DISA Logo

ACAS

PKE and CAC Implementation Guide

September 8, 2017

v14

# Change Log

|  |  |  |
| --- | --- | --- |
| Date | Version | Changes |
| 8-Sep-2017 | 14 | * Updated instructions for manually uploading custom CA certificates * Added troubleshooting procedures for uploading custom CA certificates |
| 9-May-2017 | 13 | * Updated commands for SecurityCenter 5 * Pulled out commands for RHEL 5 Kickstart (obsolete) * Replaced SecurityCenter 4 screenshots |
| 12-Jan-2016 | 12 | * Change log added * Added caveat on running **crt-inst.sh** script multiple times on page 9. * Added 'PKI/PKE Basics' section * Added 'Kickstart Overview' section * Updated Kickstart Process commands for both RHEL 5 and RHEL 6 throughout entire guide * Added 'Provide the correct details in the CSR' section * Updated information in the 'Install the DoD Root Certificates' section * Moved the 'Configure Apache to Authenticate via Client Certificates' section * Updated information within the 'Load Server Identity and Root Certificates on Nessus Scanner' section. * Updated information within the 'Create Nessus Certificate Based User for SecurityCenter' section. * Updated information within the 'Create Nessus Certificate Based User for Browser Login' section. * Revised Appendix C: Sample CSR Configuration File * Updated links in Appendix D: DoD Tools and Reference Documents * Added Appendix G: Linux File Handling section * Updated links in Appendix I: Useful links |

Table of Contents

[Change Log 2](#_Toc492632768)

[Introduction 5](#_Toc492632769)

[Understanding How DoD PKI/PKE Interacts With ACAS 5](#_Toc492632770)

[PKI/PKE Basics 5](#_Toc492632771)

[Understanding Linux Files 5](#_Toc492632772)

[Steps to Complete Installation 6](#_Toc492632773)

[Kickstart Overview 7](#_Toc492632774)

[SecurityCenter Public Key Enablement Guidance 7](#_Toc492632775)

[Generate Private Key and CSR 7](#_Toc492632776)

[Provide the correct details in the CSR 7](#_Toc492632777)

[Sending CSR Information 8](#_Toc492632778)

[Receive CSR Information 8](#_Toc492632779)

[Installing New Server Identity Certificate on SecurityCenter 9](#_Toc492632780)

[Install the DoD Root Certificates 10](#_Toc492632781)

[Configure CRL Checking and CA Revocation 11](#_Toc492632782)

[Enabling CAC Login 11](#_Toc492632783)

[Configure Apache to Authenticate via Client Certificates 11](#_Toc492632784)

[Configure SecurityCenter to Authenticate via Client Certificates 12](#_Toc492632785)

[Connect With SSL Certificate Enabled Browser 12](#_Toc492632786)

[Nessus Scanner Public Key Enablement 15](#_Toc492632787)

[Load Server Identity and Root Certificates on Nessus Scanner 15](#_Toc492632788)

[Create Nessus Certificate Based User for SecurityCenter 16](#_Toc492632789)

[Create Nessus Certificate Based User for Browser Login 18](#_Toc492632790)

[Restricting Access to the Nessus User Interface 18](#_Toc492632791)

[Enable Proper Enumeration of DoD SSL Certificates 18](#_Toc492632792)

[PVS Sensor Public Key Enablement 21](#_Toc492632793)

[Load Server and Root Certificates on PVS Server Host 21](#_Toc492632794)

[Create PVS Certificate Based User for Browser Login 22](#_Toc492632795)

[Appendixes 25](#_Toc492632796)

[Appendix A: Root Certificate Installation / Verification 25](#_Toc492632797)

[Verify Proper Root Certificate Installation for SecurityCenter 25](#_Toc492632798)

[Alternate Root Certificate Installation for Nessus/PVS (For a Windows Host) 25](#_Toc492632799)

[Appendix B: Troubleshooting 27](#_Toc492632800)

[Compare SSL Certificate, Private Key, and CSR 27](#_Toc492632801)

[Debug OpenSSL 27](#_Toc492632802)

[Issues with uploading Custom CA Certificate 27](#_Toc492632803)

[Appendix C: Sample CSR Configuration File 29](#_Toc492632804)

[Appendix D: DoD Tools and Reference Documents 30](#_Toc492632805)

[Components required for ACAS Public Key Enablement: 30](#_Toc492632806)

[Optional information to assist with ACAS Public Key Enablement: 30](#_Toc492632807)

[Appendix E: Root Certificates Detail 31](#_Toc492632808)

[Appendix F: Determining your CAC Certificate CN 32](#_Toc492632809)

[Appendix G: Linux File Handling 33](#_Toc492632810)

[SSH and SCP Client 33](#_Toc492632811)

[Text Editors 33](#_Toc492632812)

[Appendix H: Acronyms Used in This Document 34](#_Toc492632813)

[Appendix I: Useful links 35](#_Toc492632814)

# Introduction

This document provides a Department of Defense (DoD) Assured Compliance Assessment Solution (ACAS) system administrator with all steps required to enable Public Key Infrastructure (PKI) authentication using DoD Certificates (including Common Access Card (CAC)) for users to access SecurityCenter, Nessus, and the Passive Vulnerability Scanner (PVS) web user interfaces (webUI).

## Understanding How DoD PKI/PKE Interacts With ACAS

DoD PKI provides a uniform Certificate Authority (CA) infrastructure that all DoD organizations are required to use. If an application or system uses PKI for authentication, it is considered Public Key Enabled (PKE). For full PKI compliance, ACAS requires a server identity certificate and root CA certificates to be installed on all SecurityCenter instances, Nessus active scanners, and Passive Vulnerability Scanners. The identity certificate is used along with a private key to verify the identity of a server to clients. The root CA certificates are used to allow a mutual trust framework so that a client (browser) can trust a remote server, and the server can trust the certificate provided by the client’s CAC login.

## PKI/PKE Basics

While all DoD PKE implementations share common functions and core infrastructure, different organizations have specific requirements and procedures. Before creating or changing any certificates, you should engage the organization’s Registration Authority (RA) or Local Registration Authority (LRA) to ensure you are meeting their obligations and requirements. In general, for more information, please refer to <http://iase.disa.mil/pki/index.html>.

## Understanding Linux Files

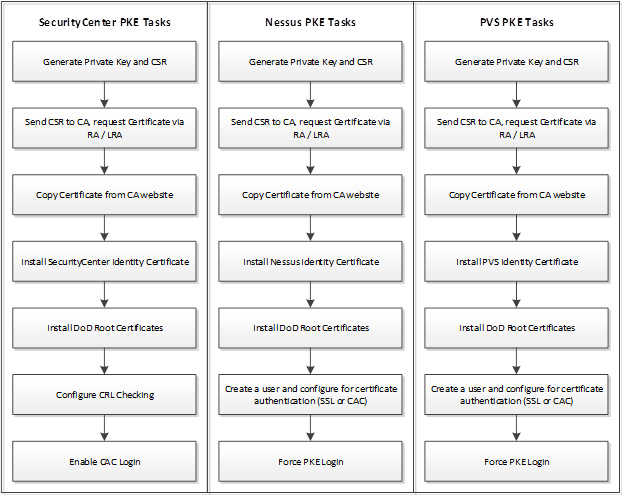
The SecurityCenter application only functions on Linux (Red Hat Enterprise Linux). Linux and Windows do not share the same End-Of-Line (EOL) characters. This is why a file that looks cleanly formatted in SecurityCenter may look incoherent on a Windows workstation. It is important to be cautious handling files in Windows that will be used on a Linux server. If you edit or create a file on a Windows computer, you should use the Notepad application for this task. Notepad was used to develop and validate this guide. Other tools are available that may be useful if Notepad proves to be too difficult, but they are not part of the ACAS baseline.

If a file does not behave properly, or if you have issues after editing a file, this command may help. To generate the **^M**, use **ctrl+v** and **ctrl+m**:

# **sed -n "s/^M//g" /path/to/file.in.question**

## Steps to Complete Installation

The ACAS PKE configuration affects the three basic components (SecurityCenter, Nessus, and PVS). The basic tasks are identified in the table below. The detailed process for accomplishing each task is provided in the following sections of this document.



