Global Information Assurance Certification Paper

Tracking single points of failure: firewall load balancing – A case study
Pierre Amoudruz
January 09, 2003
GSEC Version 1.4b Option #2 Case Study

Abstract

In today's world, availability has become more and more important as the amount of data exchanged through electronic media increases. Firewall boxes are playing a key role in the security perimeter of the company: they control traffic between users of the company and the outside world, whether it is supplier networks or the Internet. However, firewalls might be subject to failure and consequences in case of failure might be important and lead to a complete loss of connectivity for a site. Few years ago, ensuring an appropriate level of security was the main objective to achieve and availability was not really taken in account. On the contrary, we could not now define a security design without availability in mind.

I will present a case study on the set up of a high availability solution for Checkpoint firewalls using a software-based solution StoneBeat Fullcluster. We will first depict the current architecture for a common site and highlight the main weakness of the network design. Then we will detail the way StoneBeat Fullcluster performs high availability and describe key concepts in this area. This will lead us to present a detailed design that should improve the level of redundancy for this site. We will focus on firewall configuration and impacts on the local network of the new architecture. Finally, we will briefly present a testing phase and show the way the proposed design fulfills redundancy requirements for this site.

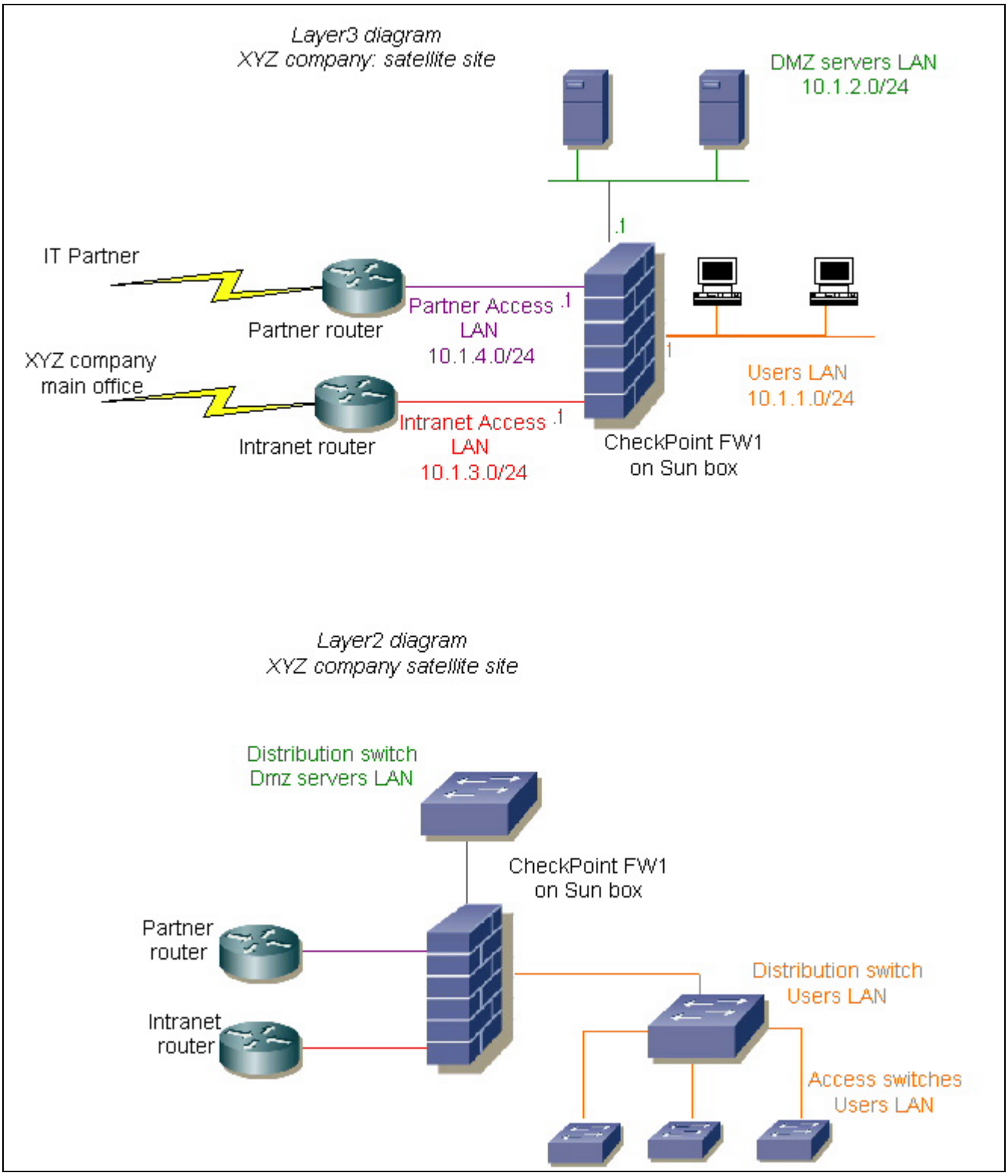
Presentation of the current architecture

Description of the business
We will base this case study on an IT company XYZ that principally deals with software development. This company includes a main office and a satellite office located in 2 different locations.
The focus is set on the satellite office that hosts almost 200 users involved in web site development. The satellite office is connected to the main office and to 1 IT partner. Development servers are hosted in the satellite office and are both accessed by local users and remote users from the main office and the IT partner.

