

Yang CHEN, Hong-chao HU, Guo-zhen CHENG, 2019. Design and implementation of a novel enterprise network defense system by maneuvering multi-dimensional network properties. *Frontiers of Information Technology & Electronic Engineering*, 20(2):238-252. <https://doi.org/10.1631/FITEE.1800516>

Design and implementation of a novel enterprise network defense system by maneuvering multi-dimensional network properties

Key words: Intranet defense; Software-defined network; Multi-dimensional maneuvering

Corresponding author: Hong-chao HU

E-mail: 13633833568@139.com

 ORCID: Yang CHEN, <http://orcid.org/0000-0001-7806-2066>

Motivation

1. The means of attack have become more complex, e.g., advanced persistent threat (APT), social engineering attacks, and zero-day exploits. Their attack behavior cannot be effectively detected by the traditional security detection system.
2. The boundaries of the enterprise network tend to be blurred, especially in recent years. With the development of the mobile Internet, bring your own device (BYOD) has become a new model for workplaces. While this model reduces enterprise costs, attackers can easily penetrate the internal resources through these devices.

Main idea

1. System architecture design:

- (1) Design principles;
- (2) General framework;
- (3) Maneuvering mechanism;

2. Implementation of cyber defense solution (SPD):

- (1) Communication protocols;
- (2) Packet classification;
- (3) IP allocation;
- (4) Path maneuvering.

1. System framework

Data plane:

This plane enforces packet forward according to the dynamic transform rules.

Management plane:

This plane is responsible for the initial system configuration and network operational status.

Control plane:

- (1) Configure resources for hosts and maintain a dynamic virtual configuration;
- (2) Establish a session and a transmission path for communication, and maintain a dynamic external network port.

