Cloud Raxak, Inc.

475 Alberto Way, Suite 130
Los Gatos, CA, 95032
USA

Getting Started with Raxak Protect™

A Step-By-Step Guide to using your Free Trial account on Amazon Web Services

January 2017

Audience

This guide is intended for end users using **Raxak Protect** for the first time. It assumes that you have background and familiarity with the concepts of security profiles, DISA STIGs, and know how to create servers in the cloud or have root access to existing servers (cloud or physical) that you want to configure. For more information on key concepts, see the section below called **More Resources**.

Prerequisites

To create a **Free Trial** account for your use on the Cloud Raxak console, go to https://www.cloudraxak.net. Click on the appropriate button to login with your **Amazon ID**. If this is the first time you are logging in, a free trial account is automatically created for you. For restrictions on the account, see the section below called **Limitations**.

CLOUD RAXAK

amazon web services | Partner Network
TECHNOLOGY PARTNER

Sections in this Guide:

1 Key Concepts: Connecting, Provisioning, Enrolling, Checking, Remediating

Before you use Raxak Protect, there are five key concepts that you should become familiar with.

Connecting

For Raxak Protect to be able to access, check, and remediate your servers, your networks need to be configured to allow Cloud Raxak's servers to reach your server and connect to it over SSH. Raxak Protect will connect over port 22 from a fixed source ID which Cloud Raxak will let you know. Raxak Protect will use SSH key pairs and a fixed **userid** (which you select) for the connection. You need to ensure that the network connectivity is correctly set up before you attempt to use Raxak Protect. We also expect that your servers have public IP addresses that are reachable from the internet. Advanced usages such as VPN are not covered in this document.

Provisioning

The provisioning step prepares your server itself by creating the **userid** and populating it with the appropriate privileges, and the SSH public key that allows Raxak Protect to connect to your server.

Enrolling

The enrolling step sets up the provisioned server in your Raxak Protect account. Once enrolled, you can then check and remediate the server as needed.

Checking

In the checking step, Raxak Protect connects to your server(s) and checks their settings against a selected **profile**. Your initial account provides out-of-the-box access to several profiles that you can use. The default operation of Raxak Protect (unless you change it) is to only check the server for compliance.

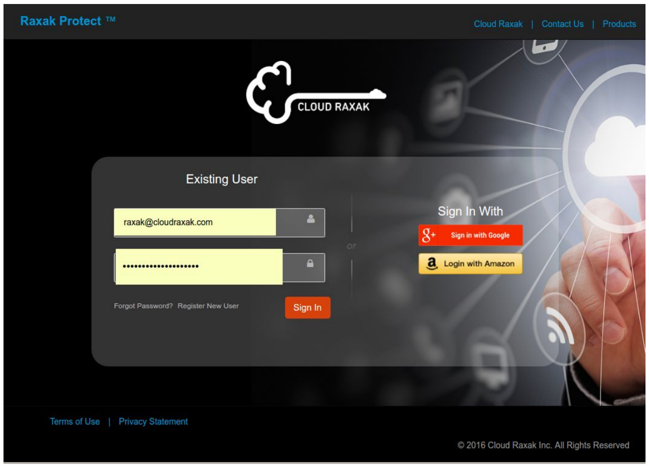
Profiles consist of sets of **rules** each of which specifies a particular OS or application specific configuration parameter and its desired setting. Rules are also tagged by **severity** from **Low** to **High** which indicates the degree of vulnerability that your server is exposed to if the configuration specified by that rule is improperly set.

Remediating

After the checking is done, configuration rules can be flagged as **Successful**, **Failed,** or **Manual.** Successful rules are those whose settings match the desired state specified in the rule. Failed rules are those whose configurations do not match the desired state. Most failed rules can be automatically remediated. Some rules however, require manual interventions--either because the configuration change requires a reboot, or because the rule refers to configuration of systems that are not accessible on the server itself (for example, offsite backup).

