Outpost24

Vulnerability Management made easy

Authenticated Scanning using SSH

Configuration Guide

Version: 2.5

2018-07-30

Table of Contents

About This Guide

The main purpose of this document is to provide users a comprehensive overview of the Linux configuration required to succeed with authenticated scans using OUTSCAN or HIAB. This document has been elaborated under the assumption the reader has access to the OUTSCAN/HIAB account and Portal Interface.

For support information, visit https://www.outpost24.com/support

Copyright

Trademark

1 Authenticated Scanning

This guide will provide you with a technical step-by-step procedure in order to succeed with authenticated scanning through SSH, along with the different setups supported within OUTSCAN and HIAB.

1.1 Prerequisites

The targets need to have at least one from the lists configured for ciphers, kex, and macs matching the supported ones on https://www.libssh.org/features/

Option	Description
Ciphers	aes256-ctr aes192-ctr aes128-ctr aes256-cbc aes192-cbc aes128-cbc 3des-cbc blowfish-cbc
Key Exchange Methods	curve25519-sha256@libssh.org ecdh-sha2-nistp256 diffie-hellman-group1-sha1 diffie-hellman-group14-sha1
MAC hashes	hmac-sha2-512 hmac-sha2-256 hmac-sha1 none

sshd_config example:

```
Ciphers aes256-ctr,aes192-ctr

KexAlgorithms curve25519-sha256@libssh.org,ecdh-sha2-nistp256

MACs hmac-sha2-512,hmac-sha2-256
```

1.2 Password Authentication

This form of authentication is the simplest, as it only requires you to specify the username and corresponding password. On Unix/Linux, the username is a usually system-wide username as specified in `/etc/passwd`

To succeed with this authentication, enable the password authentication within the SSHD

configuration on the targeted system, located at `/etc/ssh/sshd_config`. Remove the hashtag before **PasswordAuthentication yes** in the SSHD configuration file.

```
# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Change to no to disable tunneled clear text passwords
PasswordAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosGetAFSToken no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
```

Thereafter, restart the SSH service within the terminal.

1.2.1 Authenticate Against the Target

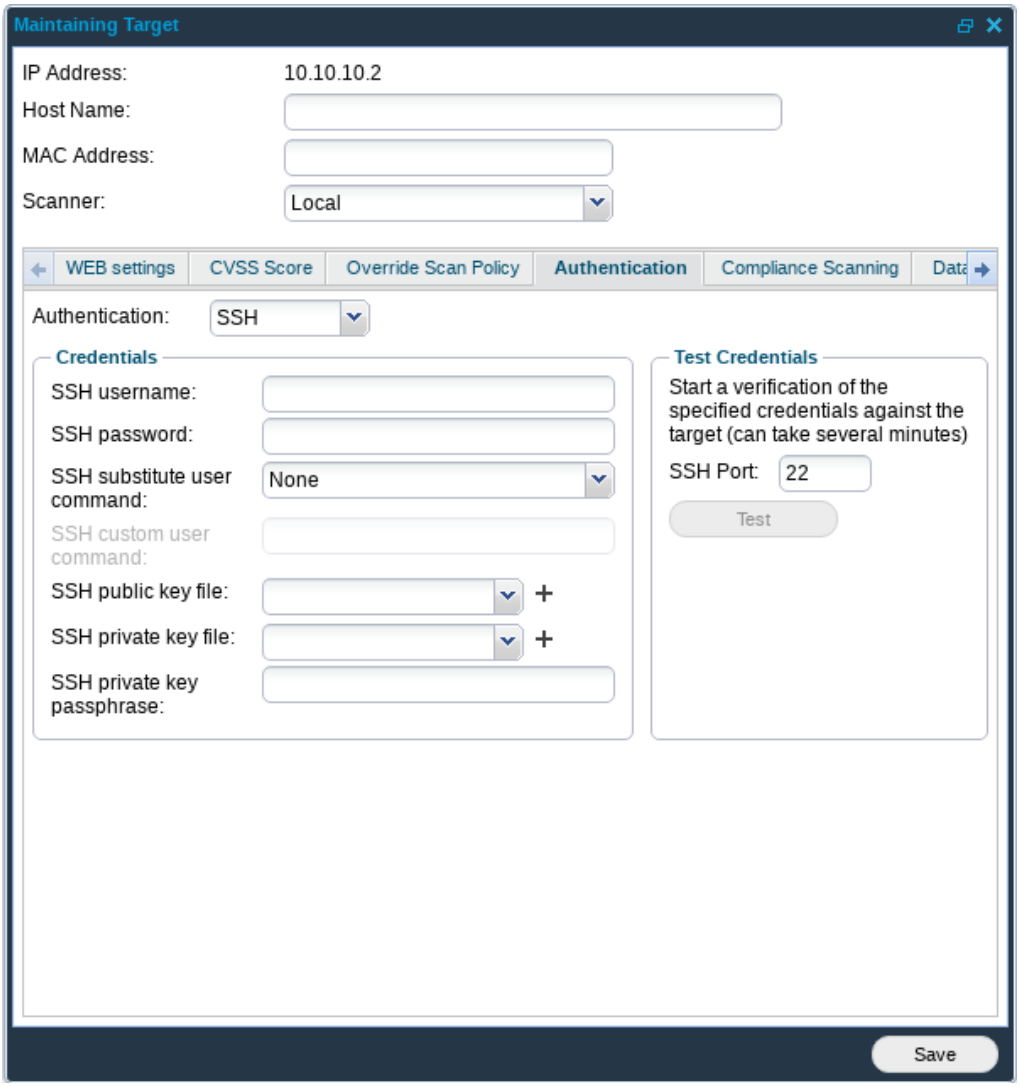
There are three available setups to use authentication against target(s).

