Juniper SRX 日本語マニュアル

SSL Forward Proxy の CLI 設定

Juniper Networks | Engineering Simplicity

はじめに

SSL Forward Proxy の CLI 設定ついて説明します。

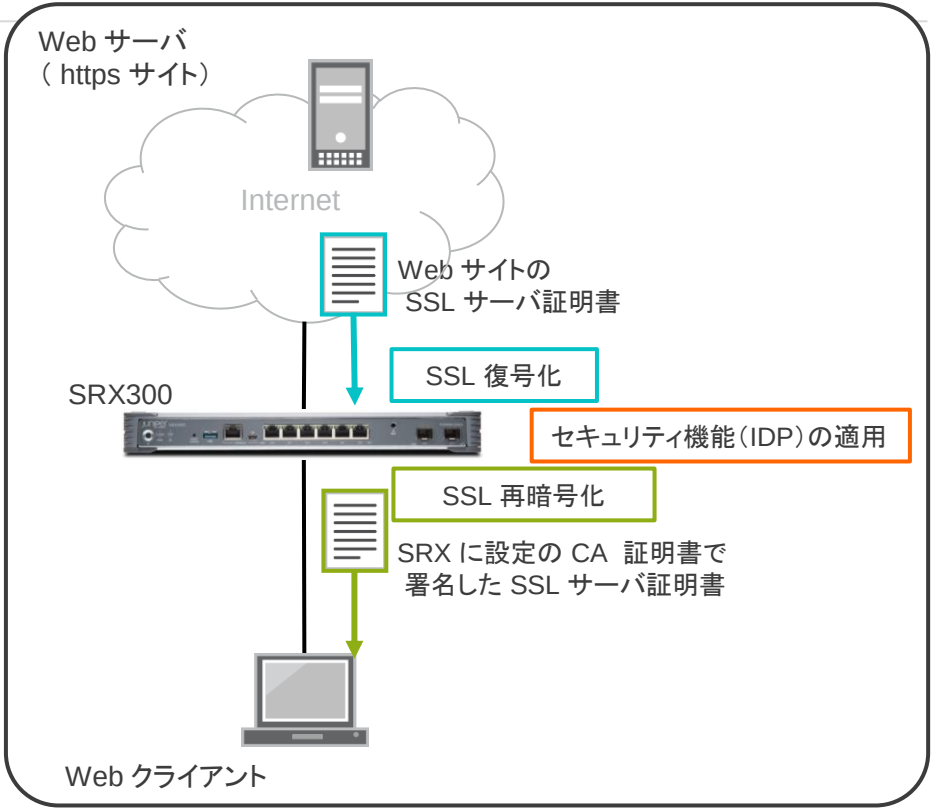
※手順内容は「SRX300」、JUNOS「19.4R3-S1」にて確認を実施しております。
　実際の設定内容やパラメータは導入する環境や構成によって異なります。
　各種設定内容の詳細は下記リンクよりご確認ください。
　https://www.juniper.net/documentation/

2021年7月

SSL Forward Proxy

以下の設定を行う場合の コマンド例となります。

SSL 暗号化通信の復号化・再暗号化
セキュリティ機能（IDP）の適用



SSL Forward Proxy

- CA 証明書の作成

```
user@host> request security pki generate-key-pair certificate-id srx-cert size 2048 type rsa
                                                    ※任意の ID（srx-cert）、鍵長を 2048、RSA 暗号を指定

user@host> request security pki local-certificate generate-self-signed certificate-id srx-cert
domain-name srx-ca.local subject "CN=srx-ca.local,OU=Sales,O=Juniper Networks,L=Tokyo,C=JP"
email admin@srx-ca.local add-ca-constraint
                ※任意のドメイン名（srx-ca.local）、任意の subject 内容 "CN,OU,O,L,C"、CA証明書オプションを適用

Self-signed certificate generated and loaded successfully
```

Key-Pair や CA 証明書の作成のやり直しが必要となった場合は、clear コマンドで作成した内容を削除

```
user@host> clear security pki key-pair certificate-id srx-cert          ※Key-Pair 削除用のコマンド
Key pair deleted successfully

user@host> clear security pki local-certificate certificate-id srx-cert  ※CA 証明書 削除用のコマンド
```

SSL Forward Proxy

CA 証明書の確認

```
user@host> show security pki local-certificate certificate-id srx-cert detail
Certificate identifier: srx-cert
  Certificate version: 3
  Serial number: 40aadcb960006223e2d11ebb76861e7f
  Issuer:
    Organization: Juniper Networks, Organizational unit: Sales, Country: JP, Locality: Tokyo,
Common name: srx-ca.local
  Subject:
    Organization: Juniper Networks, Organizational unit: Sales, Country: JP, Locality: Tokyo,
Common name: srx-ca.local
  Subject string:
    CN=srx-ca.local, OU=Sales, O=Juniper Networks, L=Tokyo, C=JP
  Alternate subject: "admin@srx-ca.local", srx-ca.local, ipv4 empty, ipv6 empty
(略)
```