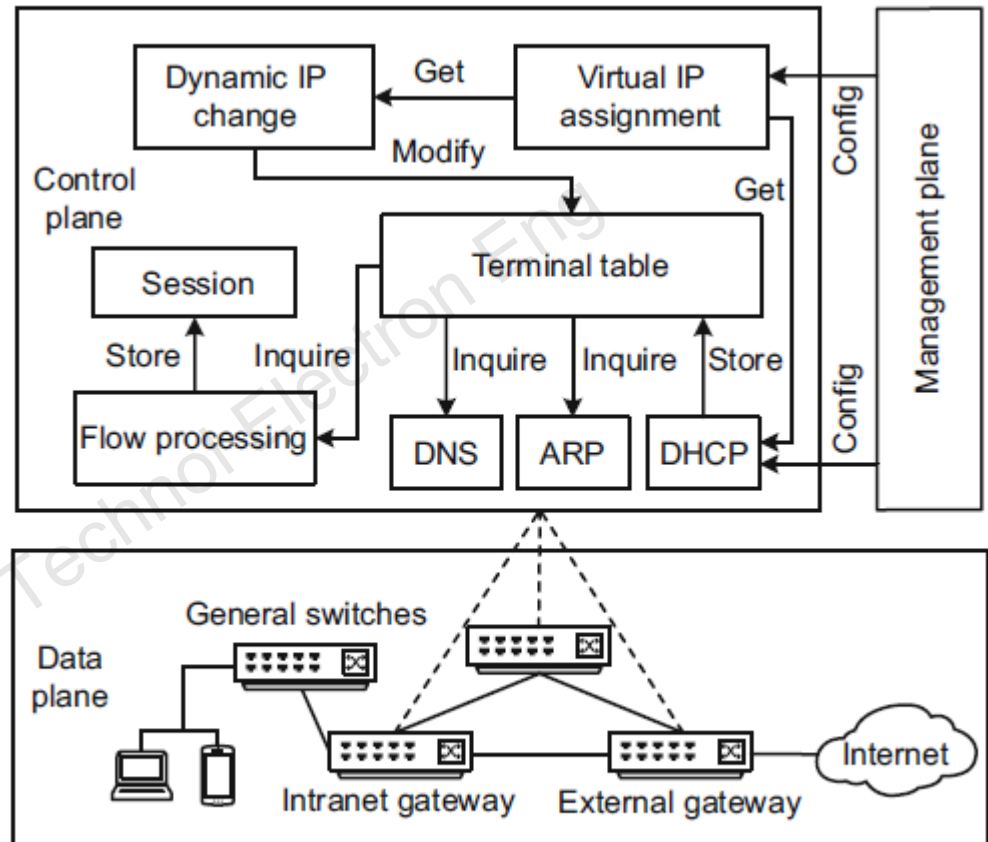


Fig. 1 System framework

2. Communication protocols

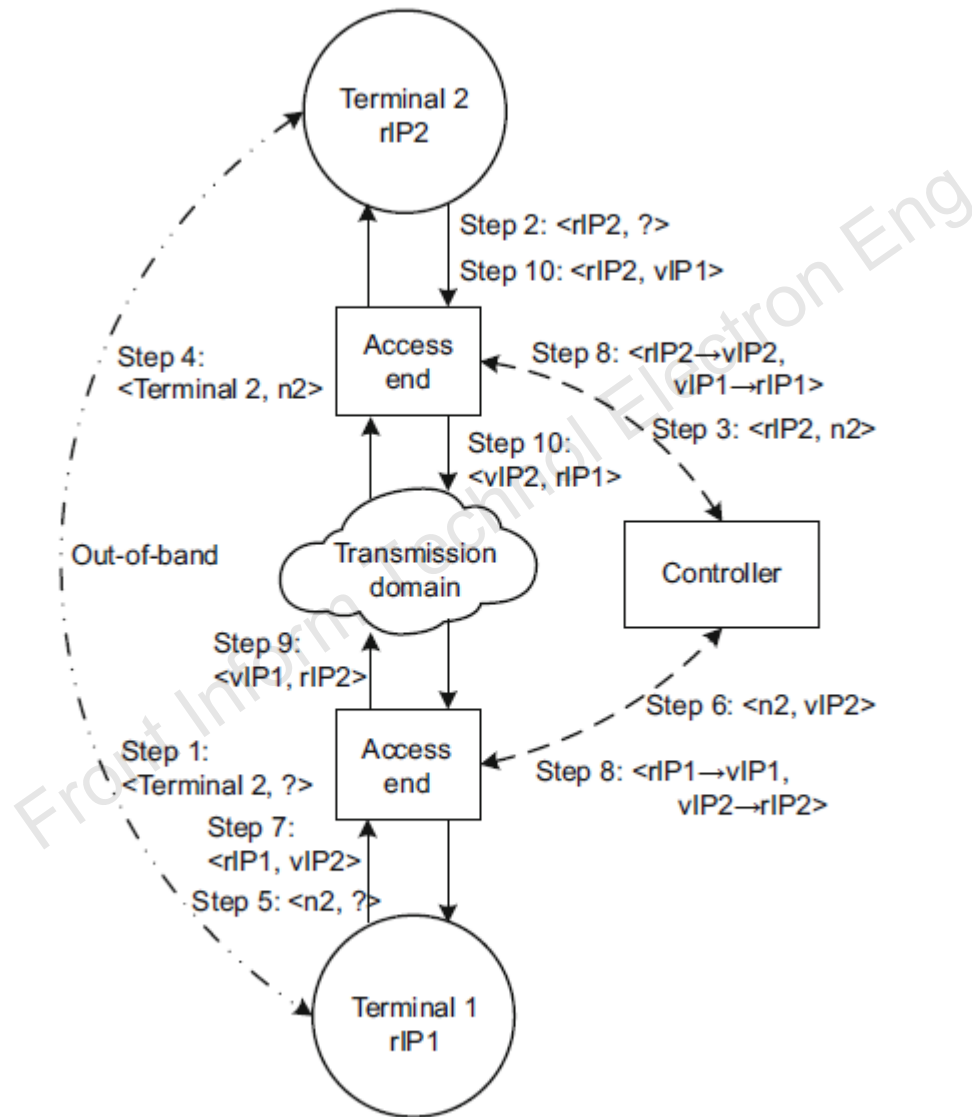


Fig. 2 Communication processing

3. Packet classification

When an IP packet is sent to the controller, the controller needs to determine to which type the packet belongs:

- (1) The sender-to-access packet, the next hop is the receiving end;
- (2) The sender-to-access packet, the next hop is not the receiving end;
- (3) The message from the non-access end, the next hop is the receiving end;
- (4) The message from the non-access end, the next hop is not the receiving end.