2 Logging In to Raxak Protect

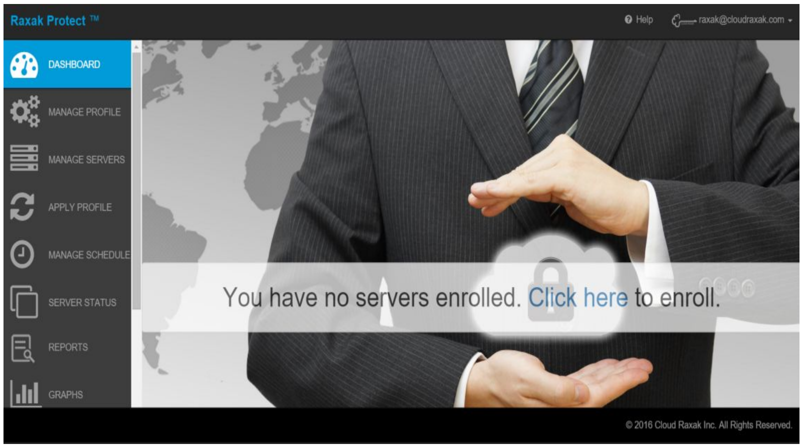
The Raxak Protect Console is accessible at https://www.cloudraxak.net. You will be presented with the following screen.


Raxak Protect ™
Cloud Raxak | Contact Us | Products
CLOUD RAXAK
Existing User
raxak@cloudraxak.com
••••••••••••••••••••
or
Forgot Password? Register New User
Sign In
Sign In With
Sign in with Google
Login with Amazon
Terms of Use | Privacy Statement
© 2016 Cloud Raxak Inc. All Rights Reserved

Click on the **Login with Amazon** button to go to Amazon's standard authentication screen. If this is the first time you are using Raxak Protect, Amazon will ask you to approve the tool's access to your basic information such as name and email address.

Log in with the password for your Amazon ID.

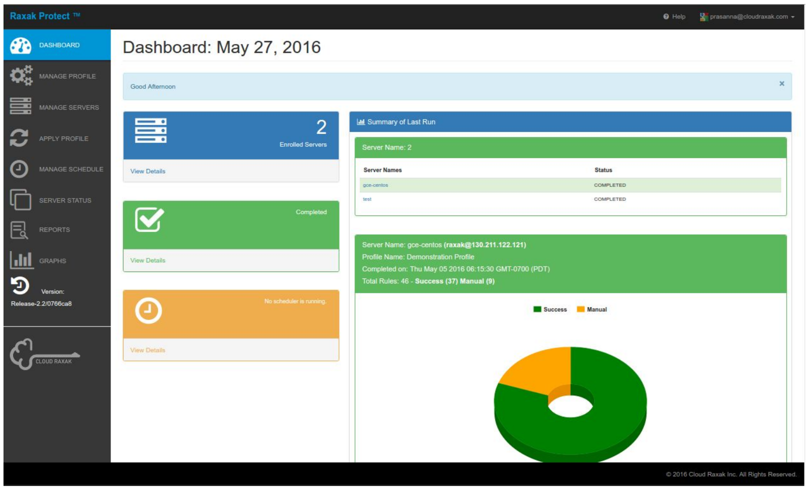
Once you are logged in, you will see the dashboard screen. The first time you log in to the console, you will see the following.



You have no servers enrolled. Click here to enroll.

© 2016 Cloud Raxak Inc. All Rights Reserved.

In the future, when you have enrolled servers, the view looks different. It shows the latest status of the servers.



Version:
Release-2.2/0766ca8

Dashboard: May 27, 2016

Good Afternoon

2
Enrolled Servers

Completed

No scheduler is running.

Summary of Last Run

Server Name: 2

Server Names	Status
gce-centos	COMPLETED
test	COMPLETED

Server Name: gce-centos **(raxak@130.211.122.121)**
Profile Name: Demonstration Profile
Completed on: Thu May 05 2016 06:15:30 GMT-0700 (PDT)
Total Rules: 46 - **Success (37) Manual (9)**

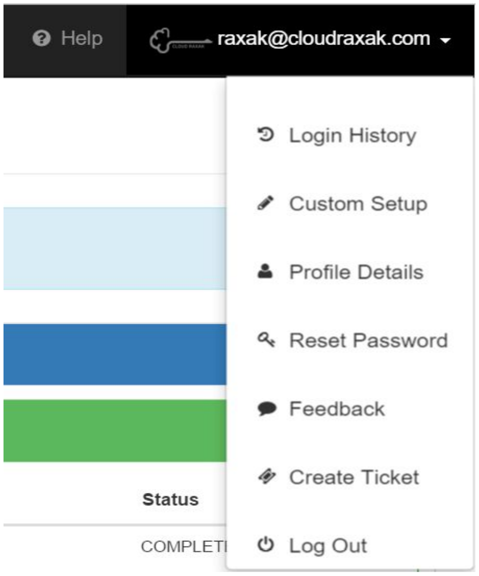
© 2016 Cloud Raxak Inc. All Rights Reserved.