- Per Target
- Per Target Group
- Per Scan Policy

1.2.1.1 Per Target

To access the setup for SSH authentication on a specific target:

1. Right click the targets entry within **Manage Targets** and choose **Edit** to display the **Maintaining Target** window.
2. Select **SSH** under **Authentication** Tab.



The necessary Authentication Credentials for password based authentication are:

- **SSH username**: Username used when authenticating against the target
- **SSH password**: Password used when authenticating against the target
- Supported **SSH substitute user command**s (optional):

***Note**: The use of the following commands is to execute commands with a different user/privilege escalation.*

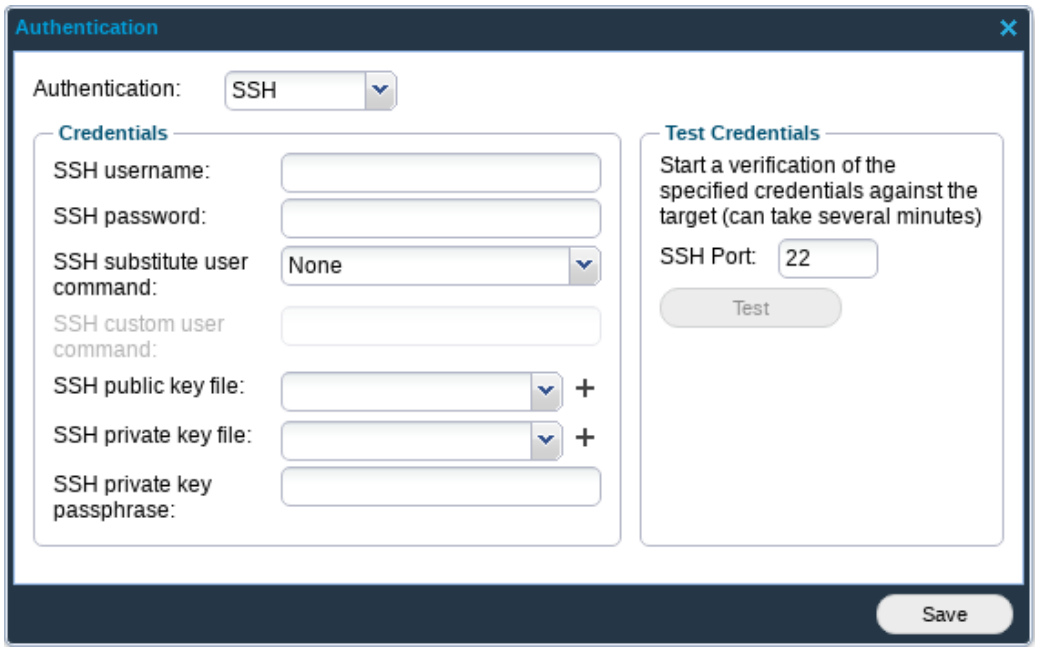
Command	Description
sudo	This command is found in most of the Linux based systems (or can be installed). Used to execute commands as a different user (other than the one used to log in). From the tools perspective, it uses root account to perform the commands.
doas	It is an OpenBSD based command. 95% of its features are similar to sudo. [Link|https://man.openbsd.org/doas]
sesu	It is an IBM implementation of su.
dzdo	Used in Linux/Unix (can be installed at will). An alternative to sudo.
pfexec	Mostly used in Solaris.
custom	It gives a flexibility to use a custom defined privilege escalation command. When this option is selected, a field labeled **SSH custom user command** is ungrayed for typing in the custom command.

