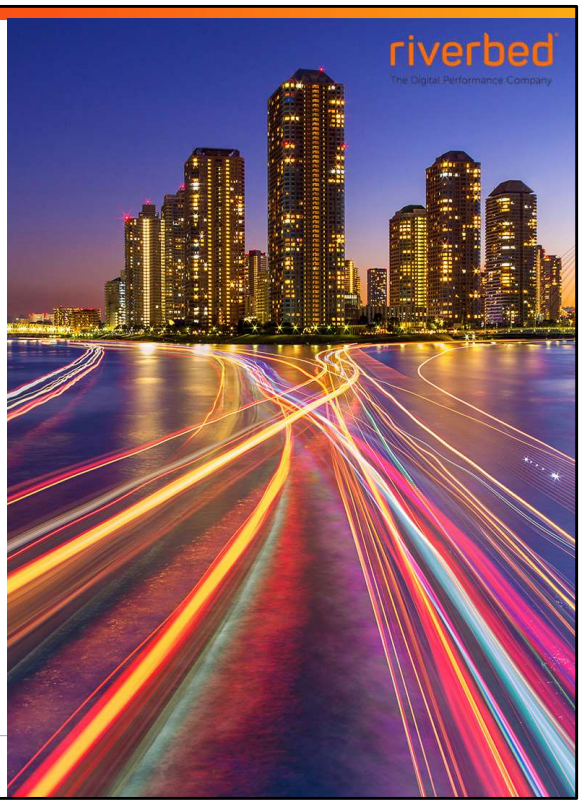
Optimize SSL Traffic

© 2020 Riverbed Technology. All rights reserved.

Learning Objectives

After completing this module, you will be able to:

- Describe the SSL protocol.
- State the purpose of digital certificates.
- Configure SSL optimization.
- Use SCC as a Certificate Authority.
- Configure advanced SSL features.
- Verify SSL optimization.



© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Key Points

- HTTPS is increasingly replacing HTTP for security reasons. For fast application access across the WAN, SSL optimization is often essential.
- Encrypted traffic is very difficult to optimize as WAN accelerators need to see clear, unencrypted traffic.
- Riverbed SteelHead appliances can be configured to optimize SSL traffic, enabling customers to have fast access to HTTPS applications as well as HTTP.

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 3

© 2020 Riverbed Technology. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

The Importance of Secure Communications

- In the digital age more and more business is performed electronically.
- Intercepting and stealing secure data is a major criminal undertaking.
- HTTP is everywhere, but is not inherently secure.
- *HTTP**S*** uses SSL/TLS to secure HTTP data.
- Percentage of HTTPS traffic rapidly increasing
  - *HTTPS is the new IP!*
- BUT …
  - To optimize this traffic you need to be able to decrypt it!
- This module will cover the mechanisms behind SSL/TLS and how we go about configuring a SteelHead to optimize the traffic.

© 2020 Riverbed Technology, Inc. All rights reserved. riverbed 5

For both private and business purposes the ability to transfer information safely and securely is important. In the digital age, more and more business is being performed electronically while at the same time there has been a concerted effort by criminal organizations and certain states to intercept and steal the information. For the individual it may be bank account details, and for the organization it may be sensitive emails, SharePoint documents, or even intranet passwords.

Increasingly, both email and general applications have been moving towards HTTP as their application layer protocol of choice (e.g., MAPI/HTTP) so to secure this, we now have secure HTTP or *HTTPS.* The technology securing this is known as *Secure Sockets Layer (SSL)* or *Transport Layer Security (TLS)*, the latter effectively being a later version of the former. However, for Riverbed SteelHead appliances to optimize this traffic they need to be able to decrypt it first and then re-encrypt it. In order to successfully configure this, an understanding of the conceptual details behind SSL/TLS is essential, so that is what we will be covering in this module.

© 2020 Riverbed Technology. All rights reserved.

SSL Encryption

Components of Encryption

- Two fundamental components:
  1. Cypher – Encryption Algorithm
  2. Key – Locking/Unlocking Mechanism
- Both sides need the key, so how do you transmit it without anyone seeing???

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 6

Imagine you are sitting at home in front of your PC and you need to have a secure conversation with someone. How do you go about doing that? You need to encrypt the data in such a way that the person you are talking to understands what you are saying and that you understand what they are saying but that nobody else does.

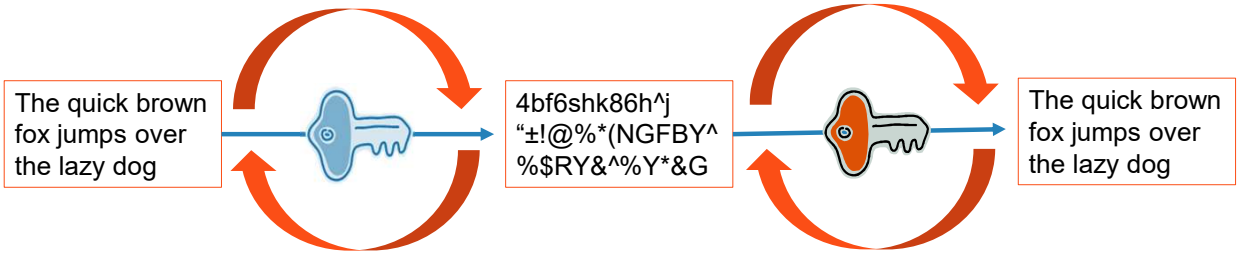
So what you need are two things: a *cypher* (a way of encrypting the data) and a *key* (the locking/unlocking mechanism which enabled you to encrypt in such a way that someone else can decrypt – the name for encryption which cannot be decrypted is *noise*).

There are different cyphers or *encryption algorithms* such as AES256, *Blowfish*, *Twofish* and *RSA.* But they are only useful if the encryption key can be negotiated securely.

Let us look at the two main types of encryption keys.

© 2020 Riverbed Technology. All rights reserved.

Symmetric Encryption



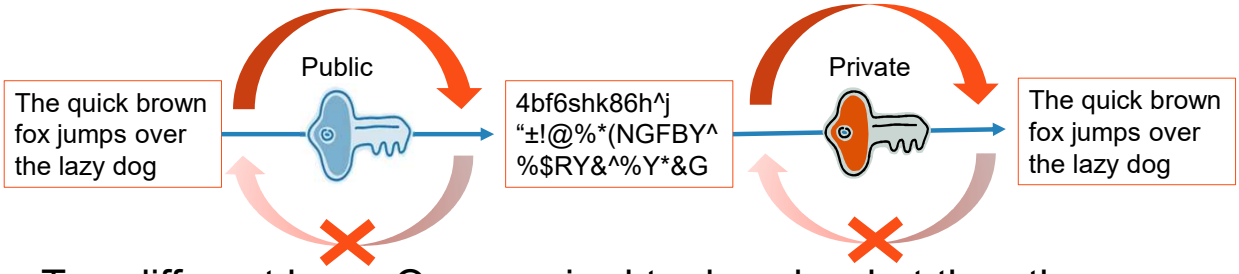
- One key to both encode and decode
- Sender encodes data with a key before sending
- Recipient decodes with *same* key to recover data

© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 7

This is the main form of encryption key. Both parties use the same key, which is used to both encrypt and decrypt.

© 2020 Riverbed Technology. All rights reserved.

Asymmetric Encryption



- Two different keys: One required to decode what the other encodes; one stays private while the other is public. The client codes with the Public key; the server decodes with the Private key.

Pros:
- No need to keep public key secret
- Senders using the public key cannot read each other’s messages

Cons:
- Need large keys (1024 – 2048 bit)
- Too slow for large messages

© 2020 Riverbed Technology, Inc. All rights reserved. riverbed 8

Here we have two keys: user1 has *key1* and user2 has *key2*. Importantly, what is encrypted with key1 can only be decrypted with key2 and what is encrypted with key2 can only be decrypted with key1, hence the term *asymmetric.*

The main point to understand here is that if I am user1 and have key1 and send a message to user2, I cannot decrypt my own message because my key can only be used to encrypt my messages and decrypt messages encrypted with key2.

© 2020 Riverbed Technology. All rights reserved.

How Does this Fit With SSL?

- SSL uses both!
- Asymmetric encryption can solve the weakness of symmetric encryption.
- Sender uses recipient's Public key to encode and send a secret, which can be used to calculate a symmetric key.
  - Secure distribution of symmetric key
- The symmetric key is chosen at random and is used exclusively for this conversation, preventing others from listening.
  - A *Session Key*

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 9

When a user in a company is connected to their corporate intranet site using SSL (or a user at home is using online banking) they are using a symmetrical key to transfer data back and forth between themselves and the web server.

The key question (so to speak) is: *how do they transfer/negotiate the symmetric key without someone else snooping in and learning the key?* The answer is *Asymmetric* encryption. The server has one key, the *private key,* which is kept secret, and then all of the users have same *public key* (hence why it is called *public*).

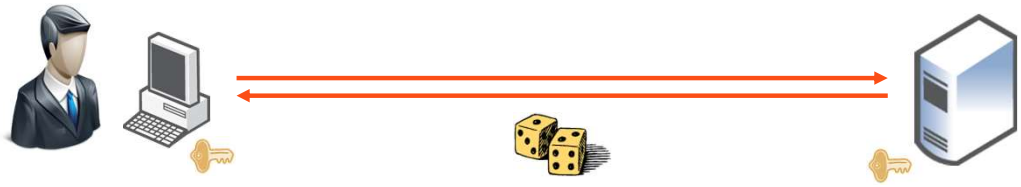
- The client can encrypt a secret message to the server using the server's public key.
- This secret can be used to calculate the symmetric key for communication, called a *session key* (highlighting its intended temporary nature).
- The client is safe in the knowledge that only the server with the corresponding private key can decrypt the secret and calculate the session key.

© 2020 Riverbed Technology. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

How do we know we have the correct Public Key?

- How does the client know he has this?



- And not this?

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 11

So, the bank gives the client their public key, the client uses this to send the bank a secret message, which is used to calculate the symmetric session key. No one knows the session key because it was encrypted in a way that only the bank could decrypt. Or was it? Consider the following situation.

In this case, something rather nasty has placed itself between the client and the bank. The client does not know about this and has inadvertently taken the attacker's public key rather than the bank's. This means that the attacker now has the session key and can read all of the user's data. This is called a *man-in-the-middle attack.*

So, the question is: *When the user accepts a public key how can he be sure it is the correct public key and not an imposter's?* The answer is: *Digital Certificates!*

© 2020 Riverbed Technology. All rights reserved.

Verify the Public Key - Digital Certificates

- Certificate contains:
  - Subject's name and Public key
  - Serial, validity dates and any limits on use
  - Issuer's (the *Certificate Authority's)* name
- The certified "Subject" (Common Name/DN) must be resolvable via DNS
- Certificates are signed with the issuer's Private key
  - Creates an encrypted hash of the original certificate so that changing any field, such as the Common Name, gives Bad Signature errors
  - Ensures certificate validity and integrity

© 2020 Riverbed Technology, Inc. All rights reserved. riverbed 12

Certificates, of any kind, validate credentials.

And, certificates typically contain certain pieces of information.

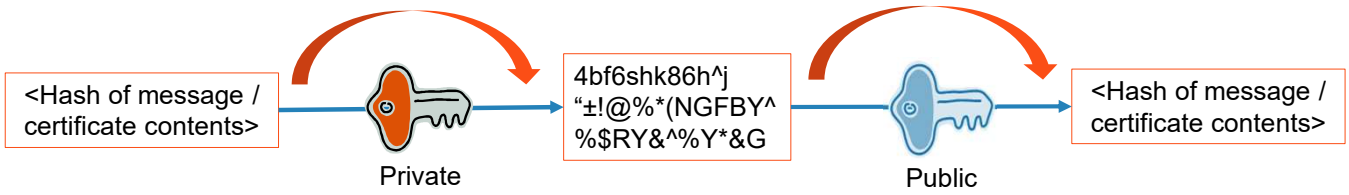
If you think about what nearly any certificate you might hold contains:
- The name of the person who has obtained the certificate
- When the certificate was issued and when it expires
- The subject or skill the certificate was awarded for
- The name of the organization who awarded the certificate, and
- A signature of a recognized authority figure

Now digital certificates work exactly the same way and have all these ingredients and more. They have a *Common Name*, identifying the server you are connecting to, an expiry date, an issuer and a signature. Let us look at the last two in a little more detail. The Common Name is important and is resolved by DNS, so it is important that you have name resolution configured correctly in order for this to work.

© 2020 Riverbed Technology. All rights reserved.

What is a Signature?

Encryption Backwards



- Messages encoded with the Private key can only be decoded with the corresponding Public key (opposite of normal).
- This is *Signing*. The Private key-holder or, in the case of Certificates, *Certificate Authority* vouches for the message.
- The *Signature* is a hash of the message body, encrypted with the Private key.
- If the Public key can successfully decode the signature and the result matches a hash of the message, then only the Private key holder could have signed it and it can not have been tampered with.

© 2020 Riverbed Technology, Inc. All rights reserved. riverbed 13

The signature is a crucial part of the certificate and serves a role similar to signatures in every day life. Think of a signature as encryption backwards.

The Certificate Authority (CA) signs the certificate by taking a hash of the certificate body (which in itself contains a hash of the public key) and then encrypts it with the private key. At first glance this sounds wrong - anything encrypted with the private key can be decrypted with the public key, which everyone has. This is true but, if the message can be decrypted with the public key, then it proves that the only person who could have encrypted it is the issuer, as they hold the private key which no one else has.