## Kickstart Overview

The Defense Information Systems Agency (DISA)-provided Kickstart provides scripts that handle the more tedious tasks. SecurityCenter, Nessus, and PVS should be configured in a manner that requires client (CAC) certificates for login to the web user interfaces (webUI).

The current Kickstart image supports Red Hat Enterprise Linux (RHEL) version 6.

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | The RHEL 6 Kickstart image contains a menu drive setup tool which can run the individual scripts called out in this guide:  # /opt/acas/setup.sh. |

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | The ACAS Kickstart is located on the DoD Patch Repository: <https://patches.csd.disa.mil/CollectionInfo.aspx?id=588&bc=442_1_15_asc,535_1_15_asc>. |

# SecurityCenter Public Key Enablement Guidance

## Generate Private Key and CSR

In order for a system to work within the DoD environment, the system must have a valid Private Key and Certificate. The Private Key is generated on the system, and is then used to create a Certificate Signing Request (CSR). It is critical to protect the Private Key’s confidentiality (more information is available in [Appendix D](#_Appendix_D:_DoD)). The system administrator needs to generate the CSR, then copy and paste the CSR data into the Certificate Authority (CA) website during the enrollment process.

**RHEL 6 Kickstart Process**

# **/opt/acas/bin/acas-pke/csr-gen.sh**

Run this process to create the private key and CSR.

**Manual Process**

ACAS uses [OpenSSL](http://www.openssl.org/), which is an open source encryption suite that is installed as part of SecurityCenter. To generate a private key, use the following command:

# **/opt/sc/support/bin/openssl genrsa -out /root/dodserverkey.key 2048**

This creates the file dodserverkey.key that contains a private key. For security purposes, do not disclose this file to anyone.

The private key is used in the command to generate a CSR as follows:

# **/opt/sc/support/bin/openssl req -new -key /root/dodserverkey.key -out /root/dodserver-pkcs10.csr -config /root/ACASpki/dod-openssl.cfg**

See [Appendix C](#_Appendix_C:_Sample) if you wish to create your own configuration file for the CSR.

### Provide the correct details in the CSR

You will be prompted for information that is required by your Certificate Authority (CA) to generate a certificate. Most information is pre-populated, some fields require input.

The ‘Military/Government Component’ should match the organization’s CC/S/A, or be ‘CONTRACTOR’. Consult with the organization’s Registration Authority (RA) or Local Registration Authority (LRA) before requesting a certificate as they determine these and other settings are determined.

Use the name of the webserver as the Common Name (CN). The common name should match the URL that users will use to access the SecurityCenter. This could be an Internet Protocol (IP) address, the server or hostname, or the Fully Qualified Domain Name (FQDN) (example: servername.domain.mil).

### Sending CSR Information

After you have created the CSR, open the dodserver-pkcs10.csr file in a text editor, and copy and paste the contents into the online enrollment form when requested.

Once you have created the CSR, you need to register the request. Contact your RA/LRA to determine the CA that will generate your certificate. Open your web browser and enter the appropriate URL (example: <https://ca-21.c3pki.chamb.disa.mil/ca/>):

1. Select new **2048-bit SSL Enrollment** form.
2. Select **PKCS#10** for the Certificate Request Type.
3. Copy the Certificate Request into the text box. You can copy the CSR by opening the dodserver-pkcs10.csr file with Notepad, selecting and copying all of the data, and pasting it into the Certificate Request box.
4. An alternate Domain Name Server (DNS) name can be added if this server has two or more DNS entries or IP addresses. For example, in a load-balanced system, the public name may be www.test.com, while the server’s internal name is web-server01.test.com. The certificate needs to reference both names. Enter the alternate name here. If there is no alternate name, leave it blank.
5. Insert your contact information in the Requestor Name, Requestor Email Address, and Requestor Phone Number text boxes.
6. Click **Submit**.
7. The Certificate Profile Window will be displayed with your request ID. Provide your RA/LRA with this request ID and any additional forms required by your organizations policy. **Note**: The RA needs this information to complete the process.

### Receive CSR Information

1. Once your certificate has been signed (i.e., approved by your Registration Authority), you will be notified by e-mail.
2. Go to the CA website and copy the Base 64 encoded certificate (PEM format), starting from “-----BEGIN CERTIFICATE-----“ through “-----END CERTIFICATE-----“ (including all content between).
3. If you use Notepad on a Microsoft Windows computer, you need to name the file and copy the file to the SecurityCenter server, preferably to the /opt/acas/private or/root directory that will help protect it from unauthorized access or use. The file can be named anything, and the file extension is not critical, however using **.crt** or **.pem** is recommended.

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | Use caution when handling certificate files or editing their content. File editors often edit the contents of files in a manner that is not seen by the user, but renders the certificate unusable by ACAS. |

## Installing New Server Identity Certificate on SecurityCenter

Once you have received your new certificates from DoD, follow the steps below to install on the SecurityCenter as the root account or a ‘super user’ account with similar permissions:

**RHEL 6 Kickstart Process**

# **/opt/acas/bin/acas-pke/sc-cer-inst.sh**

**Manual Process**

1. Backup the current certificates that are located in the /opt/sc/support/conf directory. These files are named SecurityCenter.crt and SecurityCenter.key. In the example below, we are placing the files in /root:

# **cd /opt/sc/support/conf/**

# **tar –czvf /root/SecurityCenter\_orig\_keys.tgz SecurityCenter.\***

1. Copy the new certificate (in the example below, this is shown as “host.cert”; your file may vary) and private key (dodserverkey.key) to the /opt/sc/support/conf directory and overwrite the current certificates. If prompted to overwrite, press "y".

# **cp /root/host.cert /opt/sc/support/conf/SecurityCenter.crt**

# **cp /root/dodserverkey.key /opt/sc/support/conf/SecurityCenter.key**

1. Make sure the files have the correct permissions (640) and ownership (tns:tns):

# **chmod 640 /opt/sc/support/conf/SecurityCenter.crt**

# **chmod 640 /opt/sc/support/conf/SecurityCenter.key**

# **chown tns:tns /opt/sc/support/conf/SecurityCenter.crt**

# **chown tns:tns /opt/sc/support/conf/SecurityCenter.key**

# **ls -l /opt/sc/support/conf/SecurityCenter.\***

-rw-r----- 1 tns tns 4389 May 15 15:12 SecurityCenter.crt

-rw-r----- 1 tns tns 887 May 15 15:12 SecurityCenter.key

1. Restart SecurityCenter services:

# **service SecurityCenter restart**

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | Any time the web server's certificate is changed or updated, the browser may need to be refreshed and/or its cached cleared. |

You may browse to the SecurityCenter (https://<server name or IP>) webUI. Depending on the nature of your certificate, you should not need to confirm the certificate or receive an error, and you should go directly to the application. If the server does not respond correctly, refer to [Appendix B](#_Appendix_B:_Troubleshooting) for helpful tips.

## Install the DoD Root Certificates

This example uses the Certificates\_PKCS7\_v5.0u1\_DoD package, which contains the DoD root certificates for the NIPRNet. The root certificate should be validated before installation; see [Appendix A](#_Appendix_A:_Root) for additional guidance. The certificate bundles can be downloaded from the IASE website (a link is available in [Appendix D](#_Appendix_D:_DoD)). You may need to use different CA roots specific to your environment which is discussed briefly in [Appendix E](#_Appendix_E:_Root).

**RHEL 6 Kickstart Process**

Download the DoD Root CA Certificates PKCS #7 bundle from the PKI and PKE Tools page on the IASE website. Copy the zip files to the SecurityCenter server /opt/acas/var directory.

# **/opt/acas/bin/acas-pke/sc-root-inst.sh**

**Manual Process**

Download the DoD Root CA Certificates PKCS #7 bundle from the DoD PKE Site. Copy the zip files to the SecurityCenter server /root/ACASpki directory.

