Global Information Assurance Certification Paper

Checkpoint NG Cluster Install Server

BASED ON STONEBEAT FULLCLUSTER

Author: Philipp A Müller

Option 1, version 2.1
GCUX Practical Assignment

September 20, 2004

This page is intentionally left blank.

EXECUTIVE SUMMARY

Today where more and more Internet services are offered, such as e.g. DSL provisioning services or online banking to hundred thousands of users, the protection of the service infrastructure becomes very important for such large ISPs and enterprises. They therefore choose a layered security approach.

This paper is a technical step-by-step guide for the installation, hardening, maintenance and monitoring of a tier-1 firewall cluster.

From a Failure Mode and Effect Analysis (FMEA) the risk mitigation actions were defined resulting in the server specification. We decided to use as a firewall-software Checkpoint NG FW-1 (R54) based on the FullCluster (3.0) clustering software from Stonebeat. The software is installed on the Sun Fire 280R sparc-platform, running 64-bit Solaris 2.8 as an operating system.

To mitigate the risk of misconfiguration the setup is automated by the Jumpstart install server. This scales very well for large environments and further reduces the Mean Time To Repair (MTTR).

This project is presented to fulfill the requirements of Version 2.1 Option 1 of the GIAC Certified UNIX Security Administrator (GCUX) practical assignment. We hope that this work serves as a valuable and useful contribution to the security community.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 INTRODUCTION

For ISPs and large enterprises that provide services over the internet for hundred thousands of users, such as e.g. DSL provisioning service or online banking service, protection of their infrastructure from all kinds of attack is one of the highest priorities. Therefore they choose a layered security approach as shown in Figure 1.

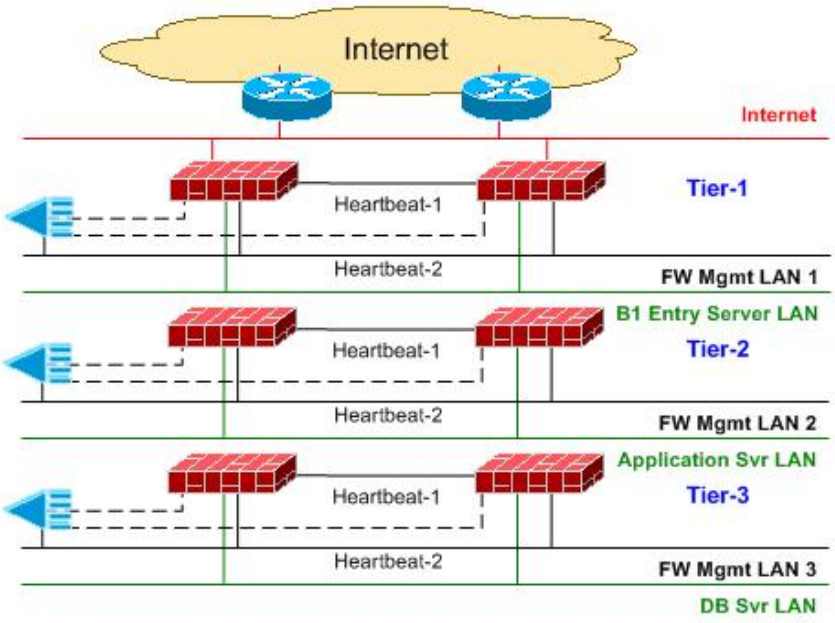

Figure 1: Online Banking Platform

The tier-1 firewall is mostly a packet filtering stateful-inspection firewall due to its superior throughput over proxy firewalls. For the 2nd tier proxy firewalls and for the 3rd tier either proxy or stateful-inspection firewalls are used. This step by step guide describes a possible solution for such a tier-1 firewall cluster, which serves multiple purposes [1]:

- It restricts people to entering at a carefully controlled point.
- It prevents attackers from getting close to your other defenses.
- It restricts people to leaving at a carefully controlled point.

The document is split into four parts. In the first part we outline the requirements, based on a risk analysis and mitigation plan. From this information we define the server requirements. The second part explains the automated setup settings based on a jumpstart install server. It also contains a step-by-step installation guide. The third part describes the maintenance design and implementation of the system and in the fourth part we test and verify the actions defined in the risk mitigation plan and implemented in part two and three.

2 SERVER SPECIFICATION AND RISK MITIGATION PLAN

2.1 Failure Mode and Effect Analysis

Before we are able to specify the server we need to know the requirements. These we get directly from the risk mitigation plan. To define such a plan we need to:

- Define the assets.
- Conduct a risk analysis.
- Define a risk mitigation plan.

There are several methods to achieve this. A simple, but successful process is the Failure Mode and Effect Analysis (FMEA) shown in Figure 2. FMEA has its origin in the aerospace industry in the mid-1960s, specifically looking at safety issues. You can get more details on the FMEA analysis from their website [2] or the FMEA Basics book [3].

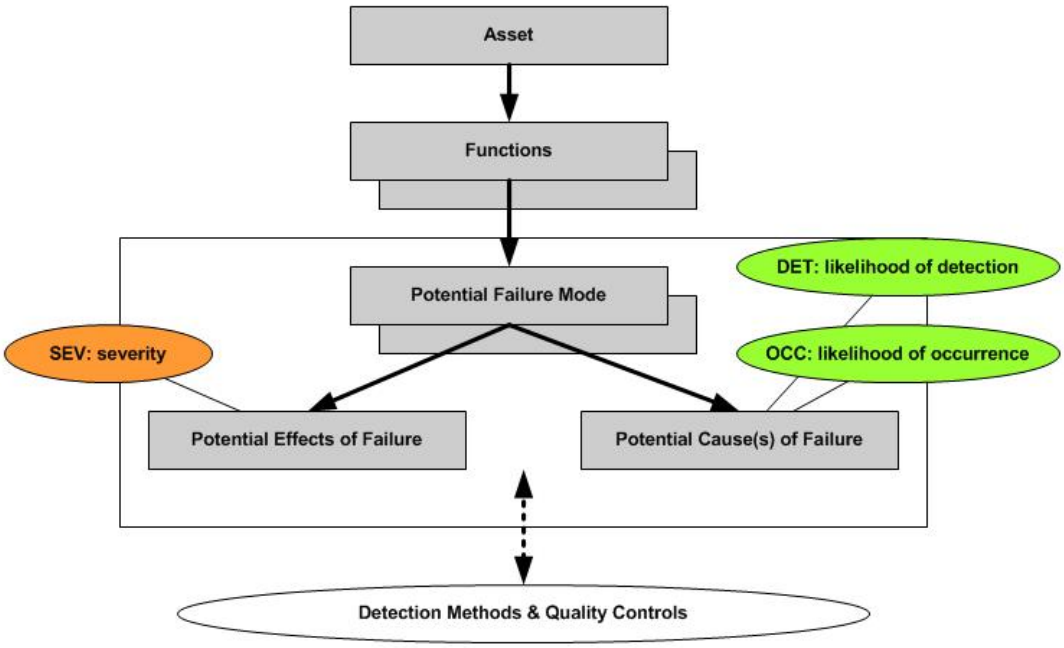

Figure 2: FMEA Overview

As a very brief summary of the FMEA Analysis we introduce the abbreviations.