Network topology
The site is segregated into 4 LANs all connected to the firewall:

- Users LAN: all the users of this site sit on this LAN. A class C network is used for IP addressing.
- Server DMZ LAN: it hosts servers used for web development purposes.

- Intranet access LAN: this LAN enables the connection of the satellite office to the main office. Developers from the main office need to connect to servers hosted on the satellite site. Internet access is provided through the main office gateway.
- Partner access LAN: this LAN enables the connection of the satellite site to the partner. Partner developers need to connect to servers hosted on the satellite site.



Security perimeter

A SUN box running Solaris 8 and CheckPoint Firewall 1 is used to protect the 4 enclaves of this site from each other. This firewall is administered remotely by a management station located in the main office, access is granted through the Intranet access LAN.

We will detail below the key points of the security policy concerning the firewall configuration:

- The generic rule that applies is Accept, Accept and then finally Drop. To be more explicit, this means that all traffic not explicitly authorized will be dropped. This method offers a good way to control traffic between the different enclaves.
- From the user LAN standpoint, all LANs but the main office user LAN is considered as non trusted. This means first that full access is granted between user LANs of the company and second that if not especially authorized, all other inbound traffic destined to the satellite user LAN will be blocked.
- Outbound traffic from the satellite user LAN and the main office LAN is not restricted.
- Inbound traffic to the server DMZ LAN is granted for the partner LAN on a per protocol basis.

Below is the default traffic table between enclaves

		Destination			
		Satellite user LAN	Server DMZ LAN	Main office user LAN	Partner LAN
Source	Satellite user LAN		Any	Any	Any
	Server DMZ LAN	Nothing		Nothing	Nothing
	Main office user LAN	Any	Any		Any
	Partner LAN	Nothing	On a per protocol basis	Nothing	

Analysis of the architecture

This architecture provides a good level of security for users located on the local LAN and is well adapted for a small to medium size site. Due to the small number of users located in the satellite office when the architecture was first built, availability was not considered as a key requirement.

However, the situation has changed. The number of users on site has grown; electronic communication between the satellite office, main office and the partner has dramatically increased. This leads to a point where availability of equipments is critical for the business of the XYZ Company.

Weakness and limitations of the current architecture
We list here the points of failure and limitations of the current architecture.

Firewall failure
Affected: local users and remote users from main office and partner office.
Severity: high, traffic between enclaves is completely blocked.

Router failure
Affected: mainly remote users from main office and partner office (as local users cross only the local firewall to access their development servers).
Severity: medium

Link failure
Affected: mainly remote users from main office and partner office (as local users cross only the local firewall to access their development servers).
Severity: medium

Server failure
Affected: both local and remote users from main office and partner office.
Severity: medium as only part of the developers will be concerned by the failure of the server on which they are working.

Performance limitation
As the number of people working on development servers of the satellite site has increased, local users start complaining about bad response times and available bandwidth when accessing the servers located on the DMZ.
Severity: medium

Switch failure
Affected: depending on the switch, mainly local users, but also remote users (in case of a DMZ switch failure).
Severity: low (a backup plan based on a re-cabling solution enables to recover connectivity in case of switch failure).

Implement high availability
Following the highlighted risks, management has decided to implement redundancy on the firewall as it is pointed to be the main point of failure. The 3 major criteria the new architecture should meet are: first provide redundancy on firewall boxes, second improve performance and third limit modifications on existing LAN infrastructure.
We had to choose between two main types of solution:

- Hardware-based architecture: separate hardware boxes are added to the existing infrastructure to provide high availability by directing traffic to a set of firewall systems.

- Software-based architecture: a software component is installed on each firewall part of a "firewall farm". This enables multiple firewalls to be viewed as a single virtual machine that receives traffic and distributes it among members of the "farm".

We only give some pointers for the choice the appropriate solution as a detailed comparison is out of the scope of the paper:

- In addition to redundant firewalls, hardware-based architecture requires additional hardware (boxes for high availability) that can be more expensive than software license.
- Software-based solution is better integrated with the firewall application. This enables to efficiently monitor performance of each member of the "firewall farm" and distributes traffic accordingly.
- Impact on the network of software-based solution is generally limited.

We have opted for an architecture using a software-based solution to set up high availability for this site. The product StoneBeat Fullcluster from StoneSoft was retained. Many vendors offer such products and we will not present a benchmark between all of these products.
Our main goal here is to provide a detailed description of the architecture based on StoneBeat Fullcluster and present how this architecture can fulfill the needs expressed by the XYZ Company.