Unzip the files so SecurityCenter can install the Root Certificates:

# **unzip –o /root/ACASpki/Certificates\_PKCS7\_v5.0u1\_DoD.zip -d /root/ACASpki**

Unpack the CA certificates from their PKCS #7 format. This will yield a single file with all of the DoD CA certificates concatenated together:

# **/opt/sc/support/bin/openssl pkcs7 -in /root/ACASpki/ Certificates\_PKCS7\_v5.0u1\_DoD/Certificates\_PKCS7\_v5.0u1\_DoD.pem\*.p7b -print\_certs -out /root/ACASpki/DoD\_CAs.pem**

Copy each CA certificate into its own file. Below is an example of a command that makes it easier to copy the CA certificates (substitute the actual CA name for <CA name>) from the DoD\_CAs.pem file. It is recommended that you remove non-certificate content and verify only a single certificate in the file produced:

# **sed -n "/<CA name>/,/-END CERTIFICATE-/p" /root/ACASpki/DoD\_CAs.pem > /root/ACASpki/<CA name>.pem**

Use the InstallCA tool to prepare the CA certificate for use by SecurityCenter:

# **/opt/sc/support/bin/php /opt/sc/src/tools/installCA.php /root/ACASpki/<CA name>.pem**

Once each of your CAs have been processed, restart SecurityCenter:

# **service SecurityCenter restart**

Restarting the service at this point is not needed, but this helps identify issues that might occur during this stage.

## Configure CRL Checking and CA Revocation

**Revocation Overview**

Certificates are often revoked when a user leaves an organization, loses a smart/CAC card, or a server’s private key is compromised. To enable revocation, RFC 5280 lists the following states that can be applied by a CA to any of its issued certificates. There are two popular certificate validation mechanisms:

• CRL – Certificate Revocation List

• OCSP – Online Certificate Status Protocol

**Certificate Revocation List (CRL)**

A certificate revocation list (CRL) is a digitally signed list issued by a CA that contains certificates that have been revoked. The list includes the serial number of the certificate, the date the certificate was revoked, and the reason for revocation.

**Certificate Revocation Checking Using Online Certificate Status Protocol (OCSP)**

The Online Certificate Status Protocol (OCSP) supplements CRL validation and enables high-performance validation of certificate status. OCSP is not compatible with ACAS at this time.

**ACAS SecurityCenter CRL Configurations**

Different revocation lists are available for different environments and operational needs. This example uses the unclass-crlautocache\_linux\_2-05\_nipr package that contains the script for downloading DoD CRLs for the NIPRNet. You should repeat this process with any other CAs needed for your environment. You need to copy the tools from the PKI and PKE Tools page (link is in [Appendix D](#_Appendix_D:_DoD)). The web services within SecurityCenter must be restarted to update the CRL lists, so it is important to ensure that the update occurs during off-peak hours. The Kickstart process uses a helper script to wait for users to disconnect from the web server. By default, the Kickstart scripts provide a **15-minute grace period before forcibly restarting the Apache service**. The outage window should take only a few minutes and the exact times will be recorded in /var/log/CRLAutoCache\_Linux.log. Because of this, the manual process below is discouraged.

**RHEL 6 Kickstart Process**

# **/opt/acas/bin/acas-pke/crl-conf.sh**

**Manual Process**

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | Manual configuration this way is not recommended. If you cannot use the provided scripts, it is recommended that you engage the ACAS PM to request assistance or recreate the functions contained within the scripts provided. |

## Enabling CAC Login

A feature in SecurityCenter allows users to use SSL client certificate authentication. This feature allows the use of SSL Client Certificates, Smartcards, and CAC authentication when the browser is configured for this method.

## Configure Apache to Authenticate via Client Certificates

This setting requires a client-side certificate. After this setting is applied, a CAC is required for login. This script modifies the Secure Sockets Layer (SSL) requirements in Apache as needed for client-side SSL certificates.

**RHEL 6 Kickstart Process**

# **/opt/acas/bin/acas-pke/sc-cac-require.sh**

**Manual Process**

Set client certificate verification in /opt/sc/support/conf/sslverify.conf:

# **sed –i ‘/SSLVerifyClient/c\SSLVerifyClient require’ /opt/sc/support/conf/sslverify.conf**

# **service SecurityCenter restart**

After the SecurityCenter server has been configured with the proper CA certificate(s), users should log into the server using client SSL or CAC certificates.

## Configure SecurityCenter to Authenticate via Client Certificates

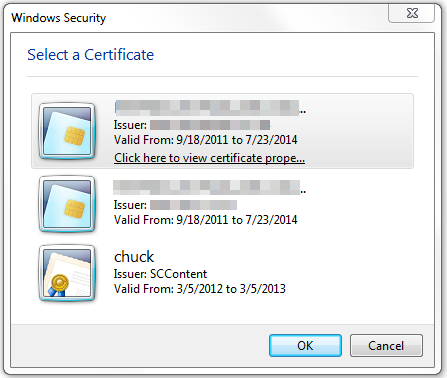
By default, SecurityCenter does not perform SSL Certificate authentication. Upon initial configuration of the user’s account, a password must be assigned. Once the user logs in via the browser, the user will be asked if they want to always use the current certificate (provided the user’s account is properly configured to use an SSL client certificate.) If “Yes” is selected, the certificate will be associated with their account and future access to the SecurityCenter will use the client certificate. If “No” is selected, the certificate will be ignored for the current session.

### Connect With SSL Certificate Enabled Browser

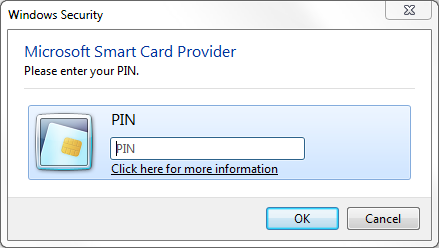
|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | The following information is provided with the understanding that your browser is configured for SSL Certificate authentication. Please refer to your browser’s instructions to configure this feature appropriately. |

When the user connects to the server for the first time, the process of binding the certificate they used to authenticate with their SecurityCenter login begins. The process is completed by the user, and does not require Administrator intervention.

1. Launch the browser and navigate to your SecurityCenter.
2. The browser displays a list of available certificate identities to select, as shown in the following example:



1. Once a certificate has been selected, a prompt for the PIN or password for the certificate is displayed, if it is required to access your certificate (as shown below). When successfully entered, the certificate will be available for the current session with SecurityCenter.

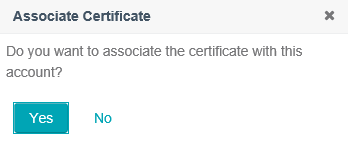


1. Upon the initial connection, log in using the username to be associated with the selected certificate.

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | Only one SecurityCenter user may be associated with a single certificate. If one user holds multiple user names and roles, a unique certificate must be provided for each login name. |