- **SEV:** is the severity of the potential failure or incident event. It represents the cost. It is measured from 1 to 10. The severity table used for our analysis can be found in appendix A.1.
- **OCC:** is the probability that such a failure or incident happens. It is measured from 1 (never) to 10 (always). The occurrence probability table for our analysis can be found in appendix A.2.

- **DET:** is the probability that such a failure or incident is detected. It is measured from 1 (easy to detected) to 10 (very hard to detect). The detection table for our analysis can be found in appendix A.3.
- **RPN:** is the Risk Priority Number and is equal to the product of SEV x OCC x DET. It determines how severe the failure or incident is.

The rest of the FMEA should become clear by following our analysis.

If we apply the FMEA to our scenario of a tier-1 firewall we get as the asset the firewall service as defined already in section 1. We recap them here again:

- It restricts people to entering at a carefully controlled point.
- It prevents attackers from getting close to your other defenses.
- It restricts people to leaving at a carefully controlled point.

2.1.1 Risk Analysis

The basic functions of the firewall service can be split up into a simplified model shown in Figure 3.

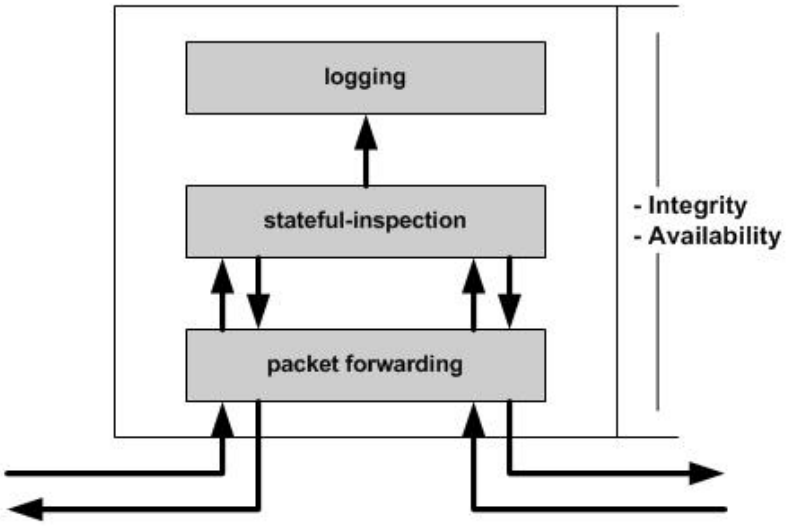

Figure 3: Simplified Firewall Model

It is important to understand that integrity and availability are critical requirements for the service, whereas confidentiality is not that much an issue.

If we use the three functions for the analysis we get Table 1. It is important to understand that the results in this table contain no risk mitigation actions.

2.1.2 Risk Mitigation Plan

From the results of the risk analysis Table 1 we can now define the risk mitigation plan shown in Table 2.

As a result we see that the following mitigation actions are required:

- **Automated installation:** Decision to use SUN jumpstart server [4]. See section 3.1.
- **Automated system monitoring:** Decision to monitor the system status 7x24x365 by the Network Management Centre via SNMP. Further monitor as much as possible. See section 4.3.
- **Change request process:** Decision to implement such a process within the company. This is briefly mentioned in section 4.1, but out of scope for this paper.
- **Disable unused ports:** Decision to turn off all unnecessary ports. See section 3.1.2.5.
- **Disk mirroring:** Decision to use RAID-1 disk mirroring to increase availability and performance. See section 3.1.2.6.
- **Enable as much logging as possible:** See section 3.1.2.10 and 4.3.
- **Harden OS:** Decision to harden kernel parameters and make system more resistant against attacks. Further uninstall all unused software packages. We use the Security Toolkit JASS from Sun [5] and in addition own hardening scripts. See section 3.1.2.5 and 3.1.2.14.
- **Harden network equipment:** Decision to harden router and switch configuration within the company. This is out of scope for this document.
- **Harden security policy:** Decision to harden the settings of the firewall policy. This is out of scope for this document, but information on how to do this can be found in [6].
- **High performance system:** Decision to use a high performance system with enough memory and CPU power. See section 2.2.2.
- **Integrity monitoring:** Decision to verify the system and file integrity with Enterasys Dragon host-based IDS [7]. See section 4.5.
- **Platform redundancy:** Decision to use a firewall cluster. As cluster software we will use Stonebeat FullCluster [8], due to its high performance, load balancing feature and good monitoring capabilities. See section 3.1.2.10.
- **Load balancing:** Decision to use a firewall cluster in load balancing mode.
- **Log file monitoring:** Decision to use LogCheck [9] as automated log monitoring tool. See section 4.4.
- **Log file rotation:** Decision to automatically rotate log files by cron job. See section 4.4.
- **Minimized network services:** Decision to turn off all unnecessary network services. See section 3.1.2.14.
- **Regular Software and OS patching:** Decision to keep track of software vulnerabilities and to upgrade or patch existing software if necessary. See section 4.2.

- **Regular system audits:** Decision to make regular system audits by external consultants and the QualysGuard scanner an automated penetration testing tool from Qualys [10]. See section 4.6.
- **Restrict directory access and file permissions:** See section 3.1.2.14.
- **Restrict physical access:** Decision to place firewall system in datacenters, which provides uninterruptible power, access control, fire-fighting system and locked racks.
- **Restrict user accounts and environment:** See section 3.1.2.14.
- **System redundancy:** Decision to use a system with duplicated power supplies and redundant architecture. See section 2.2.2.
- **Testing process:** Decision to make functional and security tests of installed systems. See section 5.
- **Time synchronization:** Decision to use Network Time Protocol for time synchronization. See section 3.1.2.5.
- **Fulfill legal requirements:** Decision to use warning banners to fulfill the legal requirements to enable prosecution of trespassers on the computer system. See section 3.1.2.14.
- **Use encrypted network access:** Decision to install OpenSSH and uninstall all other access possibilities such as telnet, ftp, etc. See section 3.1.2.7.
- **Use of pub-key authentication:** Decision to only allow public-key authentication onto the system with the only exception of the console port. There we allow the usage of passwords. See section 3.1.2.7.
- **Use strong passwords:** Decision to use passwords conforming to a password policy and change passwords and keys every 3 months.

Before we really proceed with implementing these mitigation decisions we have to check if the mitigation costs do not supercede the failure or incident costs. Therefore we recalculate the RPN, but this time we use the estimated failure/incident costs instead of a SEV value from 1 to 10. If this number is smaller than the mitigation costs for that scenario we will implement it. Since in our specific scenario the tier-1 firewall protects the core asset of company X all mitigation actions have to be implemented.

