for you to download. You can click on this alert to open the Exports page, which will list all of the export files that have been generated for the project. You can sort by the creation date to find the latest export file.
- **Problem with export** - This status indicates that there was an issue with the export and it was not able to finish. You will need to view the export log to troubleshoot the issue. For more information on export logs, see *Export Logs* on page 199.

Export Types

Metasploit Pro offers the following export types:

- **XML export** - An XML file that contains the attributes for most of the objects in a project and can be imported into another project. XML exports are particularly useful if you have a data set that you want to reuse in another project or share with another instance of Metasploit Pro. For example, you can export an XML of project data if you want to reuse the scan data from a particular project.
- **Workspace ZIP** - A zip that contains an XML export and any loot files, report files, and tasks logs. This export type is useful if you want to back up the data and contents in a project or share the project with other instances of Metasploit Pro.
- **Replay script** - A batch file that reruns tasks that opened sessions on target hosts. A replay script consists of multiple resource files (.rc). Metasploit Pro creates a resource file for each session it opens. You can run a replay script from the pro console or msfconsole.
- **PWDump** - A text file that contains all of the credentials for a project, including plaintext passwords, SMB hashes, and SSH keys. Credentials can be masked to enumerate user names only.

XML Exports

When you export your project as an XML file, it contains most of the data that you see from the Analysis area of a project--with a few exceptions. The exported XML file contains most of the objects in a project's database and their attributes; it does not include any files that are associated with the objects in a project, such as task logs, generated reports, and loot files.

When you view the XML export file, you will see the following objects:

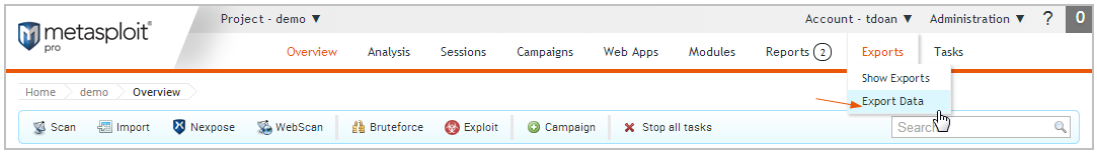
- **Hosts** - Contains the details for each host in the project, including the following attributes: notes, tags, vulnerabilities, credentials, and sessions. It also include host details, such as the host ID, IP address, MAC address, host name, OS name, OS flavor, OS service pack, and purpose.
- **Events** - Contains the event log for the project. Each event includes the workspace ID, event creation date, event name, and name of the user who launched the task.
- **Sessions** - Contains the details for each session obtained in the project, including the following attributes: host ID, session type, module used, session description, port used, and session open/close dates.
- **Services** - Contains the details for each service discovered in the project, including the service ID, host ID, port number, protocol type, state, service name, creation date, and modification date.
- **Credentials** - Contains the details for each credential stored in the project, including the credential ID, service ID, user name, password, creation date, and modification date.
- **Web sites** - Contains the details for each web server discovered, including the website ID, service ID, host address, VHOST address, HTTP port, creation date, and modification date.

- **Web pages** - Contains the details for each web page discovered, including the web page ID, HTTP response code, VHOST address, web server address, HTTP port, content type, page content, creation date, and modification date.
- **Web forms** - Contains the details for each web form discovered, including the web form ID, form path, request method, VHOST address, web server address, HTTP port, content type, page content, creation date, and modification date.
- **Web vulnerabilities** - Contains the details for each web vulnerability discovered, including the vulnerability category, vulnerability description, vulnerability confidence ranking, request method, vulnerability name, HTTP port, proof text, VHOST address, and vulnerability blame.

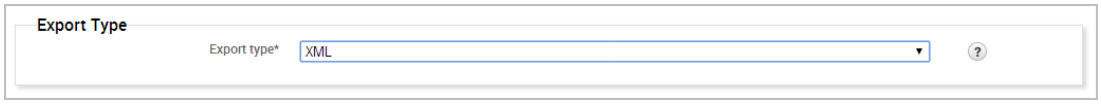
Note: Additional attributes may be available for each object; however, this list covers the most common attributes for each object.

Creating an XML Export of Project Data

1. Open the project from which you want to export data.
2. Select **Exports > Export Data** from the Project tab bar. The **Export Data** page appears.



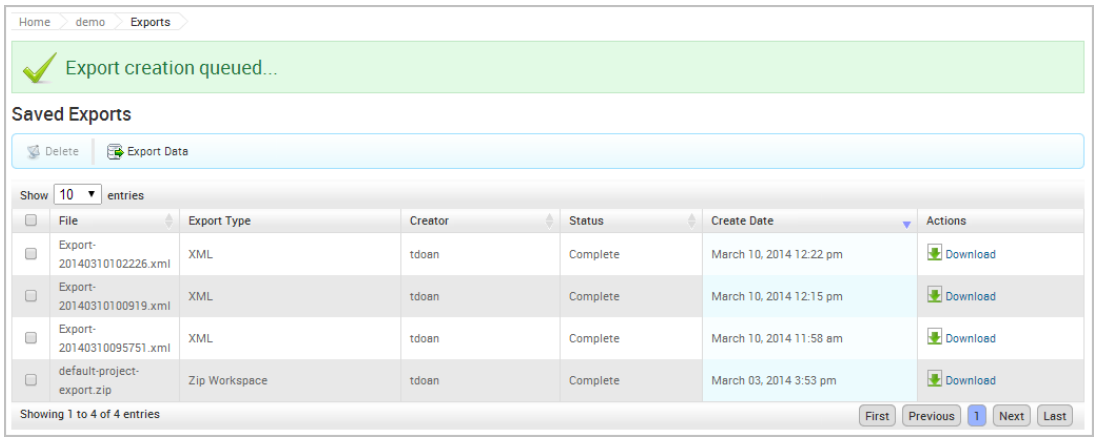
3. Select **XML Export** from the **Export Format** section.



4. Replace the export file name with a custom name, if you do not want to use the default name. (Optional)
5. Define the hosts you want to explicitly include in the **Included addresses** field. (Optional)
6. Define the hosts you want to explicitly exclude in the **Excluded addresses** field. (Optional)
7. Select the **Mask credentials** option from the **Export Options** section if you do not want to include credentials in the export.

   The credentials will be replaced with `**MASKED**` in the XML file. If you import the XML file into a project, the credentials will not be included.
8. Click the **Export Data** button.

   When the export begins, you will be taken back to the Exports page. The Exports page displays an 'Export creation queued' message.



The Notification Center icon will turn green and alert you when the export starts and completes. You can click on the Notification Center icon to view a list of system wide alerts. When the export completes, you can click on the notification message or you can select **Exports > Show Exports** from the Project tab bar to access the Exports area.

When the export is ready, it will listed be at the top of the Exports List. It will use the following naming convention: `export-[current date and time]`. If you do not see it at the top of the Exports List, click on the Create Date column name to sort the list by descending creation date.

If an error occurred during the export and the export was unable to complete, you can view the export log to identify and troubleshoot any errors that occurred. For more information on export logs, see *Export Logs* on page 199.

Workspace ZIP

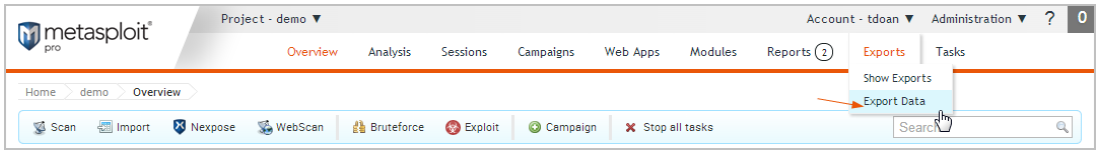
A workspace ZIP contains an XML export, which details the attributes for most of the objects in a project, and any associated directories that contain loot files, report files, and tasks logs. You can export a workspace ZIP to make a copy of a project, its data, and its files. This is useful when you want to back up your findings or when you want to import the data into other projects.

When you export a project, Metasploit Pro generates a ZIP file that contains the following:

- **Exported XML file** - Contains most of the objects in a project, including hosts, services, sessions, credentials, module details, and events.
- **Reports directory** - Contains all of the generated reports for the project.
- **Tasks directory** - Contains texts file that detail each task run.
- **Loot directory** - Contains the loot files for the project, including hashes and SSH keys.

Generating a ZIP of the Project

1. Open the project from which you want to export replay scripts.
2. Select **Exports > Export Data** from the Project tab bar. The **Export Data** page appears.

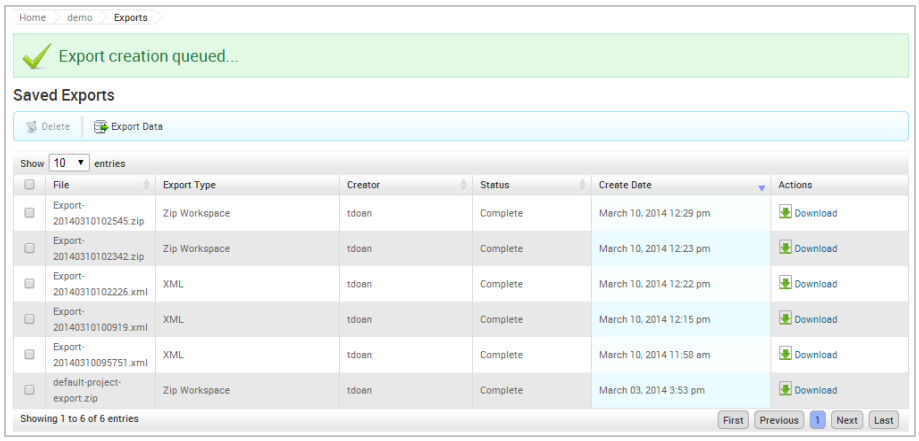


3. Choose **ZIP Workspace** from the **Export Format** section.

4. Replace the export file name with a custom name, if you do not want to use the default name. (Optional)
5. Use the **Included addresses** to explicitly define the hosts you want to include in the export. (Optional)
6. Use the **Excluded addresses** to explicitly define the hosts you want to exclude from the export. (Optional)
7. If you do not want to include credentials in the export, select the **Mask credentials** option from the **Export Options** section.
8. Click the **Export Data** button.

   When the export begins, you will be taken back to the Exports page. The Exports page displays an "Export creation queued" message.

   The Notification Center icon will turn green and alert you when the export starts and completes. You can click on the Notification Center icon to view a list of system wide alerts. When the export completes, you can click on the notification message or you can select **Exports > Show Exports** from the Project tab bar to access the Exports area.



The ZIP file will listed be at the top of the Exports List. It will use the following naming convention: `export-[current date and time]`. If you do not see it at the top of the Exports List, click on the Create Date column name to sort the list by descending creation date.

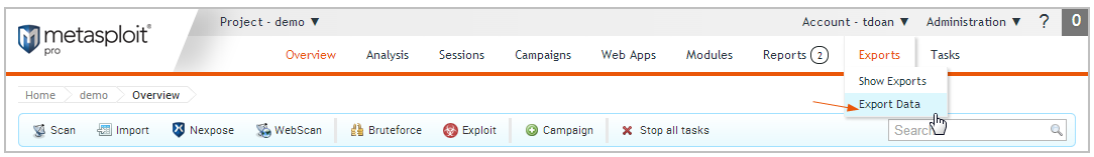
If an error occurred during the export and the export was unable to complete, you can view the export log to identify and troubleshoot any errors that occurred. For more information on export logs, see *Export Logs* on page 199.

Replay Scripts

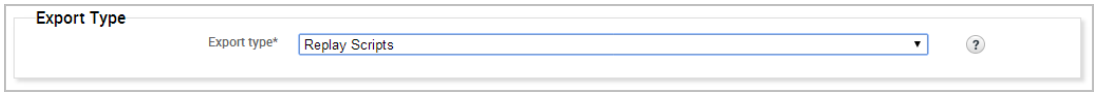
A replay script is a batch file that reruns tasks that opened sessions on target hosts. You can export a replay script to automate successful attacks through the pro console or msfconsole. When you export a replay script, Metasploit Pro creates a resource file for each opened session and compresses them into a ZIP file.

Exporting Replay Scripts

1. Open the project from which you want to export replay scripts.
2. Select **Exports > Export Data** from the Project tab bar. The **Export Data** page appears.



3. Choose **Replay Scripts** from the **Export Format** section.



4. Use the **Included addresses** to explicitly define the hosts you want to include in the replay scripts. (Optional)
5. Use the **Excluded addresses** to explicitly define the hosts you want to exclude from the replay scripts. (Optional)
6. If you do not want to include credentials in the export, select the **Mask credentials** option from the **Export Options** section.
7. Click the **Export Data** button.

   When the export begins, you will be taken back to the Exports page. The Exports page displays an "Export creation queued" message.

   The Notification Center icon will turn green and alert you when the export starts and completes. You can click on the Notification Center icon to view a list of system wide alerts. When the export completes, you can click on the notification message or you can select **Exports > Show Exports** from the Project tab bar to access the Exports area.

   The ZIP file will listed be at the top of the Exports List. It will use the following naming convention: `export-[current date and time]`. If you do not see it at the top of the Exports List, click on the Create Date column name to sort the list by descending creation date.

   If an error occurred during the export and the export was unable to complete, you can view the export log to identify and troubleshoot any errors that occurred. For more information on export logs, see *Export Logs* on page 199.

Running the Replay Script with the Pro Console or MSFConsole

To run the replay script, you need to use the `resource` command. It loads the batch files and run them through the pro console or msfconsole.The resource command needs to include the path to the replay script. For example, you can enter `resource /path/to/session_ID_IP.rc` to load the replay script and run the commands stored in the file.

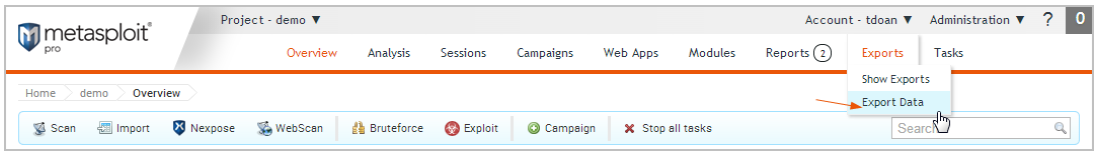
> ! Before you can run the resource file, you will need to extract them from the ZIP file.

PWDumps

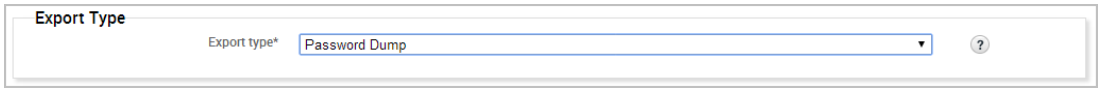
A PWDump is a text file that contains all of the credentials for a project, including plaintext passwords, SMB hashes, and SSH keys. You can export a PWDump file to perform offline password cracking with a tool like John the Ripper.

Exporting a PWDump

1. Open the project from which you want to export data.
2. Select **Exports > Export Data** from the Project tab bar. The **Export Data** page appears.



3. Select **PWDump** from the **Export Format** section.



4. Use the **Included addresses** to explicitly define the hosts you want to include in the export. (Optional)
5. Use the **Excluded addresses** to explicitly define the hosts you want to exclude from the export. (Optional)
6. Click the **Export Data** button.
7. When the export begins, you will be taken back to the Exports page.

The Notification Center icon will turn green and alert you when the export starts and completes. You can click on the Notification Center icon to view a list of system wide alerts. When the export completes, you can click on the notification message or you can select **Exports > Show Exports** from the Project tab bar to access the Exports area.

The PWDump will listed be at the top of the Exports List. It will use the following naming convention: `export-[current date and time]`. If you do not see it at the top of the Exports List, click on the Create Date column name to sort the list by descending creation date.

If an error occurred during the export and the export was unable to complete, you can view the export log to identify and troubleshoot any errors that occurred. For more information on export logs, see *Export Logs* on page 199.

Viewing Exported Data

To see a list of exported data, select **Exports > Show Exports** from the Project tab bar. The Data Exports list will display all exports associated with the project. You can click on the **Download** or **View** link to access each item.

Social Engineering

Social engineering is an attack method that typically uses a delivery tool, like e-mail, a web page, or a USB key, to induce a target to share sensitive information or perform an action that enables an attacker to compromise the system. You perform social engineering tests to gauge how well the members of an organization adhere to security policies and to identify the security vulnerabilities created by people and processes in an organization.

The data you gather from a social engineering campaign can help paint a clearer picture of the risks and vulnerabilities that exist in an organization's security infrastructure and policies. An organization can leverage the test results to strengthen their security policies, increase IT defense mechanisms and improve the effectiveness of their security training program.

In Metasploit Pro, you create and run campaigns to perform social engineering attacks. A campaign is a logical grouping of the campaign components that you need to exploit or phish a group of people. You can create a campaign using the following components:

- E-mail, web page, and portable file: The delivery mechanism for a social engineering attack.
- Template: A reusable HTML shell that contains boilerplate can be shared between campaigns in a project. You can create and use a template to quickly generate web page or e-mail content for a campaign.
- Target list - A list that defines the recipients and their e-mail addresses that will receive an e-mail.

Social Engineering Techniques

The main goal of social engineering is to entice a target to perform some illicit action that enables you to either exploit their system or to collect information from them.

Social engineering typically uses e-mail based attacks that target client-side vulnerabilities, which are exploitable through vectors that only a local user can reach. These attacks usually leverage file format exploits and client-side exploits to target the applications and information stored on a victim’s local machine or phishing scams to gather information from a human target. For example, you can attach a PDF that contains an exploit, like the Cooltype exploit, to an e-mail and send the e-mail to a group of people. When a recipient opens the infected PDF, it can create a session on their machine if it is vulnerable to the Cooltype exploit.

The method that you choose depends on the intent and purpose of the social engineering attack. For example, if you want to see how well an organization handles solicitation e-mails, you can set up a phishing attack. If you want to gauge how well an organization follows security best practices, you can

generate a standalone executable file, load it onto a USB key, and perform a USB key drop. Some of the most common social engineering methods are listed below.

Phishing

Phishing is a social engineering technique that attempts to acquire sensitive information, such as user names, passwords, and credit card information, from a human target. During a phishing attack, a human target receives a bogus e-mail disguised as an authentic e-mail from a trusted source, like a financial institution. The e-mail contains a link to open a fake web page that looks nearly identical to the official site. The style, logo, and images may appear exactly as they are on the real website. If the phishing attack is successful, the human target will fill out the web form and provide sensitive data that you can use to further compromise their system.

To set up a phishing attack in Metasploit Pro, you need to create a campaign that contains the following components:

- E-mail component: Defines the content that you want to send in the e-mail body, and the human targets that you want to receive the phishing attack. Each campaign can only contain one e-mail component.
- Web page component: Defines the web page path, the HTML content, and the redirect URL. The web page that you create must contain a form that a human target can use to submit information.

Client-Side Exploits

A client-side exploit attacks vulnerabilities in client software, such as web browsers, e-mail applications, and media players. In a client-side exploit, the victim must visit a malicious site in order for the exploit to run. A client-side exploit is different from a traditional exploit because it requires the victim to initiate the connection between their machine and an attacking machine. Traditional exploits, on the other hand, do not require human interaction.

When a human target visits the web page that contains the exploit, a session opens on the target's machine and gives you shell access to the target's system if the target's system is vulnerable to the exploit. Using the session, you can do things like capture screenshots, collect password files, and pivot to other areas of the network.

To set up a file format or client-side exploit in Metasploit Pro, you need to create a campaign that contains the following components:

- E-mail component: Defines the content that you want to send in the e-mail body and the human targets that you want to receive the e-mail. You can provide a link to the web page that serves the exploit.
- Web page component (optional): Sets the web page component to send a client-side exploit and defines the tracking URL, and the HTML content for the web page.

File Format Exploits

File format exploits are attacks that take advantage of a vulnerability in the way that an application processes data in a particular kind of file format, such as PDF, DOC, or JPEG. A file format exploit can run when a human target opens a attachment that contains the exploit. For example, you can attach a malicious Word document that contains an exploit, like MS11-006, to an e-mail. When the human target downloads and views the attachment (in thumbnail view), a session opens on the target's machine and gives you a shell to access their system.

To set up an e-mail attachment attack in Metasploit Pro, you need to create a campaign that contains the following components:

- E-mail component : Attaches a file format exploit to the e-mail and defines the content that you want to send in the e-mail body, and the human targets that you want to receive the e-mail.
- Portable file component: Generates a file format exploit that you can store on a USB key.

Java Signed Applets

The Java Signed Applet Social Engineering Code Execution module creates a jar file and signs it. You deliver the Java signed applet to a human target from a web page that contains an applet tag. When a human target visits the web page, the target's Java Virtual Machine asks the human target if they trust the signed applet. If the human target runs the applet, it creates a session on the victim's machine and gives you full user permissions to their system.