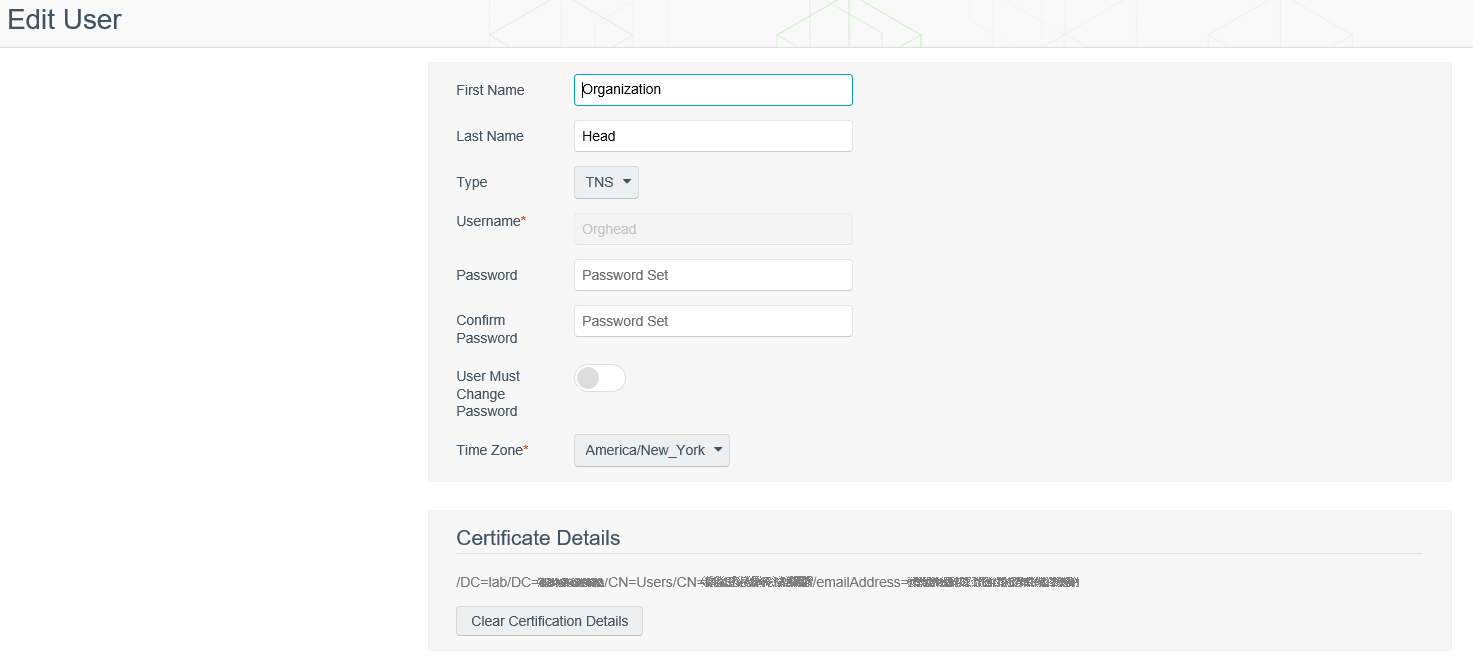
1. Once logged in, a window titled “Certificate Authentication” is presented, asking if the current certificate should be used to authenticate the current user. If “Yes” is selected, the certificate will be associated with this user. If “No” is selected, the certificate will be ignored for the current session. See the following example:

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | If the user’s browser is configured for certificate authentication but is not configured for a SecurityCenter user, then this prompt will be presented at each login. |



1. When a user’s account is associated with a certificate, it is displayed on the user’s information page, and may be viewed using the “Edit User” page on the “Basic” tab.

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | The “Certificate Details” section for a user only appears if there is an associated certificate. |



1. If a user’s certificate changes or is revoked, the current certificate may be disassociated from the user by clicking the “Clear Certification Details” button.
2. When a new certificate is available, the user logs in using their username and password and SecurityCenter will again attempt to associate the user with the certificate.

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | If you log out of the session, you will be presented with the standard SecurityCenter login screen. If you wish to login again with the same certificate, refresh your browser. If you need to use a different certificate, you must restart your browser session. |

# Nessus Scanner Public Key Enablement

## Load Server Identity and Root Certificates on Nessus Scanner

To enable the Nessus scanner in a DoD-managed environment, you need to generate a certificate request for Nessus. The recommended method for generating a certificate request involves creating the private key and CSR on the SecurityCenter server using the guidance from the "Generate Private Key and CSR" section of this document. The common name or CN needs to match the hostname or IP address that users and the SecurityCenter use to connect to the Nessus scanner.

After you receive your certificate, you need to copy the following to the Nessus scanner host:

* your private key,
* the certificate you downloaded from the CA server, and
* a copy of the DoD root certificate (Root CA-2 as of this writing).

**RHEL 6 Kickstart Process:**

1. install the DoD root certificate

# **/opt/acas/bin/acas-pke/ns-root-inst.sh**

1. Install the Nessus server certificate

# **/opt/acas/bin/acas-pke/ns-crt-inst.sh**

**Manual Process:**

1. Backup the existing certificates:

**Linux Only:**

# **cp /opt/nessus/com/nessus/CA/cacert.pem /root/OLDNESSUScacert.pem**

# **cp /opt/nessus/com/nessus/CA/servercert.pem /root/OLDNESSUSservercert.pem**

# **cp /opt/nessus/var/nessus/CA/serverkey.pem /root/OLDNESSUSserverkey.pem**

**Windows Only:**

# **copy <X>:\ProgramData\Tenable\Nessus\Nessus\CA\cacert.pem <X>:\Users\<admin account>\Downloads\OLDNESSUScacert.pem**

# **copy <X>:\ProgramData\Tenable\Nessus\Nessus\CA\servercert.pem <X>:\Users\<admin account>\Downloads\OLDNESSUSservercert.pem**

# **copy <X>:\ProgramData\Tenable\Nessus\Nessus\CA\serverkey.pem <X>:\Users\<admin account>\Downloads\OLDNESSUSserverkey.pem**

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | Do not remove or alter the default cakey.pem file on the Nessus scanner. This file is located in /opt/nessus/com/nessus/CA/ on Linux. This file is located in <x>:\ProgramData\Tenable\Nessus\nessus\CA\ on Windows. |

2. Copy the applicable DoD root certificate to the Nessus scanner. A copy of the certificate can be retrieved from the /opt/sc/data/CA directory on the SecurityCenter host. The following Step 3 refers to the copy of this file as “NEWNESSUScacert.pem”. Stop the Nessus service before overwriting the certificates and key file.

**Linux Only:**

# **service nessusd stop**

**Windows Only:**

# **<X>:\> net stop "Tenable Nessus"**

3. Copy valid certificates and keys to the Nessus scanner:

**Linux Only:**

# **cp /root/<NEWNESSUScacert.pem> /opt/nessus/com/nessus/CA/cacert.pem**

# **cp /root/<NEWNESSUSservercert.pem> /opt/nessus/com/nessus/CA/servercert.pem**

# **cp /root/<NEWNESSUSserverkey.pem> /opt/nessus/var/nessus/CA/serverkey.pem**

**Windows Only:**

# **copy <X>:\Users\<admin account>\Downloads\<NEWNESSUScacert.pem> <X>:\ProgramData\Tenable\Nessus\Nessus\CA\cacert.pem**

# **copy <X>:\Users\<admin account>\Downloads\<NEWNESSUSservercert.pem> <X>:\ProgramData\Tenable\Nessus\Nessus\CA\servercert.pem**

# **copy <X>:\Users\<admin account>\Downloads\<NEWNESSUSserverkey.pem> <X>:\ProgramData\Tenable\Nessus\Nessus\CA\serverkey.pem**

4. Start Nessus:

**Linux Only:**

# **service nessusd start**

**Windows Only:**

# **<X>:\> net start "Tenable Nessus"**

## Create Nessus Certificate Based User for SecurityCenter

You will create a user, and then configure that user for certificate authentication.

1. On the Nessus server, run the **nessuscli mkcert-client** command:

**Linux Only:**

# **/opt/nessus/sbin/nessuscli mkcert-client**

**Windows Only:**

# **<X>:\> \Program Files\Tenable\Nessus\nessuscli mkcert-client**

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | The answer to the second question “Do you want to add user to the Nessus server as soon as their certificate is created?” should be “Y”. |

2. If you already have a certificate to use for authentication to the Nessus scanner, the Nessus username should match the CN of your existing certificate. If you request a certificate after you create the user, the name should match the CN of the certificate you will be requesting.

3. You should discard the certificate and key pair generated by the mkcert-client utility, but it is not critical.

4. Generally, this user should be a Nessus Administrator, unless SecurityCenter will not be pushing plug-ins to the scanner.

5. Configure the Nessus server for certificate authentication. This disables password authentication:

**Linux Only:**

# **/opt/nessus/sbin/nessuscli fix –-set force\_pubkey\_auth=yes**

**Windows Only:**

# **<X>:\> \program files\Tenable\Nessus\nessuscli.exe –-set force\_pubkey\_auth=yes**

Now you need to request an SSL certificate where the CN (common name, typically a server name) of the certificate matches the username you specified in Step 2 above. Create the private key and CSR on the SecurityCenter scanner using the guidance from "Generate Private Key and CSR" section of this document. Then submit your CSR and retrieve your certificate.