The next step is to create a custom server setup script that provisions and enrolls your servers.

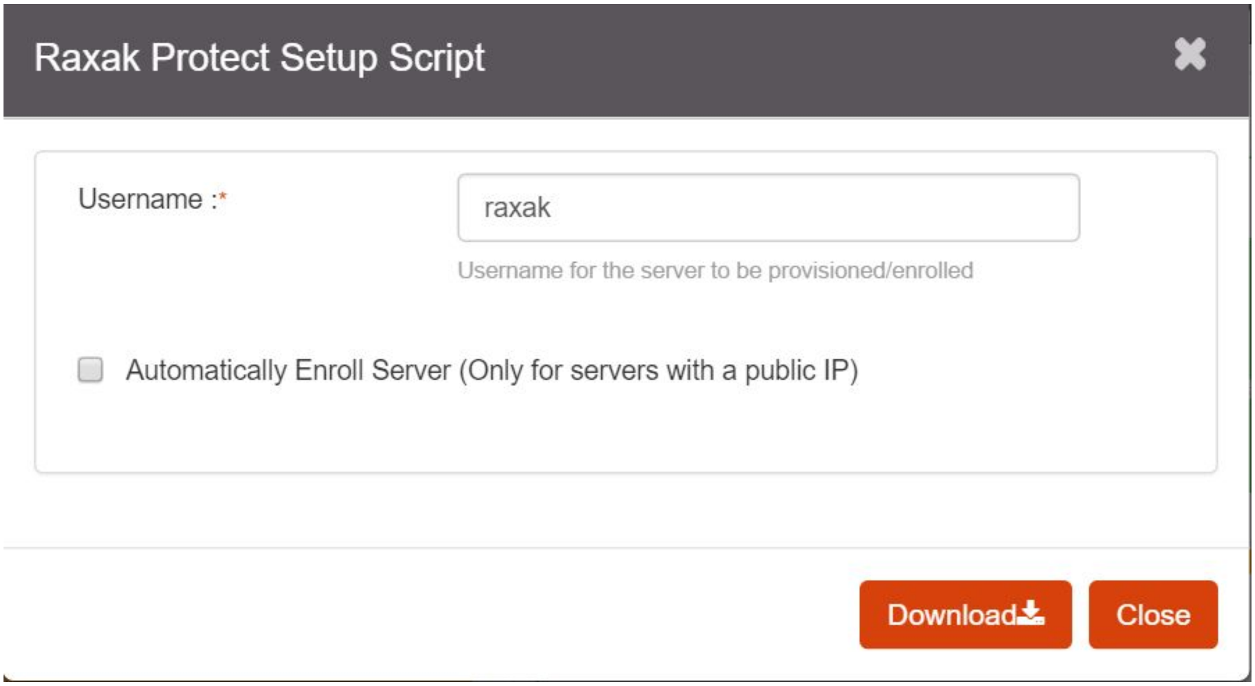
3 Custom Server Setup Script

NOTE: We assume that your server has a **public IP address** and the the network has been configured correctly to allow SSH access. For help on how to do this with the AWS console, refer to AWS online documentation.

To create a setup script, click on the down-arrow next to your login email id on the top-right of the console as shown below.



Click on the **Custom Setup** line and you will see the following window pop up.



Raxak Protect Setup Script

Username :* raxak

Username for the server to be provisioned/enrolled

☐ Automatically Enroll Server (Only for servers with a public IP)

In the **Username** box, enter the userid for the account that Raxak Protect should use to connect to the server being provisioned for enrollment in Raxak Protect. Note that if this username does not exist, an account with that name will be created. Ensure that the name you use does not conflict with other usernames in use on the server.

Check the box **Automatically Enroll Server**. For this example, leave the **Select Profile** drop-down set to **None.**

Click on the **Download** button. The script named **“RaxakProtectSetup.sh”** will be downloaded to your server, typically to your downloads folder depending on how your browser is configured. A sample RaxakProtectSetup.sh script is shown in the Appendix below.

