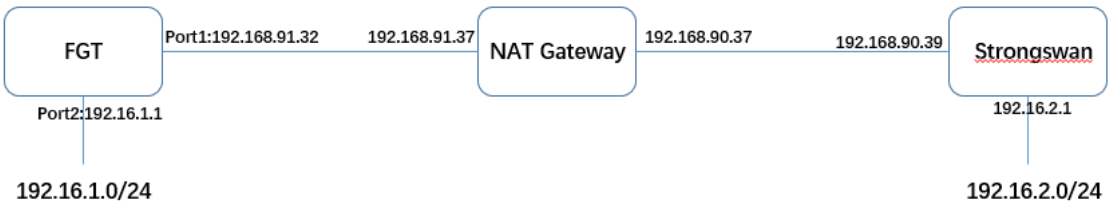
FGT 与 Strongswan 建立 IPSEC VPN

版本	V1
时间	2020 年 4 月
作者	王祥
状态	

目录

1. 网络拓扑 3
2. 设备版本 3
3. 在 Centos7 上安装 Strongswan 3
4. Strongswan 配置 4
4.1. 特别说明 4
4.2. Strongswan 配置 5
4.3. 几个参数介绍 7
4.4. Strongswan 日志 8
4.5. 系统配置 10
4.6. 启动 Strongswan 服务 11
5. FGT 配置 12
6. 状态检查 14
7. Strongswan 日志查看 15

1. 网络拓扑



拓扑说明：

FGT 访问 Strongswan 时，NAT Gateway 做源 NAT；

NAT Gateway 同时映射了 FGT port1 接口 udp 500 和 4500 端口给 Strongswan；

2. 设备版本

FortiGate： v6.2.7

Strongswan：v5.7.2

3. 在 Centos7 上安装 Strongswan

安装 epel 源：yum install epel-release

安装 strongswan：yum install strongswan

安装完成查看 Strongswan 版本

```
[root@localhost ~]# strongswan version
Linux strongSwan U5.7.2/K3.10.0-957.el7.x86_64
University of Applied Sciences Rapperswil, Switzerland
See 'strongswan --copyright' for copyright information.
```

Strongswan 更新的版本可通过官网下载，然后源码安装

Strongswan 官网：https://www.strongswan.org/

Strongswan 参数参考：https://wiki.strongswan.org/projects/strongswan/wiki/Swanctlconf

4. Strongswan 配置

4.1. 特别说明

1. 上述截图出处：

   https://wiki.strongswan.org/projects/strongswan/wiki/UserDocumentation

2. 当使用 swanctl 和 starter 时，需要的配置文件是完全不同的；
3. 当使用 swanctl 时，启动的服务是 strongswan-swanctl；使用 starter 时，启动服务是 strongswan；
4. ipsec.conf 向 sdwanctl.conf 配置迁移：

   https://wiki.strongswan.org/projects/strongswan/wiki/Fromipsecconf

5. 本文后续配置均是基于 swanctl。

4.2. Strongswan 配置

Strongswan 配置参数：

https://wiki.strongswan.org/projects/strongswan/wiki/Swanctlconf

或者 man swanctl.conf

Strongswan 配置用例:

https://wiki.strongswan.org/projects/strongswan/wiki/UsableExamples

从 swanctl.conf 配置可以看从 strongswan 会读取当前 conf.d 目录下以.conf 结尾的配置文件

[root@localhost ~]# grep -Ev '#|^$' /etc/strongswan/swanctl/swanctl.conf

include conf.d/*.conf

新建配置文件：

vim /etc/strongswan/swanctl/conf.d/swanctl.conf

```
connections {
    #IPSEC 名称
     vpn1 {
       #IKE 版本，1 表示 IKEv1，2 表示 IKEv2
        version = 1
        #本地 IPSEC VPN 地址
        local_addrs = 192.168.90.39
        #对端 IPSEC VPN 地址
        remote_addrs = 192.168.90.37
        #IPSEC 阶段一加密集
        proposals = aes128-sha1-modp2048
        # 野蛮模式 yes，主动模式 no，默认是 no
        aggressive = yes
        #dpd 检测间隔
        dpd_delay = 5s
        #DPD 检测超时时间
        dpd_timeout = 15s
        #本地认证
        local{
            #认证方式：psk 预共享秘钥
            auth = psk
            #本地 id
```

```
            id = 192.168.90.39
        }
        #  对端认证
        remote{
            #对端 id
            id = 192.168.91.32
            auth = psk
        }
       #IPSEC 阶段二设置
        children {
            #阶段二名称，如果有多条感兴趣流，可以写多个，如 vpn2{xxx}
             vpn1 {
                #阶段二加密集
                esp_proposals = aes128-sha1-modp2048
                #本地感兴趣流
                local_ts = 192.16.2.0/24
                #对端感兴趣流
                remote_ts = 192.16.1.0/24
                #IPSEC 隧道模式
                mode = tunnel
                #  当 dpd 检测失败时会立即重新发起 IKE 协商
                dpd_action = restart
                #当加载配置时执行的动作
                start_action = trap
            }
        }
    }
}
#秘钥设置
secrets {
    # IKE preshared secret section for a specific secret.
    ike-vpn1 {
        #共享秘钥的值
        secret = "fortinet"
        #该共享秘钥属于谁，如果 id 为空或者 id = %any，表示匹配所有
        id = 192.168.91.32
    }
}
```

