

Nokia HA State Synchronization and ACK Protection

Stephen Gill
E-mail: gillsr@cymru.com
Revision: 1.0, 04/18/2001

Contents

Introduction	2
What is state synchronization and why should I enable it?	2
What is ACK protection and when should I enable it?	3
How do I enable state synchronization?	4
How do I tell if state synchronization is working between my firewalls?	6
Notes	7

Introduction

The purpose of this document is to describe the implications of state table synchronization. Further, this document denotes those situations for which the state table is desirable. It also details the basics of configuring state table synchronization without interfering with s/key authentication between firewall modules and their Enterprise Management Center (EMC).

What is state synchronization and why should I enable it?

The state table tracks established connections through the firewall. It also improves flow performance by eliminating the requirement to match every packet against the rulebase. If an established *flow*¹ makes it to the firewall's connections table, the packet has been previously been allowed by the rulebase and will no longer need to be crosschecked against the rules.

Under high-availability situations, Checkpoint FireWall-1 provides the ability to maintain state synchronization between redundant firewalls. This means each firewall owns its own virtually identical copy of the same connections table. This will differ by no more than SYNC_INTERVAL. The standard interval for state synchronization is 50 milliseconds. In case one firewall fails, the other firewall should then recognize all valid connections and maintain existing state transparently.

Under normal circumstances within a datacenter, state table synchronization might be considered an unnecessary luxury, given the underlying behavior of Checkpoint FireWall-1. If the state table is cleared, Checkpoint dynamically rebuilds the connection table without affecting existing traffic. If Checkpoint sees a flow that has been established, such as ACKs flowing from end-station to end-station, which is not in its connections table, it will inspect that against its rulebase. If allowed, it will then add that entry automatically as an established connection. The above is true provided that no NAT, Encryption, or ACK protection is in place.

¹ Used here to refer to traffic associated with a specific source port, source IP, destination port, destination IP.

The necessity for state synchronization is actually quite minimal under “vanilla” circumstances, and would likely only be necessary if one of the following is true.

A. The presence of an asymmetric data flow.

Much care should be taken in one's internal network design to avoid asynchronous routing. Not only should you avoid it, but research has shown that state synchronization is not adequate enough to keep up with this because the sync interval is usually longer than the time it takes the packet to traverse both the firewalls in question.

B. You are performing NAT or Encryption in your rulebase.

C. You have implemented ACK protection.

If you have implemented ACK protection, you are effectively removing the dynamic state table rebuilding feature of Checkpoint for TCP. Thus, without synchronization, all established TCP connections would get dropped after a failover. Refer to section II for more details on ACK protection.

What is ACK protection and when should I enable it?

In recent months, an advisory regarding a DOS against the Checkpoint connection table was published at <http://www.securityfocus.com/vdb/bottom.html?vid=549>. Among other methods of minimizing the vulnerability the primary solution presented was a bit of inspect code that enables Checkpoint gateways to drop non-first TCP packets if they are not in the connections table instead of matching the rulebase. This is known as ACK protection.

When a Checkpoint firewall first receives a packet that is not in the connections table it will search the rulebase for a match. When an unregistered packet comes through the firewall such as a TCP ACK, if it is allowed by the rulebase, it will be added to the connection table with the assumption that this is an established session. The timeout for such connections will then be set to the configured duration, typically 3600 seconds by default. One could theoretically use this as a DOS by sending a steady stream of spoofed ACK packets to a host behind a Checkpoint firewall and cause its table to fill before the timeout expires. When the connections table fills up, connectivity through the firewall suffers greatly. The attack becomes harder to mitigate for inbound connections if TCP RST packets are sent in response to the rogue ACK packets because Checkpoint will decrease the timeout for such sessions to a much smaller value (typically 50 seconds). However, it is not difficult to model a few scenarios where this vulnerability could easily be used as a very effective Denial of Service attack.

Typically ACK protection has only been necessary on high profile sites. However, one should know how to prevent the effects of this vulnerability in the event of such an attack on a production network.

The associations between ACK protection and state synchronization become quite clear at this point. With ACK protection enabled and state synchronization disabled, we no longer have the luxury of allowing existing connections to be rediscovered in the event of a failover.

It will be necessary to make code modifications in '\$FWDIR/lib/code.def' to enable ACK protection. However, there exist a few variations of the inspect code that are version dependant. If you are running a version of Checkpoint other than v.4.0 (4.1 and greater, or 3.0) and require the inspect code, please reference the code changes under Nokia's knowledge base Resolution #1710 or under Checkpoint's knowledge base under ACK Protection. For Checkpoint version 4.0, one would do the following:

Insert the inspect code below at the end of the '\$FWDIR/lib/code.def' file, just before the #endif statement, then re-install the security policy. This code will also log these events.

```
----- 4.0 edit follows -----
#ifdef ALLOW_NONFIRST_RULEBASE_MATCH
    tcp, first or <conn> in old_connections or
    (src in firewalled_list, dst in firewalled_list) or
    (
#ifdef NO_NONFIRST_RULEBASE_MATCH_LOG
    (
        <ip_p,src,dst,sport,dport,0> in logged
    ) or (
        record <ip_p,src,dst,sport,dport,0> in logged,
        set sr10 12, set sr11 0, set sr12 0, set sr1 0,
        log bad_conn
    ) or 1,
#endif
    vanish
);
#endif
----- End of 4.0 insert -----
```

How do I enable state synchronization?