Running **Test** under **Test Credentials** performs authentication against the target to verify if the provided credentials are valid, the test will return with **Success** if the authentication was successful.

1.2.1.2 Per Target Group

To access the setup for SSH Authentication for a Target Group:

1. Right click on the **Target Group** entry within **Manage Targets** and choose **Set Target Authentication** to display the **Authentication** window.
2. In the *Authentication* drop-down menu, select **SSH**.



The necessary Authentication Credentials for password based authentication are:

- **SSH username**: Username used when authenticating against the target
- **SSH password**: Password used when authenticating against the target
- Supported **SSH substitute user command**s (optional):

***Note**: The use of the following commands is to execute commands with a different user/privilege escalation.*

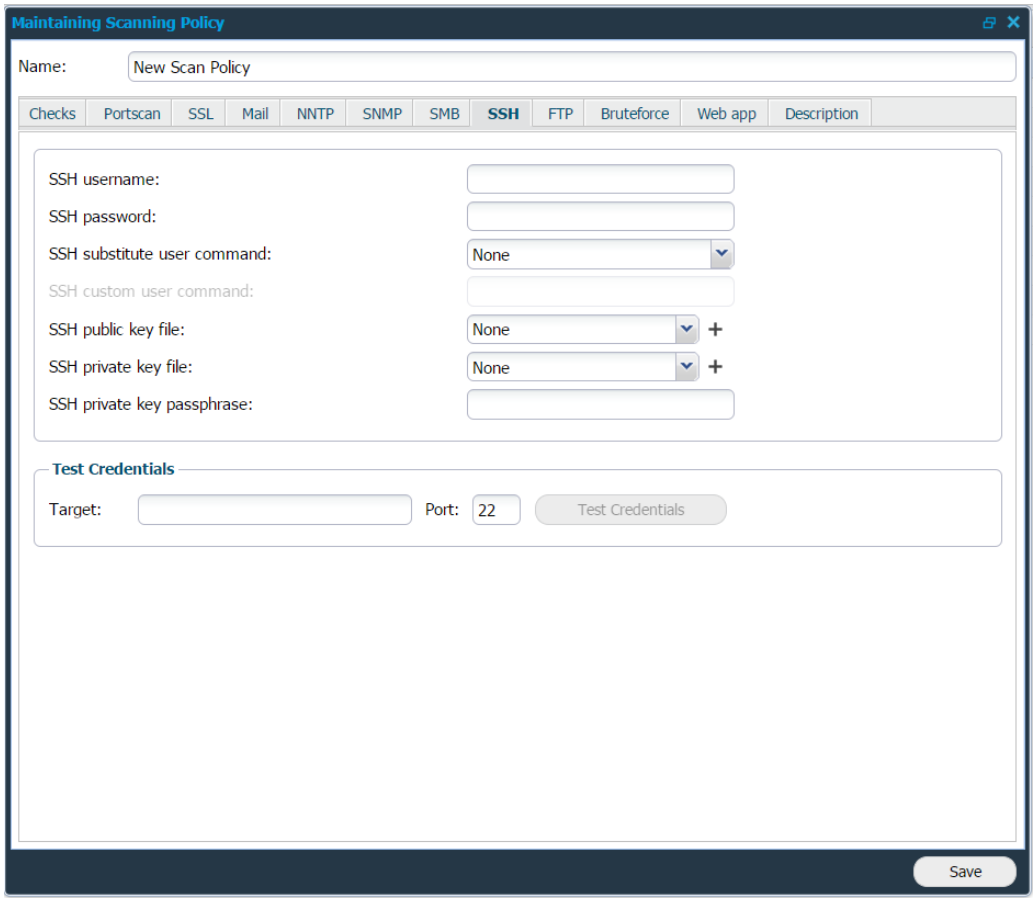
Command	Description
sudo	This command is found in most of the Linux based systems (or can be installed). Used to execute commands as a different user (other than the one used to log in). From the tools perspective, it uses root account to perform the commands.
doas	It is an OpenBSD based command. 95% of its features are similar to sudo. [Link|https://man.openbsd.org/doas]
sesu	It is an IBM implementation of su.
dzdo	Used in Linux/Unix (can be installed at will). An alternative to sudo..
pfexec	Mostly used in Solaris.
custom	It gives a flexibility to use a custom defined privilege escalation command. When this option is selected, a field labeled **SSH custom user command** is ungrayed for typing in the custom command.

