



## Cisco NCS Server Hardening

This appendix provides an instructional checklist for hardening a NCS server. Ideally, the goal of a hardened server is to leave it exposed on the Internet without any other form of protection. This describes the hardening of NCS, which requires some services and processes exposed to function properly. Think of it as NCS Best Practices. Hardening of NCS will involve disabling unnecessary services, removing and modifying registrykey entries, and applying appropriate restrictive permissions to files, services, and end points.

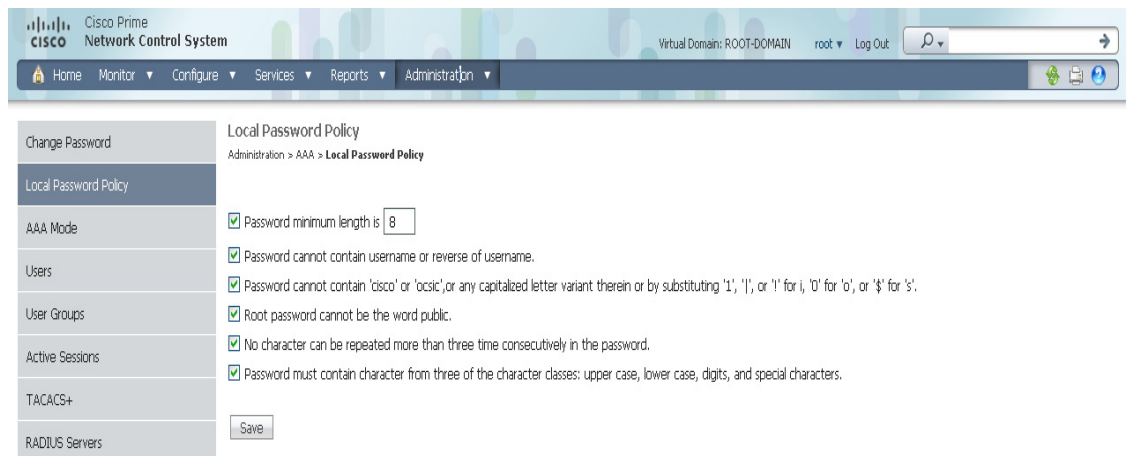
This appendix includes the following sections:

- [NCS Password Handling, page C-1](#)
- [Setting Up SSL Certification, page C-2](#)

## NCS Password Handling

You can configure additional authentication by configuring the **Local Password Policy** parameters. Select the check boxes if you want the configurations to be enabled.

**Figure C-1** Local Password Policy



The following configurations are added for additional authentication:

- You can configure the minimum length of the password.

- You can configure if you want to allow the username or reverse of the username to be part of the password.
- You can configure if the password can contain 'cisco' or 'ocsic', or any capitalized letter variant therein or by substituting '1', 'l', or '!' for 'i', '0' for 'o', or '\$' for 's'.
- You can configure if the root password can be the word **public**.
- You can configure if a character can be repeated more than three times consecutively in the password or not.
- You can configure if the password must contain character from three of the character classes: upper case, lower case, digits, and special characters.

## Setting Up SSL Certification

The Secure Socket Layer (SSL) Certification is to ensure secure transactions between a web server and the browsers. Installing the DoD Certificates will allow your Web browser to trust the identity and provide secure communications which are authenticated by Department of Defense (DoD).

These certificates are used to validate the identity of the server or web site and are used to generate the encryption key used in the SSL. This encryption protects the information being passed between the server and the client.

SSL Certification involves the following topics:

- [Setting Up SSL Client Certification, page C-2](#)
- [Setting Up SSL Server Certification, page C-3](#)

## Setting Up SSL Client Certification

Follow the below steps to setup the SSL Client Certificate Authentication using DoD certificates:



**Note** As a prerequisite, to create the SSL Certificates, you would require “KeyTool” available in JDK. KeyTool is a command line tool to manage keystores and the certificates.

**Step 1** Create SSL Client Certificate using the below command.

```
% keytool -genkey -keystore nmsclientkeystore -storetype pkcs12 -keyalg RSA -keysize 2048
-alias nmsclient -dname "CN=nmsclient, OU=WNBU, O=Cisco, L=San Jose, ST=CA, C=US"
-storepass nmskeystore
```



**Note** Provide the Key Algorithm as RSA and KeySize as 1024 or 2048.

**Step 2** Generate the Certificate Signing Request (CSR) using the below command.

```
% keytool -certreq -keyalg RSA -keysize 2048 -alias nmsclient -keystore nmsclientkeystore
-storetype pkcs12 -file <csrfilename>
```



**Note** Provide the Key Algorithm as RSA and KeySize as 1024 or 2048 and provide a certificate file name.

**Step 3** Send the generated CSR file to DoD. The DoD will issue the corresponding signed certificates.




---

**Note** The CSR reply is through dod.p7b file. In addition you should also receive the root CA certificates.

---




---

**Note** Please makes sure to retrieve the PKCS7 encoded certificates; Certificate Authorities provide an option to get the PKCS7 encoded certificates.

---

**Step 4** Import the CSR reply in the Keystore using the command:

```
% keytool -import dod.p7b -keystore nmsclientkeystore -storetype pkcs12
-storepass nmskeystore
```

**Step 5** Check the formats of root CA certificates recieved, they must be base 64 encoded. If they are not base 64 encoded, use the OpenSSL command to convert them to base 64 encoded format.

```
% openssl x509 -in rootCA.cer -inform DER -outform PEM -outfile rootCA.crt
% openssl x509 -in DoD-sub.cer -inform DER -outform PEM -outfile rootCA.crt
```




---

**Note** Convert both root CA certificate and sub-ordinate certificates recieved.

---

In case you recieved both root CA certificate and the sub-ordinate certificate, you have to bundle them together using the below command:

```
% cat DoD-sub.crt > ca-bundle.crt
% cat DoD-rootCA.crt >> ca-bundle.crt
```

**Step 6** To setup SSL Client Authentication using these certificates, enable SSL Client Authentication in Apache in the **ssl.conf** file located in <NCS\_Home>/webnms/apache/ssl/backup/ folder.

```
SSLCAcertificationPath conf/ssl.crt
SSLCAcertificationFile conf/ssl.crt/ca-bundle.crt
SSLVerifyClient require
SSLVerifyDepth 2
```




---

**Note** SSLVerifyDepth will depend of the level of Certificate Chain. In case you have only 1 root CA certificate, this should be set to 1. In case you have a certificate chain (root CA and subordinate CA), this should be set to 2.

---

**Step 7** Install the DoD root CA certificates in NCS.

**Step 8** Import the nmsclientkeystore in your browser.

## Setting Up SSL Server Certification

Follow the below steps to setup the SSL Server Certificate using DoD certificates:

**Step 1** Generate the Certificate Signing Request (CSR).

```
% keyadmin -newdn genkey <csrfilename>
```

**Step 2** Send the generated CSR file to DoD. The DoD will issue the corresponding signed certificates.




---

**Note** The CSR reply is through dod.p7b file. In addition you should also receive the root CA certificates.

---



---

**Note** Please make sure to retrieve the PKCS7 encoded certificates; Certificate Authorities provide an option to get the PKCS7 encoded certificates.

---

**Step 3** Import the Signed Certificate using the below command in the Keytool:

```
% keyadmin -importsignedcert <dod.p7>
```



---

**Note** The NCS stores the self-signed certificate at `/opt/CSCOncs/httpd/conf/ssl.crt`. The imported certificates/keys are stored at `/opt/CSCOncs/migrate/restore`.

---