Portable Files

A portable file can be used for a USB drive drop. A portable file can be a generated executable file or a file format exploit that you load onto a USB key. When a human target installs the USB drive and opens the file, a connection is created from the target's machine to the attacking machine.

To create a portable file in Metasploit Pro, you need to create a campaign that contains the following component:

- Portable file component - Generates an executable or file format exploit that you can store on a USB key.

Social Engineering Terminology

Before you start building campaigns, you should familiarize yourself with the following social engineering terms.

Campaign

A campaign is a logical grouping of components that you need to perform a social engineering attack. A campaign can contain only contain one e-mail component, but can have multiple web pages or portable files.

Click Tracking

Click tracking is a method of client-side testing that tracks the number of human targets that click on a link. The web page tracks the number of visits and helps an organization identify how susceptible members of their organization are susceptible to social engineering attacks.

E-mail Template

An e-mail template contains predefined HTML content that you can insert into an e-mail.

Executable

An executable file that automatically runs when a human target opens the file. The executable runs a payload that creates a connection from the exploited machine back to the attacking machine.

File Format Exploit

A file format exploit targets a vulnerability in a specific application, such as Microsoft Word or Adobe PDF.

Human Target

A human target is the person who receives the social engineering attack or is part of a campaign.

Phishing Attack

A phishing attack is a form of social engineering that attempts to acquire sensitive information, such as user names, passwords, and credit card information, from a human target. During a phishing attack, a human target receives a bogus e-mail disguised as an authentic e-mail from a trusted source, like the bank. Generally, the e-mail contains a link that opens a fake web page that looks nearly identical to the official site. The style, logo, and other images may appear exactly as they are on the real website.

Portable File

A generated executable file that you can attach to an e-mail or save to a USB key. When the victim opens the file, the executable runs the payload, starts a session on the victim's machine, and connects back to

your machine.

Resource File

A resource file refers to a web page template, e-mail template, or target list. It is a reusable file that you can use in a campaign. Each project has its own set of resource files. The resource files are not shareable between projects.

Target List

A target list defines the targets that you want to include in the social engineering campaign. You use the target list to specify the recipients that you want to e-mail the social engineering attack.

Tracking GIF

A tracking GIF sets a browser cookie when a human target opens an e-mail.

Tracking Link

A tracking link consists of a URL path to a web page and a tracking string. When a target clicks on the URL, the system sets a cookie to track the visit and any subsequent visits.

Tracking String

A tracking string is a 64 bit string that encodes the target and e-mail IDs. Campaigns use tracking strings to monitor the activity of a target.

Visit

A visit occurs when a target clicks on a link and opens the web page.

Web Template

An web template contains predefined HTML content that you can insert into a web page.

Managing Campaigns

In Metasploit Pro, you create and run campaigns to perform social engineering attacks. A campaign contains the e-mails, web pages, and portable files that are necessary to run a social engineering attack against a group of targets. You can set up campaigns to perform phishing attacks, launch client-side exploits, run Java signed applets, generate executables for USB key drops, and send out e-mails with malicious attachments.

The campaign tracks the number of human targets that fall victim to the attack and presents the results in a social engineering report. You can read the report to review the metrics for the campaign, learn about remediation recommendations, and determine the effectiveness of the campaign.Additionally, the campaign page shows real-time statistics that provide you with a high-level overview of the campaign results. For example, you can view the number of recipients who opened the e-mail or filled out the web form in a phishing campaign.

A campaign is a logical grouping of the campaign components that you need to exploit or phish a group of people. A campaign can be comprised of the following campaign components: e-mail, web page, or portable file. The components that you add to the campaign depend on the purpose and goal of the social engineering attack.

Campaign Restrictions

The following restrictions apply to campaigns:

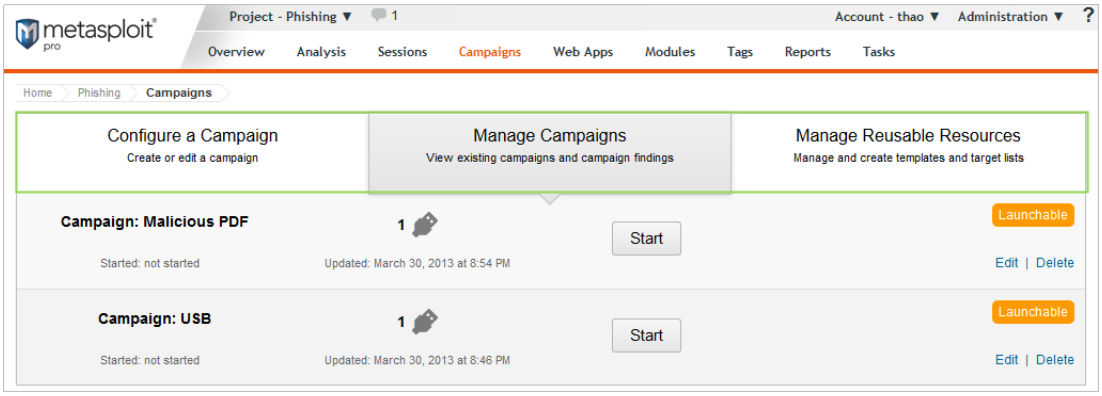
- A campaign can only contain one e-mail.
- A campaign that you build with the canned phishing campaign can only contain one e-mail and up to two web pages. One web page is used for the landing page, and the other web page is used for the redirect page. If you need additional redirect pages, do not use the canned phishing campaign to create a campaign, use the custom campaign builder instead.
- Each instance of Metasploit Pro can only run one campaign at a time.

Campaign Dashboard

The Campaign Dashboard contains the interfaces and tools that you need to set up social engineering campaigns. It provides you with access to the campaigns, target lists, and resource files that are in a project. The Campaign Dashboard is made up of the campaign tasks bar, modal windows, campaign widgets, and action links.

Campaign Tasks Bar

When you access the Campaign Dashboard, you will see the Campaign Tasks bar below the main Tasks bar. Each tab in the Campaign Tasks bar represents a major section of functionality within social engineering. Click on the tabs to switch to between the campaign configuration, campaign management, and campaign elements areas.

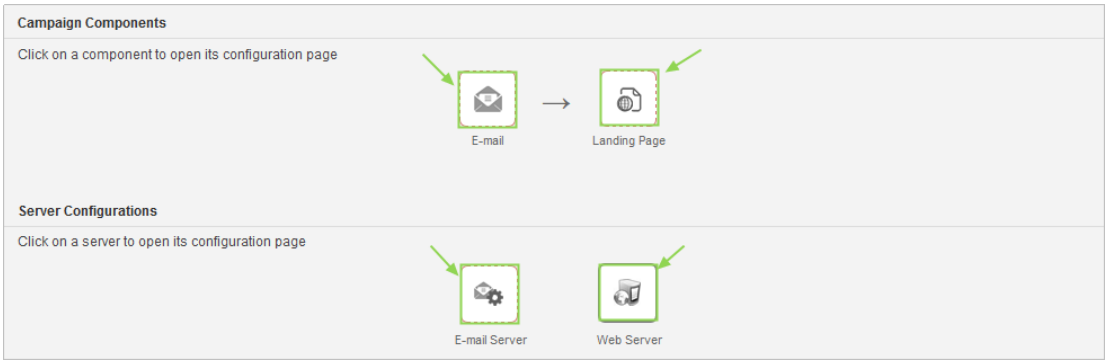


The Campaign Tasks bar contains the following tabs:

- **Configure a Campaign** - Displays the campaign editor. Use the campaign editor to create new campaigns and edit existing campaigns.
- **Manage Campaigns** - Shows a list of campaigns that are currently in the project. Next to each campaign listing is a set of action links. Use these action links to edit, delete, reset, preview, and start/stop a campaign.
- **Manage Reusable Resources**- Provides a management interface for reusable campaign resources, such as e-mail templates, web page templates, target lists, and malicious files.

Campaign Widgets

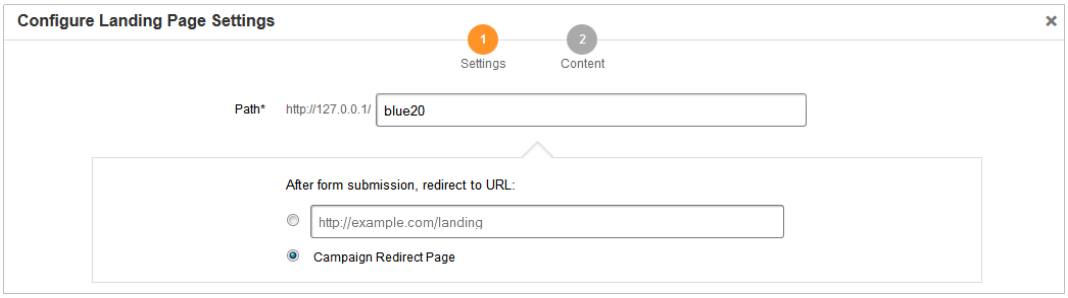
A campaign widget is an icon that represents a campaign component. When you click on the campaign widget, it opens a modal window that displays the configuration form for that campaign component.



Modal Windows

A modal window is a small pop-up window that requires you to interact with it before you can go back to the main window. Typically, modal windows are used to display alerts and confirmation windows. In Metasploit Pro, modal windows guide you through the process of setting up campaign components.

To exit a modal window, you must either complete the required form data, or you can click the 'X' to exit the screen.



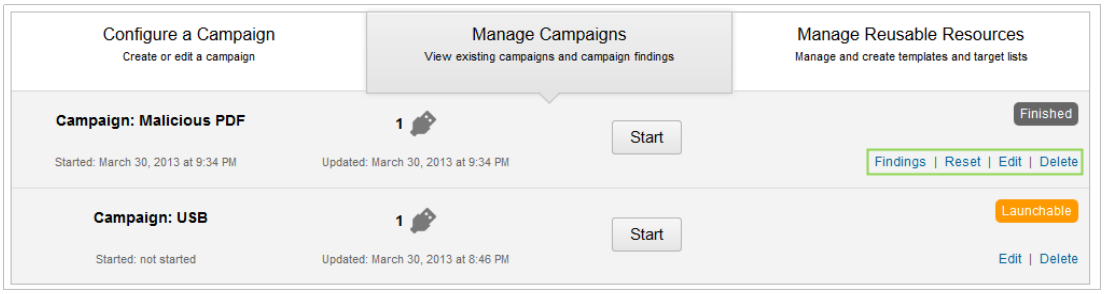
Action Links

An action link is an interactive link that you can click on to perform a specific task. Each campaign has a set of action links that are available for you to use.

The following action links are available to each campaign:

- **Start** - Launch the campaign.
- **Stop** - Stop the campaign.
- **Preview** - Generate a preview of an e-mail and web page.
- **Reset** - Reset the statistics and data in a campaign.
- **Edit** - Edit the current configuration for campaign components.
- **Delete** - Remove the campaign and its data from the project.

The following image shows the action links that are available for a campaign:



Campaign States

The state describes the current status of a campaign. At any given point in time, a campaign can be in one of the following states:

- Unconfigured - The campaign does not contain any components or contains components that have not been configured.
- Preparing - The campaign is getting ready to run.
- Launchable - The campaign is ready to be launched.
- Running- The campaign is online.

  For campaigns that have a web page, this means that the web page is online and accessible to target machines that can reach the Metasploit instance.

  For campaigns that contain an e-mail, this means that Metasploit Pro has attempted to send the e-mail to the target list through your mail server.

  For campaigns that contain portable files, this means that handler is ready and waiting for incoming connections from target machines.

- Finished- The campaign is no longer active.

  For campaigns that have a web page, this means that the web page is no longer accessible and cannot be viewed by anyone.

  For campaigns that contain portable files, this means that the handler is no longer listening for incoming connections.

Creating a Campaign

1. From within a project, select **Campaigns** from the Tasks menu.
2. When the Manage Campaigns area appears, click the **Configure a Campaign** tab.
3. When the Configure a Campaign area appears, enter a name for the campaign in the **Name** field.
4. Choose one of the following setup options:

- **Phishing Campaign** - Metasploit Pro automatically creates a campaign that has the necessary campaign components for a phishing attack. The phishing campaign contains an e-mail component and two web page components that you configure to set up the landing page and the redirect page.

- **Custom Campaign** - You manually create the campaign and add the campaign components that you need to it. For example, if you need to generate a portable file or generate a file format exploit.

Now you're ready to customize the campaign. If the campaign is empty, you will need to add a component to it. For example, if you want to generate an executable to save to a USB key, you can add a portable file component.

Editing the Campaign Name

1. From within a project, select **Campaigns** from the Tasks menu.
2. When the **Manage Campaigns** area appears, find the campaign that you want to edit.
3. Click the **Edit** link.
4. When the campaign configuration page appears, delete the existing campaign name from the **Name** field.
5. Enter the new campaign name in the **Name** field.
6. Click the **Save** button.

Running a Campaign

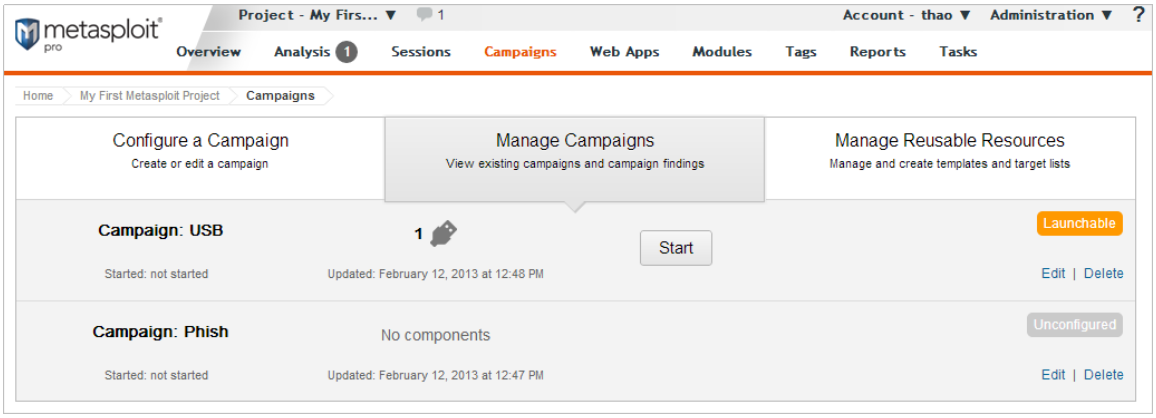
1. From within a project, click the **Campaigns** tab.
2. When the Manage Campaigns area appears, find the campaign that you want to run. The campaign status must be launchable for the campaign to run. A launchable status indicates that all necessary components of the campaign are configured.
3. Click the **Start** link.

Clearing the Data from a Campaign

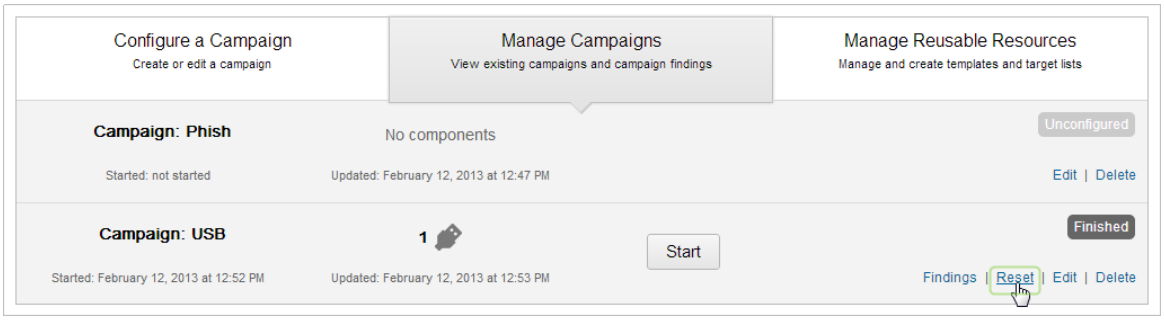
When you reset the campaign, you clear all the statistics and data collected by the campaign. A campaign reset removes any data collected through form submissions, the statistics for a phishing attack, and the statistics for e-mail tracking.

1. From within a project, select **Campaigns** from the Tasks menu.

2. When the Manage Campaigns area appears, find the campaign that you want to reset.



3. Click the **Reset** link.



4. When the confirmation window appears, click **OK** to confirm that you want to reset the data in the campaign.

Viewing the Findings for a Campaign

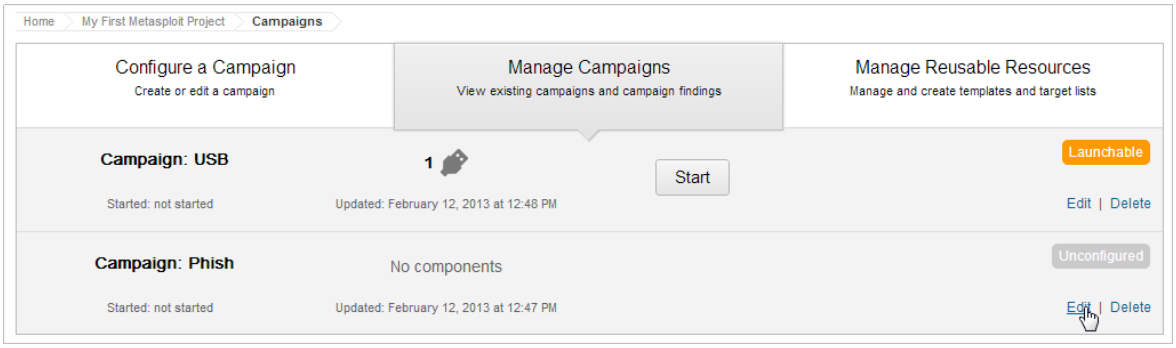
1. From within a project, click the **Campaigns** tab.
2. When the **Manage Campaigns** area appears, find the campaign whose results you want to view.
3. Click the **Findings** link. The **Findings** window appears and displays the statistics for the entire campaign. You will see the total number human targets that received an e-mail, opened the e-mail, visited the phishing web page, and submitted the web page form.
4. Click on a stat bubble to view the findings for that a list of human targets associated with that statistic.

   For example, if you view the findings for the recipients who filled out the web form, you will see the name and e-mail of the human target that submitted the web form. If you click on their e-mail address, you will see the data that they submitted.

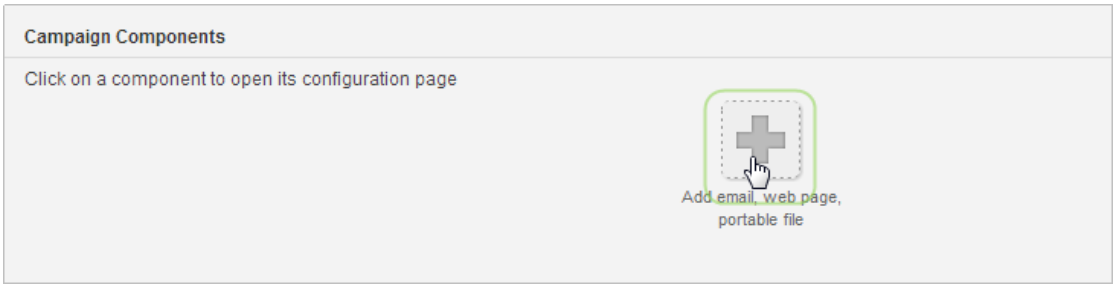
5. Click the **Done** button to close the **Findings** window.

Adding a Campaign Component

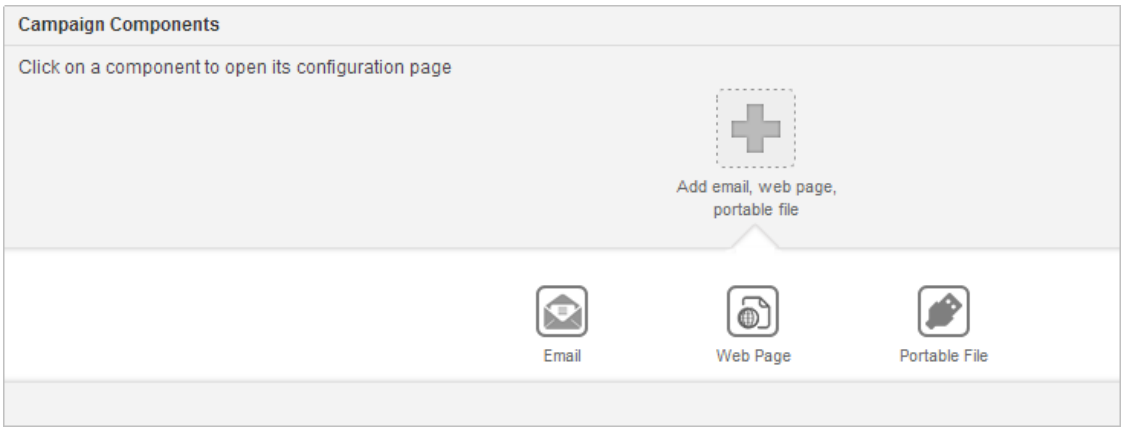
1. From within a project, click the **Campaigns** tab.
2. When the Manage Campaigns area appears, find the campaign that you want to edit and click the **Edit** link.



