SSL Yapılandırması

HTTPS kullanarak TheHive'a bağlanın

SSL katmanını yönetmek için bir ters proxy kullanmanızı öneririz; Örneğin, Nginx.

Reference: Configuring HTTPS servers on nginx.org

/etc/nginx/sites-available/thehive.conf

```
server {
  listen 443 ssl http2;
  server_name thehive;

  ssl on;
  ssl_certificate       path-to/thehive-server-chained-cert.pem;
  ssl_certificate_key   path-to/thehive-server-key.pem;

  proxy_connect_timeout   600;
  proxy_send_timeout      600;
  proxy_read_timeout      600;
  send_timeout            600;
  client_max_body_size    2G;
  proxy_buffering off;
  client_header_buffer_size 8k;

  location / {
    add_header             Strict-Transport-Security "max-age=31536000; includeSubDomains";
    proxy_pass             http://127.0.0.1:9000/;
    proxy_http_version      1.1;
  }
}
```

İstemci Yapılandırması

Uzak servislere bağlanmak için SSL yapılandırması gerekebilir. Aşağıdaki parametreler tanımlanabilir:

Parameter	Type	Description

`wsConfig.ssl.keyManager.stores`	list	Stores client certificates (see #certificate-manager)
`wsConfig.ssl.trustManager.stores`	list	Stored custom Certificate Authorities (see #certificate-manager
`wsConfig.ssl.protocol`	string	Defines a different default protocol (see #protocols)
`wsConfig.ssl.enabledProtocols`	list	List of enabled protocols (see #protocols)
`wsConfig.ssl.enabledCipherSuites`	list	List of enabled cipher suites (see #ciphers)
`wsConfig.ssl.loose.acceptAnyCertificate`	boolean	Accept any certificates *true / false*

Sertifika Yöneticisi

Sertifika yöneticisi, istemci sertifikalarını ve sertifika yetkililerini depolamak için kullanılır. Özel Sertifika Yetkilileri Kullanma#

Özel Sertifika Yetkililerini kullanmanın tercih edilen yolu sistem yapılandırmasını kullanmaktır.

Uygulama genelinde özel bir Sertifika Yetkilisi kurulumu gerekiyorsa (web proxy'lerine, LPAPS sunucusu gibi uzak hizmetlere bağlanmak için), daha iyi bir çözüm, bunu işletim sistemine yükleyip TheHive'ı yeniden başlatmaktır.

Debian :

ca-certificates-java paketinin kurulu olduğundan emin olun ve CA sertifikasını doğru klasöre kopyalayın. Ardından dpkg-reconfigure ca-certificates komutunu çalıştırın ve TheHive hizmetini yeniden başlatın.

```
apt-get install -y ca-certificates-java
mkdir /usr/share/ca-certificates/extra
cp mycustomcert.crt /usr/share/ca-certificates/extra
dpkg-reconfigure ca-certificates
service thehive restart
```

RPM :

Fedora veya RHEL'de ek pakete gerek yoktur. CA sertifikasını doğru klasöre kopyalayın, update-ca-trust'ı çalıştırın ve TheHive hizmetini yeniden başlatın.

```
cp mycustomcert.crt /etc/pki/ca-trust/source/anchors
sudo update-ca-trust
service thehive restart
```

Bir alternatif yöntem, ayrı bir güven deposu kullanmaktır; ancak bu Tercih Edilen bir seçenek DEĞİLDİR. TheHive yapılandırmasında trustManager anahtarını kullanın. Bu, uzak bir ana bilgisayarla güvenli bir bağlantı kurmak için kullanılır. Sunucu sertifikası, güvenilir bir sertifika otoritesi tarafından imzalanmış olmalıdır.

```
wsConfig.ssl.trustManager {
  stores = [
    {
      type = "JKS" // JKS or PEM
      path = "keystore.jks"
      password = "password1"
    }
  ]
}
```

İstemci Sertifikaları

keyManager, HTTP istemcisinin (sertifika tabanlı kimlik doğrulaması kullanıldığında) kendini uzak sunucuda kimlik doğrulamak için hangi sertifikayı kullanabileceğini belirtir.

```
wsConfig.ssl.keyManager {
  stores = [
    {
      type = "pkcs12" // JKS or PEM
      path = "mycert.p12"
      password = "password1"
    }
  ]
}
```

Protokoller

Farklı bir varsayılan protokol tanımlamak istiyorsanız, bunu istemcide özel olarak ayarlayabilirsiniz:

```
wsConfig.ssl.protocol = "TLSv1.2"
```

Etkin protokollerin listesini tanımlamak istiyorsanız, bunu açıkça bir liste sağlayarak yapabilirsiniz:

```
wsConfig.ssl.enabledProtocols = ["TLSv1.2", "TLSv1.1", "TLSv1"]
```

Gelişmiş seçenekler

Parolalar

Parola paketleri wsConfig.ssl.enabledCipherSuites kullanılarak yapılandırılabilir:

```
wsConfig.ssl.enabledCipherSuites = [
  "TLS_DHE_RSA_WITH_AES_128_GCM_SHA256",
  "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
  "TLS_DHE_RSA_WITH_AES_256_GCM_SHA384",
  "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
]
```

Hata ayıklama

Anahtar yöneticisi / güven yöneticisinde hata ayıklamak için aşağıdaki bayrakları ayarlayın:

```
  wsConfig.ssl.debug = {
    ssl = true
    trustmanager = true
    keymanager = true
    sslctx = true
    handshake = true
    verbose = true
    data = true
    certpath = true
  }
```

Revision #2
Created 12 April 2024 21:48:14 by Güldeniz Akca
Updated 13 April 2024 15:04:49 by Güldeniz Akca