Running Test under **Test Credentials** will perform authentication against all targets defined within the **Target Group** to verify if the provided credentials are valid, the test will return with **Success** if the authentication was successful.

1.2.1.3 Per Scan Policy

To access the setup for SSH Authentication for a Scan Policy:

1. Right click the desired entry within the **Scan Policy** Tab in **Scan Scheduling**, or create a new one to display the **Maintaining Scanning Policy** window.
2. Select the **SSH** tab to enter your SSH setup.



The following options are required to succeed with SSH password based authentication.

- **SSH username**: Username used when authenticating against the target
- **SSH password**: Password used when authenticating against the target
- Supported **SSH substitute user command**s (optional):

***Note**: The use of the following commands is to execute commands with a different user/privilege escalation.*

Command	Description
sudo	This command is found in most of the Linux based systems (or can be installed). Used to execute commands as a different user (other than the one used to log in). From the tools perspective, it uses root account to perform the commands.
doas	It is an OpenBSD based command. 95% of its features are similar to sudo. [Link|https://man.openbsd.org/doas]
sesu	It is an IBM implementation of su.
dzdo	Used in Linux/Unix (can be installed at will). An alternative to sudo.
pfexec	Mostly used in Solaris.
custom	It gives a flexibility to use a custom defined privilege escalation command. When this option is selected, a field labeled **SSH custom user command** is ungrayed for typing in the custom command.

Testing credentials against a specific target is performed within the **Test Credentials** section.

1.3 Key-Based Authentication

First generate a public/private keys pair that will identify the user on the server, and choose to protect it with password or not.
No password implies that anyone with access to the key files will have the same level of access, and password will not be asked when establishing a connection to the server. Protecting the keys with password means that every time the user attempts to establish a connection to the server using those keys, a password for decryption will be asked.

To succeed with this authentication, it is required that you specify where the authorized keys file is located within the SSHD configuration on the targeted system, located at `/etc/ssh/sshd_config`.

```
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile            %h/.ssh/authorized_keys
```

Default location is `%h/.ssh/authorized_keys`.

- Once defined, restart the SSH service within the terminal and create the file `authorized_keys` at the defined location.
  ***Note**: The authorized_keys should be a text file, and not directory.*
- Once created, copy the public SSH key previously created and paste this string within the `authorized_keys` file.