High availability using StoneBeat Fullcluster

We will detail here the principles on top of which StoneBeat Fullcluster is built.

High availability:
StoneBeat Fullcluster makes 2 firewalls part of a single machine with its own identity called a cluster. If one node of the cluster fails, the other one will take over its traffic automatically. StoneBeat Fullcluster also monitors node status for failure recovery. If the failure is no longer present on the node of the cluster that failed, it has to be brought back to a normal functioning state.
StoneBeat provides 2 ways of configuring high availability: hot standby or load balancing.

Hot standby mode
In the first case, only one node can handle traffic at a time and the other one is a pure backup. The status of the node is online for the one that handles traffic and standby for the one that is used as a backup. If the online node fails, it will be put offline; the standby node will be put online and take over the entire traffic.
Load balancing mode
In the second case, both nodes can forward traffic at a time. Both nodes are in the online state. If one online node fails, it will be forced to go offline; the other one will take over the entire traffic.

State synchronization

CheckPoint firewall 1 is using stateful inspection to filter packets. In few words, each new connection is compared to a set of rules (based on the security policy detailed above). If the connection is authorized, it will be logged in the state table. Then, all new incoming packets are compared to the state table and if they belong to one of the connections that are logged in the state table, they will be authorized through the firewall. Using stateful inspection greatly improves the filtering capacity of the firewall.

For a proper functioning of high availability, each node of the cluster should be aware of the connections that are logged in the state table of the other node. In this case, if one node of the cluster fails, the other node will effectively authorize packets belonging to connections first established through the node that has failed.

State synchronization feature on CheckPoint firewall 1 has to be enabled to ensure proper functioning of the high availability architecture.

Increase performance

Increasing available throughput is also part of the requirements. Using the load-balancing configuration detailed above, each node handles a certain amount of traffic that flows through the cluster and as a consequence the cluster works more efficiently than one single machine.

StoneBeat Fullcluster configured in load balancing mode aims at optimizing the load on the different nodes of the cluster. Here is the way StoneBeat performs this. Capacity benchmark is performed during StoneBeat initial set up. This value represents the maximum load a node can handle. In addition to this initial benchmark, StoneBeat Fullcluster monitors the load of each node on a regular basis during normal functioning. This calculation is based on several parameters like CPU, interface activity. Once a new connection is initiated through the cluster, StoneBeat compares the current load of each node to its maximum capacity in order to affect the connection to the node that has the better capacity to handle it (ie: the least loaded node among the cluster). One important thing to remember is that once a connection is directed to one node, it will stick with this node till the connection ends or the node fails.

StoneBeat Fullcluster characteristics

Clustering

All nodes of the cluster exchange periodic information on the status of the cluster (state of node -online, offline-, load of each node). This ensures that all members of the cluster maintain an identical view of the cluster. These data are exchanged using a special clustering protocol based on multicast Ethernet frames.

We set up a dedicated heartbeat network between the 2 nodes of the cluster for clustering information to be exchanged. We also use this dedicated heartbeat network for CheckPoint Firewall 1 state synchronization.

Dedicated and cluster addresses

As explained above, the cluster can be considered as a machine with its own network identity. Both nodes of the cluster have to support the identity of the cluster (cluster IP and MAC addresses) in addition to their own identity (dedicated IP and MAC addresses).

Dedicated addresses

Each interface on each node has its own dedicated IP address for node access. Apart from this dedicated IP address, each interface owns a unicast MAC address that is mapped to its dedicated IP address.

Cluster addresses

Cluster IP addresses

All nodes of the cluster share a unicast IP address per interface. In order to avoid reconfiguring equipments of the site, the cluster IP addresses should be the same of the ones used by the original firewall without the high availability configuration. For example, the default gateway of the user LAN that was 10.1.1.1 should remain the same and the cluster IP address should be set in accordance.

There is no need to define a cluster IP address for heartbeat network interfaces; only the dedicated IP addresses are used.

Cluster MAC addresses

In addition to the cluster IP address, both nodes share a cluster MAC address per interface. Equipments like routers or stations directly connected to the cluster are using a cluster MAC address to send traffic that has to flow through the cluster. For load balancing to be set up properly, Ethernet frames destined to the cluster have to be directed to both nodes of the cluster. Here come two solutions: configure the cluster MAC addresses in either unicast or multicast mode.

Use of multicast MAC addresses

In this case, all nodes share a multicast MAC address per interface. Once a switch receives a frame with a multicast address as destination, it will flood all ports of the same VLAN with the frame and therefore ensures that both nodes of the cluster will effectively receive the frame.