4. 3. 几个参数介绍

start_action：当加载配置时执行的动作

默认 none：被动模式，流量不能出发协商；

trap：流量可以触发协商，当对端主动断开后，流量依然可以出发协商；

start：当启动服务时主动发起协商，但当对端主动断开后，不会再发起协商，流量也不会触发协商；

手动发起 IPSEC 协商命令：swanctl --initiate --child <name>（阶段二的名称）

```
local{
                id = 192.168.90.39 #本地 ID
    }
remote{
                id = 192.168.91.32 #对端 ID
    }
secrets {

    ike-vpn1 {
            #该共享秘钥属于谁
            id = 192.168.91.32
        }
}
```

id 即 IPSEC 中的身份，默认是通过 IP 地址标识;

如果在有 NAT 的环境下，设备的 IP 地址是会转换，因此可以指定 id，如 id=192.168.91.32;

也可以使用 fqnd 类型，如 fqdn:test1；同样对端也要设置对应的类型和 id,若 id = %any 表示可以匹配所有。

ike-vpn1 是 secrets 中的一组秘钥，其中设置的 id 要合 remote 中的 id 对应，才能匹配的上；若 id 为空或者 id = %any，表示匹配所有。

4.4. Strongswan 日志

日志设置参考：

man strongswan.conf 中 LOGGER CONFIGURATION 章节；

vim /etc/strongswan/strongswan.conf

在 charon{}中添加设置文件日志，如果不配置，默认日志会输出到/var/log/messages

```
charon {
        #filelog 是关键字，定义类型日志
        filelog {
                #名称
                 charon {
                 #日志文件名称
                 path = /var/log/charon.log
                 #  日志时间格式
                 time_format = %b %e %T
                 #  日志中显示 ike 名称
                 ike_name = yes
                 #  是否覆盖日志文件
                 append = no
                 #  默认的日志级别
                 default = 1
                 #  刷新每一行到硬盘
                 flush_line = yes
                #  定义更高的子类日志级别，覆盖 default 对应子类的值
                 cfg = 4
                 ike = 2
                }
         }
}
```

日志级别：https://wiki.strongswan.org/projects/strongswan/wiki/LoggerConfiguration

Levels and Subsystems/Groups

The IKE daemon knows different numerical levels of logging, ranging from -1 to 4:

- -1: Absolutely silent
- 0: Very basic auditing logs, (e.g. SA up/SA down)
- 1: Generic control flow with errors, a good default to see whats going on
- 2: More detailed debugging control flow
- 3: Including RAW data dumps in hex
- 4: Also include sensitive material in dumps, e.g. keys

Each logging message also has a source from which subsystem in the daemon the log came from:

- app: applications other than daemons
- asn: Low-level encoding/decoding (ASN.1, X.509 etc.)
- cfg: Configuration management and plugins
- chd: CHILD_SA/IPsec SA
- dmn: Main daemon setup/cleanup/signal handling
- enc: Packet encoding/decoding encryption/decryption operations
- esp: libipsec library messages
- ike: IKE_SA/ISAKMP SA
- imc: Integrity Measurement Collector
- imv: Integrity Measurement Verifier
- job: Jobs queuing/processing and thread pool management
- knl: IPsec/Networking kernel interface
- lib: libstrongswan library messages
- mgr: IKE_SA manager, handling synchronization for IKE_SA access
- net: IKE network communication
- pts: Platform Trust Service
- tls: libtls library messages
- tnc: Trusted Network Connect

4.5. 系统配置

关闭防火墙

systemctl stop firewalld
systemctl disable firewalld

关闭 Selinux：

[root@localhost ~]# setenforce 0 #临时关闭
[root@localhost ~]# getenforce #查看 Selinux 状态
Permissive