3. When the campaign configuration page appears, click the **Add e-mail, web page, or portable file** button. You can only add components to a campaign that uses the custom setup. You cannot add components to a campaign that you created with the canned phishing campaign.



4. Click on the campaign component that you want to add. After you add the component, the configuration page for the component appears. Follow the onscreen instructions to configure the component.



Removing a Campaign Component

1. From within a project, click the **Campaigns** tab.
2. When the Manage Campaigns area appears, find the campaign that you want to edit and click the **Edit** link.
3. When the campaign configuration page appears, click the **Edit** button located under Campaign Components. The component icons show red X's that you can use to remove a component from the campaign.
4. Click the 'X' button for the component that you want to remove.
5. Click the **Done** button when you finish.

Stopping a Campaign

1. From within a project, click the **Campaigns** tab.
2. When the Manage Campaigns area appears, find the campaign that you want to stop.
3. Click the **Stop** link.

Sending an E-mail Notification when a Campaign Starts

Before you configure an e-mail notification, you should verify that the SMTP settings for your mail server have been configured for Metasploit Pro. Go to **Administration > Global Settings** to view your SMTP settings.

1. From the campaign configuration form, locate the Notifications area.
2. Select the **Notify others before launching the campaign** option.
3. When the Notification Settings window appears, enter the e-mail addresses of the people who you want to send the alert in the To field. To include multiple e-mail addresses, use a comma separated list of e-mail addresses. For example, you can enter a list like the following: joe@rapid7.com, mary@rapid7.com, jon@rapid7.com.
4. In the **Subject** field, enter the subject that you want the e-mail to display. By default, Metasploit Pro auto-fills the subject for you with a canned subject line.
5. In the **Message** field, enter the information, or body, that you want to send in the e-mail. For example, you may want to say something like, "This is a company wide alert to inform you that we are starting our security awareness program. If you have any questions, please contact John Smith."
6. When you are done creating the notification e-mail, click the **Save** button.

Uploading a Malicious File

1. From within a project, click the **Campaigns** tab.
2. Click the **Manage Reusable Resources** tab.
3. From the **Resource** dropdown, select **Malicious Files**.
4. Click the **New Malicious File** button.
5. In the **File name** field, enter the name of the file that you are importing. The file name must include the file extension. For example, if you are uploading an executable file, the file name should include the exe extension.
6. Click the **Browse** button to navigate to the location of the file that you want to upload. Once you have found and selected the file, click the **Open** button. The path to the file will appear in the Attachment field.
7. Click the **Save** button.

Deleting a Campaign

1. From within a project, click the **Campaigns** tab.
2. When the **Manage Campaigns** area appears, find the campaign that you want to delete.
3. Click the **Delete** button.
4. When the confirmation window appears, click **OK** to confirm that you want to permanently delete the campaign. All target lists and campaign components will be deleted from the project. You will no longer be able to view, run, or edit the campaign.

Exporting a CSV File of Campaign Findings

1. From within a project, click the **Campaigns** tab.
2. When the Manage Campaigns area appears, find the campaign that contains the data that you want to export.
3. Click the **Findings** link.
4. Click on the stat bubble that represents the data that you want to export. For example, if you want to export the list of human targets that opened the e-mail, click on the **n% recipients opened the e-mail** stat bubble. A list of human targets and the Export Data button appears.
5. Click the **Export Data** button.
6. When the Open window appears, choose the **Save File** option and click **OK**. The file saves to the Downloads folder on your system.

Exporting a CSV File of E-mail Sent from a Campaign

1. From within a project, click the **Campaigns** tab.
2. When the Manage Campaigns area appears, find the campaign that contains the data that you want to export.
3. Click the **Findings** link.
4. Click on the **#n e-mails were sent** stat bubble. A list of human targets and the Export Data button appears.
5. Click the **Export Data** button.
6. When the Open window appears, choose the **Save File** option and click **OK**. The file saves to the Downloads folder on your system.

Exporting a CSV File of Human Targets that Opened the E-mail

1. From within a project, click the **Campaigns** tab.
2. When the Manage Campaigns area appears, find the campaign that contains the data that you want to export.
3. Click the **Findings** link.
4. Click on the **%n of recipients opened the e-mail** stat bubble. A list of human targets and the Export Data button appears.
5. Click the **Export Data** button.
6. When the Open window appears, choose the **Save File** option and click **OK**. The file saves to the Downloads folder on your system.

Exporting a CSV File of Human Targets that Clicked on the Link

1. From within a project, click the **Campaigns** tab.
2. When the Manage Campaigns area appears, find the campaign that contains the data that you want to export.
3. Click the **Findings** link.
4. Click on the **%n of openers clicked on link**stat bubble. A list of human targets and the Export Data button appears.
5. Click the **Export Data** button.
6. When the Open window appears, choose the **Save File** option and click **OK**. The file saves to the Downloads folder on your system.

Exporting a CSV File of Human Targets that Submitted the Form

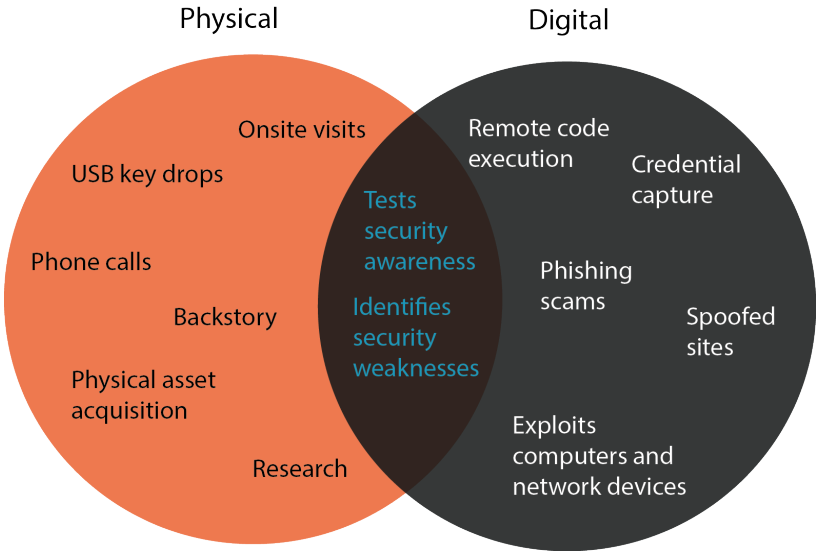
1. From within a project, click the **Campaigns** tab.
2. When the Manage Campaigns area appears, find the campaign that contains the data that you want to export.
3. Click the **Findings** link.
4. Click on the **%n of openers submitted the form** stat bubble. A list of human targets and the Export Data button appears.
5. Click the **Export Data** button.
6. When the Open window appears, choose the **Save File** option and click **OK**. The file saves to the Downloads folder on your system.

Best Practices for Social Engineering Attacks

Social engineering is an attack method that induces a person to unknowingly divulge confidential data or to perform an action that enables you to compromise their system. Typically, social engineering attacks utilize delivery-based methods, such as e-mail and USB keys, but they can also use other mechanisms, such as phone calls and onsite visits. Social engineering attacks are becoming more prevalent in the existing security landscape and are forcing many organizations to take a closer look at one of their most vulnerable targets: their employees.

As part of a penetration testing engagement or a security awareness program, you may be asked to perform social engineering tests to audit the organization's physical and IT security infrastructure. Before you can execute any type of social engineering test, you should sit down with the organization to clearly define the objectives of the engagement and to explicitly identify the goals that they wish to achieve. Most organizations will want to measure the effectiveness of their security training program or identify the weaknesses in their existing security policies and IT defense mechanisms. Once you have a clear understanding of the purpose of the assessment, you can build an attack plan that addresses all the areas of concern.

Generally, there are two distinct forms of social engineering penetration tests: digital and physical tests. A digital social engineering test focuses more on IT security and policy compliance whereas a physical social engineering test deals more with human behavior and tangible assets, like office spaces and company equipment. Depending on the goals of the engagement, you may utilize only one style of testing or you may incorporate both types.



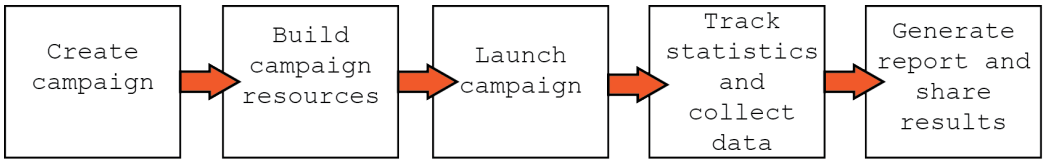
For example, if the organization wants to identify the metrics for employee security policy compliance, you may need to build a long-term plan that establishes an initial baseline before any social engineering attacks

even take place. Once you have determined the baseline, you can implement social engineering attacks, like USB key drops and phishing scams, that test both the physical security perimeter as well as the protection of digital data.

Social Engineering with Metasploit Pro

Metasploit Pro's social engineering feature mainly focuses on computer-based attacks. Most computer-based social engineering attacks utilize a delivery mechanism, like e-mail, to send links to a spoofed website or attachments that contain a malicious file. With Metasploit Pro, you can create and distribute the necessary e-mails and files that are typically associated with digital attacks.

In Metasploit Pro, a social engineering penetration test is performed through a campaign. A campaign is the workspace that you use to manage and execute all social engineering related tasks. Additionally, a campaign tracks test findings and stores the resource files that you need to create social engineering attacks, such as web page templates, e-mail templates, malicious executables, and target lists.



To understand how social engineering works with Metasploit Pro, let's go over the most common types of social engineering attacks and the processes that you will use to implement them. Along the way, we will provide you with some best practice tips that will help you set up effective and useful social engineering tests.

Phishing

If you look in your SPAM folder, you will undoubtedly find phishing e-mails that have been perfectly crafted to look like they are from your bank, your friends in Nigeria, or pretty much anyone with whom you would share your most confidential information. These e-mails may look nearly identical to the real e-mails, or they may be terrible recreations of the original. Regardless, their purpose is to trick the reader into believing in their authenticity. The e-mail may contain header information, like the sender's e-mail address, that looks absolutely legitimate. The e-mail may also contain headers, footers, and logos that are near identical matches to the real ones.

```
Hey John,

Your password is about to expire.
Please visit Netsuite to update your
account. Use this link to access your
account directly.

Thanks,
IT
```

RAPID7

Ultimately, the goal is to get the reader to click on a link provided in the e-mail. The link directs them to a spoofed site that is set up to steal data and use the stolen information for nefarious purposes.

This is where Metasploit Pro comes into the picture. One of the major capabilities of the Metasploit Pro social engineering feature is the ability to easily create and send phishing e-mails. From within Metasploit Pro, you can create and set up the components that you need to run a phishing attack - including the phishing e-mail, spoofed website, mail server settings, and target list.

Now that you have a general overview of how phishing attacks work, and how Metasploit Pro helps you phish people, let's go over some tips that will help you set up successful phishing attacks.

Phishing Tip #1: Clone, clone, clone.

One of the most useful capabilities of the social engineering feature is the ability to clone a real, live web page. To clone a web page, you simply need to provide Metasploit Pro with the URL. Metasploit Pro makes a copy of the web page's HTML and imports it into the campaign. After the HTML has been imported, you can tweak the code to further customize or perfect the page.

Since the purpose of a spoofed web page is to trick a human target into believing in its authenticity, it is absolutely vital that the spoofed web page be a near replica of the real one. Therefore, unless you are creating a unique web page for the purposes of the campaign, you should always clone an existing web page. When you clone an existing web page, resources files, such as images, will be served from the cloned website and yield less setup overhead. Overall, the cloning feature makes it extremely fast and easy for you to get a web page up and running.

Phishing Tip #2: Set up a real looking domain.

A domain name is the most obvious telltale of a suspicious website, so it's important that you use a domain name that is a close match to the real one. For example, the fake domain name for Rapid7 can be something like RAP1D7 or RAPID7. Obviously, a URL like http://www.RAPID7.com/home looks much less nefarious than http://196.184.132.24/home.

Since most people should be able to recognize a blatantly fake URL, you should use a real looking domain name. This will test a human target's ability to examine URLs and identify malicious links.

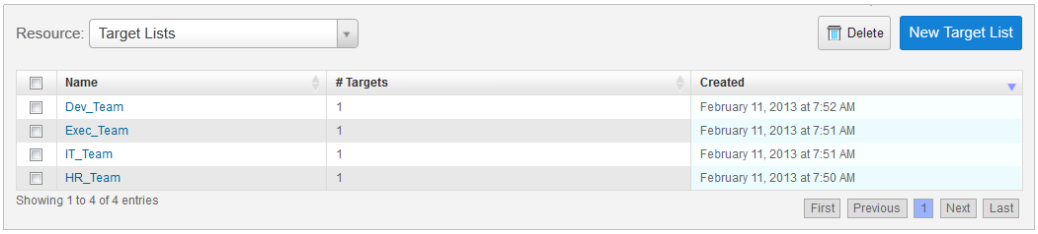
In order to set up a domain name for your Metasploit web server, you'll need to own and register the domain name. Once you have all of that set up, you'll need to point the domain name at the server running your Metasploit instance.

Phishing Tip #3: Target a smaller population.

To maintain the sanity of your IT team, you should use smaller target lists for the majority of your social engineering tests. With smaller target lists, you will be able to easily mitigate any issues and concerns that may arise.

Additionally, by limiting the number of human targets, you can control the sample of people participating in the test. For example, you may want to create a separate target list for your IT team because they may require a different type of social engineering test than the rest of the company.

However, there may be occasions where you want to run large scale tests. These tests will typically replicate a real attack scenario in which a large portion of the organization is affected. In these particular cases, you should create a large target list that includes all the targets in the organization. These large scale tests will help the organization understand their current security posture and identify where improvements need to be made in the IT and security infrastructure.



Phishing Tip #4: Use a SMTP relay service.

One of the most common issues you may encounter during a social engineering test is the inability to send e-mail through your local mail server. Most mail servers will perform a reverse DNS lookup to verify that the IP address of the server hosting Metasploit Pro matches the domain name of the e-mail that you are trying to spoof. If there's an issue with the reverse DNS lookup, the mail server will most likely reject the e-mail because it appears to originate from a suspicious source.

Since mail servers are configured to use the highest level of protection and to perform restrictive checks for spam, malicious e-mails, and e-mail abuse, it makes it very difficult to successfully deliver phishing e-mails.

To work around this issue, you should use an SMTP relay service, like Sendgrid, JangoSMTP, or Mandrill. Publicly available e-mail services, like Gmail, Hotmail, and Yahoo should not be used because they enforce the highest level of security and will most likely blacklist any e-mail that appears to be spam. Regardless of the provider you choose, always send yourself the phishing emails first to verify they get delivered with a low or zero spam rating to increase your chance of success.

Phishing Tip #5: Capture credentials.

If you intend to use a social engineering assessment to promote security awareness, you should use Metasploit Pro's phishing campaign to launch a spoofed website to capture credentials. Unfortunately, nothing affects change faster than stolen credentials.

The phishing campaign is preconfigured with the components that you will need to create the phishing e-mail, spoofed page, and redirect page. After you set up and launch the phishing campaign, you can observe the campaign findings in real-time. From the real-time findings, you can easily identify the human targets that have submitted their credentials and actually view the information that they have submitted.

Due to the open nature of spoofed page content, Metasploit Pro does not have the ability to hide credentials in the Social Engineering Campaigns Details report. Therefore, due to the sensitive nature of this content, the form submission content is not automatically included in the report. If you choose to create a custom report outside of Metasploit Pro, and opt to include the collected form submission content, please be sure to obfuscate a portion of the data - especially if you are showing sensitive data like passwords or credit card numbers.

Phishing Tip #6: Spoof the hover text.

The easiest way to identify a phishing e-mail is by hovering over the links embedded in the e-mail. To make the phishing e-mail more authentic looking, you should use the spoof hover text to URL option to modify the hover text. This option is available through the Link to web page attribute and changes the URL that displays in the hover text to any URL you want to use.

For example, if your Metasploit Pro instance runs on a web server that does not point to a DNS server, your web server URL will be something like http://1.2.3.4/blue123. If this is the case, you will want to change the hover text to display a URL that looks like it directs to a real web page, like http://www.rapid7.com.

USB Baiting

If you've ever been in an office environment, you may have noticed random USB keys scattered around. If these USB keys are left next to the copy machine or coffee machine, you may think that the owner has misplaced the key. So, your first instinct may be to install the USB key to examine its contents in order to identify the owner.

When you view the contents of the USB key, you may see a file that is aptly named to get you to open it. For example, you may be more likely to open a file name like "Joe_Resume.pdf" because it may contain useful, personal information about the owner of the USB key. Unfortunately, these files are usually not as innocuous as they seem. Opening one of these files can install malicious code onto your computer and give an attacker access to your system.

USB baiting, or a USB key drop, uses thumb drives to deliver malicious payloads and heavily relies on human curiousity to be successful. Most baiting schemes require that you have access to the comany's office facilities, which may require you to utilize some creative techniques in order to get through the front door. For example, you may need to dress up like someone from technical support or you may spend some time building a relationship with someone in the company.

During a social engineering penetration test, you should leverage USB key drops to raise security awareness, ensure adherence to security procedures, and improve defense strategies within an organization.

Aside from phishing, one of the other major capabilities of Metasploit Pro is the ability to generate and download a malicious file, such as an executable or an infected file, that can be placed onto a USB key. You can create a malicious file, such as a PDF that contains the Adobe Cooltype exploit, with the portable file component. After you create the malicious file, you will need to download the file, save it to a USB key, and drop the key off in a high traffic area.

Now that you have a general overview of how baiting works, let's go over some tips that will help you set up successful baits.

USB Baiting Tip #1: Carefully research and plan the attack.

As with any other penetration test, research and planning play a vital role in setting up a successful USB key drop. USB key drops are different from standard phishing attacks because they require you to physically access an unfamiliar location and attack systems with possibly very little reconnaissance.

Therefore, two of the most important elements you should research are the location and the potential target systems.

For a USB key drop to be successful, you need to identify an area in your targeted location that gets the most traffic. A high traffic area will most likely yield a higher possibility that someone will pick up a USB key and install it onto their system. Additionally, if you do not have direct access to the targeted location, you may need to create a back story to gain entry into the location. For example, you may want to pretend to be part of a maintenance crew or delivery service, which may require you to obtain the appropriate uniform and props to play the role.

When researching the location, you will need to ask yourself questions like:

“How will I get in?”

“What’s my story for being there?”

“Where are the high traffic areas located?”

“Who might I encounter?”

Answering questions like these will help you prepare and plan for a USB key drop.

Since you cannot control who picks up the USB key, you do not know if their system will be vulnerable to the exploit on the USB key. Therefore, it is important that you gather as much information as you can about the systems within the organization so that you can choose the most relevant and effective exploits. For example, if you know that most systems run Windows, you can tailor your attack to use Windows only exploits. Or if you know that most systems have Adobe Reader, you can use PDFs to deliver your exploits.

With extensive research, you can build an effective and strategic plan of attack that will provide clear insight into the organization.

USB Baiting Tip #2: Use descriptive and enticing file names.

When someone finds a USB key, their natural inclination may be to insert the USB key to find the owner or to view the contents of the drive. Therefore, you should always use file names that indicate that the file contains personal or confidential information. For example, a file name like “ContactInfo.pdf” or “payroll.exe” will be more likely to lure someone into opening it.

Malicious Attachments

A malicious attachment is a file format exploit or executable file that is e-mailed to a human target. The e-mail appears to come from a trusted source and always contains an attachment that must urgently be

downloaded.

Some of the most prolific social engineering attacks have started with the innocuous act of the opening an e-mail attachment. The attached file contains an exploit that delivers a malicious payload to the target's system, which in turn, makes the system vulnerable to viruses, malware, spyware, and trojans. In some cases, the attack creates a chain of events that can compromise the entire network.

Most likely, the recipient was completely unaware that the attachment was harmful because the e-mail appeared to originate from a familiar source. Similarly to phishing attacks, the attacker has manipulated the recipient into believing that the e-mail was authentic and that the attached file was trustworthy. For example, personalized corporate e-mails about stock options and health insurance are more likely to lure someone into reading them and downloading any files attached to them than generic e-mails about sales figures.

As a social engineering penetration tester, you need to identify the potential risks that malicious attachments pose to an organization and provide solutions that can mitigate those risks. It is important to provide employees with the necessary skills to reduce the risk that they pose to an organization and to identify the most pervasive vulnerabilities that the organization needs to address.

Now that you have a general overview of malicious attachments, let's go over some tips that will help create malicious e-mails and attachments.

Malicious Attachments Tip #1: Craft a convincing and legitimate looking e-mail.