Algorithm 1 Communication processing

```
1: for all Packet  $p$  comes from the OF switch do
2:   if  $p.src$  is rIP and  $p$  is from the access end then
3:     if  $p.dst$  is vIP then
4:        $R_1 \leftarrow$  the path from  $h_i$  to  $h_j$ 
5:       for all switch  $s$  in  $R_1$  do
6:         if  $s \in \text{Type.A}$  then
7:            $f_i(h_i.rIP \rightarrow h_i.vIP, h_j.vIP \rightarrow h_j.rIP,$ 
              $dst\_mac \rightarrow h_j.mac, output : s.out)$ 
8:            $f_b(h_j.rIP \rightarrow h_j.vIP, h_i.vIP \rightarrow h_i.rIP,$ 
              $dst\_mac \rightarrow h_i.mac, output : s.in)$ 
9:         else if  $s \in \text{Type.B}$  then
10:           $f_i(h_i.rIP \rightarrow h_i.vIP, h_j.vIP \rightarrow h_j.rIP,$ 
             $dst\_mac \rightarrow h_j.mac, output : s.out)$ 
11:           $f_b(h_j.vIP, h_i.rIP, output : s.in)$ 
12:        else if  $s \in \text{Type.C}$  then
13:           $f_i(h_i.vIP, h_j.rIP, output : s.out)$ 
14:           $f_b(h_j.vIP, h_i.rIP, output : s.in)$ 
15:        else if  $s \in \text{Type.D}$  then
16:           $f_i(h_i.vIP, h_j.rIP, output : s.out)$ 
17:           $f_b(h_j.rIP \rightarrow h_j.vIP, h_i.vIP \rightarrow h_i.rIP,$ 
             $dst\_mac \rightarrow h_i.mac, output : s.in)$ 
18:        end if
19:        flow_mod( $f_i, f_b$ )
20:      end for
21:    end if
22:  end if
23: end for
```