NOTE: If your browser does not download the file, but opens it in a new window, the file may be corrupted and fail to work. This is a known issue with some configurations of the Safari browser on Macs. If this happens, use a different browser (Chrome, or Firefox work best).

4 Provisioning of Existing Servers

NOTE: We recommend that you initially experiment with Raxak Protect using a server cloned from your existing configurations rather than a production server. This ensures that any configuration changes required by the security profiles do not impact your operations.

1) Log in to the server being provisioned with a privileged userid (e.g., root or some user capable of elevating privilege to root)
2) Copy the "RaxakProtectSetup.sh" script that you downloaded in the previous step to this server using **wget**, **scp**, or any equivalent command
3) If you are not logged in as root, change userid to root by doing **sudo su**
4) Change the permission of the script to allow it to be executed by doing **chmod +x RaxakProtectSetup.sh**
5) Finally, execute the script by doing **./RaxakProtectSetup.sh**

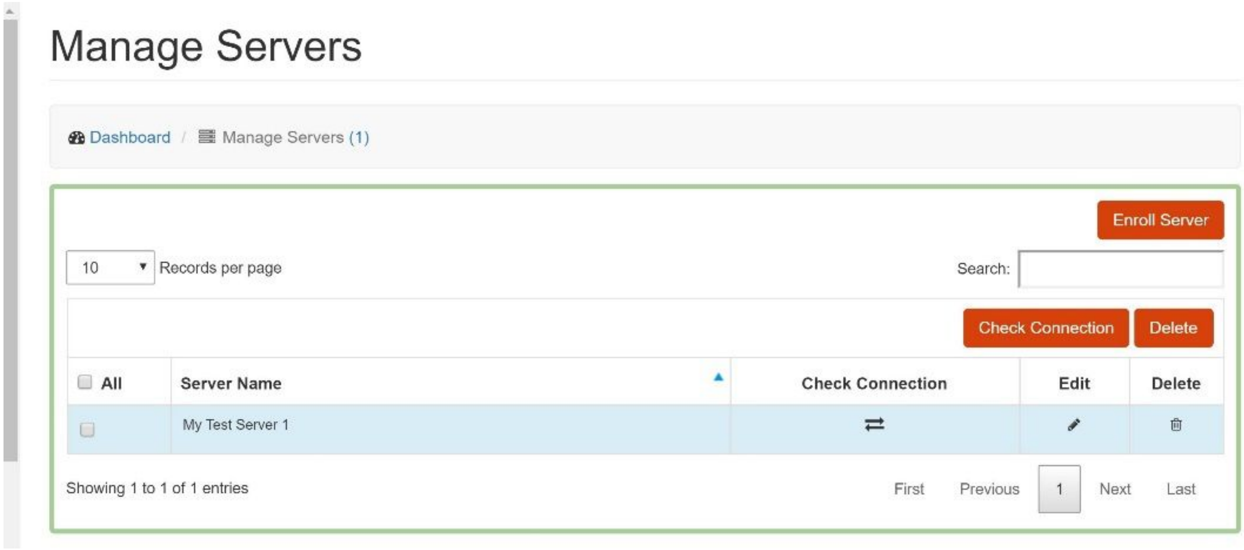
These steps provision the server with the userid specified in the previous step (**raxak** in the example) and installs the public key that Raxak Protect will use to connect to the server. After provisioning, the script also enrolls the server into the Raxak Protect system under your login id.

5 Provisioning of Newly Created Servers

If you create a new server, for example, using the AWS console, the RaxakProtectSetup.sh script can simply be provided as a post-provisioning script at the time of server provisioning. For instructions on how to do this, refer to the the AWS online documentation.

6 Verifying Server Enrollment

Return to the Raxak Protect console and click on **Manage Servers** in the left-hand navigation panel. You should see a list of enrolled servers, and the one you just added should show up in the list as well. The **Server Name** is the hostname of the server as reported by the operating system on the server. Hovering the mouse over the server name will show additional information such as the public IP of the server, and such.