To appeal to a human target's sense of trust and curiosity, you need to create an e-mail that not only looks legitimate, but contains information that is of interest to the human target.

Any e-mail you create should use the same logo, font, and colors that the real one would. If you are spoofing a corporate e-mail, you should use a real e-mail as a model so that you can accurately recreate the exact header, footer, and signature. These elements provide visual cues to the target that the e-mail comes from a trusted and familiar source.

In order to convince the human target to actually open an attachment, you need to persuade them that the attachment contains information that they absolutely need to view. Typically, people will want to view any information that they think will impact them directly. For example, an e-mail about annual bonuses with an attachment named 2013_bonus_plan.pdf will probably get more views than an e-mail about a new corporate handbook.

Malicious Attachments Tip #2: Use a common file format exploit.

Depending on the information you have gathered about the target systems, you should use exploits that are delivered using a common file format type. For example, most Windows systems in an corporate

environment will have Microsoft Windows or Adobe Reader. Therefore, when choosing a file format exploit, you should factor in the likelihood that the target will have the necessary software to open the file.

Malicious Attachments Tip #3: Zip attached files.

Most e-mail services will not deliver an executable file attached directly to an e-mail. So, if you want to attach an executable file to an e-mail, you should always send the file in a Zip file. This reduces the possibility of the attachment being flagged as a malicious file.

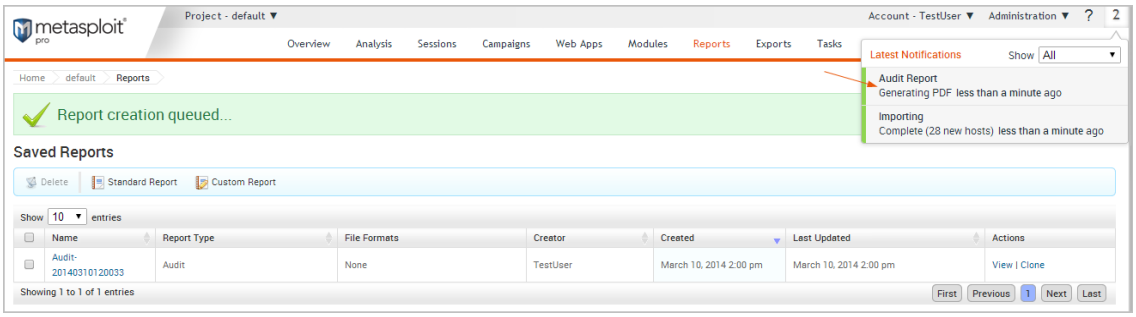
Reports

A report clearly presents project data in a distributable and tangible output format. It organizes your findings into relevant sections, displays charts and graphs for statistical data, and summarizes major findings. This is extremely useful when you need to share information with people who do not have access to Metasploit Pro or who want to quickly process your test results.

All tasks related to reports, such as generating, downloading, e-mailing, and deleting them, can be performed from the Reports area of the web interface.

Notification Center Statuses for Reports

When you generate a report, the Notification Center alerts you when a report has started generating, finished generating, or encountered an error during generation. The Notification Center appears as an icon in the upper-right corner of the global toolbar and displays the total number of unread notifications. You can click on the Notification Center icon to display a list of alerts.

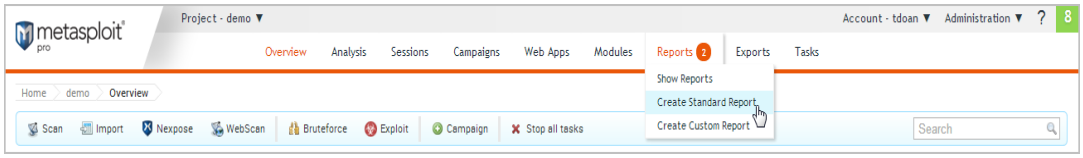


The Notification Center displays the following statuses for reports:

- Report started: This status indicates that the report has started generating.
- Report finished: This status indicates that the report was generated without errors and is ready for you to view and download. You can click on the alert to open the report. When you open the report from the Notification Center, it displays a unified view of the report and shows the formats that are available for it. You can click on any of the format icons to view the report in the selected format.
- Problem with report: This status indicates that there was an issue with the report and it was not able to finish. You will need to view the report log to troubleshoot the issue. For more information on report logs, see *Report Logs* on page 1.

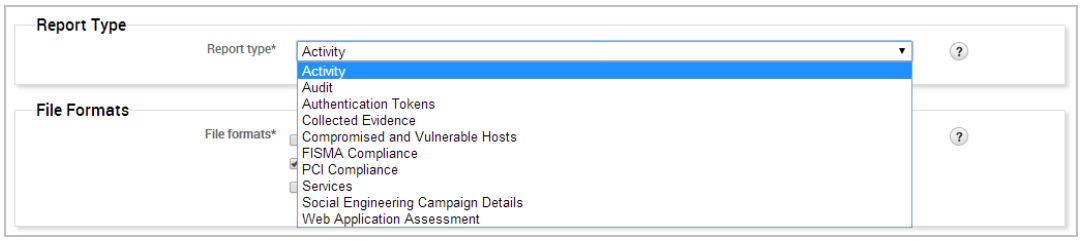
Generating a Standard Report

1. Open the project that contains the data you want to use to create a report.
2. Select **Reports > Create Standard Report** from the Project tab bar.



   The **Reports** page appears with the Generate Standard Report tab selected.

3. Click the **Report type** dropdown and choose the report you want to generate.



   For more information on the report types that are available, see *Metasploit Report Types* on page 1.

4. Choose the file formats you want to generate for the report.

   You can generate multiple formats for a report at the same time. Most reports can be generated as PDF, Word, RTF, or HTML documents; however, the Web Application Assessment Report cannot be generated as a Word file.

5. Enter a name for the report in the **Report Name** field. (Optional)

   If you do not specify a name, Metasploit Pro uses the report type and the timestamp. For example, an Audit Report will be named `Audit-20140106140552`.

6. Use the **Included addresses** to explicitly define the hosts you want to include in the report. (Optional)

   For example, if you only want to include specific hosts in the report, you should define those hosts in

the **Included Addresses** field. All other hosts will not be included in the report.

7. Use the **Excluded addresses** to explicitly define the hosts you want to exclude from the report. (Optional)

   For example, if you only want to exclude specific hosts from the report, you should specify those hosts in the **Excluded Addresses** field. All other hosts will be included in the report.

8. Click the **Campaign** dropdown and select the campaign you want to use to create a report. (Social engineering reports only)

   The report form only displays the campaigns that are stored in the project.

9. Click the **Cover Logo** dropdown and select the logo that you want to use on the cover page of the report.

   If you have not uploaded a logo to the project, you must upload the logo that you want to use to the Custom Report Collateral area of the project. For more information on uploading a logo, see *Adding a Custom Logo to a Report* on page 247.

10. Select the report sections that you want to include in the report.

    The report sections that are available will vary between reports. For more information on the sections available for each report, see *Understanding Report Content* on page 1.

11. Enable or disable any report options to manage the data that appears in the report.

    The report form displays the options that are applicable for the report type that you have selected.

    The following report options may be available:

    - Mask discovered passwords: Removes all credentials, including plain text passwords, hashes, and SSH keys, from the report. The report displays the user name and a blank password.
    - Include session details: Shows the details for each session Metasploit Pro was able to open, such as the session type and attack module that Metasploit Pro used to obtain the session.
    - Include charts and graphs: Includes visual aids, such as pie graphs, to accompany statistical findings in a report.
    - Include web page HTML (in addition to image preview): Includes the original page code as raw text as well as the rendered preview image. (Social Engineering Campaign Details Report only)

12. Enter the e-mail addresses you want to send the report to after the report generation. (Optional)

    You can use a comma or semi-colon to separate multiple e-mail addresses.

    To e-mail a report, you must have an active mail server configured through the Global Settings. For

more information on setting up a mail server, see *Defining SMTP Settings for a Mail Server* on page 1.

13. Generate the report.

    When the report generation begins, the web interface redirects you to the View Reports tab. At this point, you can navigate away from the Reports page to other areas in Metasploit Pro. The Notification Center will alert you when the report generation completes.

    When the report generation completes, you can click on the Notification Center icon to view the notification message or you can select **Reports > Show Reports** from the Project tab bar to access the Reports area.

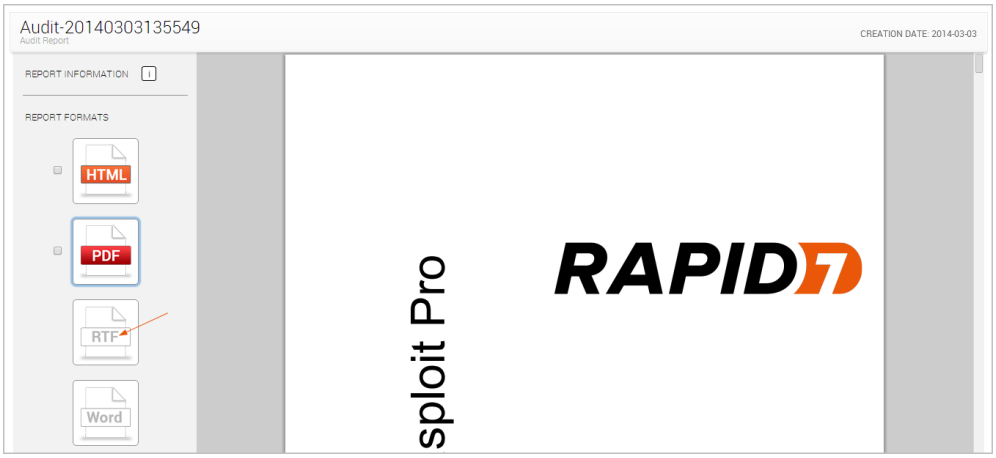
    If an error occurred during report generation, you can view the report log to identify and troubleshoot any errors that occurred. For more information on report logs, see *Report Logs* on page 1.

Generating Additional Formats for a Report

1. Open the project that contains the report for which you want to generate additional formats.
2. Select **Show Reports** from the Project tab bar. The **Show Reports** page appears.
3. Find the row that contains the report for which you want to generate additional formats.

   The row shows the metadata and the file formats that are available for the report.
4. Click on the report name to open it.

   The unified report view will open and display a preview of the report. The formats that are available for the report will be displayed in the sidebar. Formats that have a colored icon and checkbox have already been generated. Formats that are grayed out have not been generated.
5. Click on the file format that you want to generate for the report. You can only generate one format at a time.



When the report generation begins, the format button will be replaced with a progress indicator. The format button will reappear when the report is ready for you to view or download.

At this point, you can navigate away from the Reports page to other areas in Metasploit Pro. The Notification Center will alert you when the report generation completes.

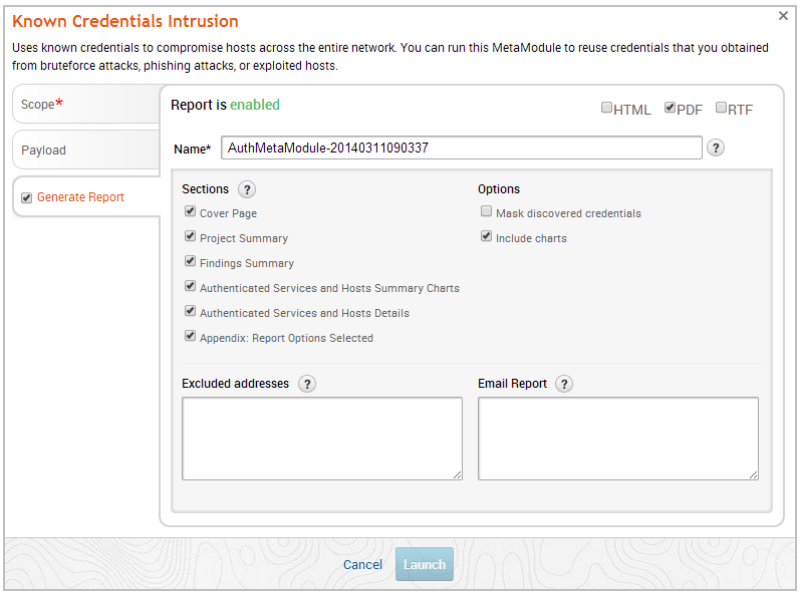
When the report generation completes, you can click on the Notification Center icon to view the latest notification message or you can select **Reports > Show Reports** from the Project tab bar to access the Reports area.

If an error occurred during report generation, you can view the report log to identify and troubleshoot any errors that occurred. For more information on report logs, see *Report Logs* on page 1.

Generating MetaModule Reports

A MetaModule provides a guided interface to walk you through a single penetration testing task. Each MetaModule leverages the core functionality of a module, such as password testing, but enables you to quickly configure and run the module with minimal set up. Each MetaModule includes a specialized report, which contains data that is specific to the MetaModule run.

MetaModule reports are configured from within the MetaModule and are generated when the MetaModule runs. After the MetaModule generates the report, you can view the report from the Reports area. For more information on MetaModule reports, see *MetaModule Reports* on page 1.



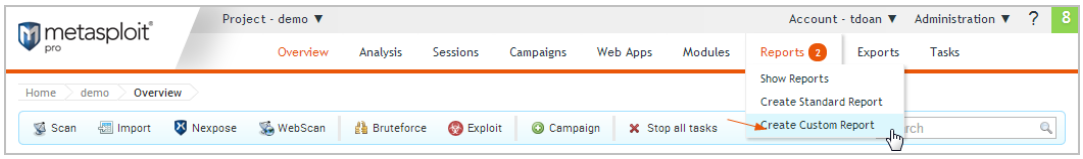
Generating a Custom Report

A custom report is created using a user-uploaded Jasper report template. The template defines the layout of the report and the sections that the report contains. You can create a report template from scratch using a tool like iReport. For more information on custom templates, see *Working with Custom Templates* on page 251.

Before you can generate a custom report, you must upload the template that you want to use to the Custom Report Collateral area of the project. If the project does not contain any custom report templates, the New Custom Report form will not load. Instead, the form displays a warning that the project does not contain any templates. You must upload a valid JRXML template to continue. For more information on uploading a custom template, see *Uploading Templates* on page 256.

To generate a custom report:

1. Open the project that contains the data you want to use to create a report.
2. Select **Reports > Create Custom Report** from the Project tab bar. The **New Custom Report** page appears.



3. Select the template you want to use to create the report.
4. Choose the file formats you want to generate for the report.

You can select multiple formats. All formats will be generated for the report at the same time.

5. Enter a name for the report in the **Report Name** field. (Optional)

If you do not specify a name, Metasploit Pro uses the report type and the timestamp. For example, an custom report will be named `Custom-20140106140552`.

6. Use the **Included addresses** to explicitly define the hosts you want to include in the report. (Optional)

For example, if you only want to include specific hosts in the report, you should define those hosts in the **Included Addresses** field. All other hosts will not be included in the report.

7. Use the **Excluded addresses** to explicitly define the hosts you want to exclude from the report. (Optional)

   For example, if you only want to exclude specific hosts from the report, you should specify those hosts in the **Excluded Addresses** field. All other hosts will be included in the report.

8. Click the **Cover Logo** dropdown menu and select the logo you want to display on the cover page of the report. (Optional)

   If you do not select a logo, the report will use the default Rapid7 logo.

9. Enter the e-mail addresses you want to send the report to after the report generates. (Optional)

   You can use a comma or semi-colon to separate multiple e-mail addresses.

   To e-mail a report, you must have an active mail server configured through the Global Settings. For more information on setting up a mail server, see *Defining SMTP Settings for a Mail Server* on page 1.

10. Generate the report.

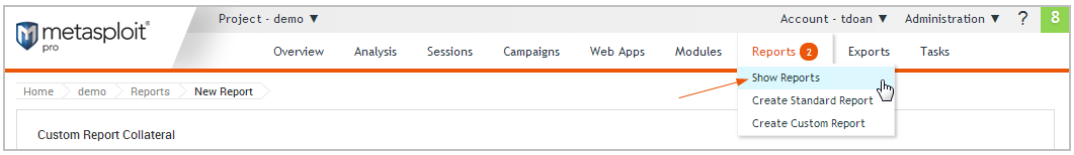
    When the report generation begins, the web interface redirects you to the View Reports tab. At this point, you can navigate away from the Reports page to other areas in Metasploit Pro. The Notification Center will alert you when the report generation completes.

    When the report generation completes, you can click on the Notification Center icon to view the notification message or you can select **Reports > Show Reports** from the Project tab bar to access the Reports area.

    If an error occurred during report generation, you can view the report log to identify and troubleshoot any errors that occurred. For more information on report logs, see *Report Logs* on page 1.

Downloading a Report

1. Open the project that contains the report you want to download.
2. Select **Reports > Show Reports** from the Project tab bar. The **Reports** page appears.



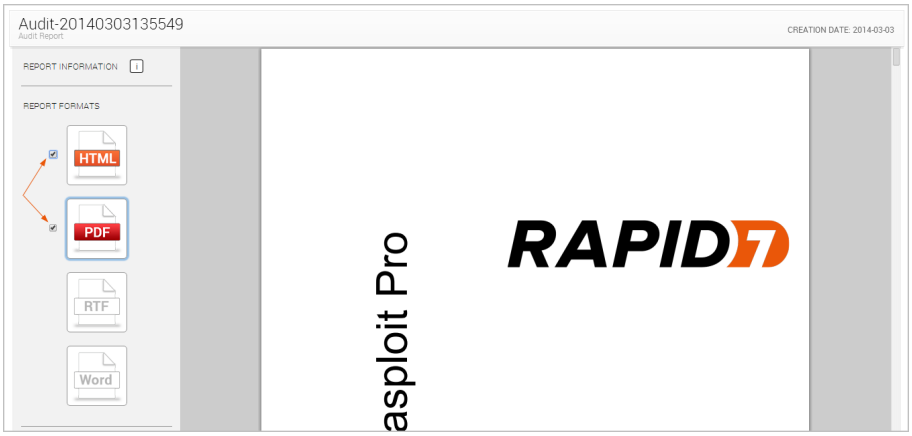
3. Find the row that contains the report you want to view.

   The row displays the metadata and the file formats that have been generated for the report.

4. Click on the report name to open it.

   The unified report view will open and display a preview of the report.

5. Select the formats you want to download.



   The formats that are available for the report will have an active checkbox located next to them.

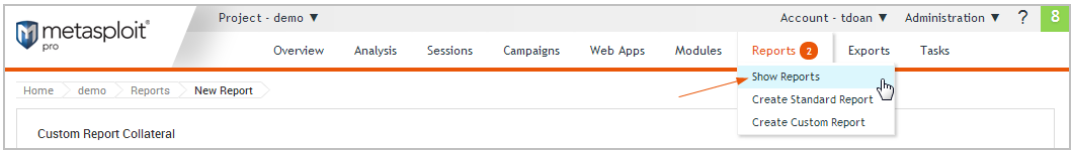
6. Click the **Download** button located under the Report Actions area.

   The download process will automatically start.

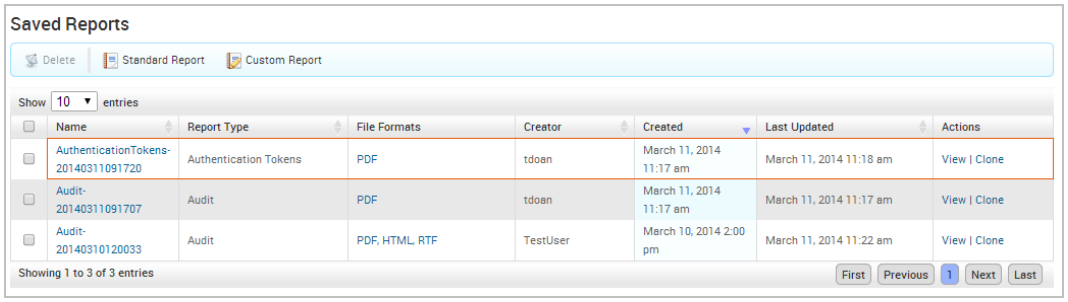
   If your browser is not configured to automatically download files, a dialog window will appear and prompt you to save or run the file. You will need to save the report to your computer.

Viewing a Report

1. Open the project that contains the report you want to view.
2. Select **Reports > Show Reports** from the Project tab bar. The **Reports** page appears.



3. Find the row that contains the report you want to view.



The row displays the metadata and the file formats that have been generated for the report.

4. Click on the format that you want to view the report in.

The report will open in your browser.

E-mailing a Report

You can quickly share reports by e-mailing them as soon as they are generated. Both the standard and custom report generation forms have an **Email Report** field that enables you to define a list of e-mail recipients.

As long as you have a valid mail server configured for your Metasploit Pro instance, the report will automatically be sent to the e-mails you have listed.

Setting Up a Mail Server

In order to utilize e-mail capabilities, you must have access to a local mail server or a web mail server. You need the address and port that the mail server runs on, the domain name that hosts the mail service, and the credentials for the mail server.