6. Combine the certificate and key to create a new file that will be uploaded into SecurityCenter:

**Linux Only:**

# **cat /root/nessus-user-cert.crt /root/nessus-user-key.pem > /root/nessus-auth-cert.pem**

**Windows Only:**

# **<x>:\> type nessus-user-cert.crt > nessus-auth-cert.pem**

# **<x>:\> type nessus-user-cert.key >> nessus-auth-cert.pem**

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | The authentication file should contain the certificate and key with no blank lines or extra characters.  -----BEGIN CERTIFICATE-----  <Large alphanumeric string>  -----END CERTIFICATE-----  -----BEGIN RSA PRIVATE KEY-----  <Large alphanumeric string>  -----END RSA PRIVATE KEY----- |

7. Copy the combined cert/key file to a location where it is accessible from a browser that you use to connect to the SecurityCenter server.

8. Restart the Nessus services:

**Linux Only:**

# **service nessusd restart**

**Windows Only:**

# **<X>:\> net stop “Tenable Nessus”**

# **<X>:\> net start “Tenable Nessus”**

9. Log into SecurityCenter as an Administrator; select **Resources->Nessus Scanners**; and select a Nessus scanner or create a new scanner. For Authentication Type, select **SSL Certificate**, and browse to the certificate you created in Step 1 and then click **Submit**.

10. Verify the configuration works properly by navigating to the Nessus scanners page. The scanners status should go to 'Updating Status' and then back to 'Working'.

11. Select the scanner again and check "Verify Hostname." This verifies the hostname or IP address used to connect to the Nessus scanner matches the CN in the Nessus scanner's SSL certificate.

12. Verify the configuration works properly by navigation to the Nessus scanners page. The scanners status should go to 'Updating Status' and then back to 'Working'

## Create Nessus Certificate Based User for Browser Login

A user should have access to the Nessus scanner for troubleshooting purposes. You need to create a user with a name that matches the common name from your CAC.

1. On the Nessus server, run the **nessuscli-mkcert-client** command:

**Linux Only:**

# **/opt/nessus/sbin/nessuscli mkcert-client**

**Windows Only:**

# **<X>:\> \Program Files\Tenable\Nessus\nessuscli mkcert-client**

2. Answer the questions presented for the **User Distinguished Name**. The only field that is critical for authentication is **Common Name**. This should match the CN (last.first.x.9999999) of the CAC card whose user you want to allow login.

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | You will be prompted to provide a username. This must match the CN of the CAC or SSL certificate you wish to use for authentication. |

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | The answer to the second question “Do you want to add user to the Nessus server as soon as their certificate is created?” should be “Y”. |

## Restricting Access to the Nessus User Interface

In addition to the web interface provided by SecurityCenter, Nessus presents a web server interface on TCP port 8834. Access to this port should be restricted to the SecurityCenter server and the local host, or analyst workstation(s) via host-based or network firewall, or equivalent network access controls.

Further guidance on configuring authentication and controlling access to the Nessus user interface is available in the Nessus Administrator Guide.

## Enable Proper Enumeration of DoD SSL Certificates

By default, Nessus considers DoD PKI Certificate Authority servers and certificates as untrustworthy. This can result in triggering plugin 51192 (SSL Certificate Cannot Be Trusted) on servers that have valid DoD SSL certificates.

Nessus uses an '**include**' file (.inc) to store PEM-formatted certificate authority certificates. A script is posted on SoftwareForge to automate the generation and packaging of the ‘include’ file. The script uses certificates that SecurityCenter recognizes as trustworthy (DoD root certificates).

**RHEL 6 Process**

1. Download the script and execute via the following command:

# **/opt/acas/bin/acas-pke/sc-customCA-inc.sh**

1. By default, the script generates a custom plugin archive with only a custom\_CA.inc file. If you have existing custom Nessus Attack Scripting Language (NASL) scripts you want to include, provide the full path to the directory as an argument (example: /opt/acas/bin/acas-pke/sc-customCA-inc.sh /root/custom-nasls).
2. The script uses the certificates that SecurityCenter trusts, so you should run this script after running the root-inst.sh script. Both scripts should be run when new Certificate Authority servers are commissioned or decommissioned.

**Manual Process**

1. Copy the contents of all trusted certificate authority certificates installed on SecurityCenter into a text file and name it custom\_CA.inc. Make sure to include the -----BEGIN CERTFICIATE----- and -----END CERTIFICATE----- lines and everything in between. (If you need to upload multiple certificates, paste them all in back-to-back.):

# **cat /opt/sc/data/CA/\*.pem > ./custom\_CA.inc**

1. Create a text file named custom\_feed\_info.inc that includes the feed type and plugin date:

NOTE: The plugin set date should be the same as the time the bundle is uploaded to SecurityCenter. It cannot be after the present date/time in SecurityCenter.

# **pluginDate=$(date +%Y%m%d%H%M)**

# **echo 'PLUGIN\_SET = "'$pluginDate'";' > ./custom\_feed\_info.inc**

# **echo 'PLUGIN\_FEED = "Custom";' >> ./custom\_feed\_info.inc**

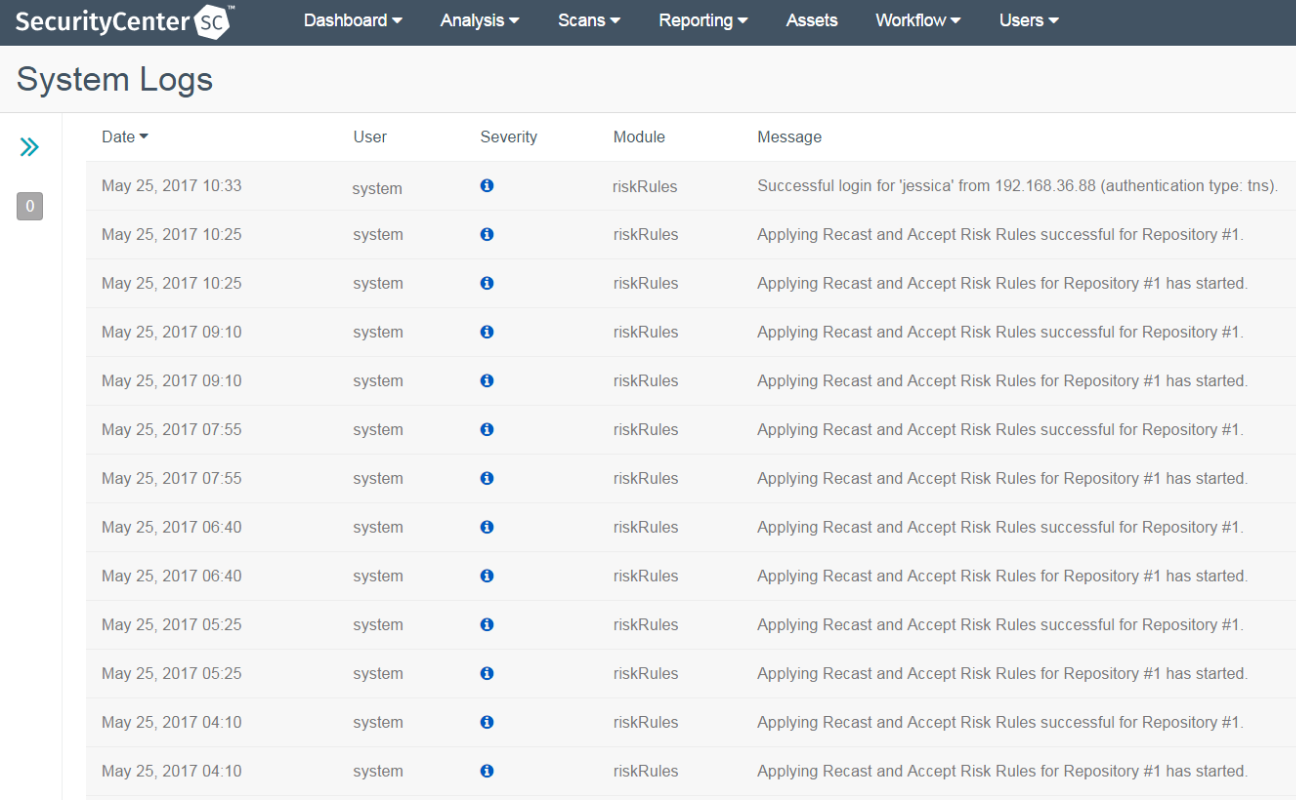
NOTE: The typical format for PLUGIN\_SET is a string of numbers in the format “YYYYMMDDHHMM” for the regular feed so that format is copied here.