永久关闭 Selinux：

vim /etc/selinux/config
[root@localhost ~]# grep dis /etc/selinux/config
# disabled - No SELinux policy is loaded.
SELINUX=disabled

如果不关闭 Selinux，Strongswan 启动将无法启动。

```
Apr 19 13:58:21 localhost setroubleshoot: SELinux is preventing /usr/sbin/swanctl from read access on the directory charon. For complete SELinux messages run:
sealert -l 7c9cb6d9-85e4-4079-887e-f1c2dfcce57e
Apr 19 13:58:21 localhost python: SELinux is preventing /usr/sbin/swanctl from read access on the directory charon.#012#012*****  Plugin catchall (100. confide
nce) suggests   **************************#012#012If you believe that swanctl should be allowed read access on the charon directory by default.#012Then you sho
uld report this as a bug.#012You can generate a local policy module to allow this access.#012Do#012allow this access for now by executing:#012# ausearch -c 'sw
```

转发相关

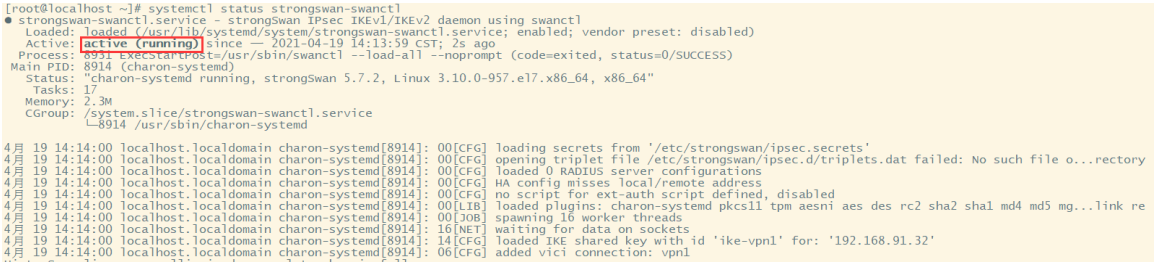
```
cat >> /etc/sysctl.conf << EOF
    ## 开启转发
    net.ipv4.ip_forward=1
    net.ipv6.conf.all.forwarding=1
     ## 禁止重定向，比如禁止 ICMP 重定向报文
    net.ipv4.conf.all.accept_redirects = 0
    net.ipv4.conf.all.send_redirects = 0
EOF
```

sysctl –p

4.6. 启动 Strongswan 服务

启动 strongswan 服务，并使其开机启动

[root@localhost ~]# systemctl start strongswan-swanctl
[root@localhost ~]# systemctl enable strongswan-swanctl



5. FGT 配置

配置 IPSEC

```
config vpn ipsec phase1-interface
    edit "to-strongswan"
        set interface "port1"
        set mode aggressive
        set peertype any
        set net-device disable
        set proposal aes128-sha1
        #如果 strongswan 指定了 id 和 id-类型，FGT 端也要明确指定
        set localid-type address
        set auto-negotiate disable
        set dpd on-idle
        set dpd-retrycount 3
        set dpd-retryinterval 5
        set dhgrp 14
        set remote-gw 192.168.90.39
        set psksecret fortinet
    next
end

config vpn ipsec phase2-interface
    edit "to-strongswan"
        set phase1name "to-strongswan"
        set proposal aes128-sha1
        set replay disable
        set src-subnet 192.16.1.0 255.255.255.0
        set dst-subnet 192.16.2.0 255.255.255.0
    next
end
```

配置策略

```
config firewall policy
    edit 1
        set name "test1"
        set srcintf "port2"
        set dstintf "to-strongswan"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
```

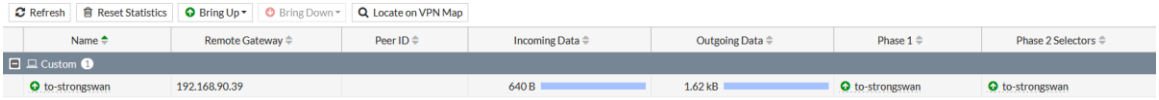
```
            set schedule "always"
            set service "ALL"
        next
        edit 2
            set name "test2"
            set srcintf "to-strongswan"
            set dstintf "port2"
            set srcaddr "all"
            set dstaddr "all"
            set action accept
            set schedule "always"
            set service "ALL"
        next
end
```