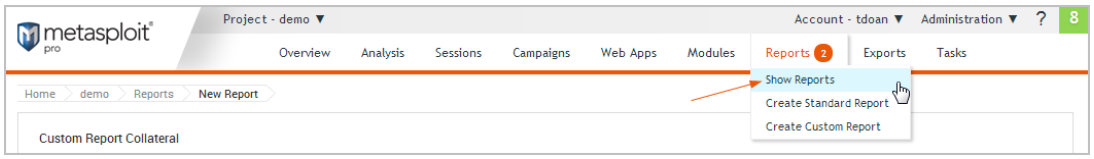
For more information on setting up a mail server, see *Defining SMTP Settings for a Mail Server* on page 1.

Cloning a Report Configuration

You can clone a report to make a copy of an existing report's configuration. Report cloning enables you to reuse and rerun a previously generated report. You can modify the configuration or run it as it is.

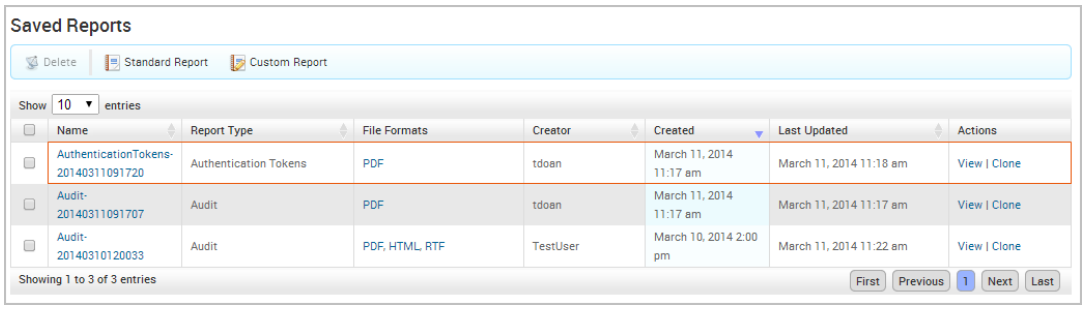
To clone a report:

1. Open the project that contains the report you want to delete.
2. Select **Reports > Show Reports** from the Project tab bar.

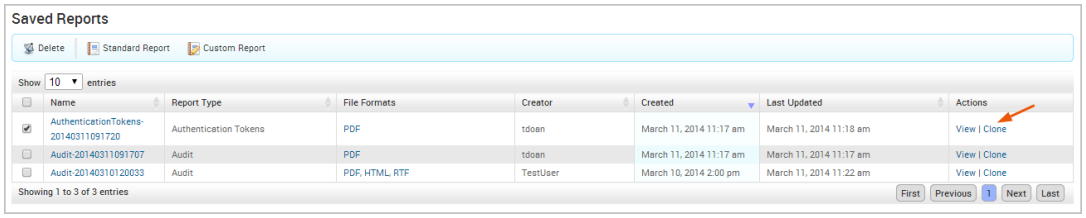


The **Reports** page appears.

3. Find the row that contains the report that you want to clone.



4. Click the **Clone** link located under the Actions column.



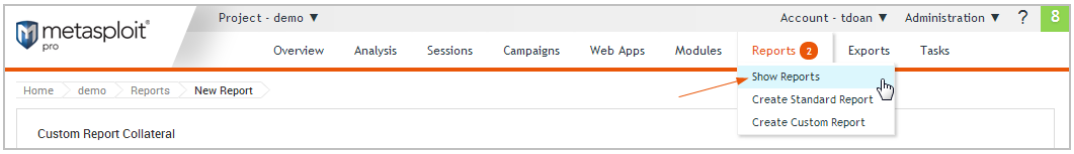
The New Report form appears. The form retains the configuration settings that you used to generate the original report.

Deleting Reports

When you delete a report, it will be permanently removed from the Reports directory, and you will no longer be able to view it from the Reports area of the web interface. Please make sure that you have the data that you need from the report before you delete it.

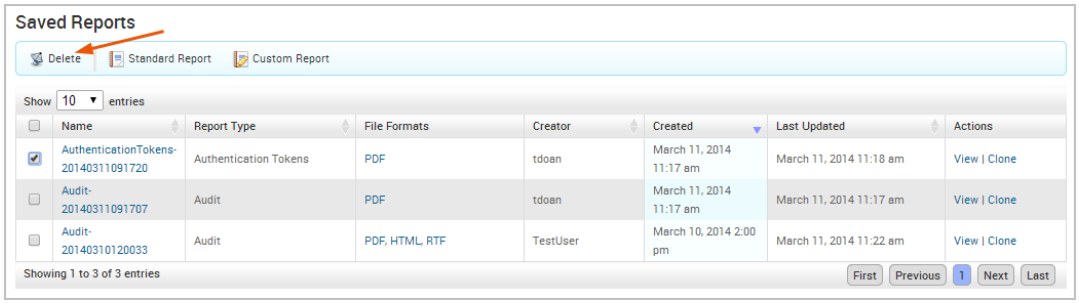
To delete a report:

1. Open the project that contains the report you want to delete.
2. Select **Reports > Show Reports** from the Project tab bar.



The **Reports** page appears.

3. Select the report or reports that you want to delete.
4. Click the **Delete** button located in the Quick Tasks bar.



The browser will ask you to confirm that you want to delete the report.

5. Select **OK** to delete the report.

Customizing Standard Reports

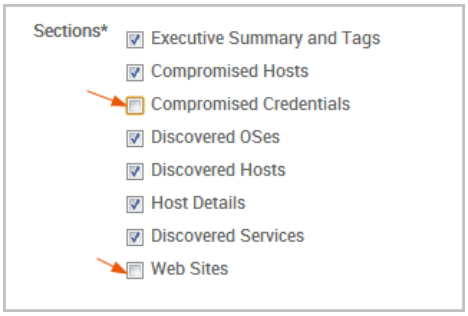
A standard report is based on a Metasploit report template, which controls the look and feel of the report. All reports have a cover page and include a set of options that enable you to manage the report data. You can customize some parts of a standard report, such as the logo and sections of content that appear in the report.

If you want to modify the layout of the report, you will need to use a custom template. For more information on custom templates, see *Working with Custom Templates* on page 251.

Excluding Report Sections

A report is made up of multiple sections. Each section divides the report content into distinct areas of information.

When you view the New Report form, you will see the sections that are available for the report you have selected. By default, all sections will be selected. If you want the report to only show certain sections of a report, you can exclude sections from the report.



To exclude specific sections, you can deselect the sections you do not want to appear in the report.

When you generate the report, you will not see the excluded sections in the report. Additionally, the report will only show content for the sections for which it has data.

For more information on report sections, see *Metasploit Report Types* on page 1.

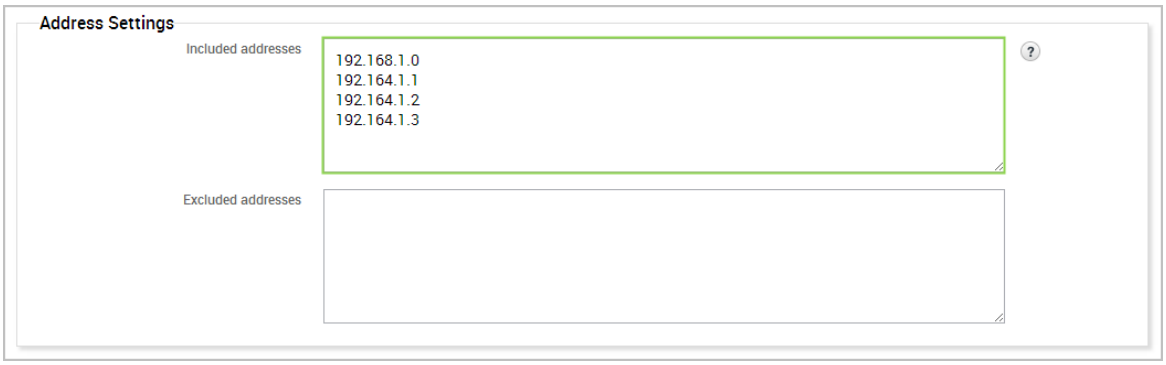
Excluding and Including Hosts from Reports

When you generate a report, Metasploit Pro automatically includes data from all hosts in the project. If you want to limit the data to a particular set of hosts, you can create an inclusion or exclusion list.

Creating Inclusion Lists

An inclusion list defines the hosts that you want to include in a report. Only the data for the hosts that you have explicitly defined will be displayed in the report.

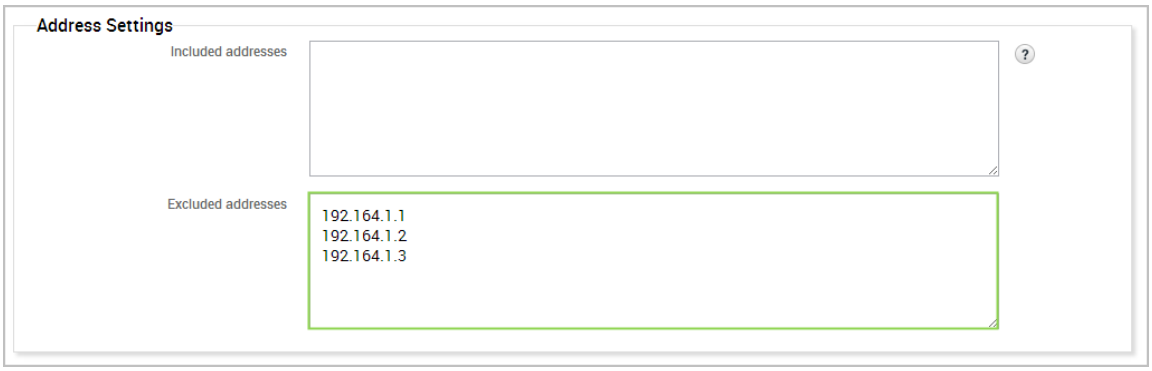
You create an inclusion list from the New Report form . Use the **Included addresses** field to define the specific hosts you want to include in the report. You can enter a single IP address, an address range, or a CIDR notation. If there are multiple addresses or address ranges, use a newline to separate each entry.



Creating Exclusion Lists

An exclusion list defines the hosts that you do not want to include in a report. The report will include data for all of the hosts in the project, except for the ones that you have defined in the exclusion list.

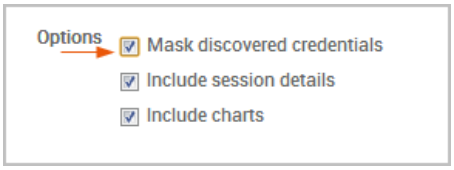
You create an exclusion list from the report generation form. Use the **Excluded addresses** field to define the specific hosts you want to exclude from the report. You can enter a single IP address, an address range, or a CIDR notation. If there are multiple addresses or address ranges, use a newline to separate each entry.



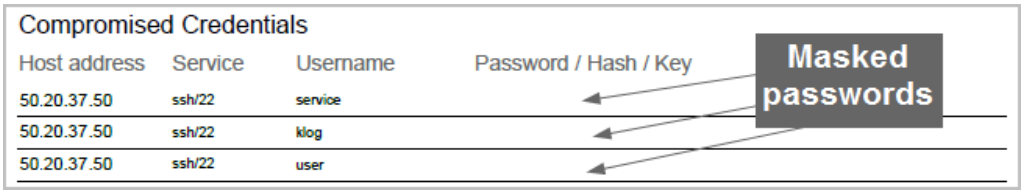
Masking Credentials from Reports

You can mask credentials if you do not want to include the plain text passwords and hashes in the Audit, Authentication Tokens, FISMA, and PCI reports.

To mask credentials from a report, you need to select the credential masking option on the New Report form. Select the **Mask discovered credentials** option to enable credential masking in your report.



When the masking option is enabled, the reports will not display plaintext credentials. For example, when you view the generated Audit report, the Compompromised Credentials section only shows the host addresses, services, and user names that were discovered. The password, hash, and key fields are blank.



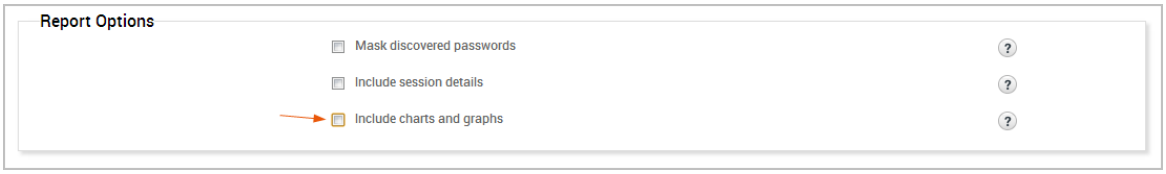
Other reports, such as the PCI and FISMA reports, replace all credentials with `<blank>`.

Removing Charts from Reports

Charts visually present numerical data. They are effective when you use to them present and compare large sets of information. You can include them in a report to simplify quantitative data and to highlight trends in your findings. Metasploit Pro reports mostly use pie charts to illustrate how data is distributed across different categories.

Most reports, with the exception of the FISMA, PCI, Social Engineering, Web Application Assessment, and Activity reports, have the option to include charts. By default, this option is enabled, so charts will be automatically generated for applicable reports. If you do not want to include charts in your report, you can disable the charts and graphs option.

To exclude charts and graphs from a report, deselect the **Include charts and graphs** option.



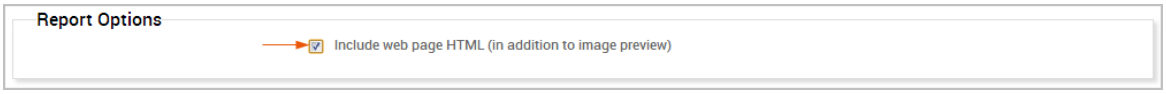
Including Web Page HTML in Social Engineering Reports

The Social Engineering Report presents the findings and data for a particular campaign. It contains the details for the campaign components that you used to build the campaign, such as the target list, e-mail, and web pages used.

The raw content for the target list and e-mail will automatically be included in the report. If you want to include the raw content for the web pages, you will need to enable the **Include web page HTML** option. If enabled, this option includes the HTML for each web page used in the campaign. A preview of the web page will render in the report if the web page was used as part of a campaign.

Note: If the web page delivered malicious code, such as a client-side exploit, Java applet, or executable file, a preview will not be rendered for the web page.

If you want to include the raw HTML that was used to create a web page and a preview of the web page, you can select the **Include web page HTML** option on the New Report form.



Customizing Report Names

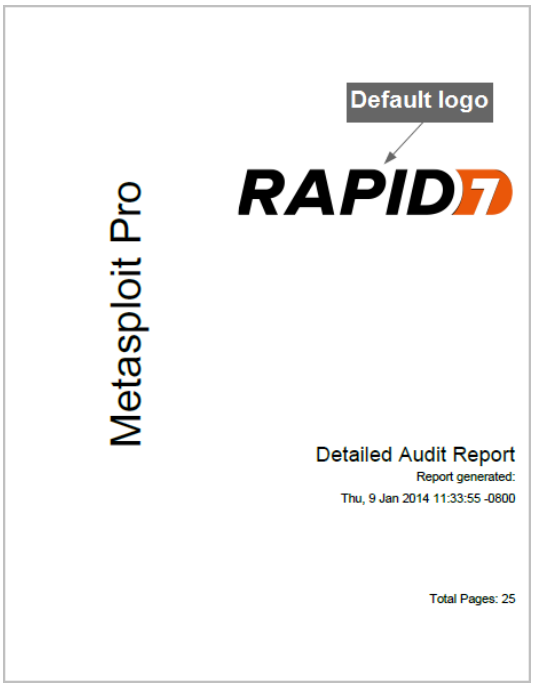
Metasploit Pro uses the following naming convention for report names: `<report type>-<timestamp>`. The report name appears in the Reports list.

You can change the name by replacing the default name in the **Report Name** field on the New Report form.

Adding a Custom Logo to a Report

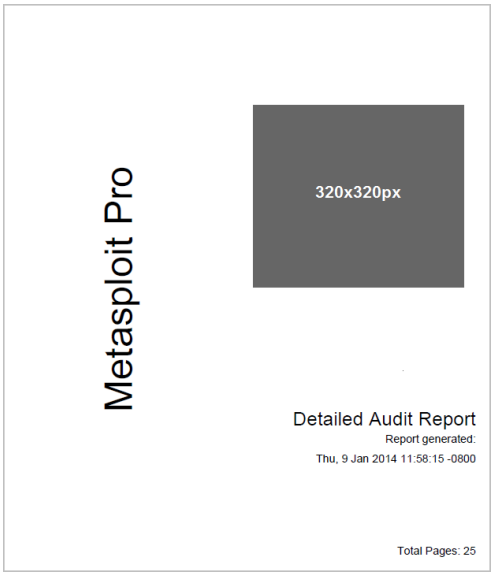
All reports include a cover page that displays the title, logo, and timestamp. The cover page displays the Rapid7 image as the default logo on all reports.

If you want to replace the default logo, you can upload a JPG, GIF, or PNG file. The uploaded logo can be used to brand a report with your organization's identity. The logo appears in the right side of the cover page and replaces the default Rapid7 logo.



Logo Requirements

The logo area on the cover page is 320 x 320 pixels. You can upload an image that is larger than the logo area, but the logo will be resized to fit the cover page.

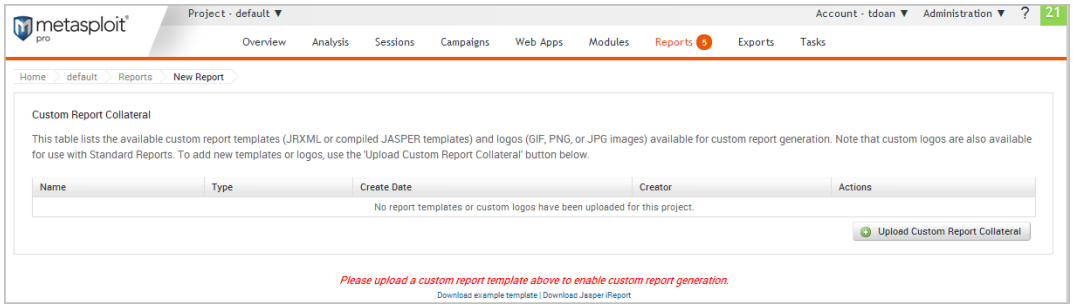


If the image is larger than the logo area, the height of the image will be preserved, but the width will be resized.

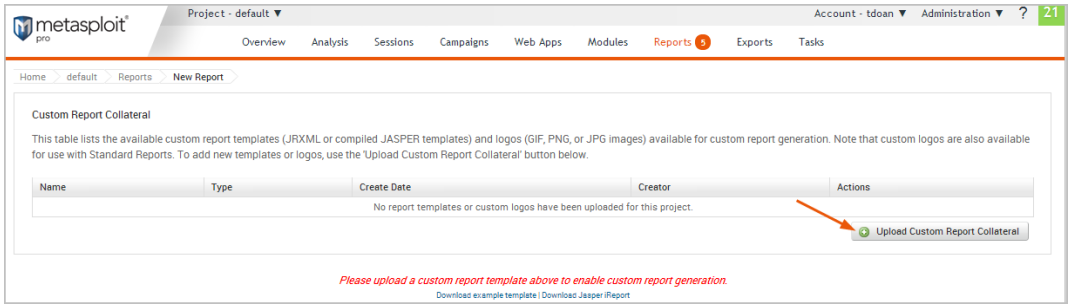
Uploading a Custom Logo

1. Open the project that you want to upload the logo to.
2. Select **Reports > Create Custom Report** from the Project tab bar.

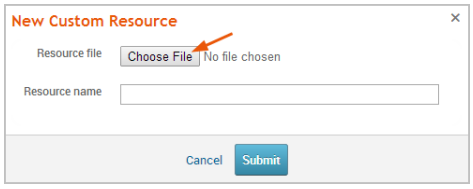
   The **Reports** page appears.
3. Find the **Custom Report Collateral** area.



4. Click the **Upload Custom Report Collateral** button.



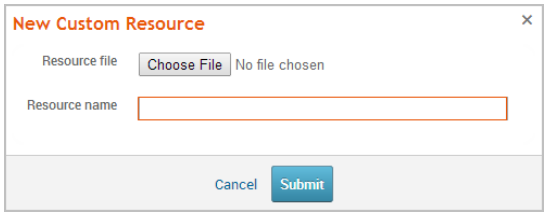
   The Upload window appears.
5. Click the **Choose File** button.



   The Open dialog window appears.
6. Browse to the location of the logo file.

   Note: You can upload a GIF, JPEG, JPG, or PNG file.

7. Select the logo file and click the **Open** button.
8. Enter a name for the file in the **Descriptive Name** field. (Optional)



If you do not specify a name, the Custom Report Collateral area shows the original file name.