Evaluation

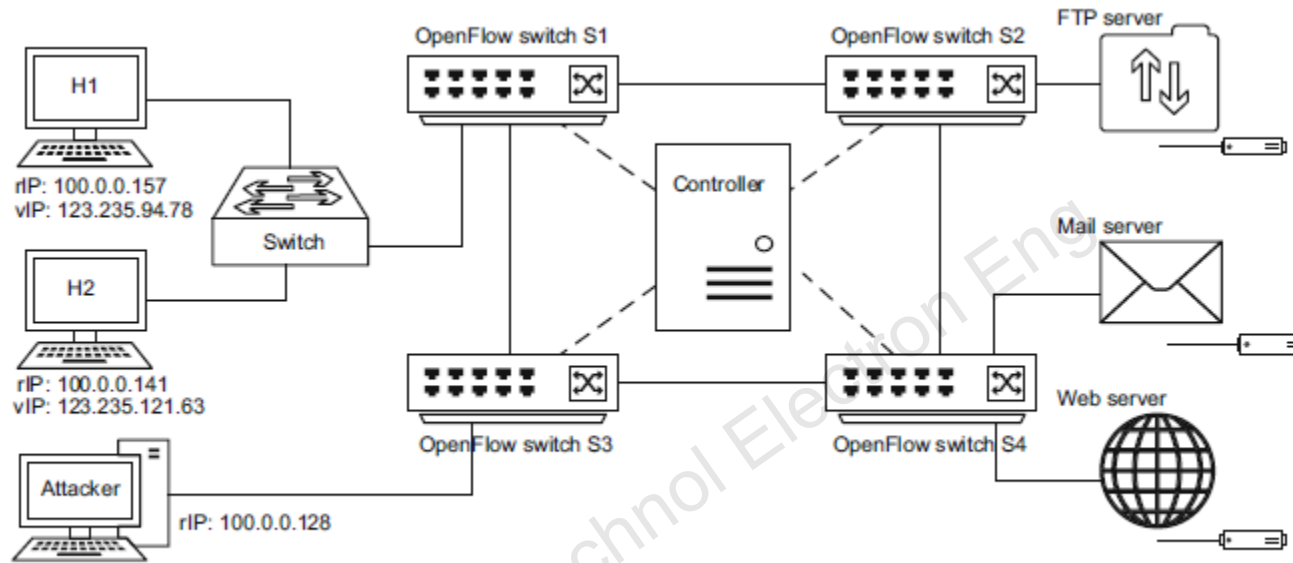


Fig. 4 Network topology

```
root@test:/home/cy# ifconfig
vnp2s0
  link encap:Ethernet  HWaddr f8:0f:41:20:0c:42
  inet addr:100.0.0.141  Bcast:100.0.0.255  Mask:255.255.255.0
  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
  RX packets:2417607 errors:0 dropped:0 overruns:0 frame:0
  TX packets:1128408 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:3608271195 (3.6 GB)  TX bytes:89461335 (89.4 MB)

root@test:/home/cy# nslookup 100.0.0.141
server:
  1.1.1.1
address:
  1.1.1.1#53

141.0.0.100.in-addr.arpa      name = ndsc737.

root@test:/home/cy# iperf3 -s
Server listening on 5201

Accepted connection from 123.235.94.78, port 4605
[ 5] local 100.0.0.141 port 5201 connected to 123.235.94.78 port 4607
ID Interval Transfer Bandwidth
[ 5] 0.00-1.00 sec 2.89 MBytes 24.2 Mbits/sec
[ 5] 1.00-2.00 sec 1.06 MBytes 8.88 Mbits/sec
[ 5] 2.00-3.00 sec 3.18 MBytes 26.7 Mbits/sec
[ 5] 3.00-4.00 sec 4.35 MBytes 36.5 Mbits/sec
[ 5] 4.00-5.00 sec 1.10 MBytes 9.18 Mbits/sec
```

(a)

```
C:\Users\Administrator\Desktop\iperf-3.1.3-win64
A nslookup ndsc737.
服务器: Unknown
Address: 1.1.1.1
名称: ndsc737
Address: 123.235.121.63

C:\Users\Administrator\Desktop\iperf-3.1.3-win64
A .\iperf3.exe -c 123.235.121.63
Connecting to host 123.235.121.63, port 5201
[ 4] local 100.0.0.157 port 4607 connected to 123.235.121.63 port 5201
ID Interval Transfer Bandwidth
[ 4] 0.00-1.00 sec 3.88 MBytes 32.1 Mbits/sec
[ 4] 1.01-2.01 sec 256 KBytes 2.10 Mbits/sec
[ 4] 2.01-3.01 sec 4.08 MBytes 33.6 Mbits/sec
[ 4] 3.01-4.01 sec 4.25 MBytes 35.7 Mbits/sec
[ 4] 4.01-5.01 sec 304 KBytes 3.15 Mbits/sec
[ 4] 5.01-6.01 sec 3.08 MBytes 32.6 Mbits/sec
[ 4] 6.01-7.00 sec 4.12 MBytes 34.7 Mbits/sec
[ 4] 7.00-8.00 sec 4.12 MBytes 34.7 Mbits/sec
[ 4] 8.00-9.00 sec 4.08 MBytes 33.6 Mbits/sec
[ 4] 9.00-10.02 sec 4.12 MBytes 34.1 Mbits/sec

ID Interval Transfer Bandwidth
[ 4] 0.00-10.02 sec 33.0 MBytes 27.6 Mbits/sec
[ 4] 0.00-10.02 sec 32.9 MBytes 27.5 Mbits/sec

iperf Done.
```

(b)