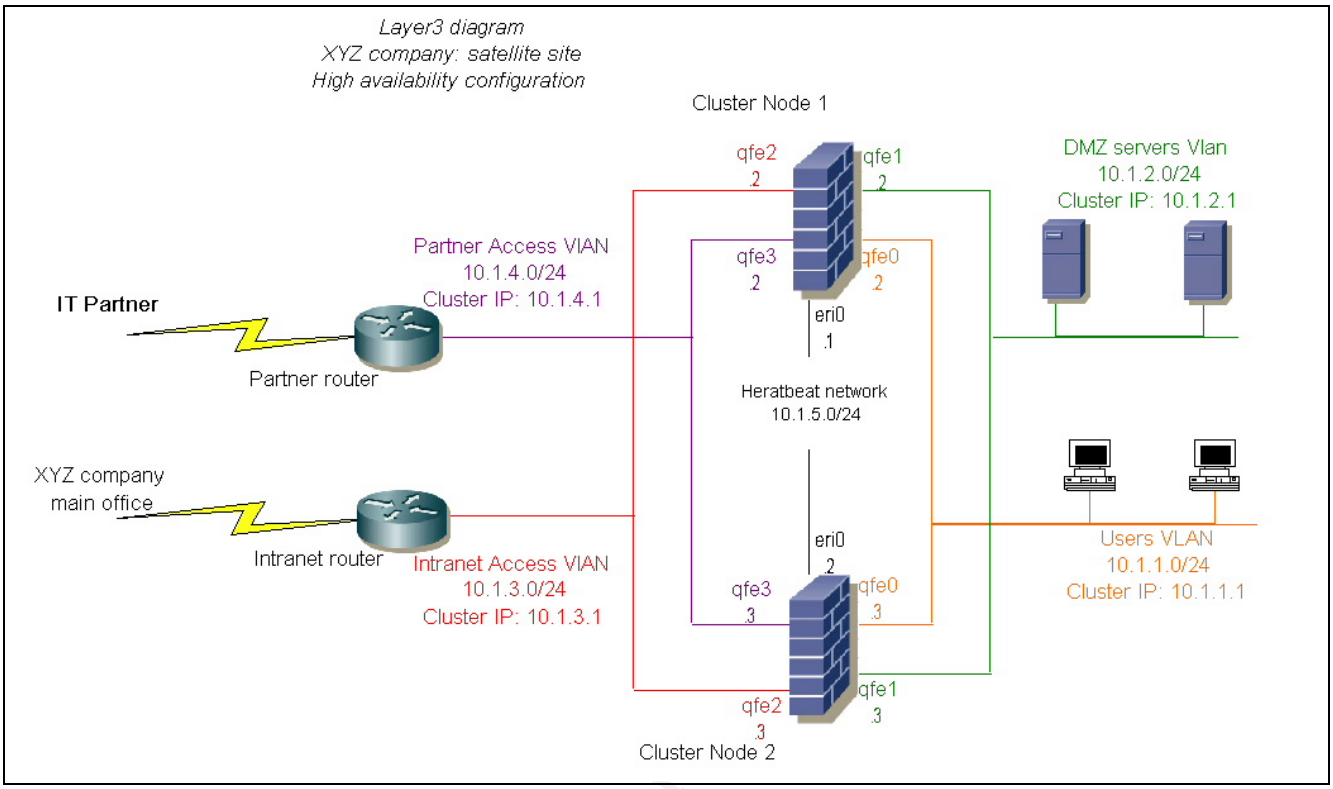
Interfaces connected to the heartbeat network have to be set up using multicast mode. For other interfaces, we prefer limiting broadcast traffic. Even if switches can be configured to forward multicast frames to selected ports and then limit broadcast traffic, we also want to limit interventions on site equipments. We opted for the second method.