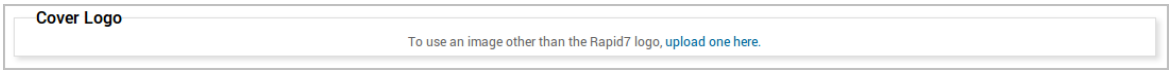
9. Click the **Upload** button.

The file appears under the Custom Report Collateral area.

Adding a Custom Logo to a Standard Report

To use a custom logo on the report's cover page, you need to click the **Custom report logo** dropdown on the New Report form and select the image you want to use. The dropdown will show the logos that have been uploaded to the project.

If the project does not contain any logos, the New Report form will display a link to the Custom Reports page where you can upload your logo.



Working with Custom Templates

Metasploit Pro ships with a set of predefined standard reports, which are created with Metasploit templates and designed to meet basic pentesting reporting requirements. However, if the standard reports do not provide you with the content or layout that you need, you can use a custom template to build your report. A custom template enables you to do things like apply corporate styles to your reports, control how and where content is displayed in your reports, and customize your reports for regional compliance needs.

A custom template is a JRXML file, which is an XML document with a JasperReport file extension. It contains the report structure, which defines where the report displays content, where it places images, and how it queries data. It can be built by directly manipulating XML or more easily by using a visual report tool for JasperReports, such as iReport Designer or the Eclipse-based Jaspersoft Studio.

Jasper Reports and iReport Designer

Metasploit Pro uses JasperReports 5.0, which is an open source Java-based reporting library, to compile JRXML templates and generate reports in output formats such as PDF, RTF, HTML, and Word. The JRXML template is a standards-based XML file that defines the elements and attributes that control where content is placed in a report. You can build the JRXML template with a visual report designer called iReport Designer, which is an open source tool maintained by Jaspersoft.

iReport Designer provides a graphical user interface that enables you to visually design your report templates without extensive knowledge of the JasperReports library, XML, and Java. You can drag and drop report elements to create layout of the report, and you can connect it to a data source, like JDBC and XML, to query data for the report. The resulting JRXML template can be imported into a Metasploit Pro project and used to create a custom report.

Downloading Jasper iReport

To download Jasper iReport, please visit the following site: http://jasperforge.org/projects/ireport.

Resources for JasperReports and iReport Designer

In order to build a custom template, you must be familiar with JasperReports and iReport Designer. There are quite a few resources available that will help you learn how to build report templates with iReport Designer and understand how JasperReports works.

To learn more about JasperReports or iReport Designer, visit the following resources:

- *JasperReports documentation list* - A list of the documentation that is available for JasperStudio, JasperReports Server, JasperReports Library, and iReport Designer. You can access this list at the

following URL: http://community.jaspersoft.com/documentation.

- *JasperReports Library materials reference* - A list of the documentation, webinars, and articles that may be helpful for working with JasperReports. You can access this list at the following URL: http://community.jaspersoft.com/wiki/jasperreports-library-reference-materials.
- *iReport Designer tutorials and help wiki* - A wiki that lists the tutorials that are available for iReport Designer. You can access this list at the following URL: http://community.jaspersoft.com/wiki/ireport-designer-tutorials-help.
- *An article on chart customizations* - A useful list of chart customizers for JasperReports, iReport Designer, and JasperReports Server. You can view this article at the following URL: http://mdahlman.wordpress.com/2011/04/17/chart-customizers-2/.
- *Groovy documentation* - Groovy is a Java-compatible scripting language that you can use in place of Java to define expressions in iReport.

  To learn more about how Groovy and iReport Designer work together, visit the iReport wiki here: http://http://community.jaspersoft.com/wiki/ireport-designer-groovy.

  To learn more about Groovy, you can view their documentation here: http://groovy.codehaus.org/.
- *Jaspersoft training* - To learn more about Jaspersoft training, you can visit https://www.jaspersoft.com/training-services or https://www.jaspersoft.com/training.

Requirements for Designing Custom Templates

To design a report template, you will need the following:

- Experience with Jasper iReport, JasperReports, XML, and SQL/XPath
- Experience with Java or a Java scripting language, like Groovy or Javascript
- A working instance of Jasper iReport
- Access to the Metasploit database

Setting Up the Metasploit Database in iReport Designer

To fill your report with data, you will need to set up a data source that points to the Metasploit postgres server. The information for the Metasploit postgres server can be found in `/path/to/metasploit/apps/pro/config/database.yml`.

You will need the following information from the database.yml file:

- *The database name* - The default database name is msf3.
- *The postgresql port* - The default postgresql port is 7337.

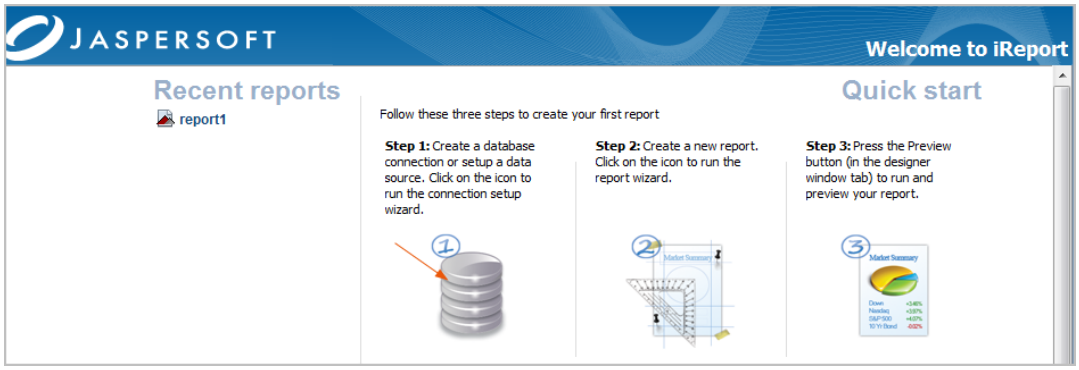
- *The user name* - The default user name is msf3.
- *The password* - Please view the database.yml file for your database password.

To set up a data source in iReport Designer:

1. Open iReport Designer.

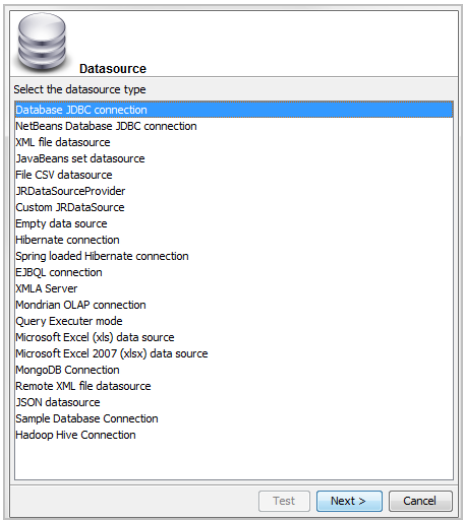
   The Quick Start window appears.

2. Click the **Database Connection** icon.

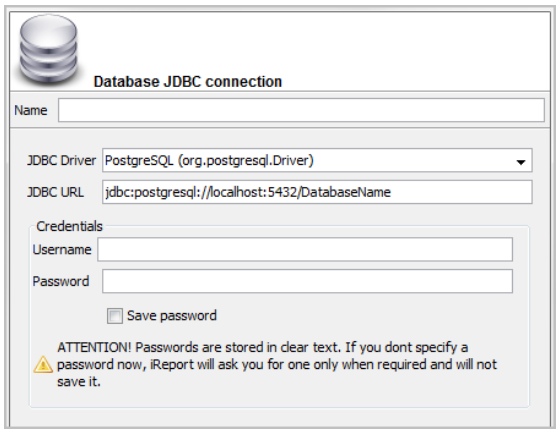


   The Datasource window appears.

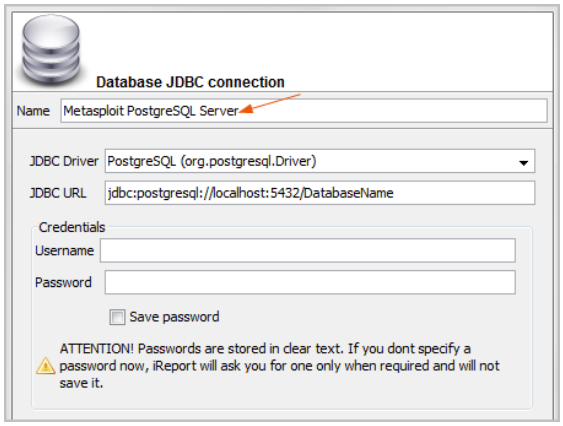
3. Select **Database JDBC connection** from the list of data sources.



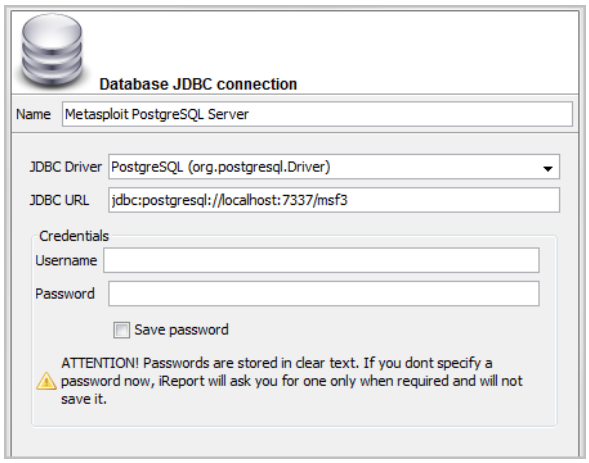
4. Click **Next**. The Database JDBC Connection window appears.



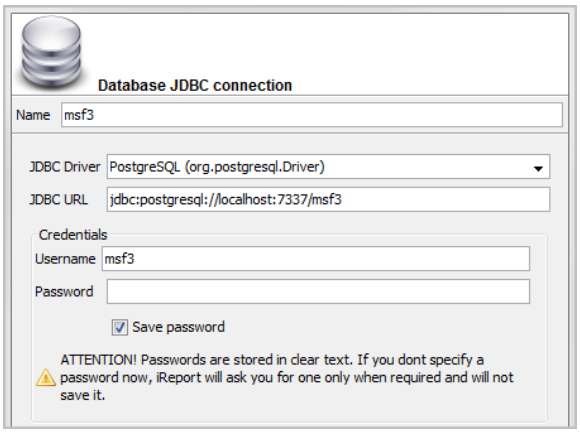
5. Enter a name for the connection in the **Name** field.



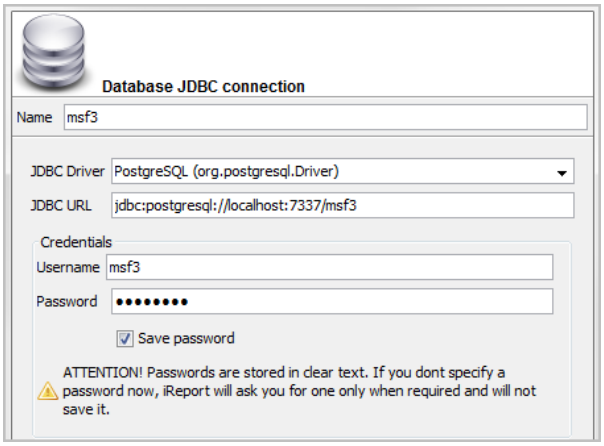
6. Replace with content in the **JDBC URL** field with `jdbc:postgresql://localhost:7337/msf3`.



7. Enter the database user name in the **Username** field.



8. Enter the database password in the **Password** field.



9. Test the connection.

   If the connection is working properly, a window appears and alerts you that the connection was successful.

   Otherwise, if the connection fails, an exception window appears and alerts you that there is an issue with your database settings. You will need to verify that your database settings match the information in the database.yml file.

10. Save the connection, if the connection was successful.

You are now ready to create your report template.

For resources on creating report templates, see *Resources for JasperReports and iReport Designer* on page 251.

Custom Resources Directory

All custom templates and logos are stored in the following directory: `/path/to/metasploit/apps/pro/reports/custom_resources`.

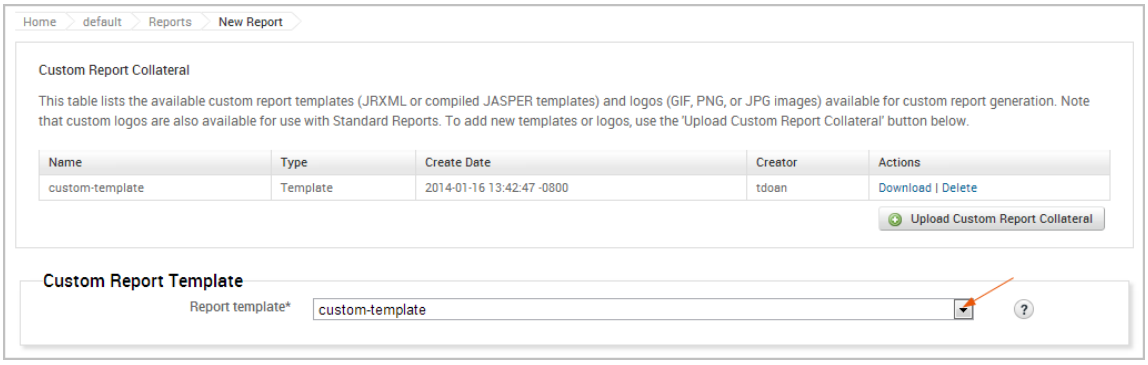
You can go to the custom resources directory to download or view logos and templates; however, you should not make any changes directly within the directory. If you need to modify your logos or templates, you should make a copy of the directory and make your changes from the new directory.

Any changes that you make directly from within the custom reports directory can cause disparities between the metadata that displays for the file in the web interface and the file itself. If you need to remove or add custom resources, you should do it from within the web interface. Do not delete them directly from the custom resources directory.

Uploading Templates

After you have created your custom template, you will need to upload it to the project you want to use to build the custom report. The template will only be available to the project that you have uploaded it to; therefore, if you want to use the template across multiple projects, you will need to import the template into each project.

When you view the New Custom Report form, the template will be available in the **Report Template** dropdown menu.

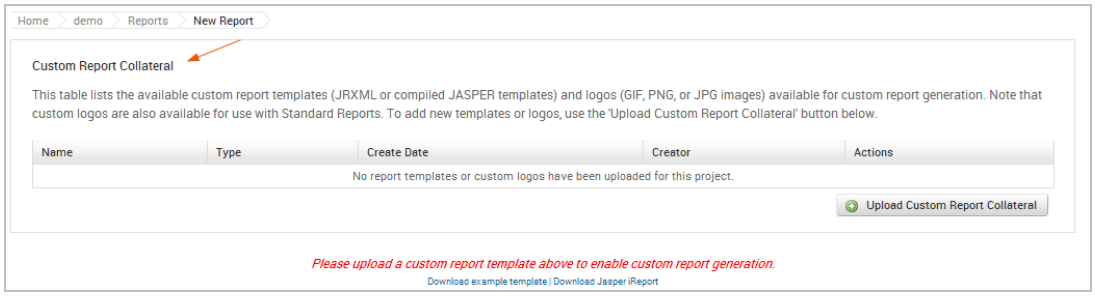


To upload a template:

1. Open the project you want to use to store the custom template.
2. Select **Reports > Create Custom Report** from the Project tab bar.

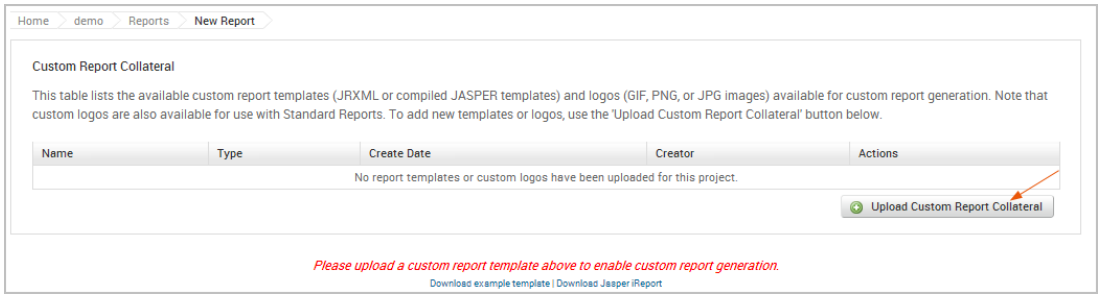
The **Reports** page appears with the Generate Custom Report tab selected.

3. Find the **Custom Report Collateral** area.



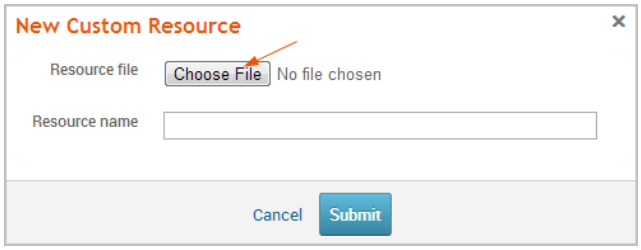
If your project does not contain any templates, the New Custom Report page will not show the form.

4. Click the **Upload Custom Report Collateral** button.



The Upload window appears.

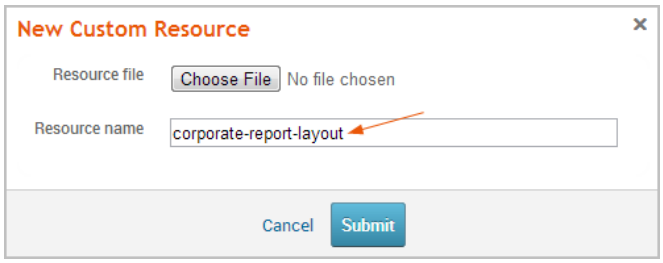
5. Click the **Choose File** button.



The Open Dialog window appears.

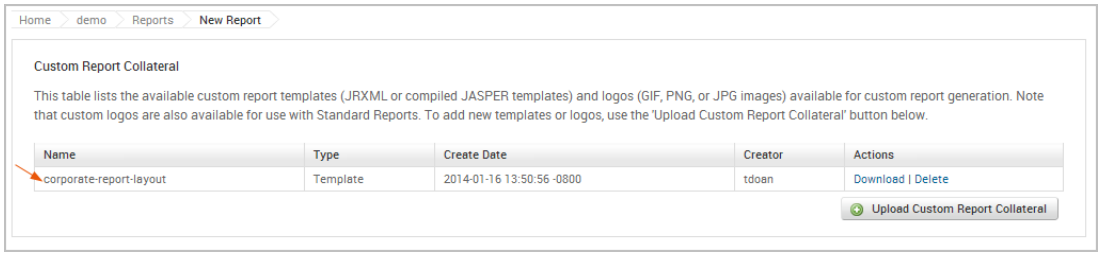
6. Browse to the location of the logo file.
7. Select the template and click the **Open** button.

   The template must have a JRXML extension.
8. Enter a name for the template in the **Descriptive Name** field. (Optional)



   If you do not specify a name, the Custom Report Collateral area shows the original file name.
9. Click the **Submit** button.

   The template appears under the Custom Report Collateral area.



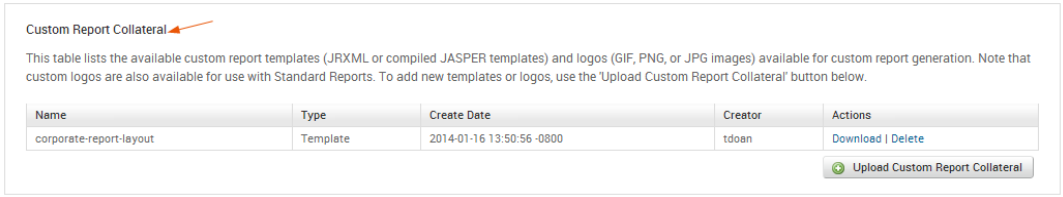
   You are now ready to generate a custom report. For more information on generating custom reports, see *Generating a Custom Report* on page 238.

Downloading a Custom Report Template

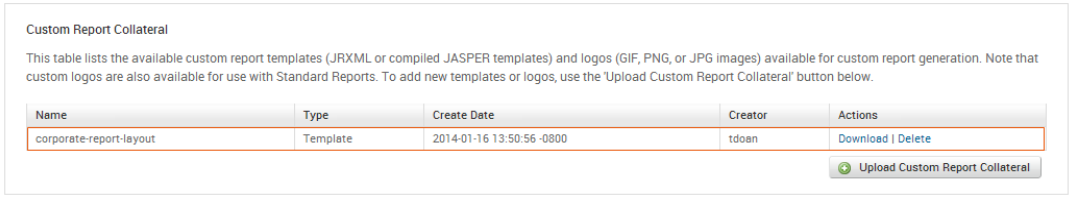
1. Open the project that contains the custom report template that you want to download.
2. Select **Reports > Create Custom Report** from the Project tab bar.

   The **Reports** page appears with the Generate Custom Report tab selected.