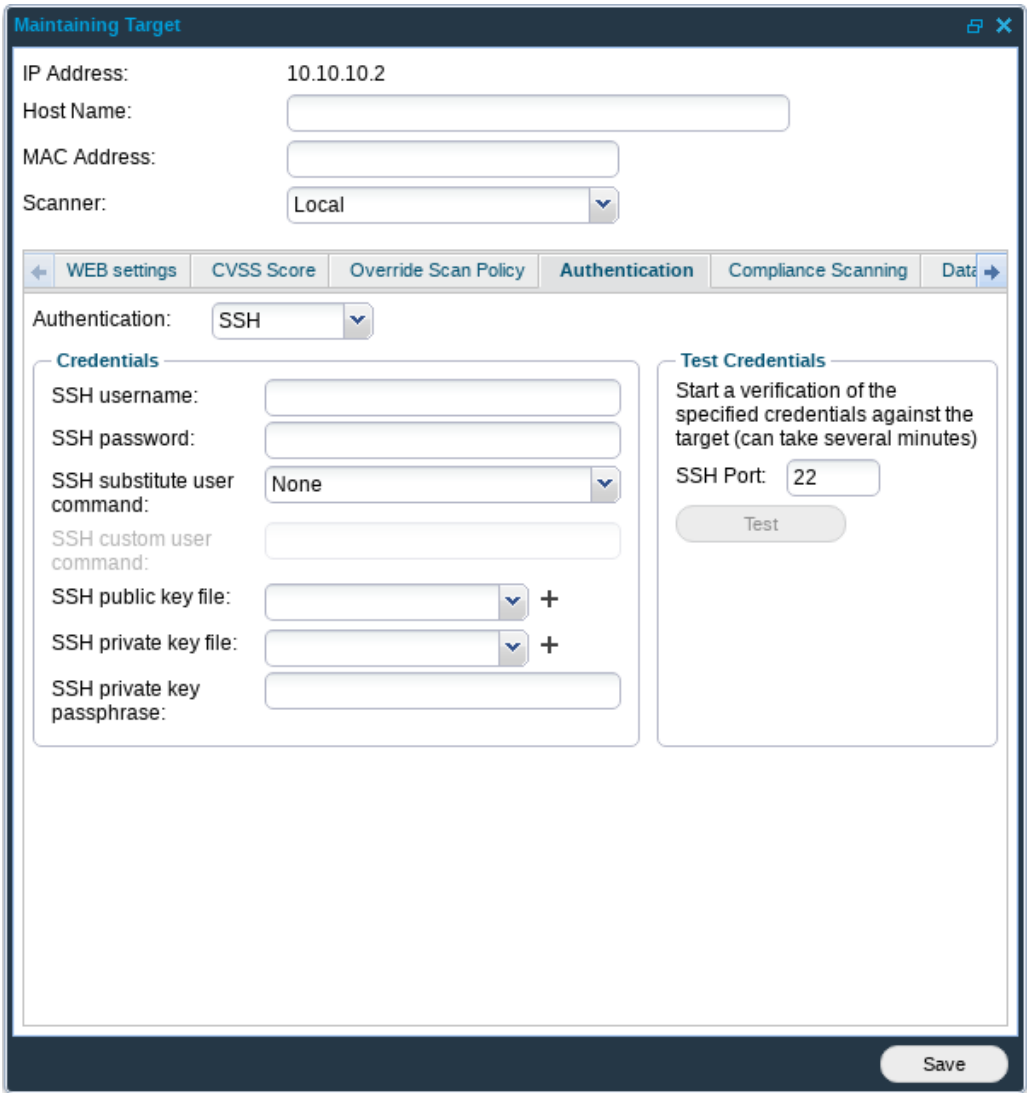
1.3.1 Authenticate against the target

There are three available setups to use authentication against target(s).

1.3.1.1 Per Target

To access the setup for SSH authentication on a specific target:

1. Right click on the target entry within **Manage Targets** and choose **Edit** to display the **Maintaining Target** window.
2. Select **SSH** under **Authentication** Tab.



The following options are required to succeed with SSH Private Key Authentication.

- Supported **SSH substitute user command**s (optional):

***Note**: The use of the following commands is to execute commands with a different user/privilege escalation.*

Command	Description
sudo	This command is found in most of the Linux based systems (or can be installed). Used to execute commands as a different user (other than the one used to log in). From the tools perspective, it uses root account to perform the commands.
doas	It is an OpenBSD based command. 95% of its features are similar to sudo. [Link|https://man.openbsd.org/doas]
sesu	It is an IBM implementation of su.
dzdo	Used in Linux/Unix (can be installed at will). An alternative to sudo.
pfexec	Mostly used in Solaris.
custom	It gives a flexibility to use a custom defined privilege escalation command. When this option is selected, a field labeled **SSH custom user command** is ungrayed for typing in the custom command.