1. Use the **tar** command to combine and compress the ‘include’ files into a single archive file:

# **tar -zcf custom\_feed$pluginDate.tar.gz custom\_feed\_info.inc custom\_CA.inc**

NOTE: If you are also uploading custom NASL scripts, you should include them, space separated, in this **tar** command after the .inc files.

After the custom feed<plugin date>.tar.gzfile has been created, you must copy it to a workstation so it can be uploaded via a browser. To upload the file, open a web browser and navigate to your SecurityCenter (https://<server name or IP>). Once logged in (As Administrator), go to the Plugins page (located in the upper right-hand corner of the screen; click on admin’s user name), and select “Plugins”. The “Upload Custom Plugins” button will be located on the top right corner. Click “Upload Custom Plugins,” then click the “Browse…” button to select the custom feed<plugin date>.tar.gz file that was created. Finally, click on the “Submit” button to upload the file.

1. Verify upload was successful. Go to “System” then “System Logs” and you should see the logs similar to the following:
2. Verify issue is resolved by running another scan including plugin 51192. You can verify that Nessus has the custom plugin bundle by checking its plugin directory.

NOTE: Updating SecurityCenter plugins to initiate a plugin push to the Nessus scanners, will only work if the plugin feed being downloaded by SecurityCenter is newer than the plugin set on the Nessus scanners. If it is still the same plugin set because Tenable has not released a newer plugin feed yet, wait a few hours for the next plugin feed to be available and then update again, or wait for the scheduled plugin update to run overnight.

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | The custom\_CA.inc file is overwritten every time it is uploaded. When adding additional CA certificates, start with a copy of the existing custom\_CA.inc and append the new certificate. If there are multiple certificates in the file, it should look like this:  -----BEGIN CERTIFICATE-----  blahblahblahblahblahblah732r  certificatestuffjsdhfgjklssahjkh  sefejhawklmkfjskmcjgkdsfmads  -----END CERTIFICATE----  -----BEGIN CERTIFICATE-----  blahblahblahblahblahblah79zc  morecertificatestuffsdg3a5tgh  fhdsthjgsfkdjt9845y6389fjsa3  -----END CERTIFICATE----- |

# PVS Sensor Public Key Enablement

## Load Server and Root Certificates on PVS Server Host

You need to generate a private key and CSR for Passive Vulnerability Scanner (PVS). This can be done on the SecurityCenter server host using the guidance from "Generate Private Key and CSR" section of this document. The common name, or CN, needs to match the hostname or IP address that users and the SecurityCenter use to connect to the PVS server.

After you receive your certificate, you need to copy the following to the PVS server host: your private key; a copy of the DoD root certificate (Root CA-2 as of this writing)’ and the certificate you downloaded from the CA server.

**RHEL 6 Kickstart Only:**

# **/opt/acas/bin/acas-pke/ps-root-inst.sh**

# **/opt/acas/bin/acas-pke/ps-crt-inst.sh**

1. Backup the existing certificates:

**Linux Only:**

# **cp /opt/pvs/var/pvs-proxy/ssl/cacert.pem /root/OLDPVScacert.pem**

# **cp /opt/pvs/var/pvs-proxy/ssl/servercert.pem /root/OLDPVSservercert.pem**

# **cp /opt/pvs/var/pvs-proxy/ssl/serverkey.pem /root/OLDPVSserverkey.pem**

**Windows Only (requires UAC/administrative escalation):**

# **copy <X>:\ProgramData\Tenable\PVS\pvs-proxy\ssl\cacert.pem <X>:\Users\<admin account>\Downloads\OLDPVScacert.pem**

# **copy <X>:\ProgramData\Tenable\PVS\pvs-proxy\ssl\servercert.pem <X>:\Users\<admin account>\Downloads\OLDPVSservercert.pem**

# **copy <X>:\ProgramData\Tenable\PVS\pvs-proxy\ssl\serverkey.pem <X>:\Users\<admin account>\Downloads\OLDPVSserverkey.pem**

2. Copy the applicable DoD Root CA certificate to the PVS server host. A copy of the certificate can be retrieved from the SecurityCenter located in /opt/sc/data/CA. The following Step 3 refers to the copy of this file as “NEWPVScacert.pem”.

3. Copy valid certificates and keys to the PVS server:

**Linux Only:**

# **cp /root/<NEWPVScacert.pem> /opt/pvs/var/pvs-proxy/ssl/cacert.pem**

# **cp /root/<NEWPVSservercert.pem> /opt/pvs/var/pvs-proxy/ssl/servercert.pem**

# **cp /root/<NEWPVSserverkey.pem> /opt/pvs/var/pvs-proxy/ssl/serverkey.pem**

**Windows Only (requires UAC/administrative escalation):**

# **del <X>:\ProgramData\Tenable\PVS\pvs-proxy\ssl\serverkey.pem**

# **copy <X>:\Users\<admin account>\Downloads\<NEWPVScacert.pem> <X>:\ProgramData\Tenable\PVS\pvs-proxy\ssl\cacert.pem**

# **copy <X>:\Users\<admin account>\Downloads\<NEWPVSservercert.pem> <X>:\ProgramData\Tenable\PVS\pvs-proxy\ssl\servercert.pem**

# **copy <X>:\Users\<admin account>\Downloads\<NEWPVSserverkey.pem> <X>:\ProgramData\Tenable\PVS\pvs-proxy\ssl\serverkey.pem**

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | Do not remove or alter the default cakey.pem file on the Passive Vulnerability Scanner. On Linux, this file is located in:  PVS 4.4 and above: /opt/pvs/var/pvs/ssl  PVS 4.2 and below: /opt/pvs/var/pvs-proxy/ssl  On Windows, this file is located in:  PVS 4.4 and above: <x>:\ProgramData\Tenable\PVS\pvs\ssl\  PVS 4.2 and below: <x>:\ProgramData\Tenable\PVS\pvs-proxy\ssl\ |

4. Restart PVS:

**Linux Only:**

# **service pvs restart**

**Windows Only:**

# **<X>:\> net stop “Tenable PVS Proxy”**

# **<X>:\> net start "Tenable PVS Proxy"**

## Create PVS Certificate Based User for Browser Login

User accounts are created in PVS using the web interface. Each account requires a password. However, once the server is configured to use certificate authentication, a certificate is required. You need to create a user with a name that matches the common name (CN) from your CAC/soft SSL certificate.

1. In the PVS webUI (https://<pvs ip/hostname>:8835/), a new user can be added by selecting 'Users' at the top of the main page. Click '+ New User' and **provide a user name that exactly matches the CN (common name) of your certificate/CAC. The user name is case sensitive**. **A password is required, and the account should be an Administrator.**

2. Requiring a client-side certificate for login can be done via the web interface or the command line interface:

**Web UI:**

* Select '**Configuration**'.
* Change Setting Type to **'PVS Web Server**'.
* Check **'Enable SSL Client Certificate Authentication**'.

**Linux Only:**

# **/opt/pvs/bin/pvs --config "Enable SSL Client Certificate Authentication" \ "1"**

**Windows Only:**

# **<X>:\Program Files\Tenable\PVS\PVS --config "Enable SSL Client Certificate Authentication" "1"**

3. Restart PVS:

**Linux Only:**

# **service pvs restart**

**Windows Only:**

# **<X>:\> net stop “Tenable PVS Proxy**

# **<X>:\> net start "Tenable PVS Proxy"**

Configuring SecurityCenter to authenticate with PVS is done in almost the exact same manner as it is done for Nessus. A file containing the SSL public certificate and private key is loaded onto the SecurityCenter web server via the SecurityCenter webUI.