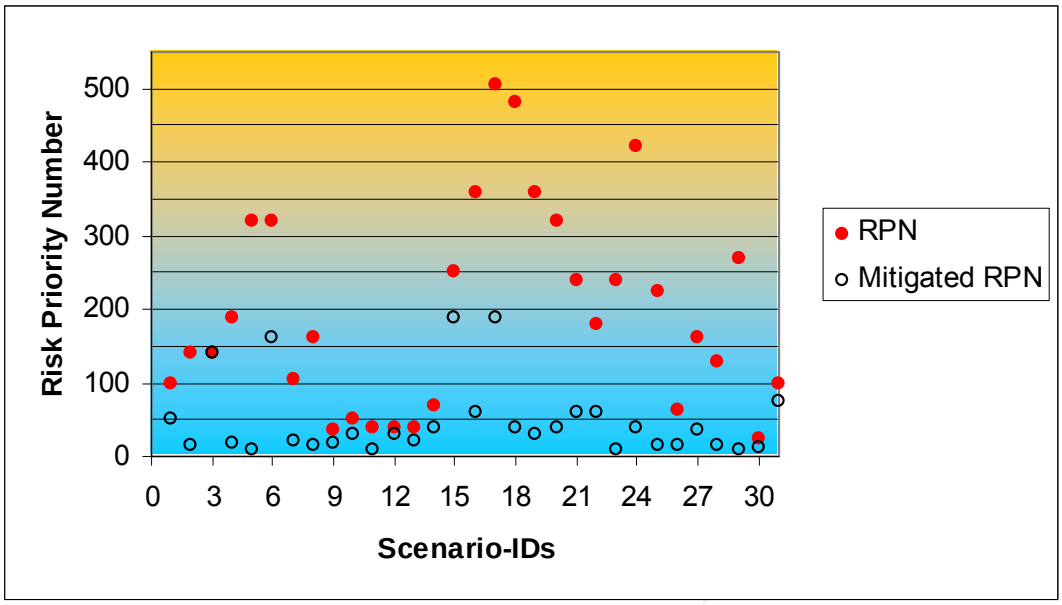

Figure 4: Risk Mitigation

In Figure 4 we can see how the mitigation actions affect the RPN values. It is also interesting to see that one major remaining threat is vulnerabilities due to software bugs. Therefore it is important to carefully read security bulletins as stated in section 4.2.

Function	Potential Failure Mode	ID	Potential Effects of Failure	SEV	Potential Cause(s) of Failure	OCC	Detection Method & Quality Control	DET	RPN
Packet forwarding.	Partially working. Loss of certain packets.	1	Moderate effect on service. The service requires repair.	5	Damaged cable.	5	Customer complaints or by chance.	4	**100**
		2	Moderate effect on service. The service requires repair.	5	Removed cable by unauthorized person.	7	Customer complaints or by chance.	4	**140**
		3	Moderate effect on service. The service requires repair.	5	Software bug memory leak, certain packets get lost.	4	By chance.	7	**140**
		4	Minor effect on service.	3	CPU overloaded.	7	By chance.	9	**189**
	Partially working. Wrong packet forwarding.	5	Small effect on service. Service does not require repair.	4	Spoofed IP packet.	8	By chance.	10	**320**
		6	Service is inoperable with loss of primary function. The system is inoperable.	8	Wrong routing information due to misconfiguration.	8	Customer complaints or by chance.	5	**320**
		7	Service performance is severely affected but functions. The system may not be operable.	7	Wrong routing information due to unpriv. user made configuration change.	3	Customer complaints or by chance.	5	**105**
		8	Small effect on service. Service does not require repair.	4	Vlan hopping attack.	4	By chance.	10	**160**
	Breakdown.	9	Failure involves hazardous outcomes and/or non-compliance with standards or regulations.	9	Power supply breakdown.	4	Customer complaints or by chance.	1	**36**
		10	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	Unpriviledged person removed power cable.	5	Customer complaints or by chance.	1	**50**
		11	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	NIC crashed.	4	Customer complaints or by chance.	1	**40**
		12	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	NIC driver crashed due to software bug.	4	Customer complaints or by chance.	1	**40**
		13	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	NIC driver crashed due to malformed packets.	4	Customer complaints or by chance.	1	**40**
		14	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	DOS attack.	7	Customer complaints or by chance.	1	**70**
Stateful-inspection.	Defective operation.	15	Failure involves hazardous outcomes and/or non-compliance with standards or regulations.	9	SW bug.	4	By chance.	7	**252**
		16	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	Weak security policy due to unpriviledged user manipulated configuration.	4	By chance.	9	**360**

Function	Potential Failure Mode	ID	Potential Effects of Failure	SEV	Potential Cause(s) of Failure	OCC	Detection Method & Quality Control	DET	RPN
		17	Failure involves hazardous outcomes and/or non-compliance with standards or regulations.	9	Weak security policy due to misconfiguration.	7	By chance.	8	**504**
		18	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	Unpriviledged system user changed configuration.	6	By chance.	8	**480**
		19	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	Manipulated by unpriviledged user who got access to the system by sniffing password.	4	By chance.	9	**360**
		20	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	Manipulated by unpriviledged user who cracked root password.	4	By chance.	8	**320**
		21	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	Manipulated by unpriviledged user who used a vulnerability in a running network service.	3	By chance.	8	**240**
		22	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	Manipulated by unpriviledged user who had physical access and connected a modem.	3	By chance.	6	**180**
	Breakdown.	23	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	Crash of firewall daemon.	4	By chance.	6	**240**
		24	Hazardous failure occurs without warning. It suspends operation of the system and/or involves non-compliance with regulations.	10	CPU overload.	6	Customer complaints or by chance.	7	**420**
Logging.	Defective log files.	25	Small effect on service. Service does not require repair.	4	Not time synchronized.	8	By incident case.	7	**224**
		26	Small effect on service. Service does not require repair.	4	Not admissible for legal actions.	8	By incident case.	2	**64**
		27	Minor effect on service.	3	Not enough information, because not all events were logged.	9	By incident case.	6	**162**
		28	Small effect on service. Service does not require repair.	4	Manipulated by unauthorized user.	4	By chance.	8	**128**
	Breakdown.	29	Moderate effect on service. The service requires repair.	5	No memory left for log file creation.	9	By chance.	6	**270**
		30	Service performance is degraded. Comfort or convenience functions may not operate.	6	HD crashed.	4	Customer complaints or by chance.	1	**24**
		31	Moderate effect on service. The service requires repair.	5	Software bug in daemon.	4	By chance.	5	**100**