3. Find the Custom Report Collateral area.

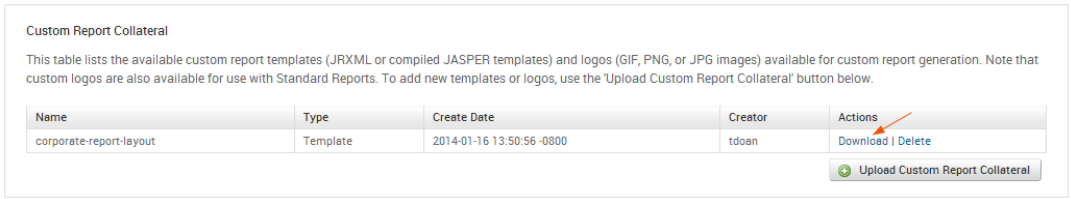


4. Find the row that contains the custom report template you want to download.



The row displays the metadata and the actions that are available for the custom report template.

5. Click the **Download** link.



The download process will automatically start.

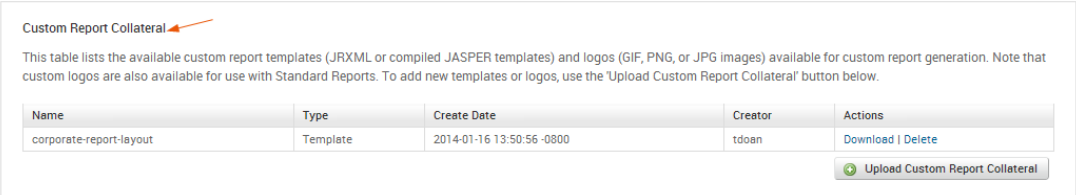
If your browser is not configured to automatically download files, a dialog window will appear and prompt you to save or run the file. You will need to save the template to your computer.

Deleting a Custom Report Template

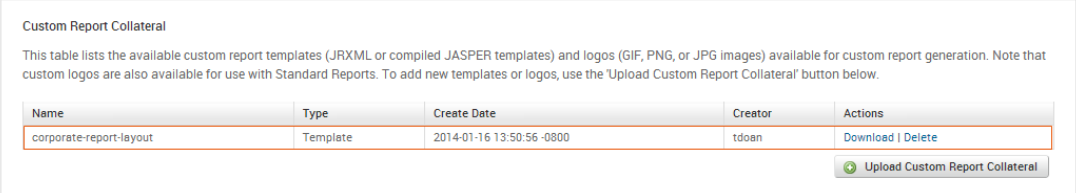
1. Open the project that contains the custom report template that you want to delete.
2. Select **Reports > Create Custom Report** from the Project tab bar.

The **Reports** page appears with the Generate Custom Report tab selected.

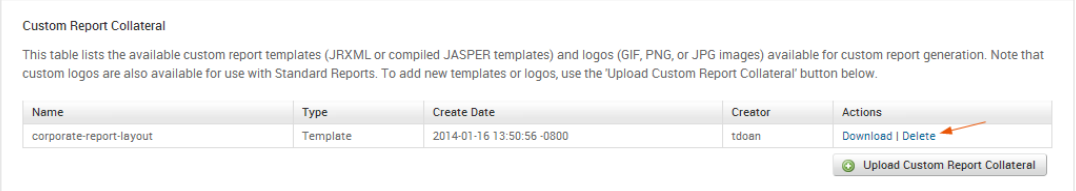
3. Find the Custom Report Collateral area.



4. Find the row that contains the custom report template you want to delete.



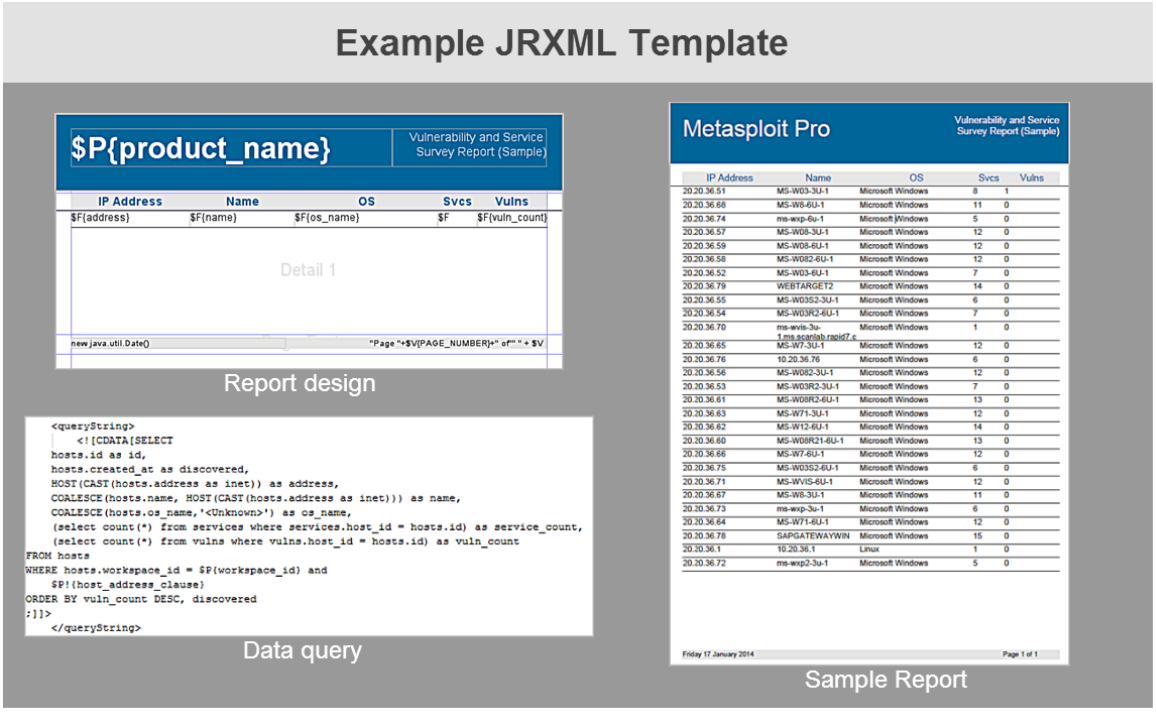
5. Click the **Delete** link.



The browser will prompt you to confirm that you want to delete the custom report template.

Downloading the Example Template

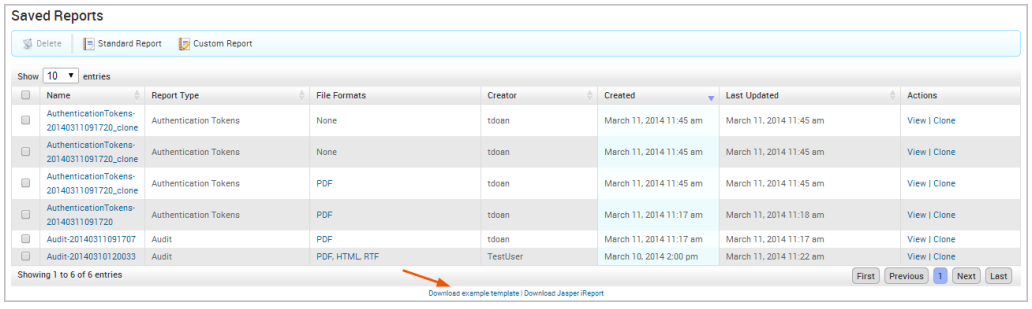
Metasploit Pro provides you with an example template that you can use as a reference when creating your own templates. The template provides simple examples that show you how you can query data, such as host IP addresses, names, operating systems, services counts, and vulnerabilities counts from a project, and display that information in a table. Additionally, you can see examples for adding a title and footer to the report.



To download the example template:

1. Open any project.
2. Select **Reports > Show Reports** from the Project tab bar.

   The **Reports** page appears.
3. Scroll to the bottom of the Reports page.
4. Click the **Download Example Template** link, which is located below the reports table.

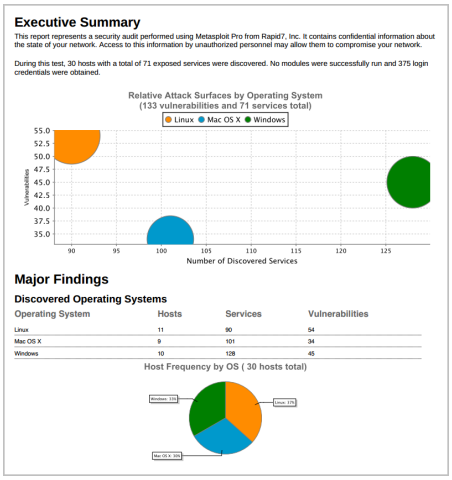


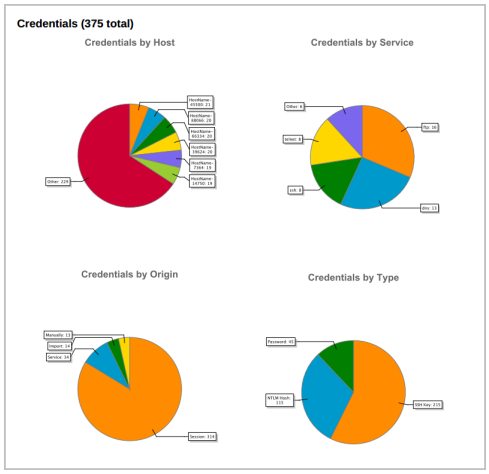
The download process will automatically start.

If your browser is not configured to automatically download files, a dialog window will appear and prompt you to save or run the file. You will need to save the report to your computer.

Audit Report

The Audit Report presents the comprehensive findings for a project. It is useful when you want to obtain a detailed look at targeted hosts in a project. The report data is divided into two sections: Major Findings and Detailed Findings.





Major Findings

The Major Findings section presents host, operating system, and compromised credential data through tables and graphs. Its purpose is to help you quickly summarize and identify important data points in the report.

The Major Findings section includes the following data:

- The potential attack surface based on the number of discovered vulnerabilities and services per operating system.
- A breakdown of the host, service, and vulnerability counts for each operating system.
- The IP address, name, operating system, service count, and vulnerability count for each host in the project.
- The public and private values, realm type, realm value, origin, host count, and service count for each type of credential found in the project. All credentials are grouped according to their type. A credential can be a plaintext password, NTLM hash, non-replayable hash, or SSH key.
- A statistical breakdown of credentials by host, origin, service, and type.

Detailed Findings

The Detailed Findings section provides granular details for each host in the project. It includes the following data:

- The host names and IP addresses of all the targets in the project.
- The details of the credentials stored in the project, such as their public (username), private (password), realm type, realm value, and origin.
- The open services that were discovered on each host.
- The vulnerabilities that were discovered on each host.
- The web vulnerabilities that were discovered on each host.
- The modules that were able to successfully exploit a vulnerability and open a session.
- The activity for each session, such as when it was opened and closed and the commands that were run during the session.

Audit Report Options

Settings	Options
Output formats	PDF, HTML, WORD, RTF
Report options	**Mask discovered credentials** - Removes all credentials, including plain text passwords, hashes, and SSH keys, from the report. The Audit report will display the user name with a blank password. **Include session details** - Shows the details for each session Metasploit Pro was able to open, such as the session type and attack module that Metasploit Pro used to obtain the session. **Include charts and graphs** - Includes visual aids, such as pie graphs, to accompany statistical findings in the report.
Report sections	Executive Summary Compromised Hosts Credentials Discovered OSes Discovered Hosts Host Details Discovered Services Web Sites

Credentials Report

The Credentials Report compiles the credential data, such as plaintext passwords, NTLM hashes, non-replayable hashes, and SSH keys, from a project and presents it in a single unified view. The Credential Report is useful if you want to take a snapshot of the credential data in a project a particular moment in time and export the data in a tangible output, such as a PDF file.

To help you navigate through the data to find key information, the report is organized into the following sections:

- Cover Page
- Project Summary
- Credentials Summary
- Credentials Details
- Login Details
- Host Details
- Module Details
- Appendix

Credential Summary

The Credentials Summary uses pie charts to visualize key findings according to the following categories:

- **Private Types** - The relative distribution of private types for all credentials in the project.
- **Credential Origins** - The relative distribution of credential origins for all credentials in the project.
- **Top Hosts by Logins** - The relative distribution of logins across all hosts in the project.
- **Top Shared Credentials by Related Hosts** - The relative distribution of credential pairs that are most commonly shared between hosts.
- **Logins by OS** - The relative distribution of logins across different operating systems.
- **Logins by Service** - The relative distribution of logins by service name.

Credential Details

The Credential Details presents the granular details of each credential that is stored in the project. Each credential will be grouped by its type: plaintext password, NTLM hash, non-replayable hash, or SSH key.

Each credential will have the following information:

- The public value
- The private value
- The realm type
- The realm value
- The origin
- The count of related hosts
- The count of related services

Login Details

The Login Details shows all validated logins that are related to the selected hosts, or validated logins that are related to all hosts in the workspace, if none are specified.Each login will have the following information:

- The service name
- The host name
- The login creation date
- The access level
- The public data
- The private data

Host Details

The Host Details lists the hosts in the project that have at least one credential or login. Each host will have the following information:

- The host name
- The IP address

- The date the host was added
- The count of logins for the host
- The number of credentials related to the host there were obtained from a login, service authentication, or looting a session

Module Details

The Module Details lists the modules that were used to obtain credentials. This section is divided into two parts: Service Origins and Session Origins.

Service Origins

The Services Origins section lists the modules that were used to authenticate to services to obtain credentials. These credentials are typically obtained by Bruteforce Guess, Credential Reuse, or Get Session.

Each module will have the following information:

- The module name
- The service name
- The number of logins related to the credential that was added by the module
- The date and time that the credential was added to the project. A credential is added when service authentication is successful.

Session Origins

The Session Origins section lists the modules that were used to obtain a session and then used to loot credentials from the compromised host.

Each module will have the following information:

- The module name
- The date and time the session was opened
- The number of credentials obtained with the module
- The number of logins that are related to the credentials that were gathered by the module

Appendix

The Appendix provides additional details about the Credentials Report, such as the options that were used to generate the report and the glossary of key terms.

Credentials Report Options

Settings	Options
Output formats	PDF, HTML, WORD, RTF
Report options	**Mask discovered credentials** - Masks all credentials, including plain text passwords, hashes, and SSH keys, from the report. It replaces the private value with *MASKED*. **Include charts and graphs** - Includes visual aids, such as pie graphs, to accompany statistical findings in the report.
Report sections	Cover Page Project Summary Credentials Summary Credentials Details Plaintext Passwords NTLM Hashes Non-replayable Hashes SSH Keys Login Details Host Details Module Details Service Origins Session Origins Appendix: Glossary Appendix: Report Options

FISMA Compliance Report

The Federal Information Security Management Act (FISMA) provides a comprehensive framework that helps federal agencies implement processes and system controls that protect the security of data and information systems. FISMA is based on a set of standards and recommendations from technology agencies like the National Institute of Standards and Technology (NIST). NIST develops standards and guidelines, like the Special Publication 800-53 revision 4 (SP800-53r4), that federal agencies can use to build their FISMA compliance programs. The guide developed by NIST defines the minimum requirements for managing, operating, controlling, and operating information systems.

The FISMA Compliance Report attempts to help you assess where an organization stands in terms of compliance with specific FISMA requirements. Metasploit Pro reports findings for select requirements from the following families and security controls:

- **Access Control** - AC7
- **Awareness and Training** - AT-2
- **Configuration Management** - CM-7
- **Identification and Authentication** - IA-2, IA-5, and IA-7
- **Risk Assessment** - RA-5
- **System and Information Integrity** - SI-2 and SI-10

The report presents compliance results by indicating a pass or fail status for each FISMA requirement. The findings should be used as an appendix for FISMA requirements testing and not as an actual audit. For more information on each of these requirements, visit the National Vulnerability Database: http://web.nvd.nist.gov/view/800-53/Rev4.

To help you navigate through the data to find key information, the report is organized into the following sections:

Executive Summary

The Executive Summary lists the pass or fail status for each FISMA requirement that Metasploit Pro tests.

Detailed Findings

The Detailed Findings section provides the technical details for each FISMA requirement that Metasploit Pro reports on. The FISMA Compliance report will list each host that did not meet the criteria defined for each requirement.

FISMA Requirement AC-7

FISMA Requirement AC-7 mandates an enforced limit on the number of invalid login attempts made by a user. This requirement dictates that this rate be set by each organization based on their security policy. However, for the purposes of this report, a host will fail this requirement if it has more than 3 failed logins within 60 seconds for a particular public. This rate is considered a reasonable default.

For each host that failed this requirement, this section reports the following information:

- The IP address and name of the host on which the login attempts were made
- The operating system running on the host
- The public value, private type, private value, origin type, and origin detail for each credential that resulted in more than 3 failed logins within 60 seconds of each other

FISMA Requirement AT-2

FISMA Requirement AT-2 mandates that security awareness training is provided to system users. The contents of the training program should be developed by the organization based on its needs and requirements. A host will fail this requirement if it has a vulnerability that was successfully exploited.

For each host that failed this requirement, this section reports the following information:

- The IP address and name of the host
- The operating system running on the host
- The vulnerability that was discovered, the module that was used to exploit the vulnerability, and the timestamp for when the exploit occurred

FISMA Requirement CM-7

FISMA Requirement CM-7 mandates that each host should have one primary function. A host will fail this requirement if it is running more than one major service, such as HTTP, HTTPS, DNS, FTP, MySQL, Postgres, DB2, and MSSQL. However, an exception to this requirement occurs when a host is running both HTTP and HTTPS. Since both services are often exposed together to support an application, they are allowed to run on the same host.

For each host that failed this requirement, this section reports the following information:

- The IP address and name of the host
- The operating system running on the host
- The major services running on the host

FISMA Requirement IA-2

FISMA Requirement IA-2 mandates that the host uniquely identifies and authenticates users. A host will fail this requirement it allowed a valid login using a common username, such as user, root, administrator, admin, tomcat, cisco, manager, sa, postgres, or guest. A host will also fail this requirement if a blank password was used to successfully authenticate to a service.

For each host that failed this requirement, this section reports the following information:

- The IP address and name of the host on which the login was made
- The operating system running on the host
- The public value, private type, private value, origin type, and origin detail for the credentials used

FISMA Requirement IA-5

FISMA Requirement IA-5 mandates that system authenticators, such as passwords and tokens, are properly created, distributed, and managed. This requirement ensures that authenticators are not shipped with default authentication credentials and enforce minimum password requirements. A host will fail this requirement if it allowed a valid login using a common username, such as user, root, administrator, admin, tomcat, cisco, manager, sa, postgres, or guest. A host will also fail this requirement if a blank password was used to successfully authenticate to a service.

For each host that failed this requirement, this section reports the following information:

- The IP address and name of the host on which the login was made
- The operating system running on the host
- The public value, private type, private value, origin type, and origin detail for the credentials used

FISMA Requirement IA-7

FISMA Requirement IA-7 mandates that mechanisms for authentication use acceptable cryptographic methods. A host will fail this requirement if it has any of the following services open: telnet, shell, rexec, rlogin, or POP3. A host will also fail this requirement if it is a Cisco device that has an open HTTP service.

For each host that failed this requirement, this section reports the following information:

- The IP address and name of the host
- The operating system running on the host
- The major services running on the host

FISMA Requirement RA-5

FISMA Requirement RA-5 mandates that vulnerability scans are performed regularly. A host will fail this requirement if it has a vulnerability that was successfully exploited.

For each host that failed this requirement, this section reports the following information:

- The IP address and name of the host
- The operating system running on the host
- The vulnerability that was discovered, the module that was used to exploit the vulnerability, and the timestamp for when the exploit occurred

FISMA Requirement SI-2

FISMA Requirement SI-2 mandates that all systems that have security flaws must be reported. All known vulnerabilities must have the latest vendor security patches applied. A host will fail this requirement if it has a vulnerability that was successfully exploited.

For each host that failed this requirement, this section reports the following information:

- The IP address and name of the host
- The operating system running on the host
- The vulnerability that was discovered, the module that was used to exploit the vulnerability, and the timestamp for when the exploit occurred

FISMA Requirement SI-10

FISMA Requirement SI-10 mandates that the syntax and semantics of information system inputs match the specified definitions for format and content. A host will fail this requirement if it has a vulnerability that was successfully exploited.