Finally, a quick hash of the message body can be compared with the decrypted hash and if they match you have confidence in both authenticity and integrity.

© 2020 Riverbed Technology. All rights reserved.

Self-Signed Certificates

A Certificate Signed with its Own Private Key

- Any host can sign their own certificate.
  - Self-signed certificates are considered trusted only when placed in a "trusted" location/directory (differs by OS/browser/application).
  - The trusted certificate signer becomes a Certificate Authority (CA).
  - Note: self-signed certificates can still be fake – what they can't be is real but tampered with.
- A company may have its own Certificate Authority, either *self-signed* or signed with a certificate bought from a public CA (can be used externally).
  - This creates a *certificate chain.*
- *Root CA* certificates are pre-loaded into modern web browsers so certificates can be verified using root CA public keys.

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 14

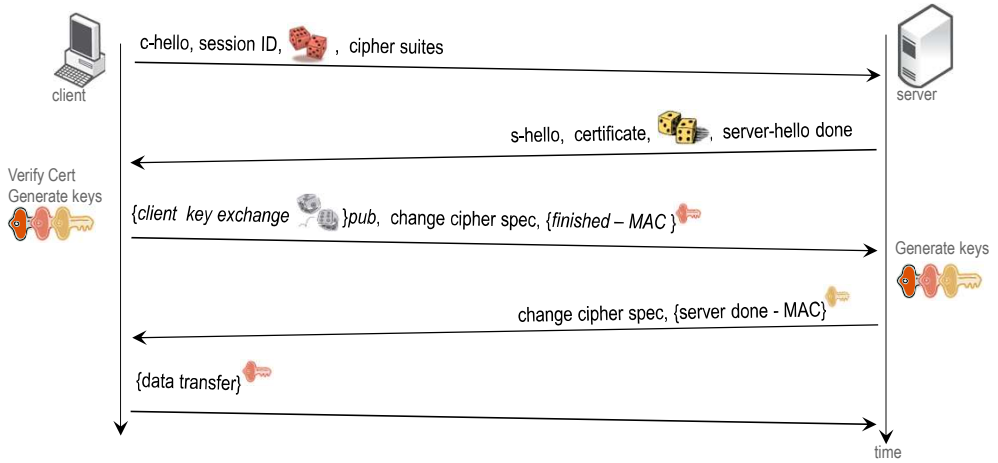
A self-signed certificate is a certificate that has been signed using its own private key – i.e., you have signed your own certificate. This does not prove that the certificate is genuine, what it does prove is that it has not been tampered with. Modern browsers have self-signed certificates of the major CAs installed. To see these in Chrome, just go to *Advanced* settings and then click *Manage Certificates.*

The issuer is really important; you need to trust who has provided the certificate. Fortunately, there are a number of *Certificate Authorities (CAs)*, such as GoDaddy and Verisign. Our browsers have these *root* CA's *self-signed certificates* installed and trust certificates signed by them.

You could pay a root CA to sign your certificates, but it would be much cheaper to setup your own, internal CA, and then just pay for the root CA to sign the certificate for your internal CA. In this way, you create a *certificate chain.*

© 2020 Riverbed Technology. All rights reserved.

Putting it all Together – SSL Handshake



- "cipher suites" include: encryption, authentication, key exchange and hash algorithms
- "change cipher spec" means: "everything after this is encrypted"
- "MAC" = Message Authentication Code: a hash of all previous handshake messages

© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 15

© 2020 Riverbed Technology. All rights reserved.

Online Certificate Status Protocol (OCSP) Stapling

Who to Trust?

- Public Key Infrastructure: policies, procedures, and protocols necessary to create, distribute, use, and revoke certificates
- Revocation procedures have traditionally not been very effective
  - Why? PKI is distributed: easy for a Certificate Authority (CA) to revoke a certificate, not easy for all clients that trust the CA to discover this in a timely fashion
- Ideas:
  - Certificate Revocation Lists (CRL)
    - Simple list consisting of serial numbers of revoked certificates
    - Problems: updated too infrequently, lists grow to unwieldy size
  - Online Certificate Status Protocol (OCSP)
    - Each CA issues certificates with an 'Authority Information Access' field pointing to a server called OCSP responder
    - Client extracts responder url from certificate and requests certificate status from OCSP responder
    - OCSP responder indicates to client whether certificate is good, revoked, or unknown
    - Problems: validation is responsibility of client, CA has to maintain an OCSP responder able to handle a large number of requests, validation would add latency to SSL handshake, little privacy

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 16

© 2020 Riverbed Technology. All rights reserved.

OCSP Stapling (continued)

Who to Trust?

- Client asks origin server for proof that offered certificate is valid
  - This is done as ‘status_request’ TLS extension in Client Hello
- Origin server periodically polls OCSP responder for OCSP response that indicates server certificate is valid
  - OCSP response signed by CA
  - OCSP responder only has to respond to origin servers
- Origin server sends OCSP response to clients that request it
  - ‘status_request’ set in Server Hello
  - OCSP response sent in ‘Certificate Status’ SSL handshake message
  - Problem: the OCSP response is timestamped, so there will be a window where the certificate could be revoked but still recognized as valid by the origin server

© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Certificate Authority (CA) Configuration

How to Configure your own CA in Linux

- At root issue the command:
  - `# apt-get install openssl`
- Once you have done this you will need to create your CA certificate::
  - `# openssl req -x509 -nodes -newkey rsa:2048 -keyout ca.key -out ca.cer -days 1095`
- At the resulting prompt fill in your CA details. For example:
  - `Country Name (2 letter code) [AU]:GB`
  - `State or Province Name (full name) [Some-State]:North Yorkshire`
  - `Locality Name (eg, city) []:York`
  - `Organization Name (eg, company) [Internet Widgits Pty Ltd]: Dave's CA Company`
  - `Organizational Unit Name (eg, section) []:Certificate Team`
  - `Common Name (e.g. server FQDN or YOUR name) []:ebor.mycompany.com`
  - `Email Address []:user@mycompany.com`
  - `#`

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 18

This part is more important than you might think as there are situations where you will need to set up your own CA and sign your own certificates. We will look at situations where this might be done in this module and also the later module on SteelHead SaaS. These commands are for Ubuntu/Debian but are broadly similar for RedHat.

At root issue the command:
apt-get install openssl (and people say Linux is hard!)

Once you have done this you will need to create your CA certificate. This is easy and is done using the following command:
 # openssl req -x509 -nodes -newkey rsa:2048 -keyout ca.key -out ca.cer -days 1095

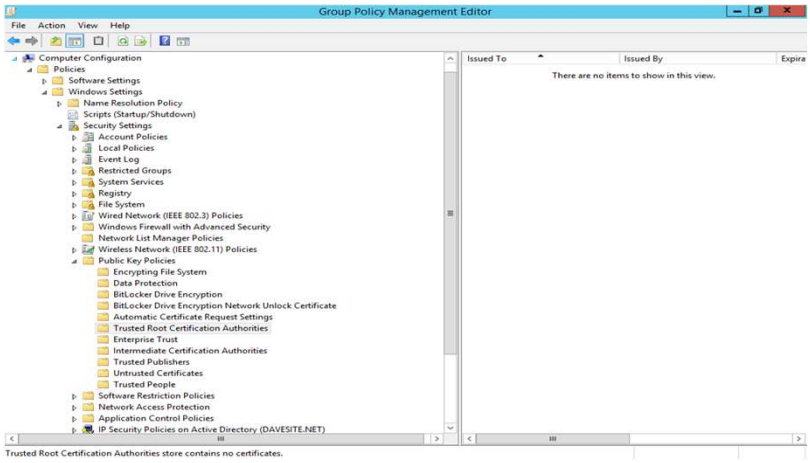
At the resulting prompt fill in your CA details.

© 2020 Riverbed Technology. All rights reserved.

Install this CA Certificate in your Windows Domain

Import the Certificate under *Default Domain Policy*

- Navigate to Tools > **Computer Configuration** > **Windows Settings** > **Security Settings**.
- Then click **Public Key Policies**.
- It can be installed as either a Root CA or an Intermediate CA.



© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 19

This point is important as the certificates you sign with this needs to be *trusted*.

In your Windows environment, import the file *ca.cer* as a Trusted Root Certification Authority by editing the Default Domain Policy.

This is done by using the Group Policy Management Editor, then navigate to the Default Domain Policy for your domain. Right-click to Edit then navigate to *Tools* > *Computer Configuration* > *Windows Settings* > *Security Settings* and click *Public Key Policies*. You can install the certificate as either a Root CA or as an intermediate CA.

The above screenshot shows where you need to be for this (in Windows2012R2).

© 2020 Riverbed Technology. All rights reserved.

Use your CA Certificate

Sign a Certificate Signing Request (CSR)

- Generate certificate signing request:
  - `# openssl req -newkey rsa:2048 -nodes -keyout mysite.key -out mysite.csr`
- Sign the request using your CA key:
  - `# openssl x509 -req -days 1095 -in mysite.csr -CA ca.cer -CAkey ca.key -CAcreateserial -out mysite.cer`
- You can now import into either IIS, Apache or even a SteelHead.

© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 20

To sign a certificate using this CA certificate the first thing you will need to do is create a *certificate signing request*. This is very easy in Linux and can be performed with just one command as follows:

openssl req -newkey rsa:2048 -nodes -mysite.key -out mysite.csr

If you need to sign for a web site using IIS, then this can be done very easily in Windows:

- Launch the Server Manager.
- From Tools, select Internet Information Services (IIS) Manager.
- In the Connections panel on the left, click the server name for which you want to generate the CSR.
- In the middle panel, double-click Server Certificates.
- In the Actions panel on the right, click *Create Certificate Request* and follow the wizard.

Now that we have the certificate request it just needs to be signed. Our CA runs in Linux so you will need the following command:
 # openssl x509 -req -days 1095 -in mysite.csr -CA ca.cer -CAkey ca.key -CAcreateserial -out mysite.cer

Now that you have the signed certificate it is very straightforward to import into either IIS, Apache or a SteelHead.

© 2020 Riverbed Technology. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Overview - SSL Acceleration Module

- Allows the SteelHead to optimize SSL encrypted traffic
- Allows optimization of traffic that was normally:
  - Pass through (according to default in-path rules)
  - Transport optimized only (TCP optimization of SSL streams in certain instances proved helpful)
- SSL encrypted traffic can now receive the full benefit of:
  - SDR for data reduction
  - HTTP protocol specific optimization when applicable

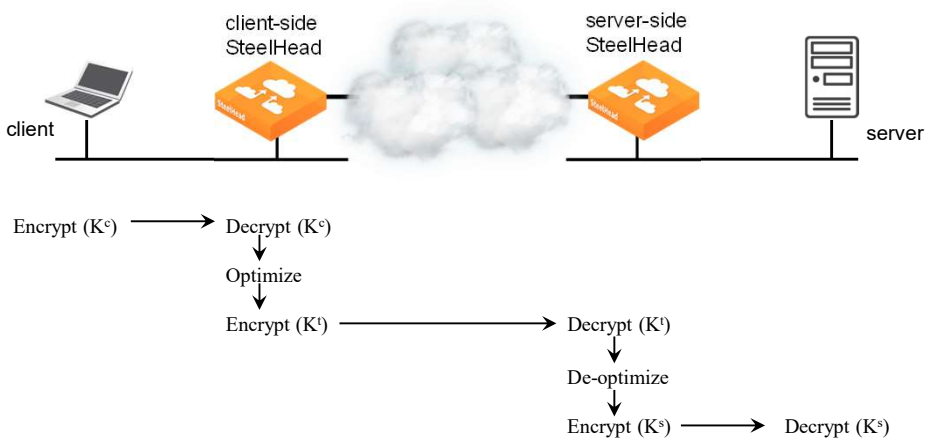
© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 22

SteelHead software uses OpenSSL
https://www.openssl.org/

© 2020 Riverbed Technology. All rights reserved.

Data Transfer

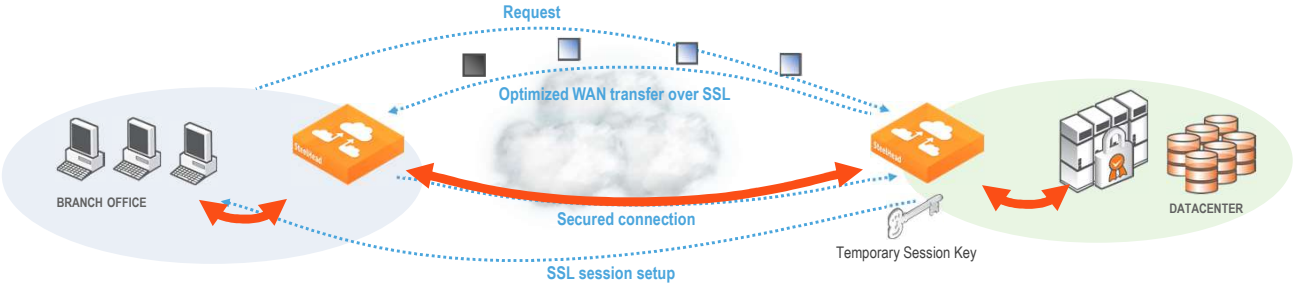


- Data transfer is always secure and optimized ~~over WAN~~ in flight.