Table 1: Risk Analysis

Function	Potential Failure Mode	ID	Potential Cause(s) of Failure	Mitigation Actions	**SEV**	OCC	**Mitig. OCC**	DET	**Mitig. DET**	RPN	**Mitig. RPN**	Detection Method & Quality Control
Packet forwarding.	Partially working. Loss of certain packets.	1	Damaged cable.	- Automated system monitoring - Platform redundancy	5	5	5	4	2	100	**50**	- System monitoring
		2	Removed cable by unauthorized person.	- Automated system monitoring - Platform redundancy - Restrict physical access	5	7	3	4	1	140	**15**	- System monitoring
		3	Software bug memory leak, certain packets get lost.	- Automated system monitoring - Regular patching	5	4	4	7	7	140	**140**	- System monitoring
		4	CPU overloaded.	- Automated system monitoring - Load balancing - High performance systems	3	7	3	9	2	189	**18**	- System monitoring
	Partially working. Wrong packet forwarding.	5	Spoofed IP packet.	- Harden security policy - Harden OS	4	8	1	10	2	320	**8**	- Log file monitoring
		6	Wrong routing information due to misconfiguration.	- Automated installation - Change request process - Testing process - Regular system audits	8	8	5	5	4	320	**160**	- Installation process - Testing process
		7	Wrong routing information due to unpriv. user made configuration change.	- Restrict user accounts & env. - Restrict file permissions - Integrity monitoring	7	3	3	5	1	105	**21**	- Integrity monitoring - Log file monitoring
		8	Vlan hopping attack.	- Restrict physical access - Harden network equipment	4	4	1	10	4	160	**16**	- Network IDS
	Breakdown.	9	Power supply breakdown.	- Automated system monitoring - Platform & system redundancy	9	4	2	1	1	36	**18**	- System monitoring
		10	Unpriviledged person removed power cable.	- Automated system monitoring - Platform & system redundancy - Restrict physical access	10	5	3	1	1	50	**30**	- System monitoring
		11	NIC crashed.	- Automated system monitoring - Platform & system redundancy - Restrict physical access	10	4	1	1	1	40	**10**	- System monitoring
		12	NIC driver crashed due to software bug.	- Automated system monitoring - Software patching	10	4	3	1	1	40	**30**	- System monitoring
		13	NIC driver crashed due to malformed packets.	- Automated system monitoring - Harden OS	10	4	2	1	1	40	**20**	- System monitoring
		14	DOS attack.	- Harden OS - High performance system - Load balancing	10	7	4	1	1	70	**40**	- System monitoring

Function	Potential Failure Mode	ID	Potential Cause(s) of Failure	Mitigation Actions	SEV	OCC	Mitig. OCC	DET	Mitig. DET	RPN	Mitig. RPN	Detection Method & Quality Control
Stateful-inspection.	Defective operation.	15	SW bug.	- Log file monitoring - Regular patching.	9	4	3	7	7	252	**189**	- Log file monitoring
		16	Weak security policy due to unpriviledged user manipulated configuration.	- Restrict user accounts & env. - Restrict file permissions - Integrity monitoring	10	4	3	9	2	360	**60**	- Integrity monitoring - Log file monitoring
		17	Weak security policy due to misconfiguration.	- Change request process - Regular system audits	9	7	3	8	7	504	**189**	- System audits
		18	Unpriviledged system user changed configuration.	- Restrict user accounts & env. - Restrict file permissions - Restrict directory access - Integrity monitoring - Log file monitoring	10	6	2	8	2	480	**40**	- Integrity monitoring - Log file monitoring
		19	Manipulated by unpriviledged user who got access to the system by sniffing password.	- Integrity monitoring - Restrict physical access - Use encrypted network access - Use of pub-key authentication	10	4	1	9	3	360	**30**	- Integrity monitoring - Log file monitoring
		20	Manipulated by unpriviledged user who cracked root password.	- Use of pub-key authentication - Use strong passwords - Log file monitoring	10	4	2	8	2	320	**40**	- Log file monitoring
		21	Manipulated by unpriviledged user who used a vulnerability in a running network service.	- Minimized network services - Hardened OS - Restrict user accounts & env. - Restrict file permissions - Restrict directory access - Integrity monitoring - Log file monitoring	10	3	2	8	3	240	**60**	- Integrity monitoring - Log file monitoring
		22	Manipulated by unpriviledged user who had physical access and connected a modem.	- Restrict physical access - Harden OS (disable serial port) - Restrict user accounts & env. - Use strong passwords	10	3	1	6	6	180	**60**	- Integrity monitoring - Log file monitoring
	Breakdown.	23	Crash of firewall daemon.	- Automated system monitoring - Software patching - Platform redundancy	10	4	1	6	1	240	**10**	- System monitoring
		24	CPU overload.	- Automated system monitoring - High performance systems - Load balancing	10	6	2	7	2	420	**40**	- System monitoring

Function	Potential Failure Mode	ID	Potential Cause(s) of Failure	Mitigation Actions	**SEV**	OCC	**Mitig. OCC**	DET	**Mitig. DET**	RPN	**Mitig. RPN**	Detection Method & Quality Control
Logging.	Defective log files.	25	Not time synchronized.	- Use NTP - Restrict user accounts & env. - Integrity monitoring - Log file monitoring	4	8	2	7	2	224	**16**	- Due to incident
		26	Not admissible for legal actions.	- Use banners - Integrity control	4	8	2	2	2	64	**16**	- Due to incident
		27	Not enough information, because not all events were logged.	- Enable as much logging as possible	3	9	2	6	6	162	**36**	- Due to incident
		28	Manipulated by unauthorized user.	- Restrict user accounts & env. - Restrict file & dir permissions - Integrity monitoring	4	4	2	8	2	128	**16**	- Integrity monitoring - Log file monitoring
	Breakdown.	29	No memory left for log file creation.	- Automated system monitoring - Log file monitoring - Log file rotation	5	9	2	6	1	270	**10**	- System monitoring
		30	HD crashed.	- Automated system monitoring - Disk mirroring - Platform & system redundancy	6	4	2	1	1	24	**12**	- System monitoring
		31	Software bug in daemon.	- Log file monitoring - Regular patching	5	4	3	5	5	100	**75**	- Log file monitoring

Table 2: Risk Mitigation Plan

2.2 Server Specification

2.2.1 Network Topology

From the requirements of section 2.1.2 we get the network topology shown in Figure 5. A tier-1 firewall cluster called c13 with two nodes, called c13f1 and c13f2, which have two redundant heartbeat connections. Both heartbeat links have to use a physical different layer-2 connection to guarantee high availability and security.