For each host that failed this requirement, this section reports the following information:

- The IP address and name of the host
- The operating system running on the host
- The vulnerability that was discovered, the module that was used to exploit the vulnerability, and the timestamp for when the exploit occurred

FISMA Compliance Report Options

Settings	Options
Output formats	PDF, HTML, WORD, RTF
Report options	**Mask discovered credentials** - Masks all credentials, including plain text passwords, hashes, and SSH keys, from the report. The FISMA Compliance Report will replace the password with `*BLANK*`.
Report sections	Executive Summary Detailed Findings

PCI Compliance Report

The PCI Compliance Report presents your findings based on Payment Card Industry Data Security Standard (PCI-DSS) 2.0 requirements, which represent a common set of industry tools and measurements that help ensure the safe handling of cardholder data. The PCI-DSS consists of 12 overall requirements, which are logically organized into the following groups:

1. Building and maintaining a secure network

2. Protecting cardholder data

3. Maintaining a vulnerable management program

4. Implementing strong access control measures

5. Monitoring and testing networks regularly

6. Maintaining an information security policy

The PCI Compliance Report describes where an organization stands in terms of compliance with PCI-DSS requirements related to groups 1, 3, and 4. The report provides coverage for a select subset of requirements within each group. It outlines the target's status for using default vendor settings, applying the latest security patches, and implementing strong user and password policies. The report presents compliance results by indicating a pass or fail status for each PCI-DSS requirement. The findings should be used as an appendix for PCI requirements testing and not as an actual audit.

To help you navigate through the data to find key information, the report is organized into the following sections:

- Executive Summary
- Requirements Status Summary
- Host Status Summary
- Detailed Findings

Executive Summary

The Executive Summary briefly describes the contents of the report.

Requirements Status Summary

The Requirements Status Summary presents a pass or fail status for the following PCI-DSS requirements:

- 2.2.1 - The organization implements only one primary function per server to prevent functions that require different security levels from co-existing on the same server.
- 2.3 - The organization encrypts all non-console administrative access such as browser or web-based management tools.
- 6.1 - The organization ensures that all system components and software have the latest vendor supplied security patches installed. Deploy critical patches within a month of release.
- 8.2 - The organization employs at least one of these to authenticate all users: password or passphrase or two-factor authentication.
- 8.4 - The organization renders all passwords unreadable for all system components both in storage and during transmission using strong cryptography based on approved standards.
- 8.5 - The organization ensures proper user authentication and password management for non-consumer users and administrators on all system components.
- 8.5.8 - The organization does not use group, shared, or generic accounts and passwords, or other authentication methods.
- 8.5.10 - The organization requires a minimum password length of at least seven characters.
- 8.5.11 - The organization uses passwords containing both numeric and alphabetic characters.

Host Status Summary

The Host Status Summary presents the pass or fail results for each host in the project. A host will have a pass status if it passes every PCI-DSS requirement that Metasploit Pro reports on; otherwise, it will have a fail status.

Detailed Findings

The Detailed Findings section provides the technical details for each FISMA requirement. For each FISMA requirement, the report lists each host that did not meet the criteria set by each standard.

PCI Requirement 2.2.1

This requirement mandates that hosts should only have one primary function. Each function should be implemented on separate servers. This section lists the hosts that have more than one listening service

defined as a major system component.

For each host that failed this requirement, this section reports the following information:

- The host IP address and name
- The operating system running on the host
- The services and ports that were discovered on the host

PCI Requirement 2.3

This requirement mandates that all non-console administrative access, such as Telnet and rlogin, be encrypted using strong cryptography, such as SSH or SSL. This section lists the hosts that do not enforce strong encryption methods or have HTTP listening on Cisco devices.

For each host that failed this requirement, this section reports the following information:

- The host IP address and name
- The operating system running on the host
- The services and ports that were discovered on the host

PCI Requirement 6.1

This requirement mandates that all known vulnerabilities must have the latest vendor security patches applied. This section displays all hosts that have exploitable vulnerabilities.

For each host that failed this requirement, this section reports the following information:

- The host IP address and name
- The operating system running on the host
- The services and ports that were discovered on the host

PCI Requirement 8.2

This section displays hosts that do not use password authentication or two-factor authentication. By failing this requirement, the target indicates that it does not enforce passwords/passphrases or authentication via token device.

For each credential that failed this requirement, this section reports the following information:

- The host IP address and name
- The operating system running on the host

- The public value, private type, private value, origin type, and origin detail for the credential

PCI Requirement 8.4

This requirement mandates that passwords should be encrypted during storage.This section displays hosts that have private data stored for validated logins.

For each credential that failed this requirement, this section reports the following information:

- The host IP address and name
- The operating system running on the host
- The public value, private type, private value, origin type, and origin detail for the credential

PCI Requirement 8.5.8

This requirement mandates that generic usernames are not used. This section displays the credentials that have the any of the following usernames: user, root, administrator, admin, tomcat, cisco, manager, sa, postgres, or guest.

For each credential that failed this requirement, this section reports the following information:

- The host IP address and name on which the credential was validated
- The operating system running on the host
- The public value, private type, private value, origin type, and origin detail for the credential

PCI Requirement 8.5.10

This requirement mandates that all passwords have a minimum character length of at least seven characters. This section displays validated passwords that contain less than seven characters .

For each credential that failed this requirement, this section reports the following information:

- The host IP address and name on which the credential was validated
- The operating system running on the host
- The public value, private type, private value, origin type, and origin detail for the credential

PCI Requirement 8.5.11

This requirement mandates that passwords contain both numeric and alphabetic characters. This section displays validated passwords that do not contain both alphabetic and numeric characters.

For each credential that failed this requirement, this section reports the following information:

- The host IP address and name on which the credential was validated
- The operating system running on the host
- The public value, private type, private value, origin type, and origin detail for the credential

PCI Compliance Report Options

Settings	Options
Output formats	PDF, HTML, WORD, RTF
Report options	**Mask discovered credentials** - Masks all credentials, including plain text passwords, hashes, and SSH keys, from the report. The PCI Compliance report will replace the password with `*BLANK*`.
Report sections	Executive Summary Requirements Status Summary Host Status Summary Detailed Findings

Credentials Domino MetaModule Report

The Credentials Domino MetaModule performs an iterative credentials-based attack to identify the attack routes that are possible when a session is obtained on or a credential is captured from a particular host. It helps you identify the targets that can be successfully compromised and the additional credentials that can be captured by leveraging a particular credential or session, and it presents the results of the attack in the Credentials Domino MetaModule Report. You can generate the report to provide a record of the results of the attack in a tangible output, such as a PDF file.

To help you navigate through the data to find key information, the report is organized into the following sections:

- Cover Page
- Executive Summary
- Project Summary
- Run Summary
- Findings Summary
- Summary Charts
- Compromised High Value Hosts
- Uncompromised High Value Hosts
- All Compromised Hosts
- All Uncompromised Hosts
- Appendix

Executive Summary

The Executive Summary provides a high-level recap of the findings from the MetaModule run, which includes the number of hosts that were targeted, the number of hosts that were compromised, and the number of high value hosts that were compromised.

Project Summary

The Project Summary lists the project name and the user who generated the report.

Run Summary

The Run Summary lists the runtime data for Credentials Domino MetaModule.

It includes the following data:

- **Runtime** - The total runtime for the Credentials Domino MetaModule.
- **Iterations** - The total number of iterations the Credentials Domino MetaModule performed.
- **Initial host** - The host that has the login or session that the Credentials Domino MetaModule used to start the attack.
- **Entry point** - The login or session that the Credentials Domino MetaModule used to start the attack.

Findings Summary

The Findings Summary provides an overview of the data captured by the Credentials Domino MetaModule.

It includes the following data:

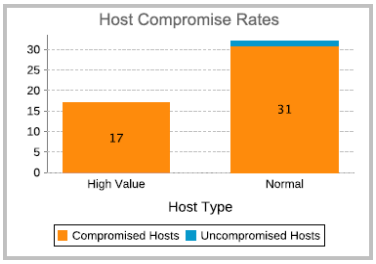
- **Hosts selected** - The number of target hosts selected for the attack.
- **High Value Hosts** - The number of High Value Hosts targeted during the attack.
- **Credentials captured** - The total number of credentials collected from the attack.
- **Hosts compromised** - The total number and percentage of hosts on which a session was opened during the attack.
- **High Value Hosts compromised** - The total number and percentage of High Value Hosts on which a session was opened during the attack.

Summary Charts

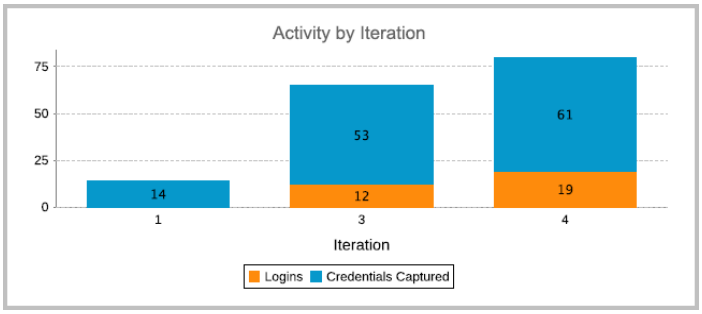
The Summary Charts section presents a graphical breakdown of the compromise rates based on hosts and services and the activity for each iteration.

This section displays the following graphs:

- **Host Compromise Rates** - Shows the relative distribution of hosts that were compromised. This graph displays findings for High Value Hosts and normal hosts.



- **Activity by Iteration** - Shows the number of logins and credentials that were captured during each iteration of the attack.



Compromised High Value Hosts

High Value Hosts identify critical hosts in an organization, such as domain controllers and servers that contain sensitive financial information. If you designated High Value Hosts when you configured the Credentials Domino MetaModule, these hosts will be included in the Compromised High Value Hosts section and will be highlighted in the report with a bold red tag. This Compromised High Value Hosts section presents the granular details for each High Value Host on which the MetaModule was able to successfully open a session and capture credentials.

The following information is included for each host:

- The host name
- The host IP address
- The host operating system
- The service that the MetaModule targeted
- The date and time the MetaModule was able to access the target
- The total number of captured credentials
- The compromise chain, which chronologically lists the series of hosts that were compromised in order to access the current host.

Uncompromised High Value Hosts

This Uncompromised High Value Hosts section presents the granular details for each High Value Host on which the MetaModule was unable to open a session.

The following information is included for each host:

- The host name
- The host IP address
- The host operating system
- The date and time the MetaModule attempted to access the target

All Compromised Hosts

The All Compromised Hosts section lists all hosts on which the MetaModule was able to successfully open a session and capture credentials.

The following information is included for each host:

- The host name
- The host IP address
- The host operating system
- The service that the MetaModule targeted
- The total number of captured credentials
- High Value Host designation

All Uncompromised Hosts

The All Uncompromised Hosts section lists all hosts on which the MetaModule was unable to open a session, and therefore, was unable to collect credentials.

The following information is included for each host:

- The host name
- The host IP address
- The host operating system

- The service that the MetaModule targeted
- The date and time the MetaModule attempted to access the target
- High Value Host designation

Appendix

The Appendix provides additional details about the Credentials Domino MetaModule Report, such as the options that were used to generate the report.

Credentials Report Options

Output formats	PDF, HTML, WORD, RTF
Report options	**Include charts and graphs** - Includes visual aids, such as pie graphs, to accompany statistical findings in the report.
Report sections	The Credentials Domino MetaModule Report includes the following sections: Cover Page, Executive Summary, Project Summary, Run Summary, Findings Summary, Summary Charts, Compromised High Value Hosts, Uncompromised High Value Hosts, All Compromised Hosts, All Uncompromised Hosts, and Appendix.
MetaModule Options	Lists the options that were configured for the MetaModule run, including the Maximum iterations, Overall timeout, Timeout per service, Included hosts, Excluded hosts, and High Value Hosts.

Known Credentials Intrusion Report

The Known Credentials Intrusion Report presents the results from using all the credentials in a project against targeted hosts and services.

To help you navigate through the data to find key information, the report is organized into the following sections:

- Project Summary
- Findings Summary
- Authenticated Services and Hosts Summary Charts
- Authenticated Services and Hosts Summary Details
- Appendix

Project Summary

The Project Summary shows the project name and the user who generated the report.

Findings Summary

The Findings Summary lists the following information:

- **MetaModule** - The MetaModule that was run.
- **Runtime** - The total duration of the MetaModule run.
- **Hosts selected** - The total number of hosts that were selected as targets for the MetaModule.
- **Hosts tried** - The total number of hosts that the MetaModule attempted to authenticate to.
- **Sessions opened** - The total number of sessions that the MetaModule opened on all targets.

Authenticated Services and Hosts Summary Charts

The Authenticated Services and Hosts Summary Charts section uses pie charts to visualize the relative distribution of the top five hosts on which the MetaModule was able to open sessions.

The following image shows the Authentication Services and Hosts Summary Charts:

Authenticated Services and Hosts Details

The Authenticated Services and Hosts Details section lists the technical details for each target on which the MetaModule was able to open a session. The report organizes targets by host name and lists the session information under each host.

Each host will have the following information:

- The timestamp for when the host was added to the project
- The type of session that was established between Metasploit and the target.
- The timestamp for when the session was opened.
- The timestamp for when the session was closed.

Appendix

The Appendix provides additional details about the Known Credentials Intrusion Report, such as the options that were used to generate the report.

Report Options

Output formats	PDF, HTML, WORD, RTF
Report options	**Mask discovered credentials** - Masks all credentials from the report. It replaces the private with *MASKED*. **Include charts and graphs** - Includes visual aids, such as pie graphs, to accompany statistical findings in the report.
Report sections	• Cover Page • Project Summary • Findings Summary • Authenticated Services and Hosts Summary Charts • Authenticated Services and Hosts Summary Details • Appendix
Selected services	Lists the services that were selected for the MetaModule to attempt to authenticate to.

Single Password Testing MetaModule Report

The Single Password Testing MetaModule Report presents the results from using a particular username and plaintext password against targeted hosts and services. At a high-level, the report displays graphs to show the relative distribution of the top five hosts and services that were authenticated using the credential pair. The report also includes the technical details for each target that was successfully authenticated to using the username and password.

To help you navigate through the data to find key information, the report is organized into the following sections:

- Project Summary
- Findings Summary
- Authenticated Services and Hosts Summary Charts
- Authenticated Services and Hosts Summary Details
- Appendix

Project Summary

The Project Summary shows the project name and the user who generated the report.

Findings Summary

The Findings Summary lists the following information:

- **MetaModule** - The MetaModule that was run.
- **Runtime** - The total duration of the MetaModule run.
- **Username selected** - The username that the MetaModule used to attempt to authenticate to a target.
- **Password selected** - The password that the MetaModule used to attempt to authenticate to a target.
- **Hosts selected** - The total number of hosts that were selected as targets for the MetaModule.
- **Services selected** - The total number of services that were targeted.
- **Credentials selected** - The total number of credentials that were provided for the MetaModule run.
- **Successful logins** - The total number of logins that the MetaModule was able to establish.

Authenticated Services and Hosts Summary Charts

The Authenticated Services and Hosts Summary Charts section uses pie charts to visualize the relative distribution of the top five hosts that the MetaModule was able to authenticate to and the top five services that the MetaModule was able to find logins for using the provided username and password.

The following image shows the Authentication Services and Hosts Summary Charts:

Authenticated Services and Hosts Details

The Authenticated Services and Hosts Details section lists the technical details for each target on which the MetaModule was able to authenticate with the provided username and password. Targets are listed by host name. For each host, the report lists the services that the MetaModule was able to successfully authenticate to and the details for each service.

Each service will have the following information:

- The port number
- The protocol
- The service name
- The timestamp for the login attempt
- The result of the login
- The access level for the login

Appendix

The Appendix provides additional details about the Single Password Testing Report, such as the options that were used to generate the report.

Report Options

Output formats	PDF, HTML, WORD, RTF
Report options	**Mask discovered credentials** - Masks all credentials from the report. It replaces the private with *MASKED*.

	Include charts and graphs - Includes visual aids, such as pie graphs, to accompany statistical findings in the report.
Report sections	• Cover Page • Project Summary • Findings Summary • Authenticated Services and Hosts Summary Charts • Authenticated Services and Hosts Summary Details • Appendix
Services selected	Lists the services that were selected for the MetaModule to attempt to authenticate to.

SSH Key Testing MetaModule Report

The SSH Key Testing MetaModule Report presents the results from using a particular username and SSH key against targeted hosts and services. At a high-level, the report displays graphs to show the relative distribution of the top five hosts and services that were authenticated using the username and SSH key. The report also includes the technical details for each target that was successfully authenticated to using the SSH key and username.

To help you navigate through the data to find key information, the report is organized into the following sections:

- Project Summary
- Findings Summary
- Authenticated Services and Hosts Summary Charts
- Authenticated Services and Hosts Summary Details
- Appendix

Project Summary

The Project Summary shows the project name and the user who generated the report.

Findings Summary

The Findings Summary lists the following information:

- **MetaModule** - The MetaModule that was run.
- **Runtime** - The total duration of the MetaModule run.
- **Username selected** - The username that the MetaModule used to attempt to authenticate to a target.
- **SSH key selected** - The SSH key that the MetaModule used to attempt to authenticate to a target.
- **Hosts selected** - The total number of hosts that were selected as targets for the MetaModule.
- **Services selected** - The total number of services that were targeted.
- **Hosts tried** - The total number of hosts that the MetaModule attempted to authenticate to.
- **Services tried** - The total number of services that the MetaModule attempted to authenticate to.
- **Successful logins** - The total number of logins that the MetaModule was able to establish.

Authenticated Services and Hosts Summary Charts

The Authenticated Services and Hosts Summary Charts section uses pie charts to visualize the relative distribution of the top five hosts that the MetaModule was able to authenticate to and the top five services that the MetaModule was able to find logins for using the provided username and SSH key.

The following image shows the Authentication Services and Hosts Summary Charts:

Authenticated Services and Hosts Details

The Authenticated Services and Hosts Details section lists the technical details for each target on which the MetaModule was able to authenticate with the provided SSH key and username. Targets are listed by host name. For each host, the report lists the services that the MetaModule was able to successfully authenticate to and the details for each service.

Each service will have the following information:

- The port number
- The protocol
- The service name
- The timestamp for the login attempt
- The result of the login
- The access level for the login

Appendix

The Appendix provides additional details about the SSH Key MetaModule Testing Report, such as the options that were used to generate the report.

Report Options

Output formats	PDF, HTML, WORD, RTF
Report options	**Mask discovered credentials** - Masks all credentials from the report. It replaces the private with *MASKED*.

	Include charts and graphs - Includes visual aids, such as pie graphs, to accompany statistical findings in the report.
Report sections	• Cover Page • Project Summary • Findings Summary • Authenticated Services and Hosts Summary Charts • Authenticated Services and Hosts Summary Details • Appendix
Selected services	Lists the services that were selected for the MetaModule to attempt to authenticate to.

Pass the Hash Report

The Pass the Hash Report presents the results from using a particular username and hash against targeted hosts and services. At a high-level, the report displays graphs to show the relative distribution of the top five hosts and services that were authenticated using the credential pair. The report also includes the technical details for each target that was successfully authenticated to using the username and hash.

To help you navigate through the data to find key information, the report is organized into the following sections:

- Project Summary
- Findings Summary
- Authenticated Services and Hosts Summary Charts
- Authenticated Services and Hosts Summary Details
- Appendix

Project Summary

The Project Summary shows the project name and the user who generated the report.

Findings Summary

The Findings Summary lists the following information:

- **MetaModule** - The MetaModule that was run.
- **Runtime** - The total duration of the MetaModule run.
- **Username selected** - The username that the MetaModule used to attempt to authenticate to a target.
- **NTLM hash selected** - The hash that the MetaModule used to attempt to authenticate to a target.
- **Hosts selected** - The total number of hosts that were selected as targets for the MetaModule.
- **Services selected** - The total number of services that were targeted.
- **Hosts tried** - The total number of hosts that the MetaModule attempted to authenticate to.
- **Services tried** - The total number of services that the MetaModule attempted to authenticate to.
- **Successful logins** - The total number of logins that the MetaModule was able to establish.

Authenticated Services and Hosts Summary Charts

The Authenticated Services and Hosts Summary Charts section uses pie charts to visualize the relative distribution of the top five hosts that the MetaModule was able to authenticate to and the top five services that the MetaModule was able to find logins for using the provided username and hash.

The following image shows the Authentication Services and Hosts Summary Charts:

Authenticated Services and Hosts Details

The Authenticated Services and Hosts Details section lists the technical details for each target on which the MetaModule was able to authenticate with the provided username and hash. Targets are listed by host name. For each host, the report lists the services that the MetaModule was able to successfully authenticate to and the details for each service.

Each service will have the following information:

- The port number
- The protocol
- The service name
- The timestamp for the login attempt
- The result of the login
- The access level for the login

Appendix

The Appendix provides additional details about the Pass the Hash Report, such as the options that were used to generate the report.

Report Options

Output formats	PDF, HTML, WORD, RTF
Report options	**Mask discovered credentials** - Masks all credentials from the report. It replaces the private with *MASKED*.

	Include charts and graphs - Includes visual aids, such as pie graphs, to accompany statistical findings in the report.
Report sections	• Cover Page • Project Summary • Findings Summary • Authenticated Services and Hosts Summary Charts • Authenticated Services and Hosts Summary Details • Appendix
Selected services	Lists the services that were selected for the MetaModule to attempt to authenticate to.