Fig. 5 Communication verification: (a) terminal H2;
(b) terminal H1

Evaluation

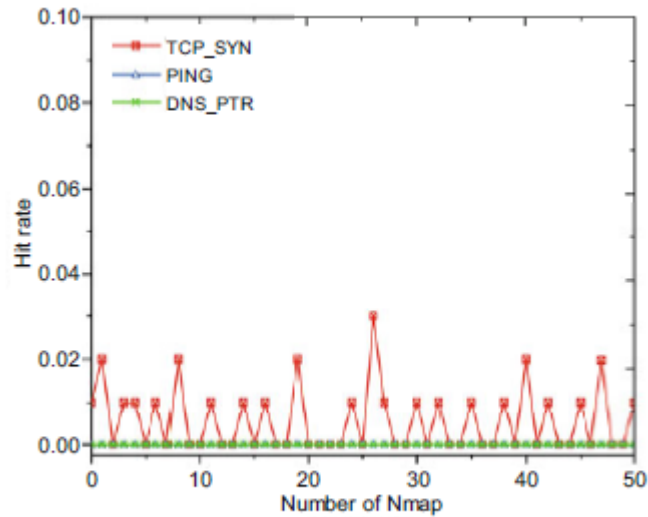


Fig. 6 Ratio of hits

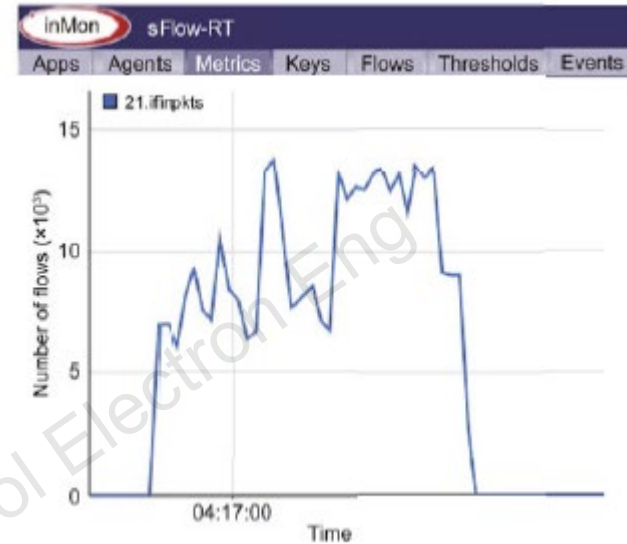


Fig. 7 Anti-denial-of-service (DoS) attack

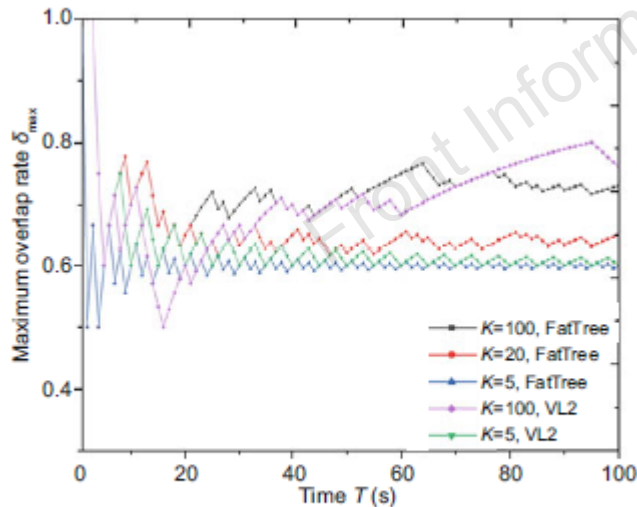


Fig. 8 Overlap rate of network

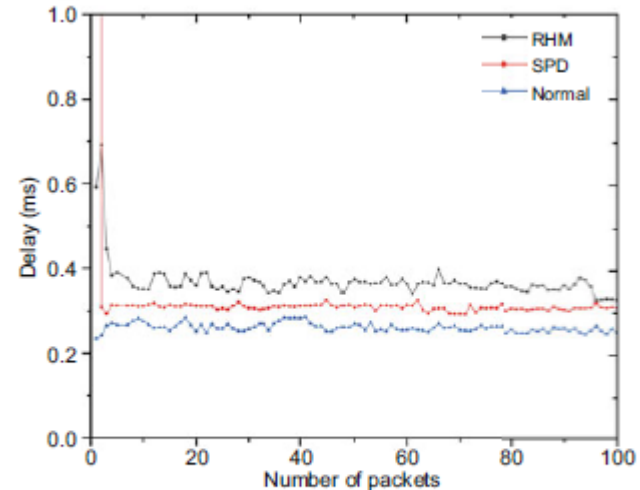


Fig. 10 Network delay caused by switch forwarding in RHM, SPD, and the normal modes

Conclusions

1. SPD with an appropriate alternation strategy can resist almost all network scanning attacks, and effectively intercept worms, DoS, and other unknown network attacks.
2. The flow overhead introduced by the system needs to be further optimized.