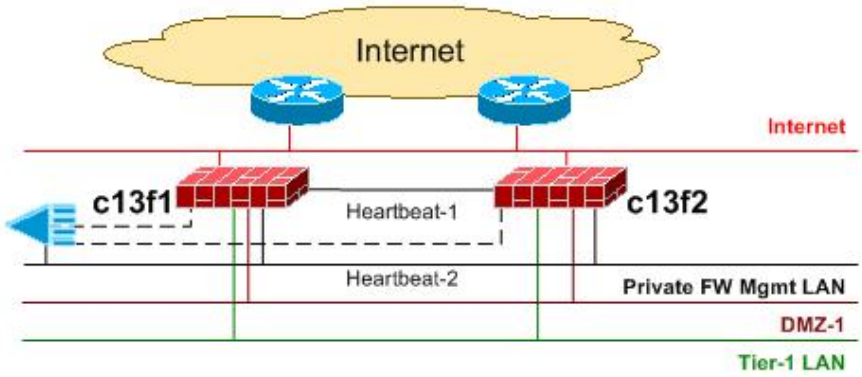

Figure 5: Network Topology

2.2.2 Hardware

To achieve the requirements defined in section 2.1.2 we use the Sun Fire 280R platform as a firewall node for this installation.

Details of one firewall cluster node – Sun Fire 280R:

- Two 1200-MHz UltraSPARC III Cu CPUs
- RAM 2 GB (4 × 512MB SDRAM DIMMs)
- Two Seagate ST336704FC (36.4GB - 10000 RPM FC-AL)
- Three Quad FastEthernet PCI (QFE/P)

2.2.3 Software

As packet filtering firewall software we decided to use Checkpoint FW-1 NG, because they are well established and showed in the past that their product protects from attacks. As already mentioned in section 2.1.2 we will use StoneBeat FullCluster as cluster software. For the OS we decided to use Solaris 8 in 64-bit mode, due to its good performance, hardening capabilities and the support of all Checkpoint features. Table 3 lists all the required software for our installation.

Software	Version		Download
Solaris OS	2.8	02/02	[4]
Checkpoint FW-1	NG	R54	[23]
Stonebeat Fullcluster	3.0	03-03	[8]
Solstice DiskSuite	4.2.1		[4]
OpenSSH	3.9	p1	[24]
Dragon HIDS	6.3.1		[28]
JASS - Solaris Security Toolkit	4.0.1		[5]
Logcheck/LogSentry	1.1.1		[9]
Stonebeat SNMP Agent	3.0.1		[8]
lsof	4.68		[25]
FixModes	1.42		[26]
MD5	1.4		[27]
CIS Benmark for Solaris	1.5.0		[22]

Table 3: Software Versions

2.2.4 Network Services

Since we only want to use minimal network services we have five applications requiring listening ports. These are:

- Checkpoint FW-1[1]: tcp/256, tcp/257, tcp/259, tcp/900, tcp/18183 – 18184, tcp/18187, tcp/18191, tcp/18192, tcp/18208, tcp 32775 - 32780
- Stonebeat FullCluster: tcp/3002
- OpenSSH: tcp/22
- Network Monitoring by snmp: udp/165
- Time-synchronization by NTP: udp/123

Dragon does not require a network service port on the node, since the node pushes the events directly to the IDS ring buffer station.

2.2.5 Processes

Besides the essential operating system daemons we need to run the following additional daemons:

- Checkpoint: fwd, cpd, cpwd, cprid, cphamcset, bootd, fwssd
- Stonebeat: sbfcd, sbfc-ela, sbfc-tester
- SSH: sshd
- Network Management Service: snmp
- NTP: xntpd
- Dragon-HIDS/dsquire: dragonctl, replicator, md5sum, cachemanager, dsquire, driderc

[1] Looks like a great design from Checkpoint having 15 ports open. There are possibilies to close some of them, but then they no longer support you.

2.2.6 User Management

The user management is rather simple and highly restrictive on a firewall node, because only firewall administrators needs to login on the system and they need root privileges. We create a single user account called ‘sans’. It satisfies our requirement since everyone is using the same account. This seems to sound strange, but it is securely setup and outline in see section 3.1.2.7. Besides the user accounts there are of course the system accounts of the OS and the described applications from above.

3 STEP-BY-STEP INSTALLATION

3.1 Jumpstart Server Configuration

3.1.1 Overview

We decided to use as a risk mitigation action an automated installation by Jumpstart server. This document does not outline how to setup such an install server. There exist good documents on the web such as Hal's Solaris Jumpstart resource [11] or Richard Braun's talk at the duke university [12].

In this section we explain how we designed our jumpstart installation process. The goal was that we could easily reuse some of the scripts for a complete other server or hardware platform. Therefore the finish scripts are split up into each software package or installation step. Figure 6 gives an overview of the different automated installation steps. In appendix B.1 the directory structure is listed of our Jumpstart server this helps you understanding the installation scripts which are explained in detail below.

3.1.2 FW-1 Cluster Installation Scripts

3.1.2.1 setup-client

The setup-client [B.6.1] is a user friendly GUI-interface for the installation. It allows you to enter hostname, MAC address, OS image file and Jumpstart profile. From this information it executes the Jumpstart add-install-client.

3.1.2.2 NG-fw-node-280R.profile

NG-fw-node-280R.profile [B.3] is the Jumpstart profile for our firewall cluster. It contains beside the list of Solaris packages to be installed, the disk partitioning.

Disk Partitioning

The two 36 G disks of our firewall node are partitioned as shown in Table 4. By isolating the /var partition we protect the root partition from overfilling. By isolating the /usr partition, we can mount it read-only, helping to protect system binaries from modification or potential remote exploit. We create a separate partition for /opt as this is where the FW-1 NG binaries are stored.

The reason for having three meta databases on disk1 and two on disk0 comes from the fact that if 50% of the databases are lost, the disk suite mirroring algorithm only allows us to boot into single user mode. In our situation where we have only two disks we have the problem that depending on which disk fails we loose at least 50% of the databases. Therefore the best solution is to make a 2/3 split as we did. In the case disk0 fails, disk1 can be rebooted without a problem, because it contains more than 50% of the databases. In the case disk1 fails the system still keeps on running, but can only be rebooted into single user mode. For more details see the Solstice DiskSuite 4.1 User's Guide.

Disk	Diskslices		Size [MB]
0	c1t0d0s0	/	8192
	c1t0d0s1	swap	2048
	c1t0d0s2	backup	
	c1t0d0s3	/var	16384
	c1t0d0s4	metadb1&2	10
	c1t0d0s5	/opt	rest
	c1t0d0s6	-	
	c1t0d0s7	-	
1	c1t1d0s0	/	8192
	c1t1d0s1	swap	2048
	c1t1d0s2	backup	
	c1t1d0s3	/var	16384
	c1t1d0s4	metadb1&2	10
	c1t1d0s5	/opt	rest
	c1t1d0s6	metadb3	6
	c1t1d0s7	-	

Table 4: Disk partitions

OS Software Packages

Beside the 64-bit Solaris 8 core installation our installation requires a few additional packages.