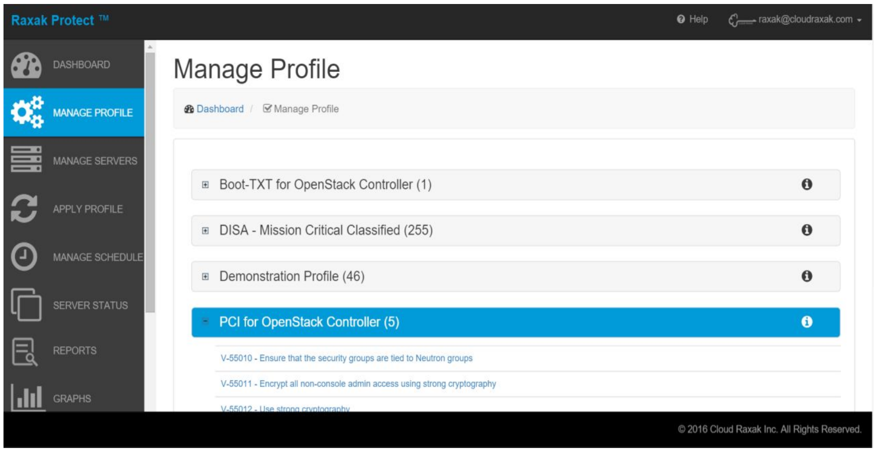
□ All	Server Name	Check Connection	Edit	Delete
□	My Test Server 1	⇄	✎	🗑

Showing 1 to 1 of 1 entries

The server should be highlighted in **green**, indicating that the Raxak Protect system could successfully connect to the server. Other colors indicate communications errors, such as inability to connect via SSH on port 22, inability to run at elevated privilege, etc. and usually indicate network configuration errors or failure of the script to run correctly. Please contact Cloud Raxak since it is not possible to diagnose all possible error conditions in this brief explanation.

7 Selecting and Applying a Profile

Raxak Protect comes populated with a few key profiles that you can get started with. They can be seen by navigating to the **Manage Profiles** pane using the left-hand navigation panel.



Manage Profile

⊞ Boot-TXT for OpenStack Controller (1)

⊞ DISA - Mission Critical Classified (255)

⊞ Demonstration Profile (46)

⊟ PCI for OpenStack Controller (5)

V-55010 - Ensure that the security groups are tied to Neutron groups

V-55011 - Encrypt all non-console admin access using strong cryptography

V-55012 - Use strong cryptography

© 2016 Cloud Raxak Inc. All Rights Reserved.

Profiles are listed along with the number of rules in the profile (the number in parentheses). Make particular note of the following profiles:

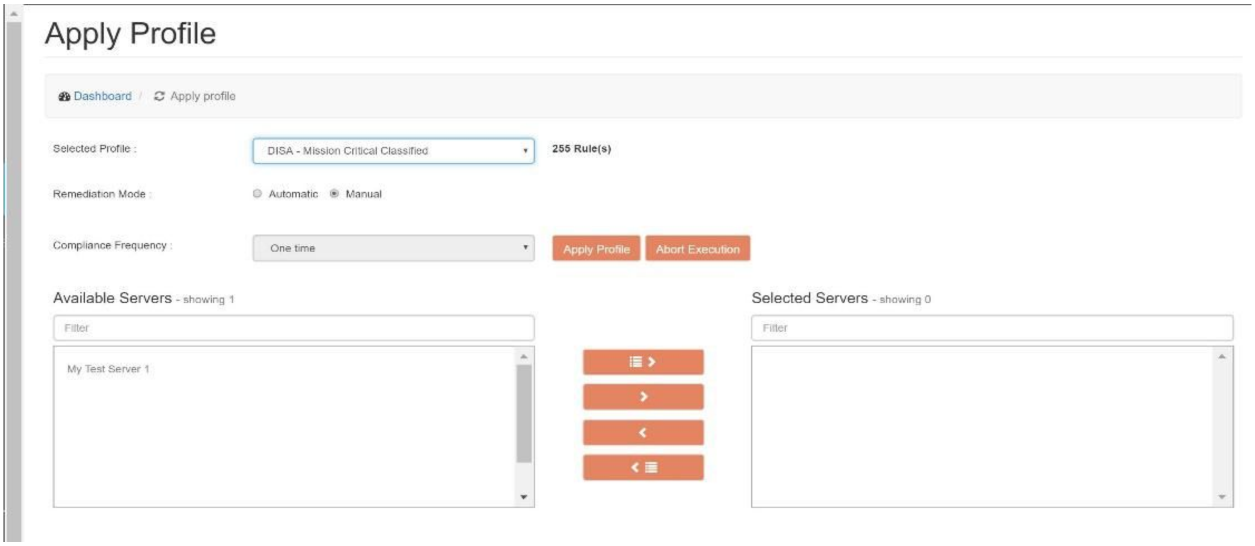
DISA Mission Critical Classified This is the profile that the US Defense Information Service Agency specifies as the gold standard for locking down sensitive assets. It is mirrored by the US National Institute of Standards and Technology (NIST) guidelines called the Security Content Automation Protocol (SCAP). This is generally a superset of many of the controls required by other regulations, such as HIPAA for medical records and PCI-DSS for the payment card industry.