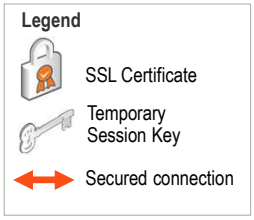
© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Optimize SSL: Overview



1. SSL Certificates and private keys are copied to server-side SteelHead appliance.
2. SteelHead appliances use their own identity certificates to establish a secure connection (proactive or on-demand).
3. On secure data request, client handshake with server is intercepted by server-side SteelHead. Server-side SteelHead established SSL connection with origin server.
4. Server-side SteelHead establishes SSL connection with client.
5. Temporary session key is migrated from server-side SteelHead to client-side SteelHead, moving client's SSL connection to client-side SteelHead.
6. Transfers are accelerated via all RiOS streamlining mechanisms.

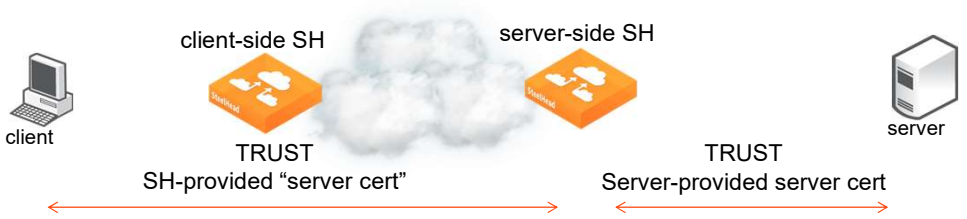


© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 24

© 2020 Riverbed Technology. All rights reserved.

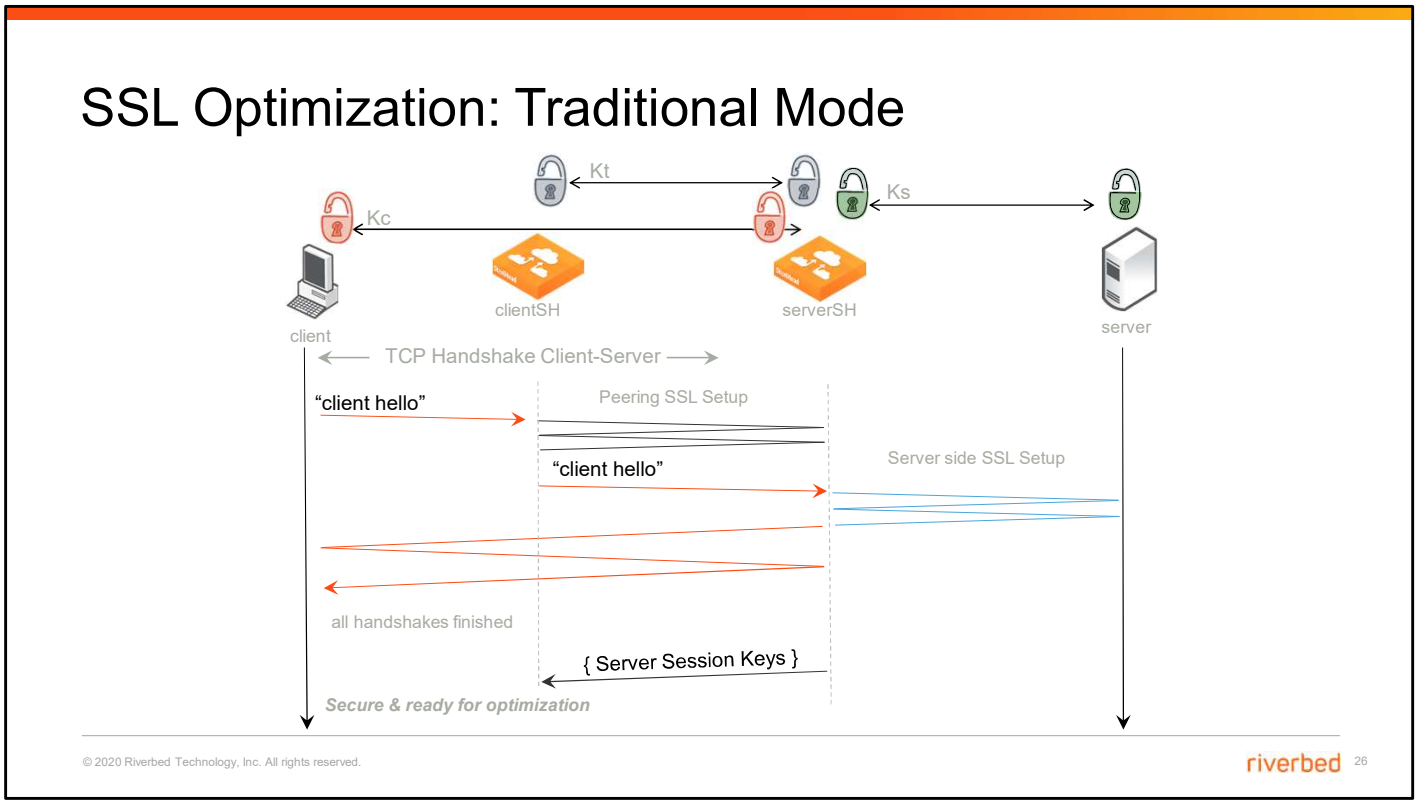
Trusts - Basic View

- Two separate trusts are established with SSL on the server-side SteelHead.
  - Client connects to the server-side SH and MUST trust the provided certificate.
  - Server-side SteelHead connects to server and MUST trust the provided certificate.



© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.



It is important to understand what is going on here. During the early stages of the SSL handshake, the server-side SteelHead is acting as a client to the server and a server to the client. So, the client performs an SSL handshake with the server-side SteelHead which, at the same time, performs a handshake with the actual SSL server. What this means is that the server-side SteelHead must trust the server and the client must trust the server-side SteelHead.

So, if we have an internal CA, which has signed the server's certificate, then we need to install this on the server-side SteelHead plus all certificates in the chain, up to the self-signed Root CA. The SteelHead will attempt to do this automatically by default.

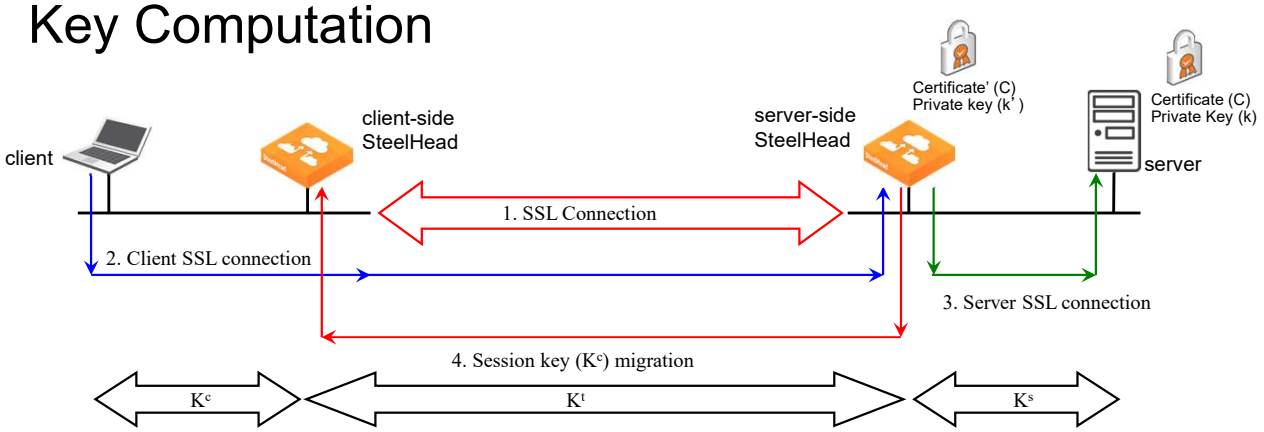
Just as important is that when the client connects to the server-side SteelHead, it needs to think it is connecting to the actual SSL server and needs to trust the certificate. There are, in reality, only two options here:

1. Import the server's certificate onto the server-side SteelHead (recommended if you can).
2. Create and install a *Proxy Certificate* on the server-side SteelHead. Set this up carefully; otherwise, the client's browser will flag that it is not the 'correct' certificate. This can be a SAN certificate if required.

Once this stage has completed we have two SSL connections, but nothing is really being optimized. It is at this stage that the server-side SteelHead gives the session key to the client-side SteelHead. Now we have three SSL connections and the two SteelHeads can happily optimize. However, the two SteelHeads must trust each other, so we need a trust relationship between the two SteelHeads, as the server-side appliance is giving the client-side appliance the session key. If this was not a trusted SteelHead (for example, a clever thief had just inserted a proxy) you would effectively be giving someone the ability to intercept and decrypt your SSL traffic.

© 2020 Riverbed Technology. All rights reserved.

Key Computation



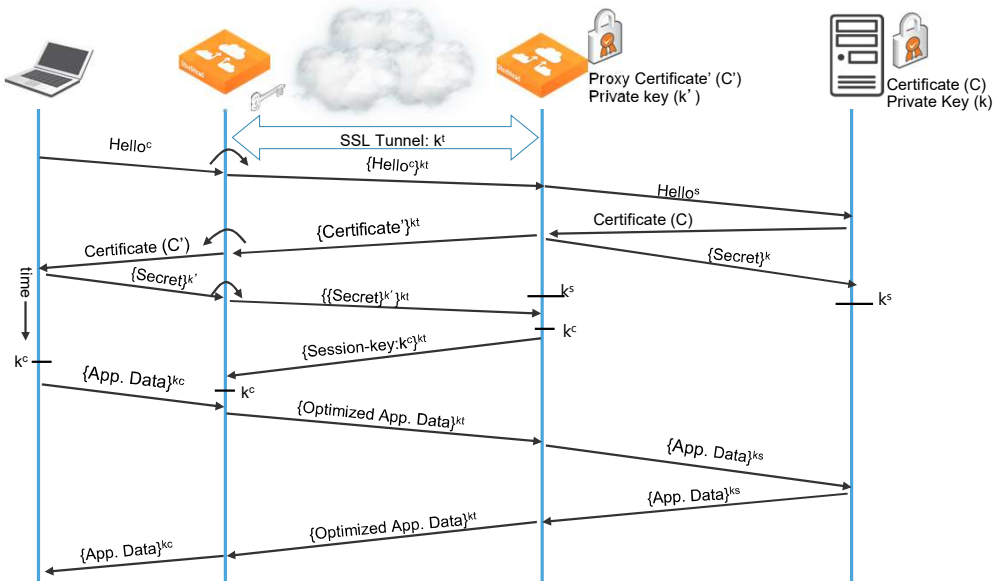
1. Inter-SteelHead SSL connection (establishment: on-demand or proactive)
2. Interception and termination of client's SSL connection at server-side SteelHead
3. SSL connection with Server
4. Inline migration of session key (Kc) to client-side SteelHead

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 27

© 2020 Riverbed Technology. All rights reserved.

SSL Optimization Sequence Diagram



© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 28

© 2020 Riverbed Technology. All rights reserved.

Configure SteelHeads for SSL Optimization

Basic Steps

- *On both client- and server-side SteelHeads*:
  1. Obtain & install SSL licenses.
  2. Enable SSL optimization.
  3. Establish a peering trust.
- *On server-side SteelHead:*
  4. Install the private or proxy key and certificate of the server you wish to optimize (which must be trusted by the clients).
  5. Ensure server-side SteelHead trusts the server certificate.
- *On client-side SteelHead:*
  6. Ensure in-path rules allow optimization of desired SSL port(s).

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 29

Now that you understand the theory, this part is fairly easy. The steps are as follows:

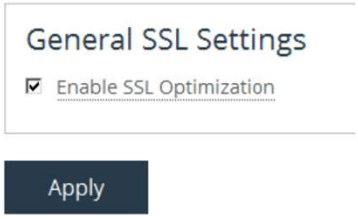
1. Obtain and install SSL licenses on both client- and server-side SteelHeads.
2. Enable SSL optimization on both of them.
3. For the server-side SteelHead, install the private key and certificate of the server you wish to optimize. This is important as during the initial phases the server-side SteelHead acts as a client to the server and a server to the actual client. This can be a wildcard or SAN certificate.
4. Ensure that the server-side SteelHead trusts the server certificate and install any necessary CA certificates. Note, if there are any intermediate CAs then you need to install every certificate in the chain, though these can be pulled automatically from the back-end server.
5. Remove port 443 from the secure port label on the client-side SteelHead or create an in-path rule.
6. Establish a peering trust between the SteelHeads.

© 2020 Riverbed Technology. All rights reserved.

Configure SSL Optimization – License & Enabling

Both SH: Obtain SSL License and Enable SSL Optimization

- The SSL license is available by default for new SteelHeads; please refer to KB article S28361 for more info.
- Once obtained and installed, then *SSL Optimization* needs to be enabled on both SteelHead appliances.



- Restart the service after you have enabled SSL Optimization.

© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 30

Riverbed no longer requires customers to complete an application process to receive an SSL License.
Effective June 3, 2016 Riverbed will no longer require an additional application process for SSL.
SSL licenses will now be configured as a build time key. The license will be available by default for all new SteelHead orders.
For existing SteelHead appliances sold prior to June 3rd, 2016, Riverbed will generate an SSL key. This license can be fetched via the autolicense feature. The license will also be available on the License Portal. To locate the license on the portal:
Navigate to the Riverbed License Portal
Input the desired serial #
Enter your email address
Click the hyperlink on the serial #
The SSL license will be displayed along all other licenses related to the serial #

© 2020 Riverbed Technology. All rights reserved.