1. Combine the certificate and key to create a new file that will be uploaded into SecurityCenter (this process is identical for PVS and Nessus):

**Linux Only:**

# **cat /root/pvs-user-cert.pem /root/pvs-user-key.pem > /root/pvs-auth-cert.pem**

**Windows Only:**

# **<x>:\> type nessus-user-cert.crt > nessus-auth-cert.pem**

# **<x>:\> type nessus-user-cert.key > nessus-auth-cert.pem**

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | The authentication file should contain the certificate and key with no blank lines or extra characters.  -----BEGIN CERTIFICATE-----  <Large alphanumeric string>  -----END CERTIFICATE-----  -----BEGIN RSA PRIVATE KEY-----  <Large alphanumeric string>  -----END RSA PRIVATE KEY----- |

2. Copy the combined cert/key file to a location where it is accessible from a browser that you use to connect to the SecurityCenter server:

3. Restart the PVS services:

**Linux Only:**

# **service pvs restart**

**Windows Only:**

# **<X>:\> net stop “Tenable PVS Proxy”**

# **<X>:\> net start “Tenable PVS Proxy”**

4. Log into SecurityCenter as an Administrator; select **Resources**->**Passive Scanners**; then select a passive scanner or create a new scanner. For Authentication Type, select **SSL Certificate** and browse to the certificate you created in Step 1. Then click **Submit**.

5. Verify the configuration works properly by navigating to the Passive scanners page. The scanners status should go to 'Updating Status' and then back to 'Working' or 'Results Ready'.

6. Select the scanner again and select "Verify Hostname." This verifies that the hostname or IP address used to connect to the Passive scanner matches the CN in the PVS scanner's SSL certificate.

7. Verify the configuration works properly by navigation to the PVS scanners page. The scanners status should go to 'Updating Status' and then back to 'Working'.

# Appendixes

## Appendix A: Root Certificate Installation / Verification

### Verify Proper Root Certificate Installation for SecurityCenter

Command line access is required if you need to check the installation of the Root certificates. Each file should be owned by tns:tns and with its permissions set at 640. In addition to the CA certificates, there also a link file with a string of characters pointing back to each certificate file (.pem) in the CA directory.

The easiest way is to view these files within the SecurityCenter CA directory is with the ‘**ls**’ command:

# **ls -la --sort=t /opt/sc/data/CA/**

The expected output should list a certificate file and link for all CAs:

lrwxrwxrwx 1 root root 9 Aug 19 02:50 12d46639.0 -> CA-27.pem

-rw-r----- 1 tns tns 3932 Aug 19 02:50 CA-27.pem

### Alternate Root Certificate Installation for Nessus/PVS (For a Windows Host)

If a Microsoft Windows computer is used for the scanner, the DoD root certificates may already be installed as part of the deployment process. If the root certificates are not installed, the appropriate InstallRoot script can be used to install them (See [Appendix D](#_Appendix_D:_DoD) –DoD Tools and Reference Documents).

To manually install the root certificate for Nessus/PVS on a Windows host, use the following instructions:

* From your desktop, open a Microsoft management console by holding down the Windows key (**)** and pressing the **’R’** key simultaneously. This will launch the Run command dialog box.
* In the text field of the Run dialog box, type ‘**mmc**’ to launch the Microsoft management console.
* From the management console, hold down the Control key (**’Ctrl’**), while pressing the **’M’** key to launch the ‘**Add or Remove Snap-ins**’ list builder window. The window includes a pair of list boxes – one with the list of available snap-ins, and the other for selected snap-ins.
* From the list of available snap-ins, select the ‘**Certificates**’ snap-in, and then click on the ‘**Add**’ button; next, select ‘**My user account**’ in the ‘**Certificates snap-in**’ window. Then click the ‘**Finish**’ button. Now, click ‘**OK**’ in the ‘**Add or Remove Snap-ins**’ window.
* Expand ‘**Certificates – Current User**’ under the ‘**Console Root’** folder. Then expand ‘**Trusted Root Certification Authorities**’ and click on ‘**Certificates**’.
* Scroll down the list and find the ‘**DoD Root CA 2**’ certificate. Double-click on this certificate. Next, click on the ‘**Details**’ tab and then click on the ‘**Copy to File…**’ button.
* Click ‘**Next**’ then select ‘**Base-64 encoded X.509 (.CER)**’ as the format you want to use, and click ‘**Next**’ again.
* You will be asked to provide a file name. Name the file ‘**cacert**’, then click ‘**Next**’, and then the ‘**Finish**’ button.
* You should be prompt with the following message in a dialog box: “The export was successful.” Click ‘**OK**’.
* Click ‘**OK**’ again in the ‘**Certificate**’ window to close it out.
* Look for the certificate that you exported. The file should have been automatically saved as ‘**cacert.cer**’ and exported to your local user account folder.
* Copy your certificate file to:

For Nessus:

**<X>:\ProgramData\Tenable\Nessus\nessus\CA**

For PVS:

**<X>:\ProgramData\Tenable\PVS\pvs-proxy\ssl**

* Rename your ‘**cacert.cer**’ file to ‘**cacert.pem**’ (so that the file format is changed from a CER file to a PEM file). If there is already a file with that same filename, replace it.

## Appendix B: Troubleshooting

### Compare SSL Certificate, Private Key, and CSR

If you receive an error that the private key does not match the certificate, or that a certificate you installed to a site is not trusted, try one of these commands:

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | Check the SHA-1 hash of the certificate to ensure it matches the hash of a CSR or private key. |

# **openssl x509 -noout -modulus -in SecurityCenter.crt | openssl sha1**  
# **openssl rsa -noout -modulus -in SecurityCenter.key | openssl sha1**# **openssl req -noout -modulus -in dodserver-pkcs10.csr | openssl sha1**

### Debug OpenSSL

If you receive an error that the private key does not match the certificate, or that a certificate you installed to a site is not trusted, try this command:

|  |  |
| --- | --- |
| 11769225_Caution_HiRes.png | Check the SSL connection. The web server should preset its identity certificate (and any intermediates). |

# **openssl s\_client -connect** [**<IP SecurityCenter is listening on>:443**](http://www.disa.mil:443)

### Issues with uploading Custom CA Certificate

If you are having issues with uploading custom CA certificates, check the following items:

1. Custom\_CA.inc format

The CA certificate should be in PEM (Base64) format. To verify, open it in a text editor. The certificate should be between -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----. If you do not see those lines, it is in the wrong format and should be changed to PEM (Base64) format.

2. /opt/sc/data/customNasl/custom\_CA.inc

If the SecurityCenter installation is not on the Appliance, check the uploaded custom\_CA.inc with the following command: # cat /opt/sc/data/customNasl/custom\_CA.inc

The output should match the custom\_CA.inc file that you checked in a text editor in step T1 above. If the file does not exist, the upload was not successful. If the file does not match, the most recent upload may not have been successful. Go over the steps above for creating and uploading upload\_this.tar.gz and ensure it is being done correctly.

3. /opt/nessus/lib/nessus/plugins/custom\_CA.inc (Linux) **or** \ProgramData\Tenable\Nessus\nessus\plugins\custom\_CA.inc (Windows)

If Nessus is not on the Appliance, navigate to the plugins folder and cat or type custom\_CA.inc to verify it exists and matches the custom\_CA.inc file contents verified in steps 1 and 2 above. If custom\_CA.inc does not exist in the plugins folder, or does not match the most recent custom\_CA.inc in SecurityCenter, it has not propagated to the scanner. Check Resources > Nessus Scanners in SecurityCenter to see if the scanner is still updating plugins. If it is in a Working state, try updating the active plugins in SecurityCenter to prompt a plugin push. See note N2 above about plugin feed versions. If the plugin feed version has not incremented and the customer really has to push plugins right now, see the following article: Force plugin update on scanner managed by SecurityCenter (Comparable to nessus-update-plugins -f)

4. Plugin output

Adding the custom CA certificate to custom\_CA.inc will not resolve the issue if the issue is something else - the service is missing intermediate certificate(s), the service has a self-signed or default certificate (if not self-signed with the server name, it may be issued by a vendor name like "Nessus Certification Authority") and not a certificate signed by their custom CA at all, the certificate is expired, etc. Look at the detailed plugin output of 51192 to see exactly why the certificate is untrusted. If custom\_CA.inc will fix it, the output will say that the certificate at the top of the certificate chain is unrecognized, and the certificate it shows will be either issued by the custom CA (matching the name \*exactly\*) or the actual custom CA self-signed certificate.

Delete this text and replace it with your own content.