配置路由

```
config router static
        edit 1
            set dst 192.16.3.0 255.255.255.0
            set device "to-strongswan"
        next
end
```

6. 状态检查

FGT IPSEC 状态



Strongswan IPSEC 状态

```
[root@localhost ~]# swanctl --list-sas
vpn1: #2, ESTABLISHED, IKEv1, 51262afc0fac59e5_i 3d33cc26d81ed84c_r*
  local  '192.168.90.39' @ 192.168.90.39[4500]
  remote '192.168.91.32' @ 192.168.90.37[4500]
  AES_CBC-128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
  established 662s ago, rekeying in 12860s
  vpn1: #8, reqid 8, INSTALLED, TUNNEL-in-UDP, ESP:AES_CBC-128/HMAC_SHA1_96/MODP_2048
    installed 3s ago, rekeying in 3292s, expires in 3957s
    in  c6efc138,      0 bytes,     0 packets
    out 678fbdff,      0 bytes,     0 packets
    local  192.16.2.0/24
    remote 192.16.1.0/24
```

IKEv1 主模式/野蛮模式，IKEv2 测试都能正常建立连接。

但是，Strongswan 使用野蛮模式 PSK 认证提示不安全，默认不允许建立连接。

```
Apr 20 11:12:02 15[CFG] <1> selecting proposal:
Apr 20 11:12:02 15[CFG] <1>   proposal matches
Apr 20 11:12:02 15[CFG] <1> received proposals: IKE:AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
Apr 20 11:12:02 15[CFG] <1> configured proposals: IKE:AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
Apr 20 11:12:02 15[CFG] <1> selected proposal: IKE:AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
Apr 20 11:12:02 15[IKE] <1> Aggressive Mode PSK disabled for security reasons
Apr 20 11:12:02 15[IKE] <1> queueing INFORMATIONAL task
Apr 20 11:12:02 15[IKE] <1> activating new tasks
Apr 20 11:12:02 15[IKE] <1>   activating INFORMATIONAL task
Apr 20 11:12:02 15[ENC] <1> generating INFORMATIONAL_V1 request 540149239 [ N(AUTH_FAILED) ]
Apr 20 11:12:02 15[NET] <1> sending packet: from 192.168.90.39[500] to 192.168.90.37[500] (56 bytes)
Apr 20 11:12:02 15[IKE] <1> IKE_SA (unnamed)[1] state change: CONNECTING => DESTROYING
Apr 20 11:12:05 07[NET] <2> received packet: from 192.168.90.37[500] to 192.168.90.39[500] (600 bytes)
```

需要在 vim /etc/strongswan/strongswan.conf 中添加如下选项：

i_dont_care_about_security_and_use_aggressive_mode_psk = yes

```
charon {
        load_modular = yes
        i_dont_care_about_security_and_use_aggressive_mode_psk = yes
        plugins {
                include strongswan.d/charon/*.conf
        }
```

7. Strongswan 日志查看

通过 tail -f /var/log/charon.log 观察 IPSEC 日志信息输出，来判断 IPSEC 的问题

①当发给 Strongswan 加密集不一致时，其日志信息

```
Apr 19 21:04:13 13[IKE] <1> 192.168.90.37 is initiating a Aggressive Mode IKE_SA
Apr 19 21:04:13 13[IKE] <1> IKE_SA (unnamed)[1] state change: CREATED => CONNECTING
Apr 19 21:04:13 13[CFG] <1> selecting proposal:
Apr 19 21:04:13 13[CFG] <1>   no acceptable INTEGRITY_ALGORITHM found
Apr 19 21:04:13 13[CFG] <1> received proposals: IKE:AES_CBC_128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_2048
Apr 19 21:04:13 13[CFG] <1> configured proposals: IKE:AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
Apr 19 21:04:13 13[IKE] <1> no proposal found
Apr 19 21:04:13 13[IKE] <1> queueing INFORMATIONAL task
Apr 19 21:04:13 13[IKE] <1> activating new tasks
Apr 19 21:04:13 13[IKE] <1>   activating INFORMATIONAL task
Apr 19 21:04:13 13[ENC] <1> generating INFORMATIONAL_V1 request 3635494359 [ N(NO_PROP) ]
Apr 19 21:04:13 13[NET] <1> sending packet: from 192.168.90.39[500] to 192.168.90.37[500] (56 bytes)
Apr 19 21:04:13 13[IKE] <1> IKE_SA (unnamed)[1] state change: CONNECTING => DESTROYING
```