PCI-DSS Compliance This profile reflects the settings compatible with the PCI-DSS-3.2 requirements under server configuration.

Demonstration Profile This is a simple profile that runs quickly and can be used to validate that all elements of the system and your server setup are complete and are operating correctly.

The next step is to manually apply a profile to your server, check the results, and manually remediate any failures (also known as **findings**).

Application of a given security profile is done in Raxak Protect via the **Apply Profile** pane. The application of the profile to a server means that the security rules included in that profile are checked against that server. You have the choice of picking the frequency of the check as well as what action you would like Raxak Protect to take in case any of the rules in that profile fail for that server. However, for now leave the **Remediation Mode** as **Manual** and the **Compliance Frequency** as **One Time**.



Apply Profile

Dashboard / Apply profile

Selected Profile : DISA - Mission Critical Classified 255 Rule(s)

Remediation Mode : ○ Automatic ● Manual

Compliance Frequency : One time

Available Servers - showing 1

Filter

My Test Server 1

Selected Servers - showing 0

Filter

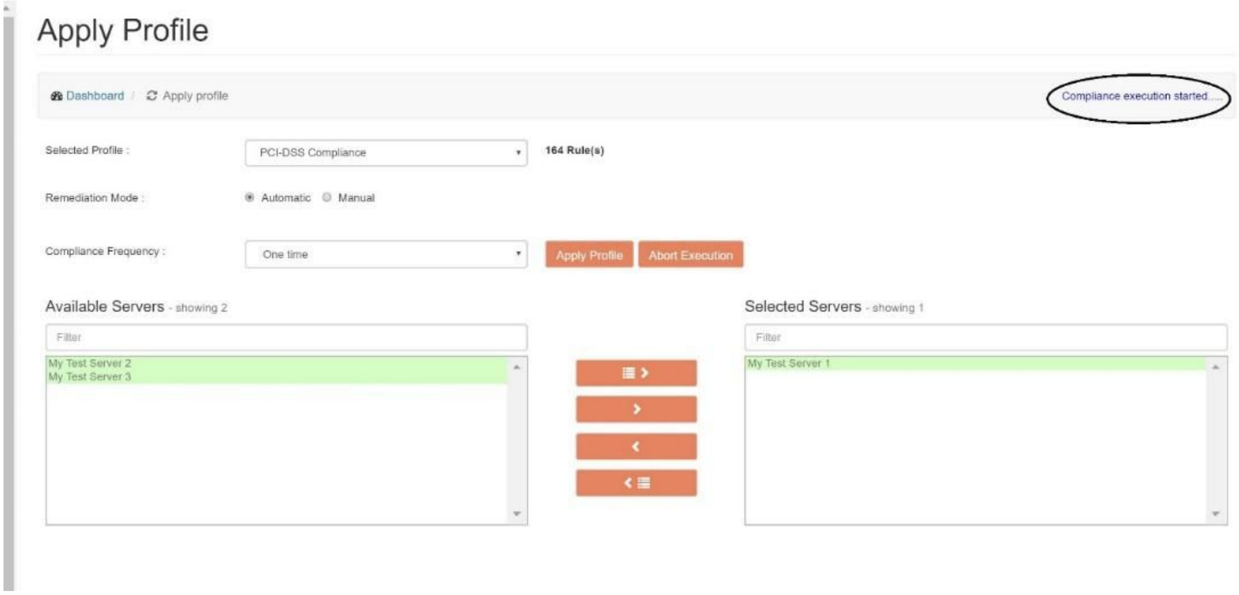
In **Selected Profile**, choose the profile that you would like to apply to one or more of the enrolled servers.