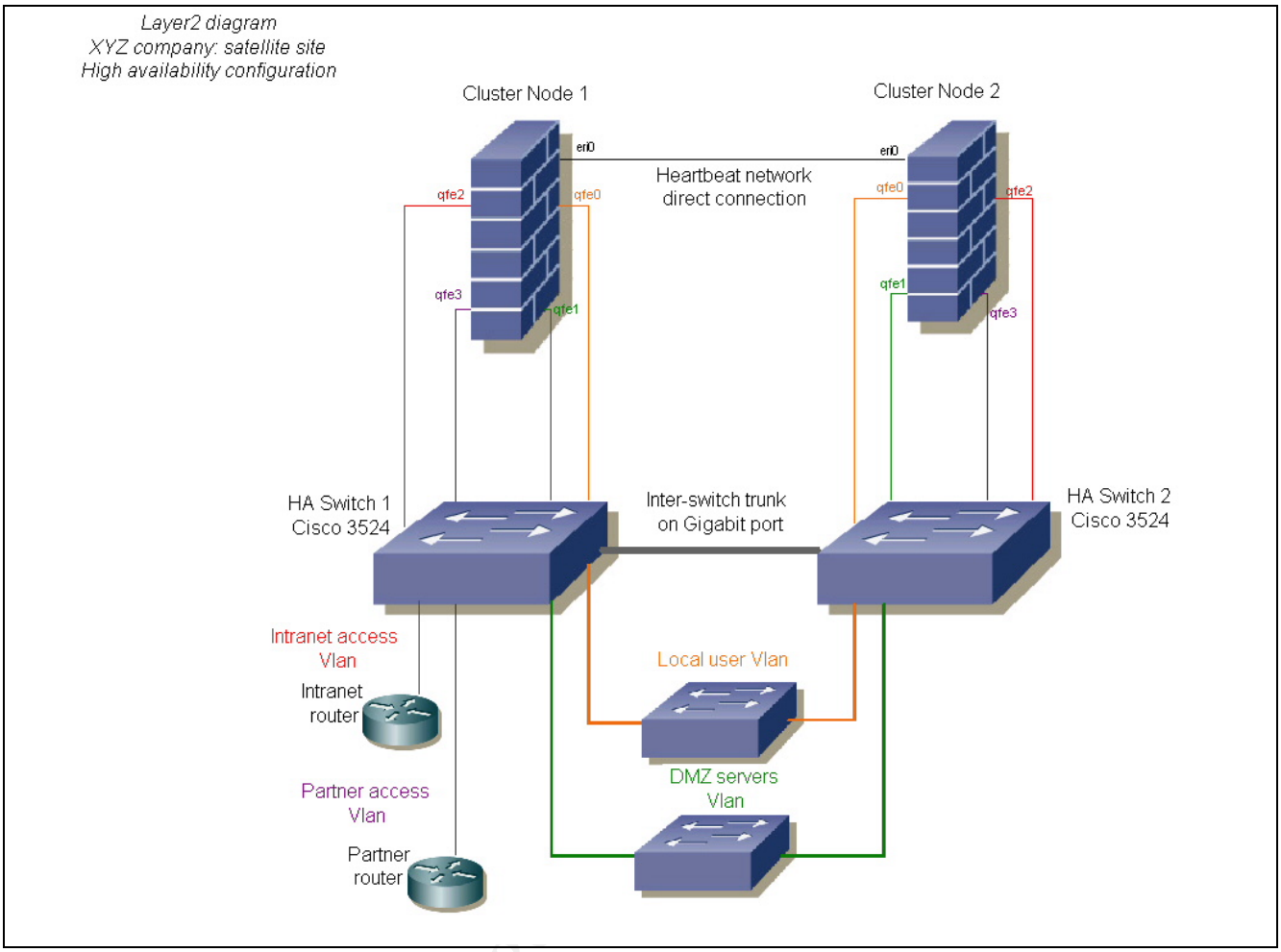
Use of unicast MAC addresses

In this case, all nodes of the cluster share the same unicast cluster MAC addresses per interface. HA switches will receive frames with unicast cluster MAC addresses that have to be directed to both node of the cluster. HA switches have to be chosen in accordance to this point: being able to forward unicast frames to more than one port.

As our requirements demand both high availability and increase of performance, a load balancing solution using StoneBeat Fullcluster in unicast mode was deployed to fulfill the expressed needs.

Major steps to set up the new architecture





We will briefly describe the set up of the load balancing architecture. The goal here is not to provide a "how to" but highlight key points of the set up.

Sun boxes

The first step is configuration of the 2 new Sun boxes:

- Installation of Solaris 8 with latest patches. A secure procedure should be followed on the 2 boxes. (Remove unneeded packages, disable unneeded services…). A good way to secure Solaris 8 can be found here: Yet Another Solaris Security Package (www.yassp.org).
- Installation of Checkpoint firewall 1 Feature Pack 1.
- Installation of StoneBeat Fullcluster 3.0 plus Feature Pack 1 and Hot fix 5.

System configuration

On each node, we need to configure both the dedicated IP addresses along with the cluster IP addresses. Cluster IP addresses are set up using Solaris 8 sub-interfaces.
MAC configuration (multicast/unicast) for the cluster will be performed through the StoneBeat interface.

Node 1					
VLAN	Network	System interfaces	Dedicated IP	System sub Interfaces	Cluster IP
Users VLAN	10.1.1.0/24	qfe0	10.1.1.2	qfe0:1	10.1.1.1
DMZ server VLAN	10.1.2.0/24	qfe1	10.1.2.2	qfe0:2	10.1.2.1
Intranet access VLAN	10.1.3.0/24	qfe2	10.1.3.2	qfe0:3	10.1.3.1
Partner access VLAN	10.1.4.0/24	qfe3	10.1.4.2	qfe0:4	10.1.4.1
Heartbeat network	10.1.5.0/24	eri0	10.1.5.1	eri0:1	N/A as there is no cluster IP for heartbeat network

Node 2					
VLAN	Network	System interfaces	Dedicated IP	System sub Interfaces	Cluster IP
Users VLAN	10.1.1.0/24	qfe0	10.1.1.3	qfe0:1	10.1.1.1
DMZ server VLAN	10.1.2.0/24	qfe1	10.1.2.3	qfe0:2	10.1.2.1
Intranet access VLAN	10.1.3.0/24	qfe2	10.1.3.3	qfe0:3	10.1.3.1
Partner access VLAN	10.1.4.0/24	qfe3	10.1.4.3	qfe0:4	10.1.4.1
Heartbeat network	10.1.5.0/24	eri0	10.1.5.2	eri0:1	N/A as there is no cluster IP for heartbeat network