②当身份 id 不匹配时，其日志信息：

```
Apr 19 20:49:20 12[IKE] <1> 192.168.90.37 is initiating a Aggressive Mode IKE_SA
Apr 19 20:49:20 12[IKE] <1> IKE_SA (unnamed)[1] state change: CREATED => CONNECTING
Apr 19 20:49:20 12[CFG] <1> selecting proposal:
Apr 19 20:49:20 12[CFG] <1>   proposal matches
Apr 19 20:49:20 12[CFG] <1> received proposals: IKE:AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
Apr 19 20:49:20 12[CFG] <1> configured proposals: IKE:AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
Apr 19 20:49:20 12[CFG] <1> selected proposal: IKE:AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
Apr 19 20:49:21 12[CFG] <1> looking for pre-shared key peer configs matching 192.168.90.39...192.168.90.37[192.168.91.32]
Apr 19 20:49:21 12[CFG] <1> peer config "vpn1", ike match: 3100 (192.168.90.39...192.168.90.37 IKEv1)
Apr 19 20:49:21 12[CFG] <1>   local id match: 1 (ID_ANY)
Apr 19 20:49:21 12[CFG] <1>   remote id match: 0 (ID_IPV4_ADDR: c0:a8:5b:20)
Apr 19 20:49:21 12[IKE] <1> no peer config found
Apr 19 20:49:21 12[IKE] <1> queueing INFORMATIONAL task
Apr 19 20:49:21 12[IKE] <1> activating new tasks
Apr 19 20:49:21 12[IKE] <1>   activating INFORMATIONAL task
```

③当共享秘钥不匹配时，其日志信息：

```
Apr 19 21:10:18 04[CFG] <vpn1|1> selecting proposal:
Apr 19 21:10:18 04[CFG] <vpn1|1>   proposal matches
Apr 19 21:10:18 04[CFG] <vpn1|1> received proposals: IKE:AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
Apr 19 21:10:18 04[CFG] <vpn1|1> configured proposals: IKE:AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
Apr 19 21:10:18 04[CFG] <vpn1|1> selected proposal: IKE:AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
Apr 19 21:10:18 04[IKE] <vpn1|1> calculated HASH does not match HASH payload
Apr 19 21:10:18 04[IKE] <vpn1|1> queueing INFORMATIONAL task
Apr 19 21:10:18 04[IKE] <vpn1|1> activating new tasks
Apr 19 21:10:18 04[IKE] <vpn1|1>   activating INFORMATIONAL task
Apr 19 21:10:18 04[ENC] <vpn1|1> generating INFORMATIONAL_V1 request 1922189643 [ HASH N(AUTH_FAILED) ]
Apr 19 21:10:18 04[NET] <vpn1|1> sending packet: from 192.168.90.39[500] to 192.168.90.37[500] (92 bytes)
Apr 19 21:10:18 04[IKE] <vpn1|1> IKE_SA vpn1[1] state change: CONNECTING => DESTROYING
Apr 19 21:10:18 06[CFG] vici client 2 disconnected
```

④当感兴趣流不匹配时，其日志信息：

```
Apr 20 11:29:26 03[NET] <vpn1|1> received packet: from 192.168.90.37[4500] to 192.168.90.39[4500] (364 bytes)
Apr 20 11:29:26 03[ENC] <vpn1|1> parsed QUICK_MODE request 3657100888 [ HASH SA No KE ID ID ]
Apr 20 11:29:26 03[CFG] <vpn1|1> looking for a child config for 192.16.2.0/24 === 192.16.3.0/24
Apr 20 11:29:26 03[CFG] <vpn1|1> proposing traffic selectors for us:
Apr 20 11:29:26 03[CFG] <vpn1|1>  192.16.2.0/24
Apr 20 11:29:26 03[CFG] <vpn1|1> proposing traffic selectors for other:
Apr 20 11:29:26 03[CFG] <vpn1|1>  192.16.1.0/24
Apr 20 11:29:26 03[IKE] <vpn1|1> no matching CHILD_SA config found for 192.16.3.0/24 === 192.16.2.0/24
Apr 20 11:29:26 03[IKE] <vpn1|1> queueing INFORMATIONAL task
Apr 20 11:29:26 03[IKE] <vpn1|1> activating new tasks
Apr 20 11:29:26 03[IKE] <vpn1|1>   activating INFORMATIONAL task
Apr 20 11:29:26 03[ENC] <vpn1|1> generating INFORMATIONAL_V1 request 1956105377 [ HASH N(INVAL_ID) ]
```