A. Assuming you have already reserved a port on each Firewall for this purpose, attach an Ethernet crossover cable to both ends and be sure to set hardcode the interfaces to 100/Full Duplex. Next pick two private

RFC 1918 IP addresses in the same subnet. It may be preferable to pick from a pre-assigned subnet for this purpose, and then allocate one to each interface.

Do not forget to add a firewall rule that will allow bi-directional connections between the chosen IP addresses. If for some reason firewall clusters ever need to be joined, IP pairs should be reassigned in the same subnet so as not to collide with each other.

B. Create a file '\$FWDIR/conf/sync.conf' on both modules. The file should contain the IP address of the peer module's crossover interface.

Run 'fwstop' on both modules

Run 'fw putkey -n <local module IP address> <remote module IP address>' on both modules

Run 'fwstart' on both modules

C. In addition to enabling synchronization, you will also need to make sure the system clocks of the two modules are in sync. The built-in NTP client and server in IPSO 3.1 and later can be used to ensure that the clocks do not stray too far.

On the Primary Nokia

Click CONFIG | NTP

Click YES in the ENABLE NTP field, APPLY.

Click YES under NTP Master, and select "Local Clock" under Clock Source.

Click APPLY, then SAVE.

On the Secondary Nokia

Click CONFIG | NTP

Click YES in the ENABLE NTP field, then APPLY.

Enter the IP address of the primary Nokia's crossover link (IP address of NTP server), in the "Add new server Address" box

Click APPLY, then SAVE.

You can refer to the NTP FAQ resolution #1446 for more information on how to configure NTP between Firewalls. If possible, secure NTP servers should be utilized as the NTP servers for the firewalls.

D. You should increase the default MSS size of IPSO's TCP/IP stack (512 bytes). This is done with the following command (this should be added to /var/etc/rc.local so it is performed on start up):

```
ipsctl -w net:ip:tcp:default_mss 1460
```

The above command will need to be issued on both Firewalls in order for it to have any effect. To probe the current value of the MSS and confirm your change, use the following command:

```
ipsctl -a net:ip:tcp:default_mss
```

For more information on how the Maximum Segment Size (MSS) works, refer to RFC 879.

E. Authentication on the Nokias can be unreliable, and you may experience sporadic authentication issues between the modules and the EMC or between modules themselves. The standard “fw putkey” commands may sometimes need to be entered constantly. There is a known issue in versions of FireWall-1 prior to SP5 whereby authentication can get out of sync if you are using s/key. If you cannot upgrade to SP5 you should either try using fwn1 or turn off authentication all together - both of these changes are made in ‘\$FWDIR/lib/control.map’. To disable authentication between two hosts, one would simply add two lines to the ‘control.map’ file of both hosts above the MASTERS line like so:

```
a.b.c.d, e.f.g.h : */none  
MASTERS ...  
CLIENTS ...
```

Where Host 1’s IP is a.b.c.d and Host 2’s IP is e.f.g.h. Disabling authentication over the crossover connection should certainly not be a concern since that is an isolated network between two modules. In a similar manner, authentication between the EMCs and the modules can be sacrificed since it is a private isolated network.

If authentication is completely broken and the “fw putkey” commands don’t resolve it, you may consider trying a few of the tips laid out on phoneboy’s website <http://www.phoneboy.com/fw1>.

How do I tell if state synchronization is working between my firewalls?

There are a few basic methods of determining if sync is working properly on a pair of firewalls.

First, look at the file ‘\$FWDIR/conf/sync.conf ’ on both firewalls. This file lists what IP address each firewall will attempt to sync to. This should be the IP address of the crossover interface on the other side of the state sync link.

Next, run `'netstat -na | grep 256'` on each machine. You should see something like:

```
tcp 0 0 10.0.0.1.256 10.0.0.2.1056 ESTABLISHED
tcp 0 0 10.0.0.1.1054 10.0.0.2.256 ESTABLISHED
```

The important numbers in the netstat output are the crossover IP addresses of each of the two and the port 256. This is the port used between firewalls for state sharing purposes – if both ends are open on port 256 it is a good sign that state synchronization is working.

You can also confirm the size of the connections table on each firewall by using the command:

`'fw tab -t connections -s'`. Be sure to enter this command at approximately the same time on both ends. If the numbers are roughly equivalent on both firewalls then state synchronization should be working. If there are differences, wait a few seconds and try it again.

The above can be referenced on Nokia's support website in the knowledge base under resolution #1699.

Notes

Nokia's knowledge base can be reached from the support website at <http://support.iprg.nokia.com> and requires a valid login name and password. Most of the information in this document has been gleaned directly from the knowledge base or from Checkpoint's website at <http://www.checkpoint.com/services> which may also require a login and password.