Configure SSL Optimization – Peering Trust

Both SH: Create a Peering Trust

- Create policy on SCC to 'auto-magically' create trust relationships, or
- Manually, either by:
  - Exporting each SH cert into its peers' *Secure Peering (SSL) > Peering Trust*, *or*
  - Leveraging SSL auto-discovery (requires SSL enabled on both SHs and in-path rule allowing optimization of SSL port(s) on C-SH)
    - Create a test HTTPS connection to the server
    - Each SH should appear in the other appliance's Secure Peering page *Gray List*
    - Under **Actions** select **Trust** to move to the **White List**

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 31

In terms of the peering trust, this is easy. You can either do this manually (by swapping SteelHead peering certificates) or by using the grey- and white- lists on the GUI, though the latter means that the SSL connection will not be optimized at first and will be in the grey list until you *en-lighten* it.

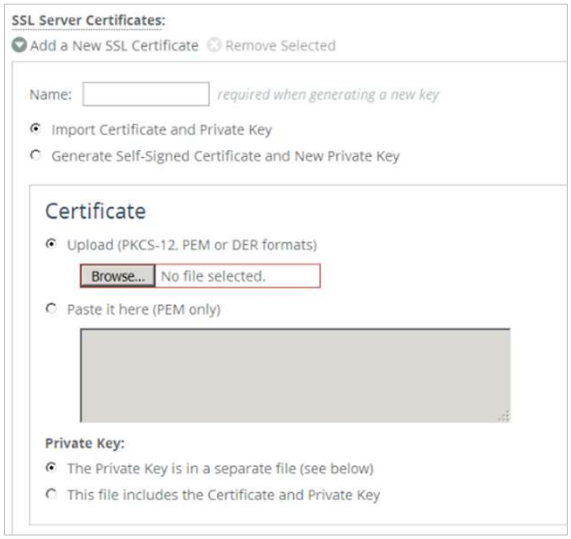
In large enterprise environments, the SCC greatly reduces the workload involved in configuring peering trusts and can quickly replace the peering certs with a few simple clicks.

© 2020 Riverbed Technology. All rights reserved.

Configure SSL Optimization – Import Server Cert & Key

Server-Side SH: Import Server Key and Certificate

- Recall: this cert must be trusted by client machines/browsers
- Under SSL Main Settings:
  - Import the signed certificate and private key from your SSL server here, *or*
  - Import the SSL server's Proxy Certificate & key
  - For demo/lab environments you could generate a self-signed certificate and private key here. *(Though your browser will usually not like it!)*



© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Configure SSL Optimization – Import CA Certs

Server-Side SH: Import any Necessary CA Certificates

- The server-side SteelHead must trust the server's certificate.
- If necessary, add any required CA certificates on the server-side SteelHead.

Certificate Authorities:

Add a New Certificate Authority Remove Selected

Certificate Authority	Issued To	Expiration Date
AC1_RAIZ_MTIN	AC1 RAIZ MTIN	Nov 3 16:17:45 2019 GMT
AC_Camerfirma_S.A._Chambers_of_Commerce_2008	Chambers of Commerce Root - 2008	Jul 31 12:29:50 2038 GMT
AC_Camerfirma_S.A._Global_Chambersign_2008	Global Chambersign Root - 2008	Jul 31 12:31:40 2038 GMT
AC_Camerfirma_SA_CIF_A82743287_Chambers_of_Commerce	Chambers of Commerce Root	Sep 30 16:13:44 2037 GMT
AC_Camerfirma_SA_CIF_A82743287_Global_Chambersign	Global Chambersign Root	Sep 30 16:14:18 2037 GMT

© 2020 Riverbed Technology, Inc. All rights reserved.

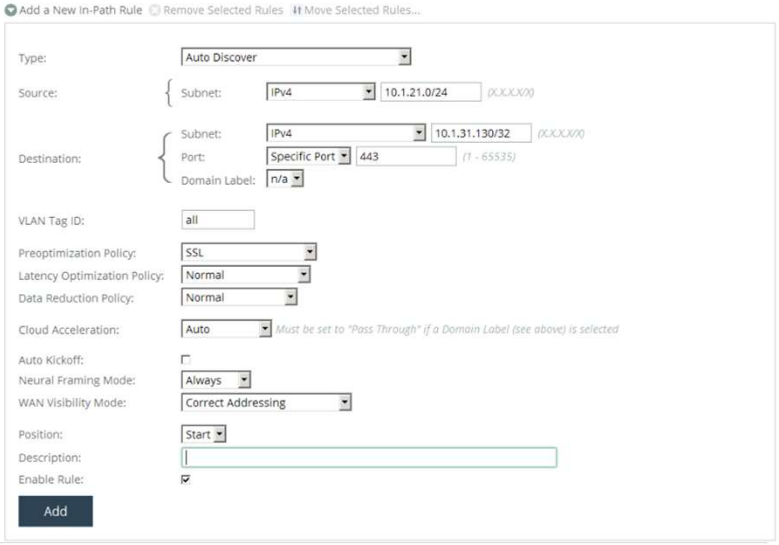
riverbed 33

© 2020 Riverbed Technology. All rights reserved.

Configure SSL Optimization – SSL In-Path Rule

Configure an In-Path Rule on the Client-Side SteelHead

- Can either:
  - Remove port 443 from the *Secure Port* label; *or*
  - Create a specific in-path rule for the SSL server:
    - Specify the SSL server as the destination on port 443
    - Set a *Preoptimization Policy* of SSL



© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 34

Alternatively, you could just remove port 443 from the secure port label. The default in-path rules contain a pass-through rule for all traffic destined to the ports defined in the secure port label.

RiOS automatically intercepts SSL traffic on the well-known TCP port 443 once you have enabled SSL optimization.

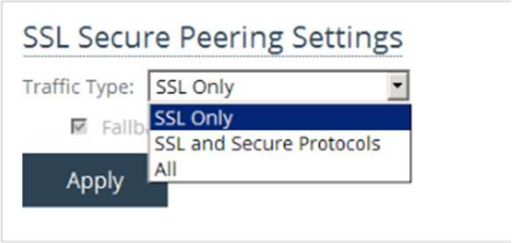
Alternatively, use an in-path rule with a custom port and preoptimization policy set to SSL to intercept SSL traffic on another TCP port. This is always the safer, recommended option, but not strictly necessary.

© 2020 Riverbed Technology. All rights reserved.

Secure Inner Channel

Also called “SSL Secure Peering"

- Important to note: Secure Inner Channel will only encrypt *optimized* traffic
- Available settings:
  - SSL Only – the default, only encrypts optimized SSL traffic
  - SSL and Secure Protocols – a best practice
  - All – all *optimized* traffic



“SSL and Secure Protocols” is important when optimizing encrypted MAPI and/or signed/encrypted SMB, as *optimized inner channels are not by default re-encrypted*.

© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 35

It is important to grasp that all three TCP connections are now SSL encrypted. By default the inner channel is only SSL encrypted for optimized SSL traffic. However, you can also configure the option *Secure Inner Channel* to SSL-encrypt other traffic. Your options are:

- SSL Only
- SSL and Secure Protocols (i.e., SSL, encrypted MAPI, signed/encrypted SMB and Citrix)
- All (Optimized Traffic)

The middle option is of particular note. When you stop to think about it, to optimize something like encrypted MAPI it needs decrypting, but the inner channel is not, by default, re-encrypted. Secure inner channel is a nice way of plugging this security loop-hole, so that, say encrypted SMB3.1.1 is securely transmitted end-to-end. For this case you do not need to import any server certificates, just enable (and license!) SSL and configure a peering trust between the SteelHeads.

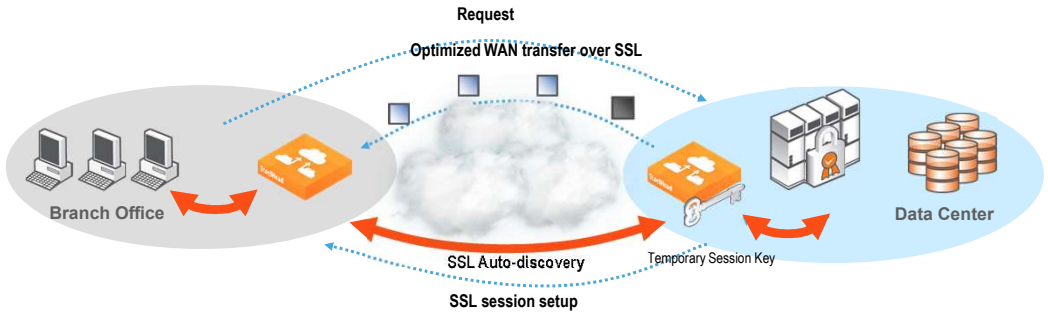
For SSL/TLS1.2 particulars on securing the inner channel, see KB S23358 “SSL/TLS 1.2 Steelhead Optimization”

© 2020 Riverbed Technology. All rights reserved.

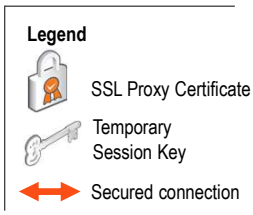
© 2020 Riverbed Technology. All rights reserved.

SSL Optimization

Overview of Trust Relationships



- For SSL Optimization three important trust relationship need to be maintained:
  - The server-side SteelHead needs to trust the server
  - The client needs to trust the server-side SteelHead
  - The SteelHeads need to trust each other



© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 37

© 2020 Riverbed Technology. All rights reserved.

SCC Certificate Authority (SCC-CA) – Overview

Simplicity and Scalability

- Using the SCC hugely simplifies the configuration of these trust relationships and gives the following benefits:
  - Allows a single secure relationship to the SCC instead of multiple peering relationship between the SteelHeads
  - Allows for central visibility and strong trust between peers
  - Does not require manual effort to renew the certificates

© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

SCC-CA – Configuration Tasks

- Configuring certificate management on the SCC consists of four tasks:
  1. Enable SCC as a Certificate Authority
  2. Manage Peering Certificates
  3. Make the SCC a trusted peer on the SteelHead
  4. Configure SSL-signed proxy certificates
- We will now look at each of these in turn

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 39

Task 1 is a global setting, establishing the SCC as a CA.

Tasks 2 and 3 are two sides of the same coin and enable the SteelHead appliances to trust each other for SSL optimization. This is all about configuring a peering trust. In task 2 we replace the certificate on the SteelHead with one signed by the SCC and in task 3 we configure the SteelHeads to trust all certificates signed by the SCC.

Task 4 is all about the server. Remember, for SSL optimization we need to import the private certificate and key of the server to the server-side SteelHead. This is all fine and splendid if you can, in fact, obtain the private certificate and key, but when you cannot, you need a proxy certificate and key. Using the SCC can greatly simplify this process and this task takes us through the steps of doing this.

© 2020 Riverbed Technology. All rights reserved.

Enable SCC as a CA – Enable CA Feature

1. Enable the Certificate Authority

- Go to *Administration* > *Security* > *Certificate Authority.*
- Select *Enable the certificate authority*.

- Click *Apply.*

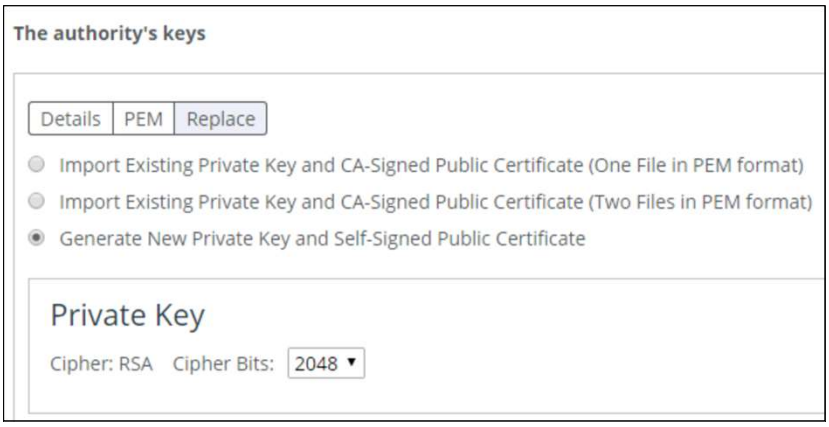
© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Enable SCC as a CA – Generate CA Key & Cert (1/2)

2. Generate New Key and Certificate

- Scroll down and click on the *Replace* tab.
- Click *Generate New Private Key and Self-Signed Public Certificate.*



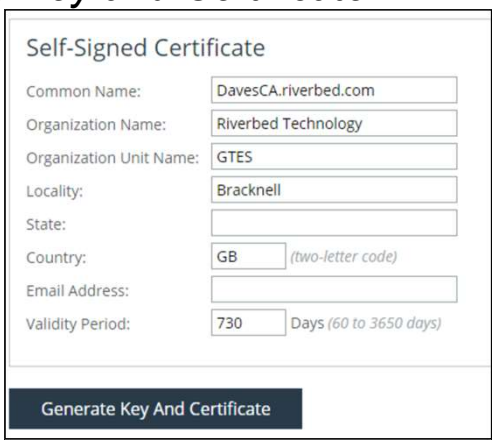
© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Enable SCC as a CA – Generate CA Key & Cert (2/2)

2. Generate New Key and Certificate

- Scroll down and fill in the certificate details.
- Click *Generate Key and Certificate.*



© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Enable SCC as a CA – View CA Cert Details

3. View the Certificate Details

Details | PEM | Replace

Certificate Details

Issued To	
Common Name:	DavesCA.riverbed.com
Organization:	Riverbed Technology
Organization Unit:	GTES
Locality:	Bracknell
Country:	GB
Issued By	
Common Name:	DavesCA.riverbed.com
Organization:	Riverbed Technology
Organization Unit:	GTES
Locality:	Bracknell
Country:	GB
Validity	
Expires On:	Mar 19 13:02:12 2020 GMT
Signature Algorithm	
Signature Algorithm:	RSA-SHA256
Fingerprint	
SHA1:	DA:B3:CF:12:90:8E:FF:75:45:79:78:17:32:F6:EA:AE:56:3B:30:CC
SHA256:	BB:3F:D7:83:12:7D:EB:92:90:44:B9:C8:AF:A2:FA:64:DB:FE:3A:F1:1B:4F:2C:5D:BC:5E:36:D4:C2:9F:8E:7C
Key	
Type:	RSA
Size:	2048

© 2020 Riverbed Technology, Inc. All rights reserved.