For example, on node 1:

```
/etc/hostname.qfe0
qfe0-dedicated-interface #dedicated IP address

/etc/hostname.qfe0:1
qfe0:1-cluster-interface #cluster IP address

/etc/hosts
10.1.1.2 qfe0-dedicated-interface
10.1.1.1 qfe0:1-cluster-interface

/etc/netmasks
10.1.1.0 255.255.255.0 #connection to users VLAN
```

CheckPoint firewall 1 configuration

- CheckPoint must be configured to filter traffic on SBIF interfaces. StoneBeat Fullcluster creates SBIF interfaces on top of system interfaces and sub-interfaces. You can see below on StoneBeat configuration.

```
/etc/fw.boot/ifdev
#Define interfaces recognized by CheckPoint firewall1
sbif accept
```

…

- State synchronization has to enable on both nodes using the CheckPoint high availability module. The heartbeat network will also be used for state synchronization.

```
$FWDIR/sync.conf
10.1.5.1 #synchronization IP on node 1
10.1.5.2 #synchronization IP on node 2
```

- State synchronization traffic has to be allowed through each node of the cluster. As a consequence, port TCP 256 used for synchronization by CheckPoint has to be allowed between 10.1.5.0 networks on the 2 nodes.

Source	Destination	Service	Action	Comment
10.1.5.0/24	10.1.5.0/24	TCP-256	Accept	Firewall 1 state synchronization

StoneBeat configuration
StoneBeat provides a good interface to configure the nodes of the cluster.
The major steps are:

Node configuration: node ID, capacity benchmark for load balancing, high availability mode (balancing), license string.

Interface configuration

Node 1						
VLAN	System Interfaces	StoneBeat interfaces	Dedicated MAC	System sub-interfaces	StoneBeat interfaces	Cluster MAC
Users VLAN	qfe0	Sbif0	08.00.20.x.x.x	qfe0:1	Sbif0:1	08.08.08.08.08.01 (unicast)
DMZ server VLAN	qfe1	Sbif1	08.00.20.x.x.x	qfe1:1	Sbif1:1	08.08.08.08.08.02 (unicast)
Intranet access VLAN	qfe2	Sbif2	08.00.20.x.x.x	qfe2:1	Sbif2:1	08.08.08.08.08.03 (unicast)
Partner access VLAN	qfe3	Sbif3	08.00.20.x.x.x	qfe3:1	Sbif3:1	08.08.08.08.08.04 (unicast)
Heartbeat network	eri0	Sbif4	00.03.ba.x.x.x	eri0:1	Sbif4:1	01.02.03.04.05.06 (multicast)

Node 2						
VLAN	System Interfaces	StoneBeat interfaces	Dedicated MAC	System sub-interfaces	StoneBeat interfaces	Cluster MAC
Users VLAN	qfe0	Sbif0	08.00.20.y.y.y.y	qfe0:1	Sbif0:1	08.09.09.08.08.01 (unicast)
DMZ server VLAN	qfe1	Sbif1	08.00.20.y.y.y.y	qfe1:1	Sbif1:1	08.09.09.08.08.02 (unicast)
Intranet access VLAN	qfe2	Sbif2	08.00.20.y.y.y.y	qfe2:1	Sbif2:1	08.09.09.08.08.03 (unicast)
Partner access VLAN	qfe3	Sbif3	08.00.20.y.y.y.y	qfe3:1	Sbif3:1	08.09.09.08.08.04 (unicast)
Heartbeat network	eri0	Sbif4	00.03.ba.y.y.y.y	eri0:1	Sbif4:1	01.02.03.04.05.06 (multicast)

StoneBeat Fullcluster is using specific interfaces called SBIF interfaces. It performs the mapping between system interfaces / sub-interfaces (eriX/qfeX, eriX:1/qfeX:1) and StoneBeat interfaces (SBIF).
For example, interface system qfe0 interface is mapped to StoneBeat sbif0 interface.

Heartbeat interface
Eri0 connects to the heartbeat network that is used for StoneBeat Fullcluster state information. Cluster multicast MAC addresses should be the same on node 1 and node 2.

Operative interfaces
These interfaces connect to regular LAN (user LAN, DMZ…). StoneBeat Fullcluster performs the mapping between unicast cluster IP addresses and unicast MAC addresses.