- To support CheckPoint FW-1 NG:
  SUNWter, SUNWlibC, SUNWlibCx, SUNWadmc, SUNWadmfw
- To support the Network Time Protocol: SUNWntpr, SUNWntpu
- To support Compression: SUNWgzip
- To support bash shell: SUNWbash
- To support snoop: SUNWfns, SUNWfnsx
- To support Secure Shell: SUNWzlib, SUNWzlibx
- To support system accounting: SUNWaccr, SUNWaccu
- To support disk mirroring: SUNWmdr, SUNWmdu, SUNWmdx

3.1.2.3 NG-fw-node-280R.driver

NG-fw-node-280R.driver [B.4.1] contains the list of finish scripts which the jumpstart server will execute.

3.1.2.4 NG-delete-unneeded-FCarch.fin

The core OS installation has still too many packages. NG-delete-unneeded-FCarch.fin [B.5.2] removes the following packages:

SUNWadmr, SUNWmdi, SUNWmdix, SUNWnamow, SUNWluxdx, SUNWluxop, SUNWluxox, SUNWpcelx, SUNWpcmci, SUNWpcmcu, SUNWpcmcx, SUNWpcmem, SUNWpcser, SUNWpsdpr, SUNWnisr, SUNWnisu, SUNWcg6, SUNWcg6x, SUNWdfb, SUNWauda, SUNWaudd, SUNWauddx, SUNWm64, SUNWm64x, SUNWrmodu, SUNWsndmr, SUNWsndmu, SUNWtleux, SUNWwsr2,

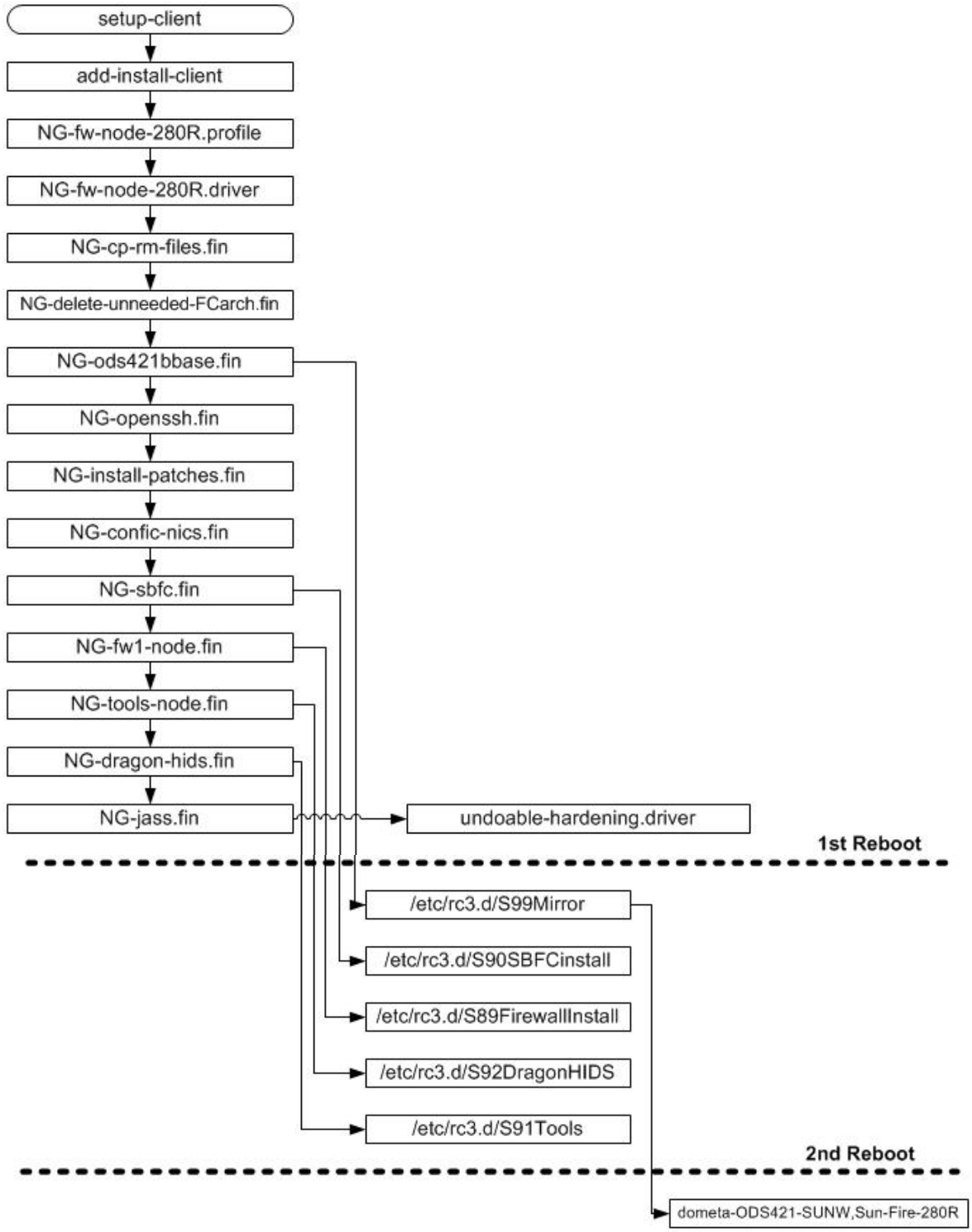

Figure 6: Installation Flow

SUNWuaud, SUNWuaudx, SUNWudf, SUNWudfr, SUNWudfrx, SUNWusb, SUNWusbx, SUNWatfsr, SUNWatfsu, SUNWpl5u, SUNWsolnm, SUNWxwdv, SUNWxwdvx, SUNWxwmod, SUNWxwmox, SUNWftpr, SUNWftpu, SUNWi15cs, SUNWi1cs, SUNWkey, SUNWdtcor, SUNWged.

In B.2 you find the list of all installed packages on the firewall node (including the ones we will install further down). This list of required packages is based

on information from Lance Armoring Solaris paper [13], from Checkpoint [14] and own research.

3.1.2.5 NG-cp-rm-files.fin

NG-cp-rm-files.fin [B.5.1] copies, removes and changes different files for hardening and installation purposes to the firewall nodes. The major hardening tool for the Solaris OS is JASS as defined in the risk mitigation actions. Since JASS leaves out a couple of hardening tweaks, we do them in this script. Like this we can easily upgrade JASS and do not have to modify the scripts. The files copied and modified are the following:

Serial port

To disable the serial port, which is unused, we copy a modified version of /etc/inittab [C.7.3] in which the line 'sc:234:respawn …' is removed. Like this an unauthorized person with physical access can attach a modem to the port and misuse it. We further disabled the serial 'login:' prompt by executing 'pmadm -d -p zsmon -s ttya' and 'pmadm -d -p zsmon -s ttyb'.