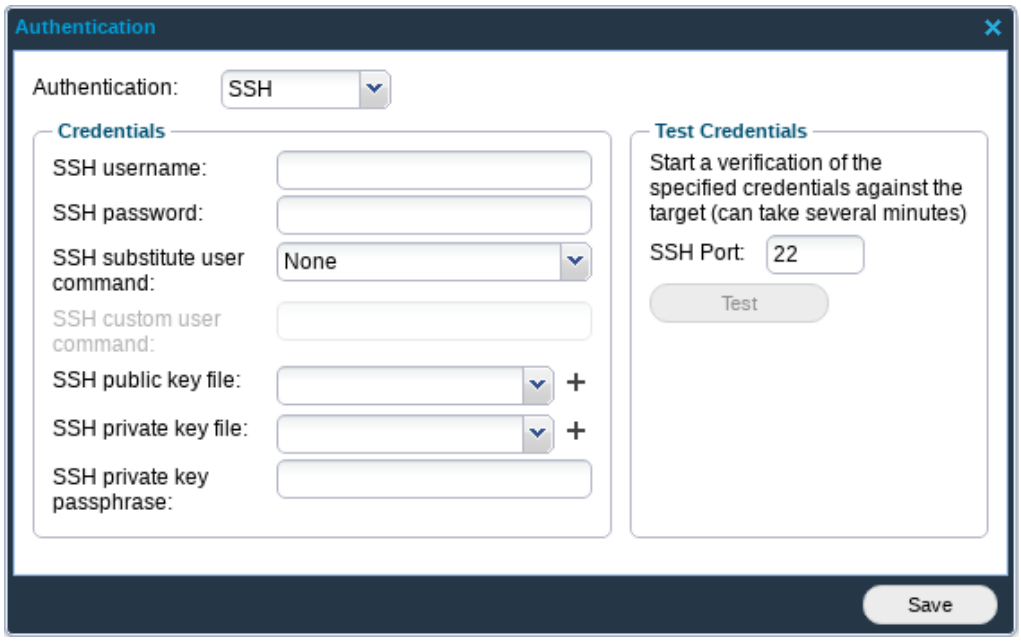
- **SSH public key file**: Provide the scanner with the public key that should be used during the authentication
- **SSH private key file**: Provide the scanner with the private key that should be used during the authentication
- **SSH private key passphrase**: Enter the passphrase for the private key. Can be left blank if the private key has no passphrase.

Running Test under **Test Credentials** will perform authentication against the target to verify if the provided credentials are valid, the test will return with **Success** if the authentication was successful.

1.3.1.2 Per Target Group

To access the setup for SSH Authentication for a Target Group:

1. Right click on the **Target Group** entry within **Manage Targets** and choose **Set Target Authentication** to display the **Authentication** window.
2. Select **SSH** in the drop-down menu.



The following options are required to succeed with SSH Private Key Authentication.

- Supported **SSH substitute user command**s (optional):

***Note**: The use of the following commands is to execute commands with a different user/privilege escalation.*

Command	Description
sudo	This command is found in most of the Linux based systems (or can be installed). Used to execute commands as a different user (other than the one used to log in). From the tools perspective, it uses root account to perform the commands.
doas	It is an OpenBSD based command. 95% of its features are similar to sudo. [Link|https://man.openbsd.org/doas]
sesu	It is an IBM implementation of su.
dzdo	Used in Linux/Unix (can be installed at will). An alternative to sudo.
pfexec	Mostly used in Solaris.
custom	It gives a flexibility to use a custom defined privilege escalation command. When this option is selected, a field labeled **SSH custom user command** is ungrayed for typing in the custom command.