SSL Forward Proxy

- CA リストの登録

```
user@host> request security pki ca-certificate ca-profile-group load ca-group-name CA-group
filename default
                                          ※ default（ファイル名）に格納されるCA 情報を CA-group として登録
Do you want to load this CA certificate ? [yes,no] (no) yes

Loading 155 certificates for group 'CA-group'.
CA-group_1: Loading done.
CA-group_2: Loading done.
   （略）
CA-group_155: Loading done.
ca-profile-group ‘CA-group’ successfully loaded. Success[154] Skipped[1]
PKId will be un-responsive for next few minutes to set-up new Cas
                                                        ※ 処理に数分（1～2分程度）必要
```

SSL Forward Proxy

CA リストの確認

```
user@host> show security pki ca-certificates ca-profile-group CA-group
                                                    ※ CA 情報プロファイル  CA-group  の詳細を表示
Certificate identifier: CA-group_1
  Issued to: Equifax Secure Certificate Authority, Issued by: C = US, O = Equifax, OU = Equifax
Secure Certificate Authority
  Validity:
    Not before: 08-22-1998 16:41 UTC
    Not after: 08-22-2018 16:41 UTC
  Public key algorithm: rsaEncryption(1024 bits)
(略)

user@host> show configuration security pki          ※ security pki 階層の設定内容を表示するコマンド
ca-profile CA-group_1 {
    ca-identity CA-group_1;
}
ca-profile CA-group_2 {
    ca-identity CA-group_2;
}
(略)
```

SSL Forward Proxy

- SSL Proxy プロファイルの設定

```
user@host# set services ssl proxy profile SSL-Proxy root-ca srx-cert
                                                        ※ CA 証明書を指定
user@host# set services ssl proxy profile SSL-Proxy trusted-ca CA-group
                                                        ※ CA 情報リストを指定
user@host# set services ssl proxy profile SSL-Proxy actions ignore-server-auth-failure
                                                        ※ サーバエラーを無視するオプションを指定
```

オプション設定（Ciphers）

```
user@host# set services ssl proxy profile SSL-Proxy preferred-ciphers custom
user@host# set services ssl proxy profile SSL-Proxy custom-ciphers rsa-with-aes-256-cbc-sha
```

- IDP の設定（ライセンス等必要）

```
user@host# set security idp idp-policy IDP rulebase-ips rule rule1 match attacks predefined-
attacks HTTP:STC:DL:EICAR
user@host# set security idp idp-policy IDP rulebase-ips rule rule1 then action drop-connection
user@host# set security idp default-policy IDP
```

SSL Forward Proxy

- セキュリティポリシーの設定

```
user@host# set security policies from-zone trust to-zone untrust policy HTTPS match source-
address any
user@host# set security policies from-zone trust to-zone untrust policy HTTPS match destination-
address any
user@host# set security policies from-zone trust to-zone untrust policy HTTPS match application
junos-https
user@host# set security policies from-zone trust to-zone untrust policy HTTPS then permit
application-services idp
user@host# set security policies from-zone trust to-zone untrust policy HTTPS then permit
application-services ssl-proxy profile-name SSL-Proxy
```

- CA 証明書のエクスポート（Web クライアントのブラウザなどにインポート）

```
user@host> request security pki local-certificate export certificate-id srx-cert type pem
filename /var/tmp/srx-cert.pem
```

SSL Forward Proxy

設定の確認

```
user@host> show
services {
    ssl {
        proxy {
            profile SSL-Proxy {
                preferred-ciphers custom;
                custom-ciphers [ rsa-with-aes-256-cbc-sha ];
                trusted-ca CA-group;
                root-ca srx-cert;
                actions {
                    ignore-server-auth-failure;
                }
            }
        }
    }
}
```

SSL Forward Proxy

設定の確認

```
security {
    idp {
        idp-policy IDP {
            rulebase-ips {
                rule rule1 {
                    match {
                        attacks {
                            predefined-attacks HTTP:STC:DL:EICAR;
                        }
                    }
                    then {
                        action {
                            drop-connection;
                        }
                    }
                }
            }
        }
        default-policy IDP;
    }
```

SSL Forward Proxy

設定の確認

```
policies {
    from-zone trust to-zone untrust {
        policy HTTPS {
            match {
                source-address any;
                destination-address any;
                application junos-https;
            }
            then {
                permit {
                    application-services {
                        idp;
                        ssl-proxy {
                            profile-name SSL-Proxy;
                        }
                    }
                }
            }
        }
    }
}
```