In the bottom left box titled **Available Servers,** you will see the list of enrolled servers. If you have only one enrolled server, it will be automatically selected. Otherwise click on the server in the left **Available Servers** box and hit the **>** button. That server will now move to the right box titled **Selected Servers**.

Remember you can always review the details for all the profiles, the rules themselves and the associated parameters. The profile and rule description is also available in the **Manage Profiles** pane and in the Resources section at the end of this document.

Once you have made all the choices in this window, hit the **Apply Profile** button.

The security compliance check of the selected servers with the selected profile will start immediately. You will see the phrase **"Compliance execution started"** in the top right corner of the window when the compliance check execution has started:



Apply Profile

Dashboard / Apply profile — Compliance execution started.....

Selected Profile : PCI-DSS Compliance — 164 Rule(s)

Remediation Mode : ● Automatic ○ Manual

Compliance Frequency : One time

Available Servers - showing 2

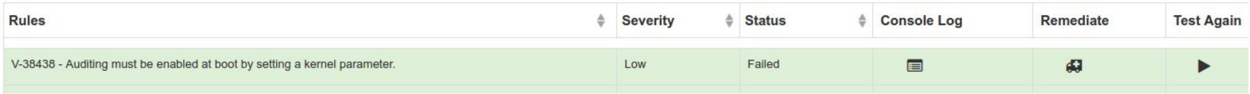
My Test Server 2
My Test Server 3

Selected Servers - showing 1

My Test Server 1

Once the execution finishes, you can check the status of the server by navigating to the **Server Status** pane.

The server status shows the compliance check that was just finished listing the rules that were run, categorized as **Successful**, **Failed**, or **Manual** as described earlier. Most of the rules in the **Failed** tab can be manually remediated by clicking the remediate icon in the row corresponding to the rule.



Rules	Severity	Status	Console Log	Remediate	Test Again
V-38438 - Auditing must be enabled at boot by setting a kernel parameter.	Low	Failed	▤	🚑	▶

If the rule is remediated, it will move to the **Successful** tab with a status of **Successfully Remediated.**

8 Next Steps

This document outlined a simple way to get started with Raxak Protect. You learned how to log on to the Raxak Protect console, create a custom provisioning script, manually provision and enroll a server, apply a profile, and manually remediate individual findings.

You are now free to experiment with other capabilities such as automatic remediation, periodic compliance checking, printing reports, and comparing the results of two runs.

Please feel free to contact us for more information, assistance, or training.

9 Limitations

Automatically created **free accounts** have the following limitations:

- They expire after **5** days, and cannot be renewed
- Users are restricted to enrolling no more than **2** servers in the system
- Detailed reports and status information pages do not show detailed console logs

10 More Resources

National Vulnerability Database

DISA STIG Viewer

PCI-DSS Requirements

For more information, assistance, or training, please contact:

info@cloudraxak.com

For white papers, case studies, and product literature, please visit:

www.cloudraxak.com/resource

Our Headquarters:

Cloud Raxak, Inc.
475 Alberto Way, Suite 130
Los Gatos, CA, 95032
USA

11 Appendix A: Sample RaxakProtectSetup.sh

This section shows a typical **RaxakProtectSetup.sh** script that is automatically created by Raxak Protect to enroll your servers. There are multiple sections in the script that may or may not be present in your own script depending on the options you select through the Raxak Protect console. **Note** that this is a sample script. **Do not** attempt to use it as is.

```
#!/bin/bash
# RaxakProtectSetup.sh
# (c) 2015-2017, Cloud Raxak, Inc.
# This script, run as a post-install script, will set up the desired userid
# and populate its public key. It will also set up no-password sudo access
# and the !requiretty flags in the sudoers file
# Usage: VMSetup.sh <userid>
# where <userid> defaults to "raxak" if not specified
#
#----------------------
# We assume that the script is running as root
# Output from the file can usually be found in /var/log
#
echo ${0}" VM setup script. (c) 2014-2016 Cloud Raxak Inc."
#
# randpw(){ < /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c${1:-16};echo;}

if [ "$(id -u)" != "0" ]; then
  echo ${0}" must be run as root."
  exit 1
fi

#   Global variables and overrides go here
username=raxak
secretcode=39F6E62A0DCE2B40
usertoken=<Hexadecimal token to validate the user>
raxakserver=http://amazon.cloudraxak.net
sshport=22
profile=None
auto=false
repeat=once
```

```
userid=kingsley.lim@redtone.com
# End customization
username=${username:-raxak}
echo Working as user `whoami`
id -u $username
result=$?;
if [ $result -ne 0 ]; then
echo "Create user "$username
echo 'Creating username '$username
useradd -m $username
echo '# Added for Raxak Protect service' >> /etc/sudoers
echo 'Defaults:'$username' !requiretty'  >> /etc/sudoers
echo 'Defaults:root !requiretty'        >> /etc/sudoers
echo $username' ALL=(ALL) NOPASSWD: ALL' >> /etc/sudoers
echo '# End section ---   Raxak Protect' >> /etc/sudoers
else
echo 'User '$username' exists -- adding public key'
fi
echo 'Switching to '$username
su - $username << "EOF"
echo User changed to `whoami`
cd ~
echo Working in directory `pwd`
mkdir -p ~/.ssh/
# Remove any old raxak key that may exist in the list
touch ~/.ssh/authorized_keys
sed -i.bak "s/.*raxak.*//g" ~/.ssh/authorized_keys
sed -i.bak "/^\s*$/d" ~/.ssh/authorized_keys  # Remove any blank lines left over from previous sed
# Replace key with active public key in ~/.ssh/raxak-key.pub (customize function)
key="ssh-rsa <RSA PUBLIC KEY INFORMATION>"
echo "Key = "$key
echo "$key" >> ~/.ssh/authorized_keys
echo ""     >> ~/.ssh/authorized_keys
chmod 640 ~/.ssh/authorized_keys
chmod 700 ~/.ssh/
EOF
#--- Next section does autoenrollment
#!/bin/bash
# This is the stub file that is  used by the createCustomVMSetup API call
#   to construct a custom VMSetup.sh
#   This section must run as root
#   $username should be defined externally
```

```
#
# Create hidden file with customization
#   This file is created as a hidden file with root access only
echo "Customization section: (c) Cloud Raxak Inc."
secretcode=${secretcode:-None}
username=${username:-raxak}
echo "Secret Code :"$secretcode
echo $secretcode > ~${username}/.raxak
chmod 600 ~$username/.raxak
#
# Now create the autoenroll capability
#
hostname=`hostname`
echo "Username: "$username
echo "Host name: "$hostname
# syntax: /raxakapi/v1/autoenroll/usertoken/nickname/username/profile/auto/periodicity
#   defaults: usertoken = ""    (will return warning)
#   username = "raxak" (set globally here)
#   nickname = hostname of the server
#   profile = None
#   auto = False (ignored if profile = None
#   periodicity = "Once" (ignored if auto = False)
#
usertoken=${usertoken:-None}
echo "User token: "$usertoken
raxakserver=${raxakserver:-http://softlayer.cloudraxak.net}
echo "Raxak SaaS server: "$raxakserver
enroll=$raxakserver"/raxakapi/v1/autoenroll/"$usertoken"/"$hostname"/"$sshport"/"$username"/"$profile"/"$auto"/"$repeat
echo "API Call: "$enroll
if [ -x /usr/bin/wget ] ; then
  echo "Using wget -q "$enroll" -O /dev/null"
  wget -q $enroll -O /dev/null
elif [ -x /usr/bin/curl ] ; then
  echo "Using curl "$enroll
  curl $enroll
else
  echo "Neither wget nor curl is installed"
  if [ -f /etc/lsb-release -o -f /etc/debian_version ]; then
     sudo apt-get -y install wget
     if [ $? -eq 0 -a -x /usr/bin/wget ]; then
        wget -q $enroll -O /dev/null
     else
```

```
            echo "wget:Install error hence unable to enroll, please contact the RAXAK administrator for further assistance."
            exit 1
        fi
    elif [ -f /etc/redhat-release ]; then
        sudo yum install -y curl
        if [ $? -eq 0 -a -x /usr/bin/curl ]; then
            curl $enroll
        else
            echo "curl:Install error hence unable to enroll, please contact the RAXAK administrator for further assistance."
            exit 1
        fi
    else
        #TODO - Redirect the logs in error file for future reference for debugging.
        echo "OS not supported. Unable to autoenroll."
        echo "Please contact the RAXAK administrator for further assistance."
        exit 1
    fi
fi
rm -f stack
exit
```