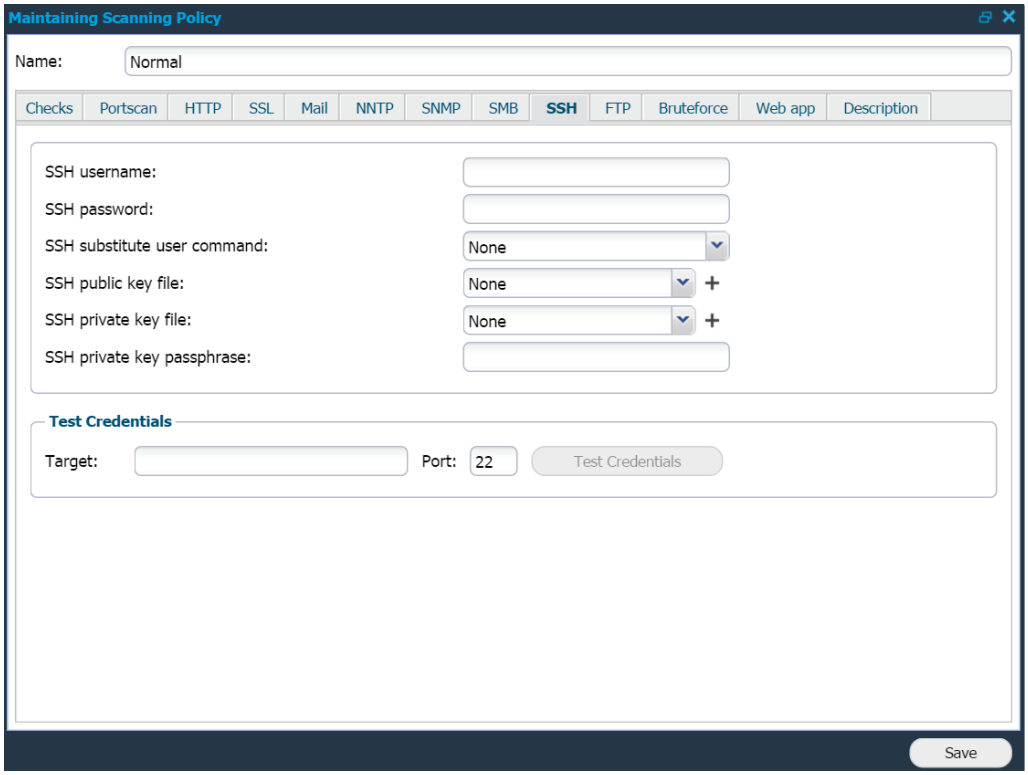
- **SSH public key file**: Provide the scanner with the public key that should be used during the authentication
- **SSH private key file**: Provide the scanner with the private key that should be used during the authentication
- **SSH private key passphrase**: Enter the passphrase for the private key. Can be left blank if the private key has no passphrase.

Running Test under **Test Credentials** will perform authentication against all targets defined within the **Target Group** to verify if the provided credentials are valid, the test will return with **Success** if the authentication was successful.

1.3.1.3 Per Scan Policy

To access the setup for SSH Authentication for a Scan Policy:

1. Right click on the desired entry within **Scan Policy** tab in **Scan Scheduling**, or create a new to display the **Maintaining Scanning Policy** window.
2. Select **SSH** in the drop-down menu and enter your SSH setup.



The following options are required to succeed with SSH Public Key Authentication.

- Supported **SSH substitute user command**s (optional):

***Note**: The use of the following commands is to execute commands with a different user/privilege escalation.*

Command	Description
sudo	This command is found in most of the Linux based systems (or can be installed). Used to execute commands as a different user (other than the one used to log in). From the tools perspective, it uses root account to perform the commands.
doas	It is an OpenBSD based command. 95% of its features are similar to sudo. [Link|https://man.openbsd.org/doas]
sesu	It is an IBM implementation of su.
dzdo	Used in Linux/Unix (can be installed at will). An alternative to sudo.
pfexec	Mostly used in Solaris.
custom	It gives a flexibility to use a custom defined privilege escalation command. When this option is selected, a field labeled **SSH custom user command** is ungrayed for typing in the custom command.

- **SSH public key file**: Provide the scanner with the public key that should be used during the authentication
- **SSH private key file**: Provide the scanner with the private key that should be used during the authentication
- **SSH private key passphrase**: Enter the passphrase for the private key. Can be left blank if the private key has no passphrase.

Testing credentials against a specific target is performed within the **Test Credentials** section.

2 SSH Commands

The following are some of the SSH commands that are run once the scanner has authenticated successfully.

***Note**: The commands list is not exhaustive and is subject to change from time to time. Contact https://www.outpost24.com/support for further details.*