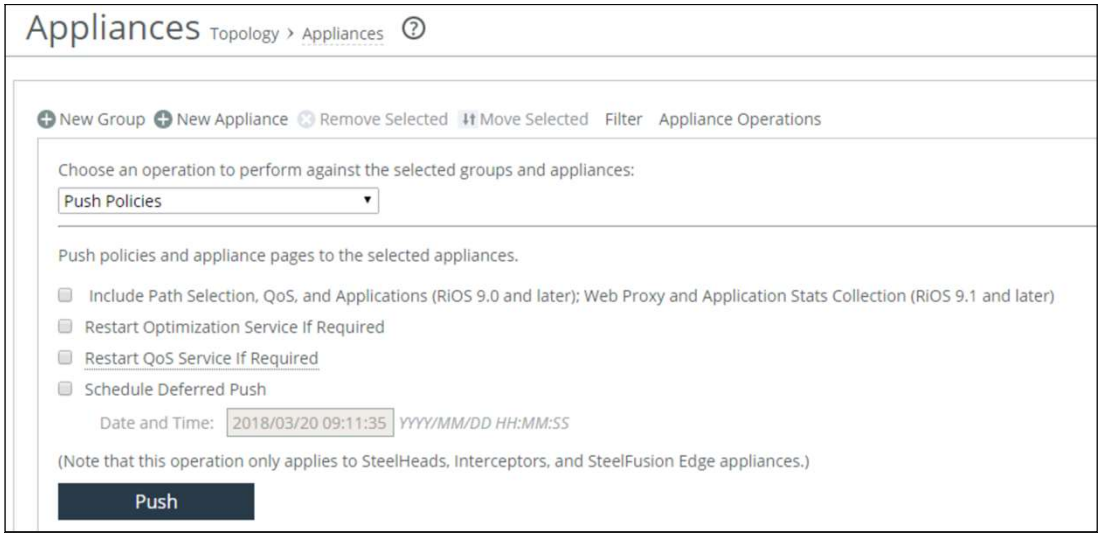
riverbed 43

ONLY IF you want to make SCC an intermediate CA (Ignore this part for SCC as Root CA and proceed ahead) you will have to go to Trusted CA store and import the Root CA certificate into that, and then on the above screen instead of choosing the option of Generating a New private Key, choose to import the Private Key & CA-Signed Public Cert of Intermediate CA. More details on this in the *SCC User Guide*.

© 2020 Riverbed Technology. All rights reserved.

Manage Peering Certificates (1/9)

1. Select *Manage > Appliances > Appliance Operations*

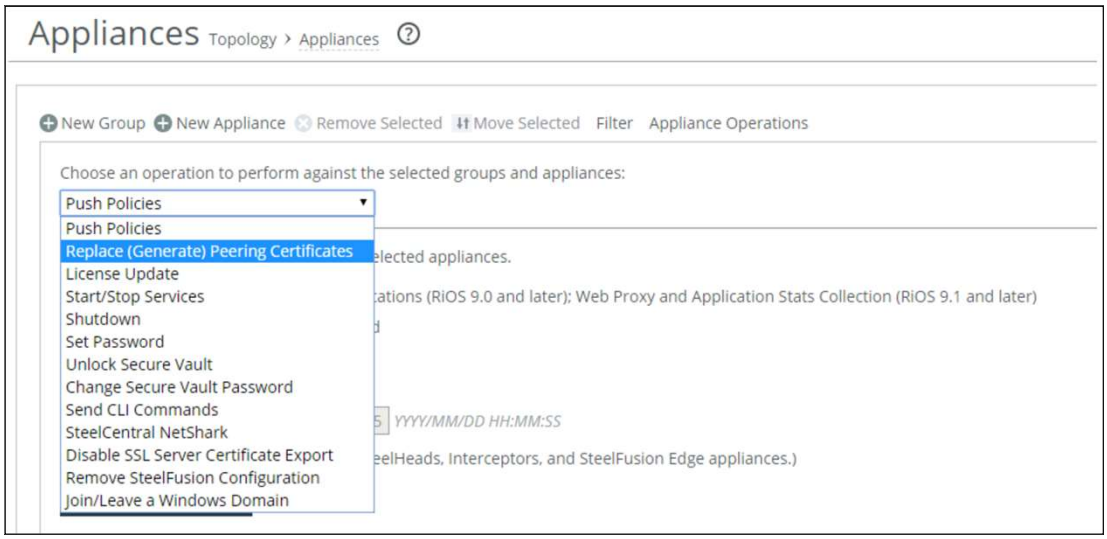


© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Manage Peering Certificates (2/9)

2. Select *Replace (Generate) Peering Certificates*



© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Manage Peering Certificates (3/9)

3. Fill in the Signing Request Information

Certificate Signing Request

Common Name:

- (•) Appliance Hostname / IP Address
- ( ) Custom: fakehostname.example.com

Organization Name:	Riverbed Technology, Inc.
Organization Unit Name:	SteelHead
Locality:	San Francisco
State:	California
Country:	US *(two-letter code)*
Email Address:	
Validity Period:	730 Days *(60 to 3650 days)*

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 46

© 2020 Riverbed Technology. All rights reserved.

Manage Peering Certificates (4/9)

4. Select the *Global* group and click *Replace*

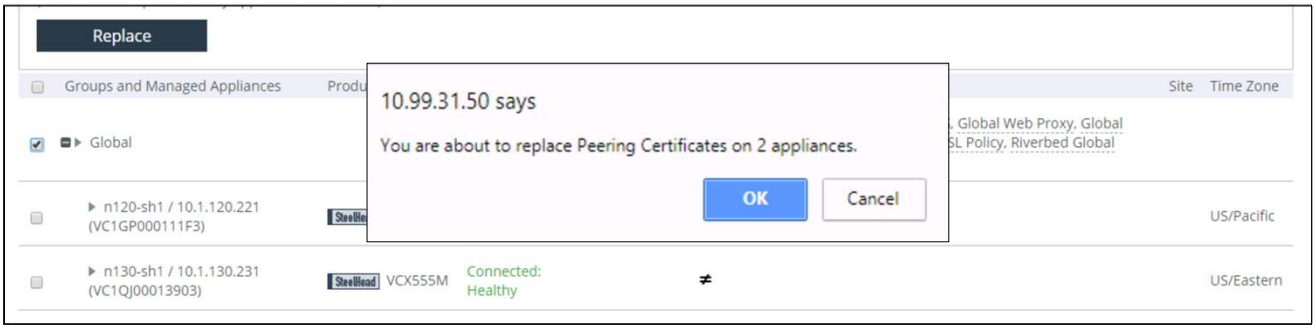
© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 47

© 2020 Riverbed Technology. All rights reserved.

Manage Peering Certificates (5/9)

5. Click *OK* at the Resulting Warning

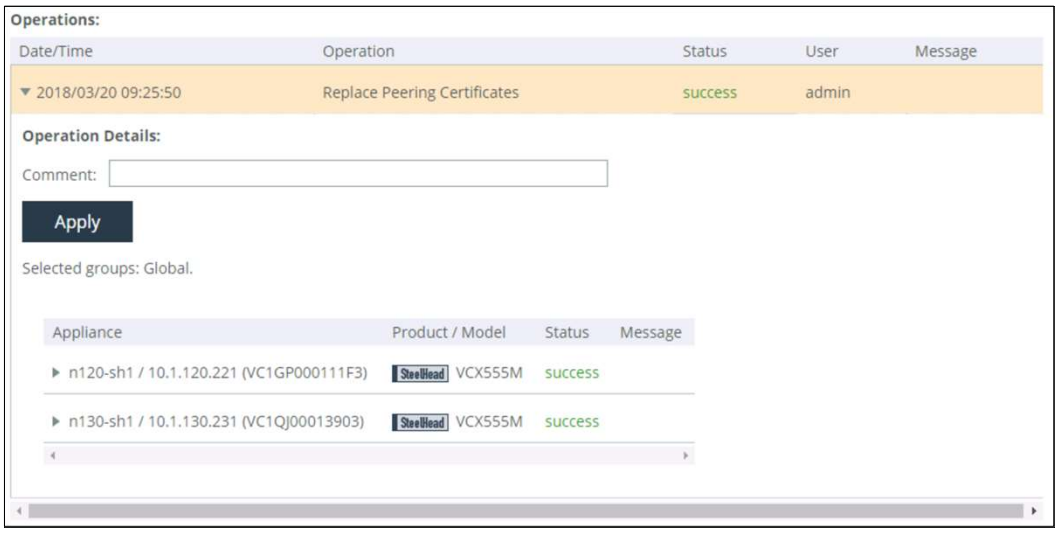


© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 48

© 2020 Riverbed Technology. All rights reserved.

Manage Peering Certificates (6/9)

6. Click on *Operation History* to Verify the Success



© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 49

© 2020 Riverbed Technology. All rights reserved.

Manage Peering Certificates (7/9)

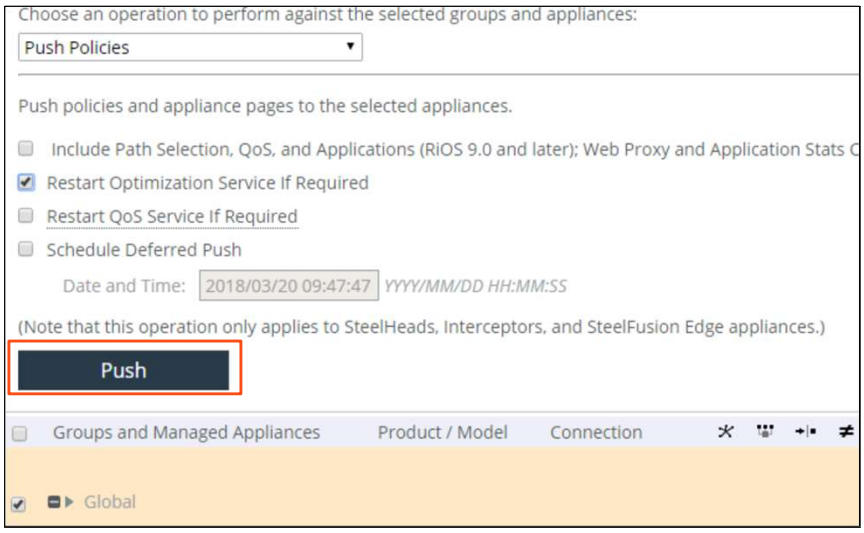
7. Verify the Certificate on the Appliance Page

© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Manage Peering Certificates (8/9)

8. Push the Policy under *Appliance Operations*



© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 51

© 2020 Riverbed Technology. All rights reserved.

Manage Peering Certificates (9/9)

9. On the SteelHead – Verify the Certificate has been Pushed

Secure Peering (SSL) SSL › Secure Peering (SSL)

Certificate Details

Issued To	
Common Name:	n120-sh1 / 10.1.120.221
Organization:	Riverbed Technology, Inc.
Organization Unit:	SteelHead
Locality:	San Francisco
State:	California
Country:	US
Serial Number:	2 (0x2)
Issued By	
Common Name:	DavesCA.riverbed.com
Organization:	Riverbed Technology
Organization Unit:	GTES
Locality:	Bracknell
Country:	GB
Validity	
Issued On:	Mar 20 13:25:51 2018 GMT
Expires On:	Mar 19 13:25:51 2020 GMT
Signature Algorithm	
Signature Algorithm:	RSA-SHA256

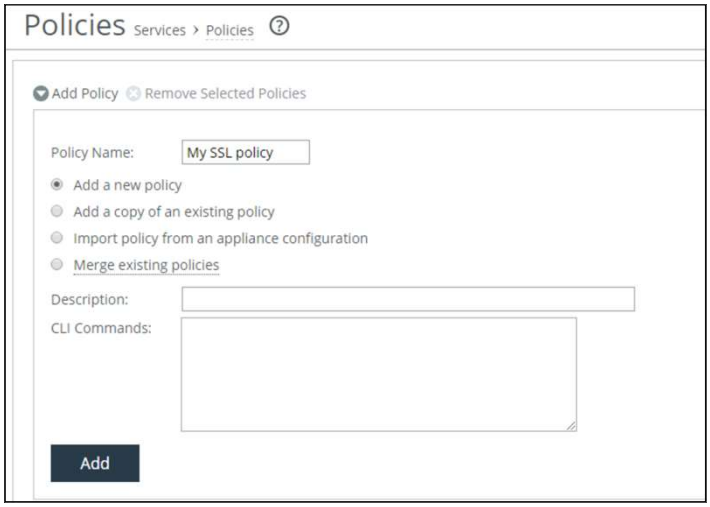
© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 52

© 2020 Riverbed Technology. All rights reserved.