StoneBeat files configuration
The checklist file
This file impacts the behavior of StoneBeat in case of node failure. As a consequence, this file is crucial for the proper functioning of the cluster. A node that encounters any types of failure has to be brought offline. If it stays online after failure, it can greatly disrupt the behavior of others nodes and open security holes within the cluster.
We intend to monitor 2 types of failures:

- Firewall failure: the node should go offline if the firewall is not running (process failure) or if no policy is loaded on the firewall.
- Network failure: the node should go offline is one of its interfaces crashes or if any port of the switch to which it connects is down

To ensure correct functioning of the cluster test system, 3 types of tests should be performed:

- Tests run in online mode: check if no failure has happened on the node. If a failure is noticed, the node should go offline.
- Tests run in offline mode: check if the failure that caused the node to go offline is solved.
- Recovery test: check if all offline tests are successful. If they are, this means that the node has no longer a problem and can be safely be brought back online.

Let's illustrate this on the following example:
The link between the interface qfe1 of node 1 and the switch goes down. StoneBeat Fullcluster runs its tests ("link_qfe1_up") every 30 seconds and notices the failure. As a consequence, it forces node 1 to go offline. No traffic can flow through it and connections first established through node 1 are balanced to node 2 (Firewall 1 state synchronization).
Tests are run in offline mode ("link_qfe1_down") every 30 seconds and confirm that the link is still down, which means that this node is not ready for recovery.

Now, the switch is working properly and the status of the link is up again. StoneBeat runs offline tests and notices that the failure has been fixed. As a consequence, the recovery test ("recovery_test") that followed is successful and node 1 is put back online.
Here is a sample of the checklist file:

```
# Tests run in online node
link_qfe1_up    30  online offline 2 1000 networkinterface-linkstatus sbif1
# tests run in offline node
link_qfe1_down    30  offline alert 2 1000 networkinterface-linkstatus sbif1
# recovery test
recovery_test 60 offline recoverstandby 1 1 all-offline-tests-ok 120
```

Filter.conf file
This file is used to configure nodes that will be part of the balancing mode.

Physical architecture
2 new switches Cisco Catalyst 3524 are used to interconnect equipments of this site (2 firewalls, intranet router, partner router, user LAN switch, DMZ server switch LAN). Several points drove this out:

- Switches need to have the capacity to forward unicast MAC addresses to several ports.
- Reconfiguration of local equipments should be limited.
- The architecture should be scalable and the next step of redundancy on this site should be integrated easily.

Site switches to high availability switches (HA switches) connection
The local user switch and the DMZ server switch are connected to the 2 HA switches. In case of failure of one of the HA switch, this ensures that traffic flows using the remaining available switch. The Spanning Tree Protocol (STP) ensures that no loop is present in the architecture.
Etherchannel links are used for the connection between the local user switch and HA switches and between the DMZ server switch and HA switches. Basically in this case, we aggregate 2 * 100 Mbits/s Ethernet ports to create one logical 200 Mbit/s ether-channel link between switches. This was driven by a performance criterion. As traffic is load balanced on the cluster, the total traffic that can flow through the cluster is theoretically double as in the single firewall configuration. As a consequence, bottleneck between switches on LANs should be avoided by the use of 200 Mbit/s Etherchannel link.

HA switches port configuration
4 VLANs are set up on each switch, one per local network (Intranet access, Partner access, DMZ server and Local user). This mainly ensures that broadcast traffic is well isolated between VLANs instead of flooding all ports of the switch.
A trunk link is set up between the 2 switches to carry all VLANs between them. This trunk uses a Gigabit port on each switch for performance reasons.

Static MAC entries have to be configured on the switches to direct unicast traffic to several ports.

HA switch 1		
Ports	VLAN	Connected devices
fa 0/1	VLAN 1 (orange)	Local user LAN
fa 0/2	VLAN 1 (orange)	Cluster node 1
Ga 0/1	Trunk	HA switch 2

HA switch 2		
Ports	VLAN	Connected devices
fa 0/1	VLAN 1 (orange)	Local user LAN switch
fa 0/2	VLAN 1 (orange)	Cluster node 2
Ga 0/1	Trunk	HA switch 1

Let's take a look at the configuration of HA switches based on this example: traffic originated from one local user destined to the cluster should reach both nodes of the cluster.
Frames destined to the cluster (unicast MAC address 08.08.08.08.08.01) come from port fa 0/1 on HA switch 1 (user VLAN). It should go to fa 0/2 on HA switch 1 (node 1 of the cluster) and Ga 0/1 (connection to HA switch 2 where the second node of the cluster is connected). Then on HA switch 2, traffic that comes from the trunk link on VLAN 1 (Local user LAN) should go to fa 0/2 (second node of the cluster). That's it: both nodes of the cluster receive the unicast frame.

The Cisco commands that enable this are simple to implement

```
# on HA switch 1:
mac-addresse-table static 0808.0808.0801 fa 0/1 fa 0/2 ga 0/1 vlan1

#on HA switch 2:
mac-addresse-table static 0808.0808.0801 ga 0/1 fa 0/2 vlan1
```

The same type of configuration has to be applied to other VLANs.