- *find /var/db/pkg/ -mindepth 2 -maxdepth 2 -printf "%P\\n"*
- */usr/sbin/pkg_info -q*
- *xl info*
- *openssl version*
- */bin/rpm -qa --qf '%{NAME}|%{EPOCH}:%{VERSION}|%{RELEASE}|%{SOURCERPM}\n'*
- */bin/rpm -qa --qf '%{NAME} %{EPOCH}:%{VERSION}-%{RELEASE}\n'*
- *apk info -v*
- *sw_vers*
- *cd /tmp; rm -f /tmp/bash_outpost24_wkeqrhqrqpq; env 'x=() { (a)=>\' bash -c "bash_outpost24_wkeqrhqrqpq echo vulnerable"; cat /tmp/bash_outpost24_wkeqrhqrqpq*
- *ls /lib/libkeyutils.so.1.9 /lib64/libkeyutils.so.1.9 2> /dev/null*
- *uname -r*
- *grep '"version"' /opt/google/chrome/resources/chromeos/chromevox/manifest.json*
- */usr/bin/model*
- */usr/bin/dpkg-query -W -f='${Package} |${Source} |${Version} |${Status} \n'*
- *for f in /Applications/*/Contents/Info.plist; do awk -v FILENAME='$f' '/CFBundleShortVersionString/ { getline; print FILENAME $0 }' "$f"; done*
- */bin/rpm -qa --qf '%{NAME} %{EPOCH}:%{VERSION}-%{RELEASE}\n' | grep 'redhat-release-'*
- *convert --version*
- */usr/sbin/swlist -a revision -l fileset*
- *instfix -i*
- */usr/bin/ipcs -pm*
- *ps ax*
- */usr/bin/ftp about:version*
- *for f in /Library/Frameworks/*/Resources/Info.plist; do awk -v FILENAME='$f' '/CFBundleVersion/ { getline; print FILENAME $0 }' "$f"; done*
- *show platform*
- */usr/bin/dpkg-query -W -f='${Package} |${Source} |${Version} |${Status} \n'*
- *pkg info -l | grep FMRI*
- */usr/bin/ldd --version*
- *bash -version*
- */usr/sbin/pkg info -q*
- *docker images --format*

{{.ID}},{{.Repository}},{{.Tag}},{{.Digest}},{{.CreatedSince}},{{.CreatedAt}},{{.Size}}

- *cat /etc/version*
- *cat /etc/gentoo-release*
- *show system info*
- */bin/rpm -Vv keyutils-libs && echo 'SUCCESSFUL COMMAND'*
- *file /opt/google/chrome/chrome 2>&1 | grep -q ELF; echo $?*
- *uname -a*
- *uname -p*
- *xe patch-list*
- *cat /etc/SuSE-release*
- *uname -s*
- *apk info*
- *echo*
- *cat /etc/product*
- *show hostinfo 1*
- */usr/bin/wget --version*
- *show module*
- *show version*
- *showrev -p*
- *cat /etc/slackware-version*
- *cat /etc/ssh/ssh_config*
- *pkgutil --pkgs | grep com.apple.pkg.update.security*
- *bash -c 'f() { if ["$1" == 0]; then return; fi; sed -n "s/^\([%a-z0-9_-]\+\)\s\+.*/\1/p" "$2"; sed -n "s/^\s*User_Alias\s\+\S\+\s*=\s*\(.*\)/\1/p" "$2" | tr "," "\n"; sed -n "s/^#include\s\+\(.*\).*/\1/p" "$2"; while read -r file; do f $(($1 - 1)) "$file"; done < <(sed -n "s/^#include\s\+\(.*\)/\1/p" "$2"); while read -r dir; do local d=$dir; while read -r file; do f $(($1 - 1)) "$d/$file"; done < <(ls -1 "$dir"); done < <(sed -n "s/^#includedir\s\+\(.*\)/\1/p" "$2"); } && f 128 /etc/sudoers | sed "s/^\s*//;s/\s*$//" | sort -fu'*
- *pkg info -l chef*
- *cat /etc/redhat-release*
- *cat /etc/os-release*
- *apk --print-arch*
- *grep '"version"' /usr/share/oem/pepper/flash_installed*
- */bin/rpm -qa --qf '%{NAME} %{EPOCH}:%{VERSION}-%{RELEASE}\n' | grep 'centos-release'*
- *procmail -v*
- *systemd-detect-virt -q; echo $?*
- *cat /proc/self/cgroup*
- *lsb_release -a*
- *cat /etc/release*
- *docker ps --format {{.ID}},{{.Image}},{{.Command}},{{.RunningFor}},{{.Ports}},{{.Status}},{{.Names}},{{.Mounts}},{{.Networks}}*
- *oslevel -r*
- *show ver*

- *env x='() { :;}; echo vulnerable' bash -c "echo this is a test"*
- */usr/bin/dpkg-query -W -f='${package}|${version}|${source:package}|${source:version}|${status}\n'*

2.1 Sudo

To run sudo from Scanner, the following configuration is required within the targets `/etc/sudoers` file:

```
Defaults:username !requiretty
```