/etc/system.add-on

We copy and then append /etc/system.add-on to /etc/system to add several important parameters, such as:

set nfssrv:nfs_portmon=1 to configuring the NFS server service to accept connections/requests originating from privileged ports only – even so the NFS server will be disabled.

set noexec_user_stack=1 to enable kernel level stack protection.

set noexec_user_stack_log=1 to enable kernel level stack logging.

set tcp:tcp_conn_hash_size=16384 to increase the TCP hash table size. Since the firewall has to handle several thousand connections this improves performance. Default value is 512.

set rlim_fd_max=16384 this is the process open file descriptors limit. Default value is 1024. This improves performance.

We also set the interfaces to 100Mbit, full-duplex, no auto-negotiation to prevent problems:

```
set hme:hme_adv_autoneg_cap=0
set hme:hme_adv_100fdx_cap=1
set hme:hme_adv_100T4_cap=0
set hme:hme_adv_100hdx_cap=0
set hme:hme_adv_10fdx_cap=0
set hme:hme_adv_10hdx_cap=0
set qfe:qfe_adv_autoneg_cap=0
set qfe:qfe_adv_100T4_cap=0
set qfe:qfe_adv_100fdx_cap=1
set qfe:qfe_adv_100hdx_cap=0
set qfe:qfe_adv_10fdx_cap=0
set qfe:qfe_adv_10hdx_cap=0
set eri:adv_autoneg_cap=0
```

```
set eri:adv_100T4_cap=0
set eri:adv_100fdx_cap=1
set eri:adv_100hdx_cap=0
set eri:adv_10fdx_cap=0
set eri:adv_10hdx_cap=0
```

/etc/default/inetinit

To force the system to use a better randomized initial TCP sequence [15] we modified 'TCP_STRONG_ISS=2'.

/etc/profile and /etc/.login

To restrict files and directories created by the users from other users we modified the UMASK to a more restricted value of 077. To further improve the permission on the user's tty device we disable talk/write by setting 'mesg n' in the files.

/etc/default/login

To restrict files and directories created by the users from other users we modified the UMASK to a more restricted value of 077.

/etc/default/ftpd

To fulfill legal requirements and enable prosecution of trespassers on the computer system we put a warning banner. BANNER="Authorized use only. Activities are monitored and reported." Additionally we restrict the files and directories created by the users from other users by setting the UMASK to a more restricted value of 077.

/etc/ftpusers

We added all the system users to /etc/ftpusers. These users are not allowed to access the system via FTP.

/etc/default/telnetd

To fulfill legal requirements and enable prosecution of trespassers on the computer system we put a warning banner. BANNER="Authorized use only. Activities are monitored and reported."

eeprom oem-banner

To fulfill legal requirements and enable prosecution of trespassers on the computer system we put a warning banner. eeprom oem-banner="Authorized use only. Activities are monitored and reported."

/etc/default/power

We restrict normal user to have access to the power management functions and set 'PMCHANGEPERM=-' and 'CPRCHANGEPERM=-'.

/etc/default/sys-suspend

We restrict that only super-user can execute the sys-suspend command by setting 'PERMS=-'.

/etc/inetd.conf

To disable all network services we remove the /etc/inet/inetd.conf.

/etc/init.d/newinetsvc

We disable the start of the inetd daemon, by removing the /etc/rc2.d/S72inetsvc and creating a link to /etc/init.d/newinetsvc in which we removed the start of /usr/sbin/inetd.

/etc/syslog.conf

Copies /etc/syslog.conf [C.7.3] to the node, which has a special entry to log the ssh authentication 'local1.info /var/adm/sshd.log'

/etc/dfs/dfstab

Since we do not use NFS we remove its related config files.

/etc/rc2.d/S93cacheos.finish and /etc/rc2.d/S73cachefs.daemon

To disable unused services we move /etc/rc2.d/S93cacheos.finish and /etc/rc2.d/S73cachefs.daemon into /etc/rc2.d/_S93cacheos.finish and /etc/rc2.d/_S73cachefs.daemon.

/var/core

We create the directory /var/core and set the correct owner and file permission since JASS is doing this not appropriate. We still use the JASS enable-coreadm.fin script for the coreadm configuration – see below 3.1.2.14.

/var/spool/cron/crontabs

We set the permissions of /var/spool/cron/crontabs files more restrictive since JASS is leaving this out.

NTP configuration

We copy /inet/ntp.conf [C.9], ntp.drift (empty file) and ntp.keys [C.9.2] – the ntp configuration, drift and key file. To assure time synchronization as requested in 2.1.2. We configured the nodes to synchronize with two time-sources. We do not permit the time-source to query or modify the service on the nodes 'restrict <IP-@ time-source-1> noquery nomodify notrap'. We enable authentication based on the keys in /etc/inet/ntp.keys.node.

/var/adm/loginlog

To enable logging of failed login attempts we need to create /var/adm/loginlog. Without the file in place no such logging would be done. We add furthermore 'SYSLOG_FAILED_LOGINS=0' to log right after the first failed login.

/var/adm/sulog

To enable logging of 'su' attempts we create /var/adm/sulog.

/etc/shells

In /etc/shells we defined all the valid shells on the system. This file is called by 'getusershell' and therefore can restrict access to the system.

/.ssh/authorized_keys [C.8.3] and /home/sans/.ssh/authorized_keys [C.8.4]

We copy the public ssh-keys for root and the user 'sans' to enable ssh pub-key authentication.

/opt/fullcluster/etc/stonebeat_ela.conf

We copy /opt/fullcluster/etc/stonebeat_ela.conf to enable Stonebeat FullCluster to send its events to the Checkpoint log server via the ELA API.

/opt/fullcluster/etc/metatest.sh

We copy the external Stonebeat test script /opt/fullcluster/etc/metatest.sh, with which we control the health of the disk mirroring.

/var/tmp/ODS/dometa-ODS421-SUNW,Sun-Fire-280R

We copy /var/tmp/ODS/dometa-ODS421-SUNW,Sun-Fire-280R the disk mirroring config file. More details under 3.1.2.6.

3.1.2.6 NG-ods421base.fin

NG-ods421base.fin [B.5.3] installs the Solstice DiskSuite packages and patches. It further copies S99Mirror [C.1] in /etc/rc3.d, which gets executed only once after the 1st reboot and runs the 'dometa-ODS421-SUNW,Sun-Fire-280R' [C.1.1] script, which installs the mirrors. You may wonder why the sub-mirrors are not directly attached in the dometa script. We did that initially but experienced several times SCSI errors, which resulted in corrupt sub-mirrors. Therefore we decided to attach them manually one after each other – as explained in 3.2.3.2.

Mirror	Submirror	Disk
d0		
	d10	c1t0d0s0
	d20	c1t1d0s0
d1		
	d11	c1t0d0s1
	d21	c1t1d0s1
d3		
	d13	c1t0d0s3
	d23	c1t1d0s3
d5		
	d15	c1t0d0s5
	d25	c1t1d0s5

Table 5: Disk Mirroring

To provide high availability and high throughput within the system we use RAID level-1, as required in 2.1.2. Table 5 shows the disk mirroring layout.