Make the SCC a Trusted Peer on SteelHead (1/6)

1. Create a New Policy

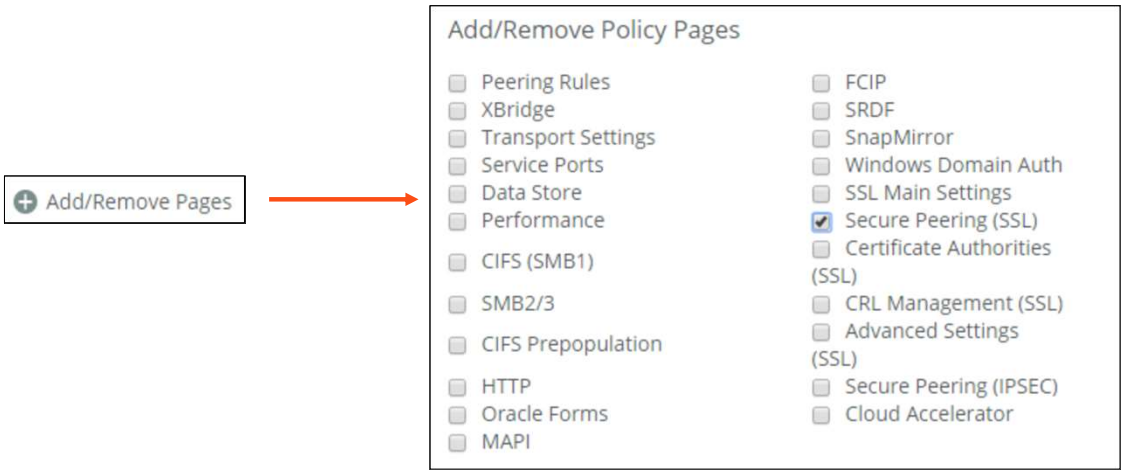


© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Make the SCC a Trusted Peer on SteelHead (2/6)

2. Add *Secure Peering (SSL)* to the Policy

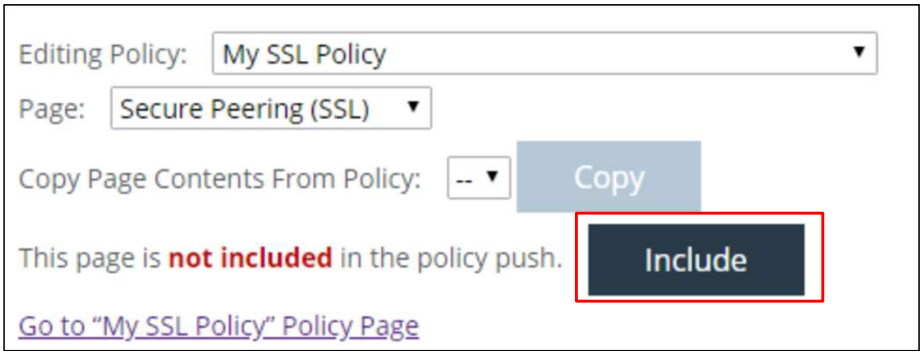


© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Make the SCC a Trusted Peer on SteelHead (3/6)

3. Edit the Policy and Include it in the Policy Push



© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 55

You need to ensure that this page is INCLUDED in a policy push.

© 2020 Riverbed Technology. All rights reserved.

Make the SCC a Trusted Peer on SteelHead (4/6)

4. Add the Policy to the Global Group

No need to change any details!

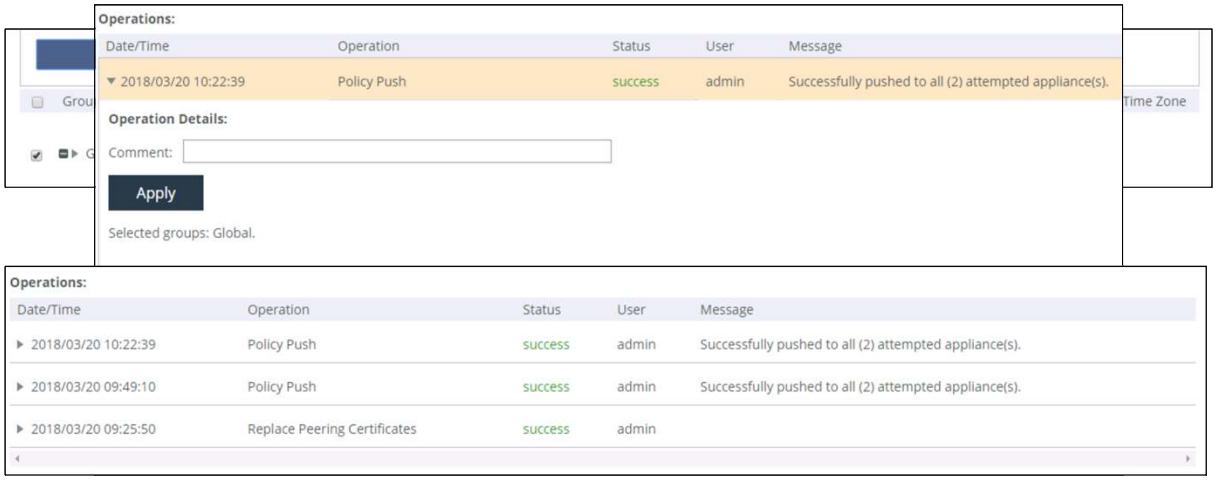
© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 56

You might think that you need to manually add the SCC as a Trusted Entity in the policy but you don't, it will be automatically be added.

© 2020 Riverbed Technology. All rights reserved.

Make the SCC a Trusted Peer on SteelHead (5/6)

5. Push the Policy and Verify the Operation History



© 2020 Riverbed Technology, Inc. All rights reserved.

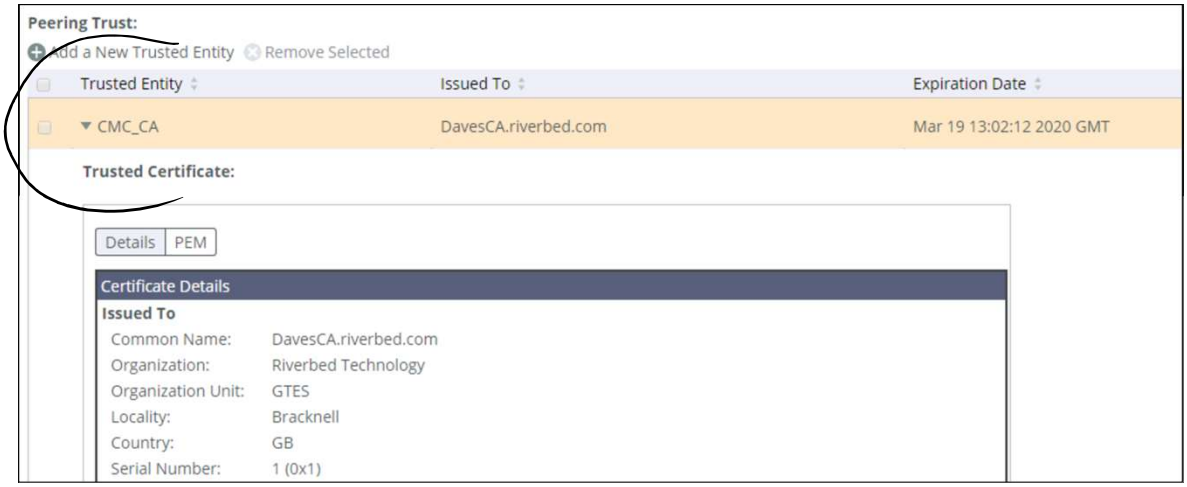
riverbed 57

Just push it!

© 2020 Riverbed Technology. All rights reserved.

Make the SCC a Trusted Peer on SteelHead (6/6)

6. Verify the SCC is Added as a Trusted Entity on the SteelHead

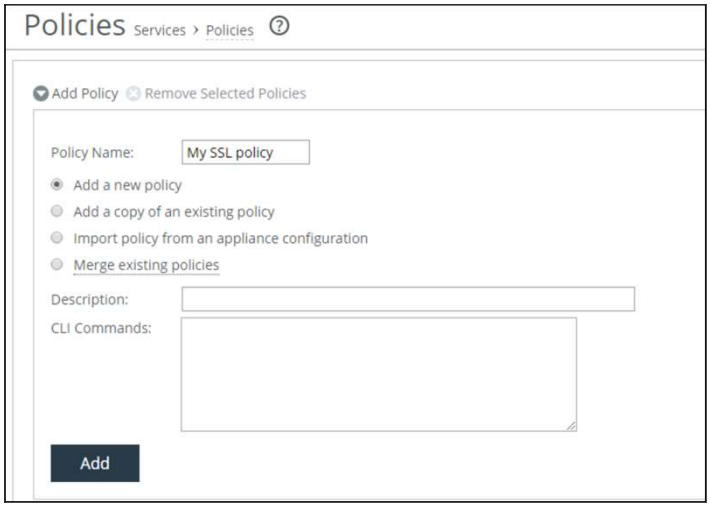


© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Configure SSL-signed Proxy Certificates (1/6)

1. Create a New Policy

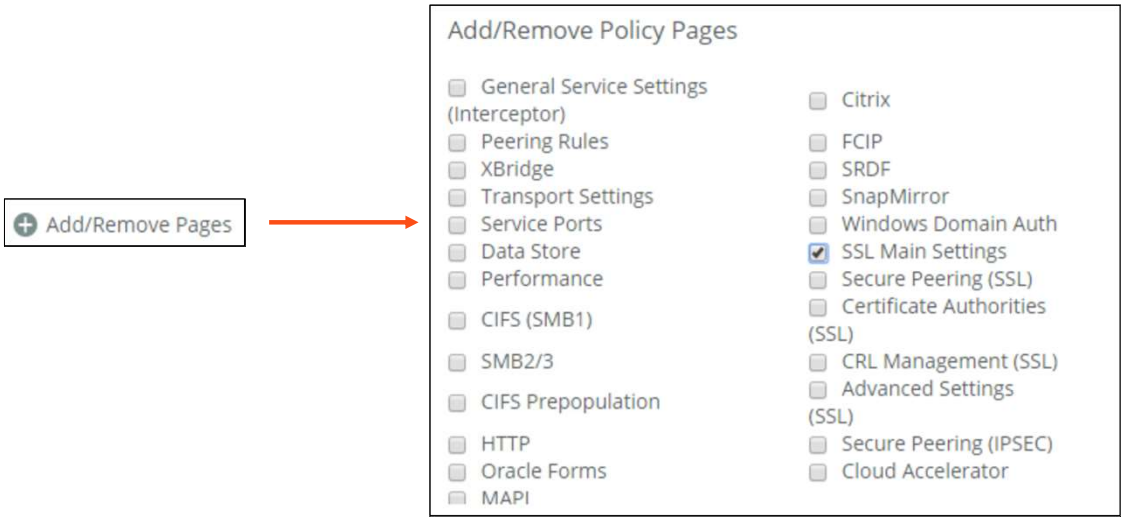


© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Configure SSL-signed Proxy Certificates (2/6)

2. Add SSL Settings to the Policy

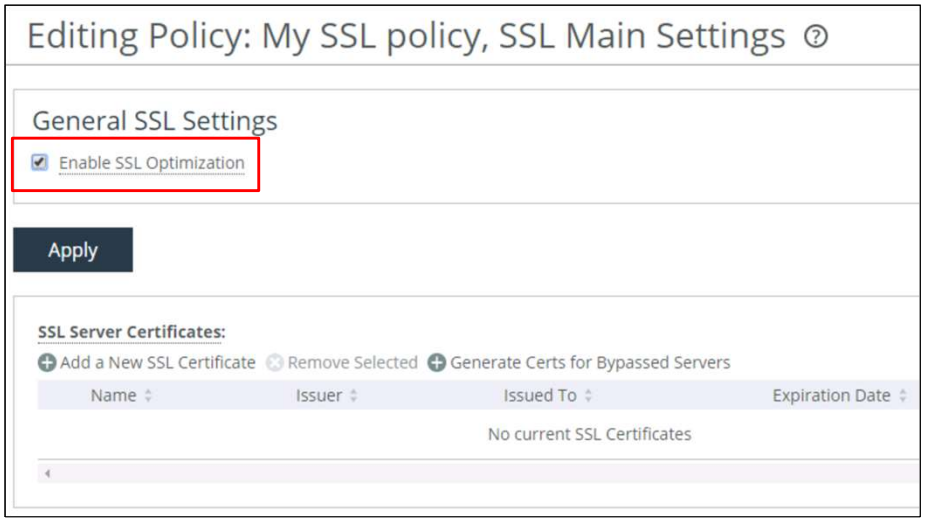


© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 60

© 2020 Riverbed Technology. All rights reserved.

Configure SSL-signed Proxy Certificates (3/6)

3. Edit the Policy, Include in the Policy Push and Enable SSL Optimization



© 2020 Riverbed Technology, Inc. All rights reserved.

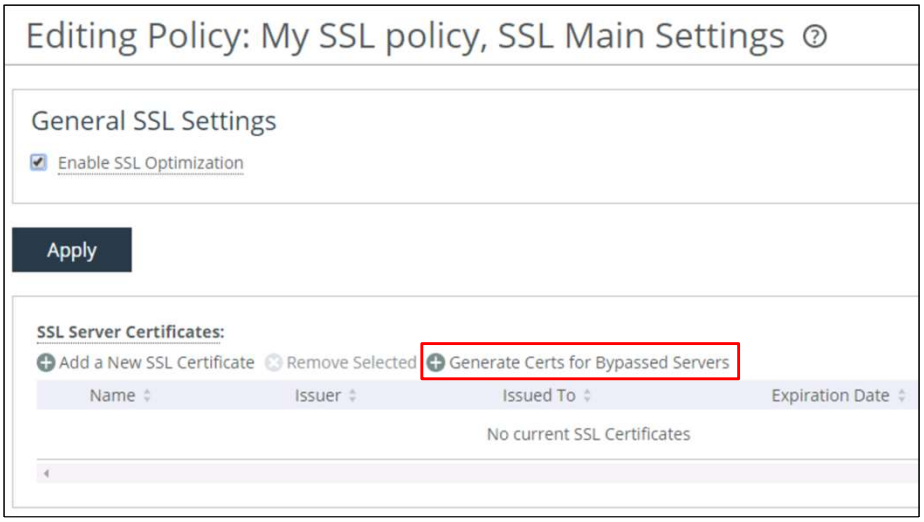
riverbed 61

Enable SSL Optimization and click Apply. You need to ensure that this page is INCLUDED in a policy push.