## Appendix C: Sample CSR Configuration File

This sample .cfg file is a configuration file that is used in the CSR generation process. If you wish to create your own instead of using the dod-openssl.cfg file in the “[Generate Private Key and CSR](#_Generate_Private_Key)” section, use the template below as a reference. Save this file as dodcert.cfg. When a CSR is generated, this file will be referenced to ensure that the required questions are asked. It also provides the default values. It ensures that the proper DN is generated for your certificate request.

[ req ]

default\_bits = 2048

default\_keyfile = dodserverkey.key

distinguished\_name = req\_distinguished\_name

default\_md = sha256

dirstring\_type = nobmp

[ req\_distinguished\_name ]

C = Country Name (2 letter code)

C\_default = US

C\_min = 2

C\_max = 2

O = Press Enter

O\_default = U.S. Government

O\_max = 64

0.OU=OU=DoD Press Enter

0.OU\_default = DoD

0.OU\_max = 64

1.OU=OU=PKI Press Enter

1.OU\_default = PKI

1.OU\_max = 64

2.OU = Military/Government Component (i.e. USN, USAF, USA, etc)

2.OU\_max = 64

CN = Common Name (FQDN or routable IP of server)

CN\_max = 64

## Appendix D: DoD Tools and Reference Documents

### Components required for ACAS Public Key Enablement:

PKI and PKE Tools (PKCS #7 Bundles):

<http://iase.disa.mil/pki-pke/Pages/tools.aspx>

CRLAutoCache\_Linux-2.0.5 NIPR:

<https://powhatan.iiie.disa.mil/pki-pke/downloads/unclass-crlautocache_linux_2-05_nipr.tar.gz>

CRLAutoCache-Linux-2.0.5 SIPR:

<https://powhatan.iiie.disa.mil/pki-pke/landing_pages/downloads/unclass-crlautocache_linux_s-2-05_sipr.tar.gz>

### Optional information to assist with ACAS Public Key Enablement:

Public Key Enabling Apache 2.2 HTTP Web Server for Linux:

<https://powhatan.iiie.disa.mil/pki-pke/landing_pages/downloads/unclass-rg_public_key_enabling_apache.pdf>

Key Store Procedures:

<https://powhatan.iiie.disa.mil/pki-pke/downloads/pdf/fouo_RG-KeyStoreTools_Procedures.pdf>

Protecting Your Private Key:

<https://powhatan.iiie.disa.mil/pki-pke/landing_pages/downloads/unclass-ss_protecting_private_key.pdf>

CSR Request Procedures:

<https://powhatan.iiie.disa.mil/pki-pke/downloads/pdf/fouo_RG-Windows_CertRequest-Openssl.pdf>

## Appendix E: Root Certificates Detail

SecurityCenter uses the installed root certificates to authenticate client certificates presented to the SecurityCenter web server, and to authenticate the DISA web servers used for plugin updates. As such, the appropriate root certificates need to be installed for automatic plugin updates to function properly.

The guidance in ‘[Install the DoD Root Certificates](#_Install_the_DoD)’ section of this document provides the required certificates to update from the NIPRNet plugin server. SIPR root certificates are required to update from the SIPRNet plugin server.

Four certificate roots are tested and automated scripts provided by ACAS development teams. These include the DoD root certificates (A), ECA root certificates (E), JITC root certificates (J), and DoD Secret root certificates (S). Other root certificates can be installed using similar procedures.

To manually install ECA root certificates, you need to replace the A/a in the file and directory names provided in the according to the following directions:

# **unzip –o unclass-installroot\_v\*a.zip -d .**

OR

# **unzip -o Certificates\_PKCS7\_\*\_DoD.zip -d .**

Would need to be updated:

# **unzip –o unclass-installroot\_v\*e.zip -d .**

OR

# **unzip -o Certificates\_PKCS7\_\*\_ECA.zip -d .**

At this time, there is no **CRLAutoCache** script provided for ECA or JITC certificates, so caution is recommended if these certificates are used in a production environment.

## Appendix F: Determining your CAC Certificate CN

Every PKI certificate has a Common Name or CN. The CN of a CAC certificate consists of a user’s last name, first name, middle initial, and a 10-digit identifier called an EDIPI. It should look like this: **DOE.JANE.Q.123456790**

The best way to find the CN on your CAC:

1. Open Internet Explorer. From the toolbar, select ‘**Tools**’ then select ‘**Internet options**’.
2. Select the ‘**Content**’ tab; then select ‘**Certificates**’.
3. Select the ‘**Personal**’ tab.
4. Double click one of your certificates.
5. Select the ‘**Details**’ tab; then select the field labeled ‘**Subject**’.
6. Inside the box in the lower half of the window, you will see ‘**CN = LASTNAME.FIRSTNAME.MI.EDIPI**’.

## Appendix G: Linux File Handling

### SSH and SCP Client

Linux utilizes Secure Shell (SSh or SSH) as the primary mechanism for remote administration. SSH provide cryptographic confidentiality and integrity for running a shell (CLI) on a remote host. Red Hat Linux provides a SSH client, but Microsoft Windows does not provide a SSH client and the ACAS project does not recommend / endorse a specific product. While these software packages are not popular with Windows administrators, they are frequently used by network administrators. ACAS administrators may want to consult with their network admins prior to selecting a SSH client. As a starting point, the most popular windows SSH client is probably a program called putty; the program is free but no support is available from the vendor (freeware).

PuTTY: <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

Moving files is best done using Secure Copy (SCP) or Secure File Transfer Protocol (SFTP), which is a sub function of SSH. A number of SCP clients are available, and it is encouraged that ACAS administrators consult with network administrators prior to selecting a SCP/SFTP client. As a starting point, the most popular windows SCP client is probably WinSCP; the program is free but no support is available from the vendor (freeware).

WinSCP: <https://winscp.net/eng/index.php>

### Text Editors

There are two primary text editors on the kickstart images provided to ACAS customers: vi and nano. The vi editor has a comparatively steep learning curve, while nano utilizes many of the same keyboard combination / shortcuts (ctrl+s to save ctrl+c to copy). Neither editor will alter the file in an unexpected manor which can happen when files are edited on windows workstations. The RHEL 6 image includes a version of vi called vim, which provides ‘syntax highlighting’ this may prove helpful or hinder casual use, if the vi command is calling vim and you do not want to use it, you can call vi explicitly:

# **/bin/vi /path/to/file.to.edit**

## Appendix H: Acronyms Used in This Document

ACAS Assured Compliance Assessment Solution

CA Certificate Authority

CAC Common Access Card

CN Common Name

COTS Commercial Off-The-Shelf

CRL Certificate Revocation List

CSR Certificate Signing Request

DoD Department of Defense

EOL End-Of-Line

FQDN Fully Qualified Domain Name

LRA Local Registration Authority

OCSP Certificate Revocation Checking Using Online Certificate Status Protocol

PKE Public Key Enabled

PKI Public Key Infrastructure

PVS Passive Vulnerability Scanner

RA Registration Authority

webUI web user interfaces

## Appendix I: Useful links

***ACAS Customer Support/OKC Helpdesk***

* Toll-free: 844-347-2457 (Select options 1,5, and 3)
* DSN: 850-0032 (Select options 1,5, and 3)
* Email: [disa.tinker.esd.mbx.okc-disa-peo-service-desk@mail.mil](mailto:disa.tinker.esd.mbx.okc-disa-peo-service-desk@mail.mil)

***ACAS Homepage***

<https://disa.deps.mil/ext/cop/mae/netops/acas/SitePages/Home.aspx>

Contains access to the following: ACAS License Request Portal, ACAS Build 1 Request Portal, ACAS Working Group information, ACAS Training, and other important ACAS related information.

***Approved documentation/binaries are located on DoD Patch Repository***

Posted at: <https://patches.csd.disa.mil/CollectionInfo.aspx?id=442>

(CAC is required for access). Click on ACAS > ACAS Software > then whichever application you need. All of our latest Plugins files and Red Hat patches can be found here also.

***SoftwareForge***

<https://software.forge.mil/sf/projects/acas>

All of our test and development efforts are located here and do not represent the approved baselines.

***Certification and Accreditation Artifacts***

Posted at ACAS SIPR Wiki: <https://www.intelink.sgov.gov/wiki/ACAS>

***ACAS Front Door:***

<http://www.disa.mil/Cybersecurity/Network-Defense/ACAS>