To subscribe to updates watch my GitHub repository. New posts get added as pull requests.

Set up SSL on Jenkins

04 May 2016

Jenkins can be an integral piece of software used to speed up development and deliver production services to customers. Because of this, security in Jenkins must be carefully implemented. This brief guide will walk a Jenkins sysadmin through setting up SSL for Jenkins.

Assumptions

This guide assumes:

- Prior knowledge of public key infrastructure (PKI).
- RedHat or CentOS GNU/Linux for file paths. Jenkins specific knowledge translates across distros.
- The hosting server has a fully qualified domain name.
- You installed Jenkins via a system package.
- You're using Jenkins Winstone (the Jenkins built-in web server) which is the default.
- You're running Jenkins as a normal user (e.g. `jenkins`). Normal users can't listen on ports less than 1024 in Linux. *Note: If you wish to use 443 instead of 8443 I recommend using `iptables` to forward ports.*

Create a Java Keystore

Generating keys and obtaining signed certificates is outside of the scope of this article. What you need first:

1. A private SSL key (`example.com.key`).
2. An SSL certificate signed by a certificate authority (`example.com.crt`).
3. The SSL certificate of the certificate authority which did the signing (`ca.crt`).

To generate a Java Keystore requires:

1. Reference your SSL certificates and key (listed above).
2. Convert the SSL certificates into an intermediate format (PKCS12 keystore).
3. Convert the intermediate format into a Java Keystore (JKS keystore).

The following command will convert SSL certificates into the intermediate format (PKCS12).

```
openssl pkcs12 -export -out jenkins_keystore.p12 \
  -passout 'pass:changeit' -inkey example.com.key \
  -in example.com.crt -certfile ca.crt -name example.com
```

Then convert the intermediate format (PKCS12) to Java Keystore (JKS). This way Jenkins can use the JKS.

```
keytool -importkeystore -srckeystore jenkins_keystore.p12 \
  -srcstorepass 'changeit' -srcstoretype PKCS12 \
  -srcalias example.com -deststoretype JKS \
  -destkeystore jenkins_keystore.jks -deststorepass 'changeit' \
  -destalias example.com
```

Some notes:

- The `-name`, `-srcalias`, and `-destalias` must be the same as the domain name (e.g. `example.com`).
- The password for the PKCS12 store and the JKS store must be **exactly the same**. Jenkins will fail to start if it is not the case.
- You should probably use a different password than `changeit` (which is conventional knowledge for Java keystores). It will protect you if the keystore is compromised and downloaded but the password is not known. Though, in a system breach, it is wise to revoke and request new certificates.
- The final keystore file is `jenkins_keystore.jks`.

Move the keystore to a place which can be referenced in Jenkins configuration.

```
mkdir -p /etc/jenkins
cp jenkins_keystore.jks /etc/jenkins/
```

```
#configure permissions to secure your keystore
chown -R jenkins: /etc/jenkins
chmod 700 /etc/jenkins
chmod 600 /etc/jenkins/jenkins_keystore.jks
```

Configure Jenkins to use SSL

Edit `/etc/sysconfig/jenkins` and set the following variables.

```
#disable HTTP
JENKINS_PORT="-1"

#configure HTTPS
JENKINS_HTTPS_PORT="8443"
JENKINS_HTTPS_KEYSTORE="/etc/jenkins/jenkins_keystore.jks"
JENKINS_HTTPS_KEYSTORE_PASSWORD="changeit"
JENKINS_HTTPS_LISTEN_ADDRESS="127.0.0.1"
```

To learn more about these options you can execute (while Jenkins is running):

```
java -jar /usr/lib/jenkins/jenkins.war --help
ps aux | grep 'jenkins\.war'
```

See `/var/log/jenkins/jenkins.log` for potential troubleshooting.

Configure iptables to port forward

`iptables` can be used to forward port 443 to 8443. This provides a better user experience since this is the standard port for HTTPS connections. To do this on a RedHat system a few things must occur:

1. The kernel must allow forwarding.
2. iptables needs rules for forwarding.

Some notes:

- The recommendations are relevant to interface `eth0`. Double check you're using the correct interface for your system.