© 2020 Riverbed Technology. All rights reserved.

Configure SSL-signed Proxy Certificates (4/6)

4. Associate with an Appliance then run the Wizard

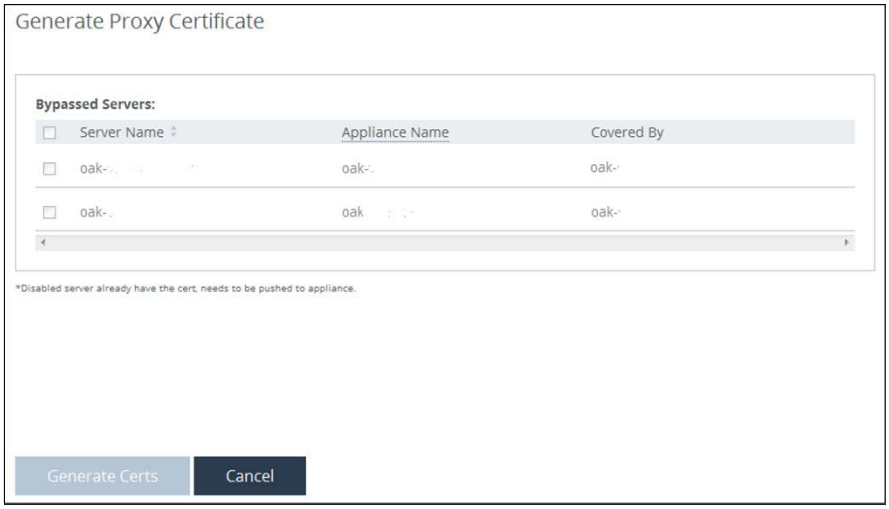


© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Configure SSL-signed Proxy Certificates (5/6)

5. Select the Servers and click **Generate Certs**

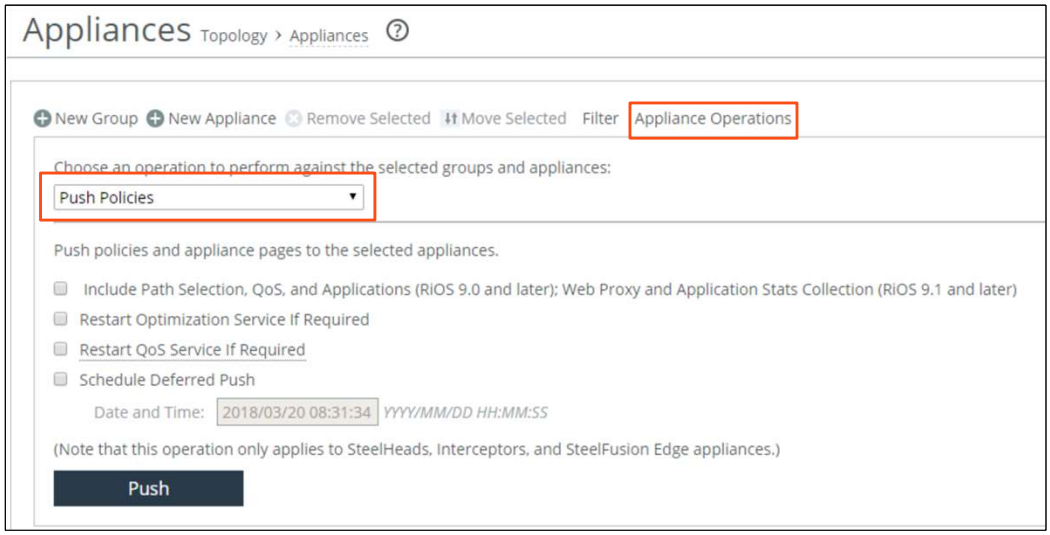


© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Configure SSL-signed Proxy Certificates (6/6)

6. Push the Policy

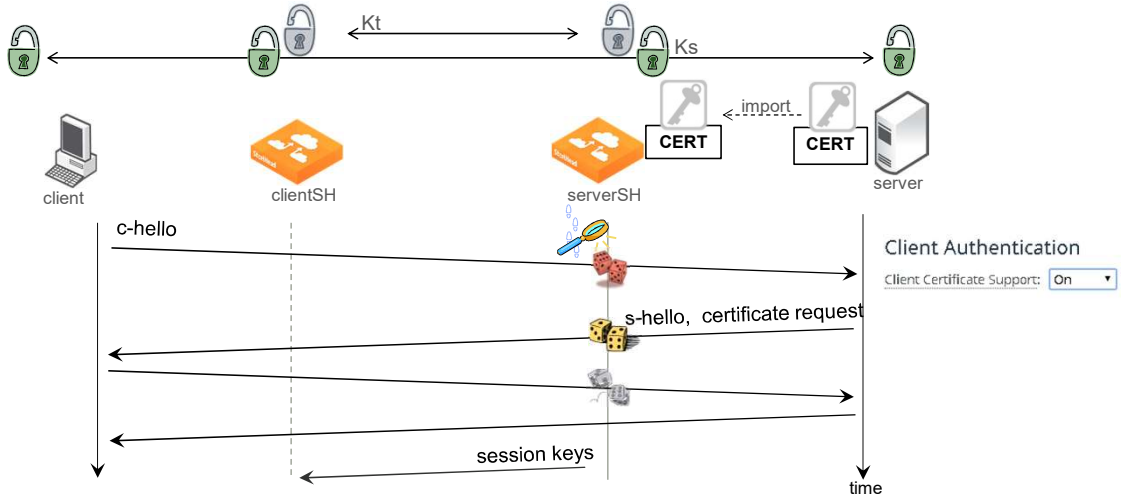


© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Client Authentication: Passive Key Derivation



- Requires Server Cert + Private Key on the server SH
- Default is to silently listen in on conversation: “Passive Key Derivation”
- Does not work with Ephemeral Diffie-Hellman key exchange

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 66

Client certificates are no different from server certificates, the only difference being that they authenticate clients rather than servers. The key process is that the client must present its own certificate for each SSL connection between the client and server.

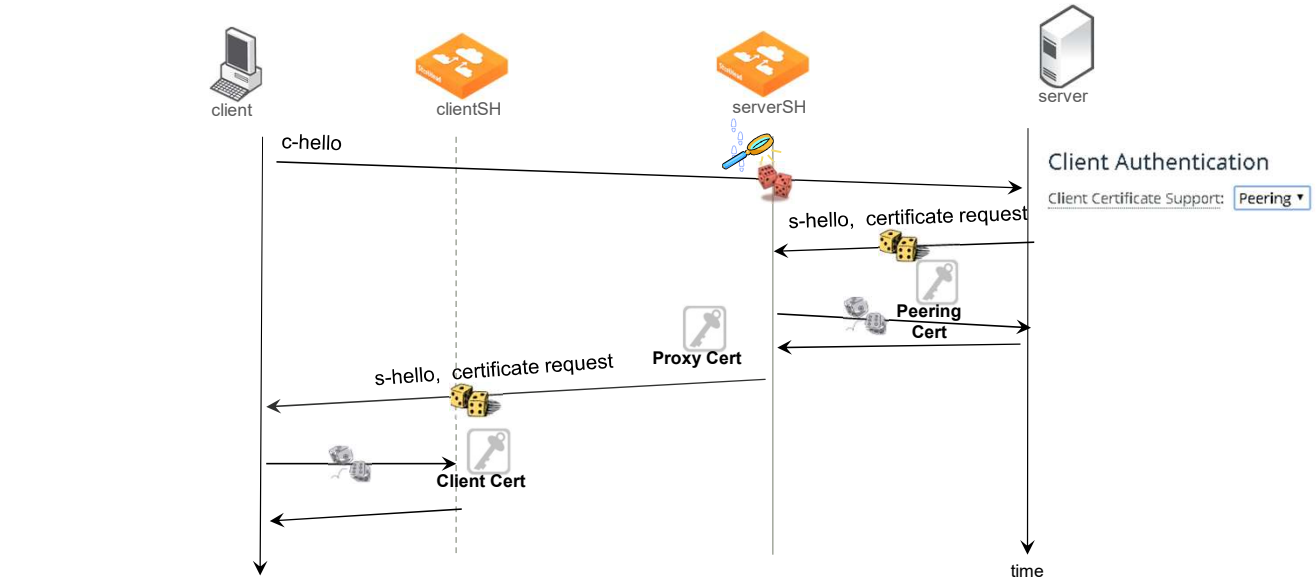
By enabling the client authentication feature, an SSL-configured (together with the server’s private key) server-side SteelHead can observe the initial SSL handshake between server and client, compute the session key, and then allow the server to authenticate the client, just as it would without SteelHeads, so there is no change in the trust model.

This just needs enabling on the server-side SteelHead. This is only possible if:

- The origin server private key is known to the SteelHead.
- The SteelHead can support the cipher suite that the client and origin server negotiate.

© 2020 Riverbed Technology. All rights reserved.

Traditional Client Authentication



© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 67

In Traditional Client Authentication, the SteelHead acts as a man-in-the-middle. This means that the Steelhead needs a proxy certificate for the connection, but it does not need the server's private key.

The client makes a client hello. If the origin server requires a client certificate it will send a client certificate request. This will be intercepted by the SSH, which will then offer its own peering cert as a client cert to the origin server. If the origin server accepts this peering certificate, then it will complete the connection with the server-side SteelHead.

The server-side SteelHead will then make a separate connection to the client and make a cert request. The client will then send its real certificate to the client-side SteelHead, which will then verify that this certificate is correct.

This mode supports the Ephemeral Diffie-Hellman key exchange.

The following trust relationships are required:

- As we are no longer using the actual server certificate, the client needs the CA certificate that issued the proxy certificate so that the client can verify the server-side SteelHead.
- The SteelHeads need the CA cert that issues the client certificate so that they can validate the client.
- Install the CA cert that issued the server cert on the server-side SteelHead so that it can validate the server.
- Import the self-signed peering cert of the server-side SteelHead as a CA onto the server so that the server can validate it.

© 2020 Riverbed Technology. All rights reserved.

HTTP SSL Proxy Interception

- The CA from the proxy server must be added to the SteelHead trusted certificate authority (CA) list.
- *Enable SSL Proxy Support* must be checked under the SSL *Advanced Settings*.
- On the client-side SteelHead, add an in-path rule, pointing to the proxy server's IP address and port, specifying an *SSL* preoptimization policy.

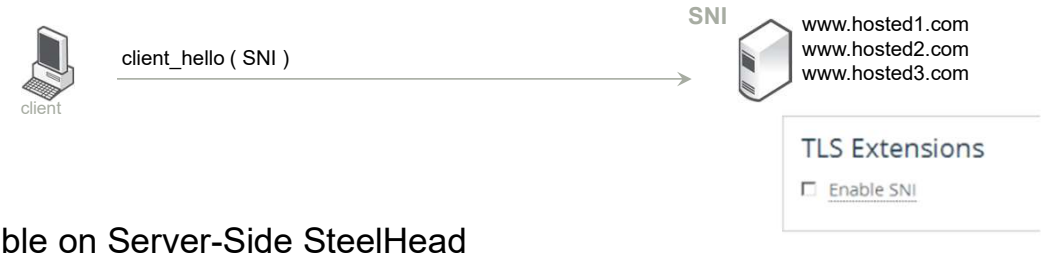
© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 68

In some cases, a proxy server will terminate the SSL session. This is done by certain security devices such as Websense. Here, the proxy will generate an SSL certificate on the fly and then resign it. In order to optimize this, the SteelHeads must be configured for SSL optimization as previously described, but also requires these additional steps:

- The CA from the proxy server must be added to the SteelHead trusted Certificate Authority (CA) list.
- *Enable SSL Proxy Support* must be checked under the SSL advanced settings.
- On the client-side SteelHead, add an in-path rule, pointing to the proxy server's IP address and port, specifying an *SSL* preoptimization policy.

© 2020 Riverbed Technology. All rights reserved.

Server Name Indication (SNI)



- Enable on Server-Side SteelHead
- Requests to a single service may result in multiple certificates; SNI allows the server-side SteelHead to determine correct host certificate
- Without SNI, poor performance is experienced because SSL has to reinitialize data structures

© 2020 Riverbed Technology, Inc. All rights reserved. riverbed 69

In modern web servers using name-based virtual hosting, one IP address can map to multiple hostnames. The distinction is made by the client setting the Host HTTP Header in the request.

For HTTPS, this requires the client to indicate the hostname at the start of the TLS handshake process so that the server knows which SSL/TLS certificate to present and then to present the correct site. This is known as *Server Name Indication* and must be enabled on the server-side SteelHead.

With SNI enabled, RiOS provides the hostname. Knowing the hostname enables the server to determine the correct named virtual host for the request and set up the connection accordingly from the start, helping ensure the browser does nt reject the proxy certificate from the server-side SteelHead.