Production environment

Testing phase

An extensive test plan was performed on the architecture before putting the solution in production. As these tests were successful, the architecture was put in production and the same tests were performed under real conditions.
Here are the main bullets of the test plan:

Link failure

Initial state: initiate several connections (ftp, windows file sharing…) to one server in DMZ. Be sure that one of them passes through node 1 (for example sniff traffic on node 1).
Failure simulation: unplug the cable between node 1 and the HA switch 1.
Functionality tested: StoneBeat failure detection (network failure) + StoneBeat recovery test + CheckPoint Firewall 1 state synchronization
Results: OK, the node 1 goes offline, transfer stopped 15/20 seconds and then restarts through node2.

Firewall module failure
Initial state: initiate several connections (ftp, windows file sharing…) to one server in DMZ. Be sure that one of them passes through node 1 (for example sniff traffic on node 1).
Failure simulation: unload the policy on node 1; wait for the node to go offline and then reload the policy.
Functionality tested: StoneBeat failure detection (firewall application failure), StoneBeat recovery test + CheckPoint Firewall 1 state synchronization.
Results: OK, the node 1 goes offline, transfer stopped 15/20 seconds and then restarts through node2.

HA switch failure
Initial state: initiate several connections (ftp, windows file sharing…) to one server in DMZ. Be sure that one of them passes through node 1 (sniff traffic on node 1).
Failure simulation: unplug the power cable of HA switch 1.
Functionality tested: physical architecture redundancy + StoneBeat features + CheckPoint Firewall 1 state synchronization.
Results: OK with limitations. The time the Spanning Tree Protocol (STP) takes to converge is about 40-50 seconds and can cause connections to time out.

Complete node failure
Initial state: initiate several connections (ftp, windows file sharing…) to one server in DMZ. Be sure that one of them passes through node 1 (for example, sniff traffic on node1).
Failure simulation: shutdown node 1.
Functionality tested: StoneBeat features + CheckPoint Firewall 1 state synchronization.
Results: OK, transfer stops 15/20 seconds and then restarts through node2.

Performance test
Initial state: initiate several connections (windows file sharing…) to one server in DMZ. Be sure that one of them passes through node 1 (sniff traffic on node 1).
Failure simulation: Not applicable for this test.
Functionality tested: StoneBeat load balancing.
Results: OK, both node handles part of the traffic (sniff traffic on node 1 and node 2 to confirmation).

Benchmark compared to initial requirements
Based on the tests presented above, the new high availability architecture responds to initial requirements in the following ways:

- The level of security is guaranteed as previously.
- High availability is guaranteed in case of a broad range of failures. Established connections are preserved upon failures.
- Performance is increased as each node of the cluster handles a certain amount of traffic. Acceptable user throughput is conserved for a more important number of connections that flow through the cluster compared to the single firewall configuration.

- Intervention on local equipments was avoided and only basic re-cabling was necessary to shift to the new architecture.
- A still higher level of redundancy can be easily applied as explained in the conclusion.

Conclusion

The main limitation of the original architecture of this site is its firewall as it controls traffic between all enclaves. Though not critical when the original architecture was implemented, availability has become mandatory to implement. As a consequence, we deployed a new architecture based on a software-based product StoneBeat Fullcluster. Key advantages of the new architecture are high firewall availability, increased performance and minimum reconfiguration of equipments. The tests performed show that the deployed architecture meets all of these requirements.
However, some single points of failure remain and among them routers and lines that connect the site to its main office and partner. A backup of these equipments can be integrated in this design without important modifications (connection of redundant Cisco routers on the second high availability switch, set up of a protocol like Hot Standby Router Protocol (HSRP) between them…). This type of solution, though efficient, can become quickly expensive especially due to the recurrent cost of the second line and might not be applicable to our site.

References

YASSP-Yet Another Solaris Security Package. URL: www.yassp.org

SANS: The New Firewall Design Question
URL: http://rr.sans.org/firewall/new_design.php

HAC High-Availability.Com. “HAC technology brief – firewall load balancing”. URL: http://www.high-availability.com/Secure/PartnerInfo/FirewallLoadBalancingPrinciples.php

StoneSoft. “StoneBeat Fullcluster for Firewall 1”. Administrator’s guide. 12 December 2002
URL: ftp://download.stonesoft.com/web/Support/StoneBeat/PublicDocs/FullCluster/fw-1/3.0/manual/SBFCAG30Online.pdf

Stonesoft. “StoneBeat Fullcluster for Firewall 1”. Switch configuration examples for Cisco Catalyst 2900 and 3500 Series XL.
URL: ftp://download.stonesoft.com/web/Support/StoneBeat/Technical Notes/SGSB-TECNSwitches3.pdf

Securitywatch. “High availability firewall solutions”.
URL: http://www.securitywatch.com/EDU/ency/highavailability.html

Securitywatch. “CheckPoint firewall 1”.
URL: http://www.securitywatch.com/EDU/ency/checkpoint.html

Cisco. “White paper Fast Etherchannel.
http://www.cisco.com/en/US/tech/tk389/tk213/technologies_white_paper09186a0080092944.shtml

Last Updated: May 22nd, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event