Configure kernel to allow port forwarding

To enable forwarding during runtime:

```
#allow forwarding
sysctl -w net.ipv4.ip_forward=1
#allow forwarding to localhost on eth0
sysctl -w net.ipv4.conf.eth0.route_localnet=1
```

or alternatively

```
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv4/conf/eth0/route_localnet
```

For forwarding settings to persist on reboot, add or change the following settings in `/etc/sysctl.conf`.

```
net.ipv4.ip_forward = 1
net.ipv4.conf.eth0.route_localnet = 1
```

Configure iptables

Depending on how `iptables` is configured you may have to adapt how you insert the rules into the runtime firewall. It may be simpler to write your rules in `/etc/sysconfig/iptables` and reload the firewall.

```
#for remote connections
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j DNAT --to-destination 127.0.0.1:8443
iptables -A FORWARD -i eth0 -m state --state NEW -m tcp -p tcp -d 127.0.0.1 --dport 8443 -j ACCEPT

#for localhost connections
iptables -t nat -A OUTPUT -p tcp --dport 443 -d 127.0.0.1 -j DNAT --to-destination 127.0.0.1:8443
```

Use `iptables-save` to update `/etc/sysconfig/iptables` rules.

Verify SSL is running

Ensure a correct certificate chain order. The following command also accounts for Server Name Indication (SNI).

```
openssl s_client -connect example.com:443 -servername example.com < /dev/null
```

View certificate information of the remote service.

```
openssl s_client -connect example.com:443 -servername example.com < /dev/null | openssl x509 -text | less
```

By passing a null byte to stdin in the above commands, `openssl` is signaling an immediate close of the connection. For more information about the above commands, see the man pages.

- `man x509`
- `man s_client`

If you visit your secured Jenkins and find your browser is not showing a secure connection, then you're likely missing the certificate authority certificate in your browser trust store.

Cited references

These are some links I referred when doing my research to formalize and write up this post.

- Running Jenkins on Port 80 or 443 using iptables.
- RHEL Security Guide iptables forward.
- ServerFault: iptables port redirect not working for localhost.
- StackExchange: iptables redirect outside requests to 127.0.0.1.

Advanced Resources

These resources and scripts are intended for the initiated. I can do follow-up blog posts on any of them. Just ask in the comments.

- genSAN.sh - A script to generate a private key and a single domain or SAN certificate signing request (CSR) to be signed by a certificate authority.
- provision_jenkins.sh - A script to quickly play and provision Jenkins with ease.
- keystore.sh - My reference for converting X.509 certificates to JKS. It is a script I wrote quite a while ago when I was more familiar with the topic.
- repo.jenkins-ci.org - A Maven repository where the Jenkins community publishes the Jenkins WAR file and plugin HPI files. Search `*jenkins*VERSION.war` to download an exact version of Jenkins where `VERSION` is the version you're running in production.

This article was last updated Nov 19, 2017.

Posted in Engineering *with tags: Jenkins, Security*

Tweet

Comments:

Diana • 7 months ago

Hello,
I am new to SSL and Jenkins as well, I have openssl 1.0 and Jenkins running on rhel 7. I have configured ssl with a CA signed cert and can successfully browse to my system and view an Apache test page via https.
Do I use my existing crt and key, in the process above to create the keystore?

Reply • Share ›

Sam Gleske Mod ➚ Diana • 4 months ago

Hi Diana,
The process outlined in this blog post is referring to running Jenkins without Apache at all. If you're running Apache out front then the configuration would be different. Either by using Apache to pass raw TCP connections (which can easily be done by iptables, instead); or by terminating SSL with Apache which would also look different than the configurations discussed in this post.

Reply • Share ›

Steven Gardner • 9 months ago

I'm fairly new to working with SSL certificates. After creating the Java keystore from the private key and two SSL certs, do you need to add the SSL certificate from the certificate authority to cacerts? When starting Jenkins with the new configuration, the jenkins.log says that SSHD has been started and I don't see any errors; however when I try to access Jenkins in a browser at port 8443, I get ERR_CONNECTION_REFUSED

Reply • Share ›

Steven Gardner ➚ Steven Gardner • 9 months ago

Oops, figured it out, forgot to put the https prefix when accessing the site!

Reply • Share ›

Sam Gleske Mod ➚ Steven Gardner • 9 months ago

Ah, glad you got it.

Reply • Share ›