Without SNI, an HTTPS server returns a default certificate that satisfies hostnames for all virtual hosts. The SSL connection setup will use the default virtual host configuration for the address where the connection was received, and the client's browser messages warn that certificates have the wrong hostname.

Session reuse table is slightly altered, now includes Hostname

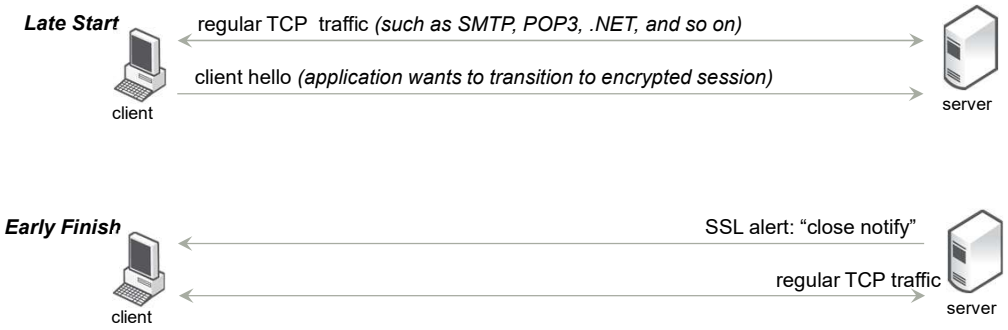
IP Address	Hostname	Certificate
192.168.10.10	server1	cert1

© 2020 Riverbed Technology. All rights reserved.

192.168.20.30	serverABC	cert2

© 2020 Riverbed Technology. All rights reserved.

SSL: Late Start / Early Finish



- Some applications send TCP data:
  - Before "client hello"
  - After "close notify" (server's finished with SSL)
  - Explicitly handle these situations using "midsession support"

© 2020 Riverbed Technology, Inc. All rights reserved. riverbed 70

The Midsession SSL feature allows optimization of connections that *transition into* SSL. Net effect of this is a delayed start to the Transport Layer Security (TLS) handshake because clients are transitioning into SSL after the initial handshake occurs.

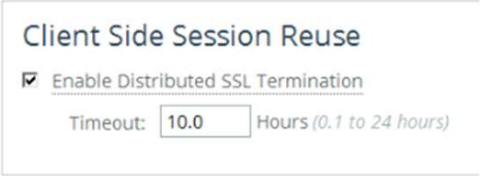
Client examples include SMTP/POP/IMAP-over-TLS and Microsoft .NET Windows Communication Foundation (WCF)-based TLS applications. This feature also enables SSL communication with protocols like Exchange-Hub to Exchange-Hub replications (for example, the SMTP-over-TLS protocol).

Must be enabled on both the client-side and server-side SteelHead appliances .

© 2020 Riverbed Technology. All rights reserved.

Distributed Termination/Session Reuse

- On by default
- Reduces the CPU load on the server-side SteelHead
- Greatly reduces number of WAN round trips
- Reuses the session key on the client-side SteelHead (if the client agrees)
- Initial full handshakes are terminated on the server-side SteelHead
- Subsequent reuse handshakes are terminated on the client-side SteelHead
- Greatly enhances the optimization performance for SSL sites



© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 71

This feature was introduced in RiOS 6.0 and is now on by default. It both reduces the CPU load on the server-side SteelHead and greatly reduces the number of WAN round trips by reusing the session key on the client-side SteelHead (if the client agrees), so that, although initial full handshakes are terminated on the server-side SteelHead, subsequent reuse handshakes are terminated on the client-side SteelHead. This greatly enhances the optimization performance for SSL sites.

© 2020 Riverbed Technology. All rights reserved.

OCSP Stapling Support

- Only required for the server-side SteelHead.
- Specify one of these stapling levels:
  - **Off** – OCSP is disabled.
  - **Strict** – If the origin server does not support OCSP, the connection is bypassed.
  - **Strict AIA** – If the certificate included an Authority Information Access (AIA) field, but the server did not send an OCSP response, the connection is bypassed. If the server did not include an AIA field the connection is not dropped.
  - **Loose –** The connection is not bypassed if OCSP is not supported.

© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Hardware Security Module (HSM) Support

- Cryptographic device that secures and manages cryptographic keys offering accelerated cryptographic operations.
- Appliances that need the private key communicate with the HSM and retrieve the required certificate and/or session key.
- The private keys from the HSM are used for proxy certificates in SSL optimization.
- The server-side SteelHead and HSM communicate through a Network Trust Link (NTL) connection.
- The server-side SteelHead must be accessible from the HSM on either the auxiliary or management interfaces.
- RiOS 9.2 and later supports SafeNet Network HSM.

© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 73

Hardware Security Module (HSM) is a cryptographic device that secures and manages cryptographic keys offering accelerated cryptographic operations. Appliances that need the private key (for example, servers, load balancers, and WAN optimization appliances such as the SteelHead) communicate with the HSM and retrieve the required certificate and/or session key. The private keys from the HSM are used for proxy certificates in SSL optimization.

The server-side SteelHead and HSM communicate through a Network Trust Link (NTL) connection. NTLs are secure, authenticated network connections between the HSM server and its clients (for example, a server-side SteelHead), which use two-way digital certificate authentication and SSL data encryption. Initial configuration steps are needed to create the two-way certificate trust between the server-side SteelHead and HSM.

The server-side SteelHead must be accessible from the HSM on either the auxiliary or management interfaces. Riverbed recommends that the server-side SteelHead and HSM be on the same LAN because high latency between them will adversely affect the SSL handshake between the SteelHead and the clients.

RiOS supports Luna HSM Client version 5.4.2 that ships preinstalled with RiOS 9.2. SafeNet Network HSM is a product from Gemalto/Safenet. You need a SafeNet support account to log in to the documentation at: https://kb.safenet-inc.com/kb/link.jsp?id=DOW3161

The following is the workflow of commands to configure HSM support:

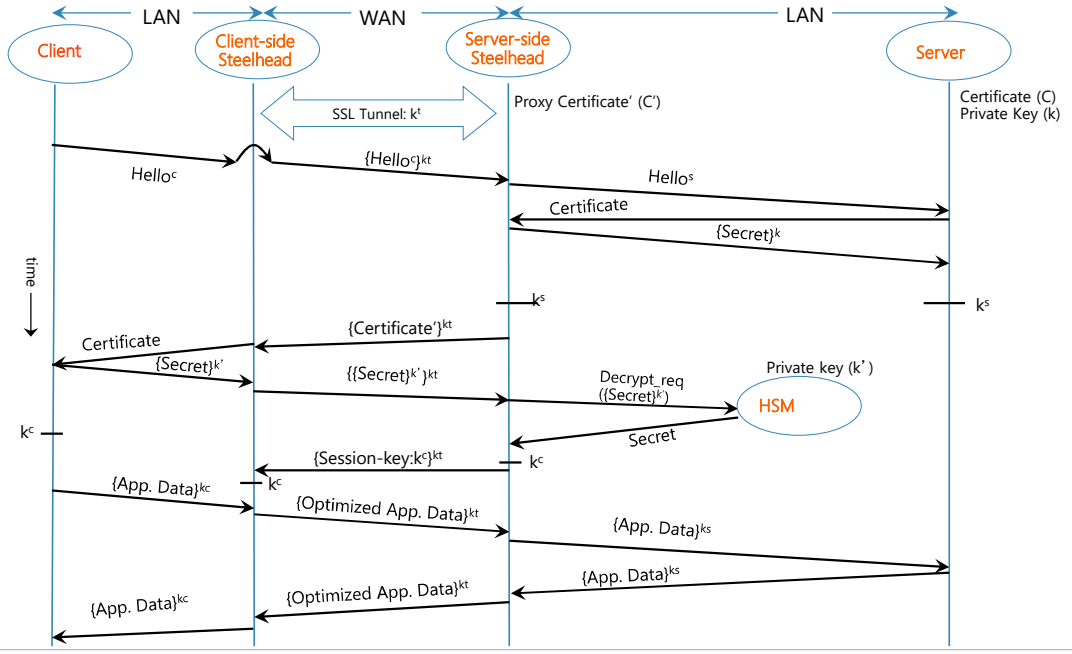
- How to set up the initial configuration to create the two-way certificates of trust between the server-side SteelHead and HSM:
  - “protocol ssl hsm safenet generate-cert”
  - “protocol ssl hsm safenet export-cert”
  - “protocol ssl hsm safenet hsm-server import-cert”

- How to configure proxy certificates and private keys (you need to assign a slot on the HSM dedicated to the server-side SteelHead before you run these commands. See the SafeNet knowledge base documentation for instructions on how to configure the HSM server):
  - “protocol ssl hsm server-cert import-cert”
  - “protocol ssl hsm slot”
  - “protocol ssl ca cert”
  - “protocol ssl hsm server-certs flush”

- How to display HSM information:
  - “show protocol ssl hsm safenet”
  - “show protocol ssl hsm server-cert”
  - “show protocol ssl hsm server-certs”

© 2020 Riverbed Technology. All rights reserved.

SSL Optimization with Keys on HSM



© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 74

© 2020 Riverbed Technology. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

Verify SSL Optimization

- You should see something like the following under *Current Connections:*

CT	Notes	Source:Port	Destination:Port	LAN kB	WAN kB	Reduction	Start Time	Application
		10.1.21.110:62666	10.1.31.130:443	5,672	5,404	4%	2016/11/09 13:10:47	HTTP

- And something similar to the following under *Traffic Summary:*

443 (SSL)	(44.88%)	21.3 MB	11.8 MB

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 76

Actually, the *Traffic Summary* report is generally a better indication as many HTTPS connections are very short lived and do not stay under *Current Connections* for very long.

© 2020 Riverbed Technology. All rights reserved.

Verify SSL Optimization – Considerations

Beware of 'Optimizing SSL'

- If SSL optimization has not been properly configured the SteelHeads may attempt to optimize the raw SSL traffic.
- The *Current Connections Report* should show zero reduction and an Application of *TCP*.

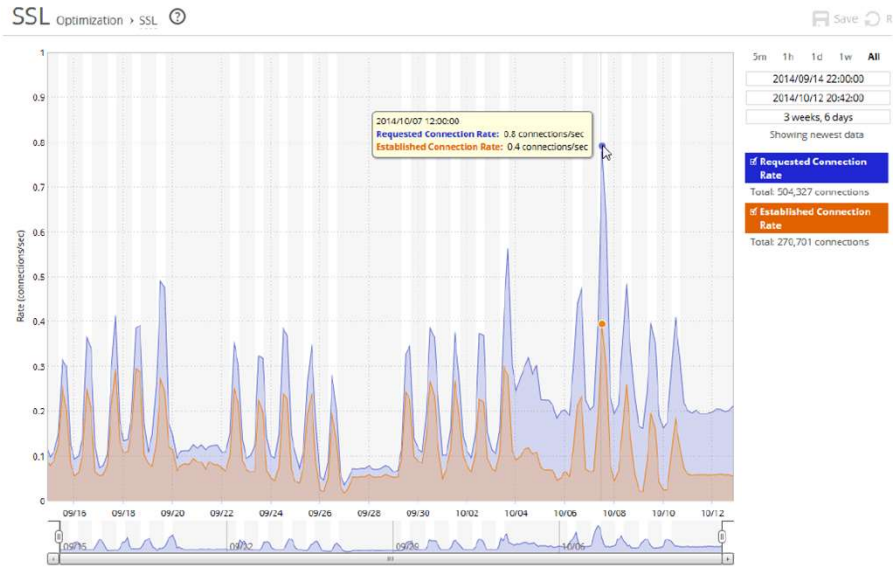
CT	Notes	Source:Port	Destination:Port	LAN kB	WAN kB	Reduction	Start Time	Application
		10.1.21.110:50531	10.1.31.130:443	6,346	6,346	0%	2019/05/29 06:27:32	TCP

© 2020 Riverbed Technology, Inc. All rights reserved.

© 2020 Riverbed Technology. All rights reserved.

View SSL Reports

- The SSL report answers these questions:
  - What is the rate of established SSL connections?
  - What is the rate of requested SSL connections?



© 2020 Riverbed Technology, Inc. All rights reserved.
riverbed 78

© 2020 Riverbed Technology. All rights reserved.

Putting it all Together – Real World Use Case

Scenario	A large design company is connected world-wide to their application in London using HTTPS. Large documents were being transferred across, often with poor WAN links.
Issue	Due to the poor WAN performance, the application was so slow it was almost unusable, causing a loss of productivity for sites outside of Europe. This was causing so much frustration amongst operators that many were leaving, causing high staff turnover and loss of time trying to hire and retrain replacements.
Solution	The customer purchased the Riverbed solution. SSL optimization was configured via an SCC, enabling users in Australia to access the application as quickly as those in London. Operators were now much happier as they could now perform their jobs correctly, increasing productivity and reducing staff turnover.

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 79

© 2020 Riverbed Technology. All rights reserved.

HOL1142

Optimize SSL Traffic

In this lab, you will:

- Optimize HTTPS (SSL) Traffic with SteelHead

Duration: **45 minutes**

eLab system: link and access details provided in your course confirmation email

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 80

© 2020 Riverbed Technology. All rights reserved.

Module Review

You should now be able to:

- Describe the SSL protocol.
- State the purpose of digital certificates.
- Configure SSL optimization.
- Use SCC as a Certificate Authority.
- Configure advanced SSL features.
- Verify SSL optimization.

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed
The Digital Performance Company

© 2020 Riverbed Technology. All rights reserved.

© 2020 Riverbed Technology, Inc. All rights reserved.

riverbed 82

This slide intentionally left blank.

© 2020 Riverbed Technology. All rights reserved.