3.1.2.7 NG-openssh.fin

NG-openssh.fin [B.5.4] installs the OpenSSH package[2], which contains the configuration files. The finish script further adds privilege separation and adds the 'sans' user. The idea of our setup is to highly restrict network access and only allow authorized people to connect on the firewall node by a single user account called 'sans'. They first have to login to a secured jump-server, which is accessible from everywhere within the company and which is the only entry point into the FW Management LAN. Authentication is only possible with ssh keys. To get root privileges it is then only possible to 'ssh root@localhost'. To limit root access from localhost we use the capability 'from=127.0.0.1' in the root authorized_key file. To still be able to identify the individual persons we increase the LogLevel to verbose in the sshd_config file. Now we get the following log entries in the log file specified in syslog.conf:

```
Sep 19 21:38:11 c13f2 sshd[991]: [ID 800047 local1.info] Connection from 127.0.0.1 port
   32804
Sep 19 21:38:11 c3f2 sshd[991]: [ID 800047 local1.info] Found matching RSA key:
    xx:xx:xx:xx:xx:xx
```

From this entry and the knowledge of who has which public ssh-key we can uniquely identify the person who logged in. This setup has the major advantage that only a small group of people need to know the super-user password and the others without the knowledge may still work as root. If somebody is leaving the team one just has to remove his ssh key. Table 7 shows the changes on the ssh client config [C.8.2] and Table 6 shows the changes on the ssh server config [C.8.1].

Old config value	New config value	Explanation
#Protocol 2,1	Protocol 2	Only allow version 2.
#ListenAddress ::	ListenAddress localhost	For root login.
#ListenAddress ::	ListenAddress <Node-Ctrl-IP>	Listen only on the FW Mgmt LAN interfaces.
#SyslogFacility AUTH	SyslogFacility LOCAL1	Log into ssh.log file.
#LogLevel INFO	LogLevel VERBOSE	We want to see the ssh keys in the logs.
#LoginGraceTime 2m	LoginGraceTime 1m	Prevent DOS attacks.
#PermitRootLogin yes	PermitRootLogin yes	We allow root to login only from localhost.
#Password-Authentication yes	PasswordAuthentication no	We only allow pubkey authentication.
#UsePAM yes	UsePAM no	No use of PAM.
#UseDNS yes	UseDNS no	No use of DNS.

Table 6: sshd_config

[2] This is an own built OpenSSH package for Solaris called Myopenssh.pkg. The package built process for Solaris is out of scope for this paper. Please refer to [29].

Old config value	New config value	Explanation
#ForwardAgent no	ForwardAgent yes	Enable key forwarding to easen operation.
# Protocol 2,1	Protocol 2	Only allow version 2.
# Cipher 3des-cbc, …	Ciphers aes256-cbc	Use AES-256 to encrypt.

Table 7: ssh_config

The super-user is allowed to directly login from the console. We allow this, because if someone has physical access to the firewall node which stands in an access restricted computer room, as stated in 2.1.2, he could damage the system by other means. A brute-force attack on the console is rather unpleasant and highly time consuming (do not forget is badly cold in computer rooms).

3.1.2.8 NG-install-patches.fin

NG-install-patches.fin [B.5.5] is the script, which installs the Solaris OS patches located in /export/install/jumpstart/files/Patches. Make sure that you downloaded and MD5 checked the newest Solaris patch cluster to that directory. You maybe have to add further required Solaris OS patches for the additional 3rd party software. You can get the information on which OS patches are required from the release notes. In our setup:

- Checkpoint requires:
  108528-17, 113652-01, 109147-18, 109326-07, 108435-01
- Stonebeat requires:
  108806-17, 108528-29, 117000-05, 117350-04

3.1.2.9 NG-config-nics.fin

NG-config-nics.fin [B.5.6] configures the firewall node interfaces, creates a /etc/hosts and /etc/netmask file.

3.1.2.10 NG-sbfc.fin

NG-sbfc.fin [B.5.7] copies the 'S90SBFCInstall' [C.3] script into /etc/rc3.d, which is executed only once after the first reboot. This script installs the Stonebeat FullCluster package and its configuration files, which are alert.sh [C.3.6], online.sh [C.3.9], offline.sh [C.3.8] and checklist [C.3.7] into the $SBFCHOME/etc directory on the firewall node. The sbfc-tester daemon monitors the system based on the settings in the checklist file. In our setup we control:

- If the operational and the heartbeat interfaces are up and running.
- If the load average is within our pre-defined limits.
- If the CPU is not overloaded.
- If there is enough room for the firewall logs.
- If there is enough room for the system logs.
- If the Checkpoint firewall daemon is running.

- If a firewall policy is installed on the node.
- If the disk mirroring is working fine. (Done by external test metatest.sh [C.3.4])

These parameters are constantly watched and a state change triggers an event. These events are processed and depending on the status of the node itself, the number of active nodes in the cluster, the node can be taken offline or online depending on its previous state and the severity of the event. This mapping is configured in alert.sh [C.3.6]. We also send a trap to inform the network management centre.

The S90SBFCInstall script adds installs the Stonebeat SNMP daemon and copies the snmpd.conf [C.3.10] file into /opt/stonebeat/snmp/etc. Over this daemon the network management centre controls the state of the firewall node. The daemon unfortunately only supports SNMP version 2, if required we could tunnel it over ssh, but in our case the network management centre has only read access and no read/write.

3.1.2.11 NG-fw1-node.fin

NG-fw1-node.fin [B.5.8] copies the S89FirewallInstall [C.4.1] script into /etc/rc3.d, which is executed only once after the first reboot. This script installs the Checkpoint software packages.

3.1.2.12 NG-tools.fin

NG-tools.fin [B.5.9] copies the S91ToolsInstall [C.6.1] script into /etc/rc3.d, which is executed only once after the first reboot. This script installs the lsof package and the LogCheck files. The detailed LogCheck configuration is explained in section 4.4.

3.1.2.13 NG-dragon-hids.fin

NG-dragon-hids.fin [B.5.10] copies the S92DragonHIDSInstall [C.5.1] script into /etc/rc3.d, which is executed only once after the first reboot. This script installs the Dragon hostbased IDS software packages and the config file.

3.1.2.14 NG-jass.fin

NG-jass.fin [B.5.11] installs the JASS security toolkit from Sun [5]. It is as defined in the risk mitigation actions the major hardening tool we use for our system. It has the big advantage that it can be easily re-run after having installed new patches or software on the system. The finish script executes the 'undoable-hardening.driver' [C.2.1], which calls the JASS driver.init which itself calls ther driver.run script. The driver.run script copies the files defined by the JASS_FILES variable to our system. These are:

/etc/dt/config/Xaccess

X configuration which denies all remote access to this server.

/etc/issue

To fulfill legal requirements and enable prosecution of trespassers on the computer system we put a warning banner "Authorized use only. Activities

are monitored and reported." in the /etc/issue. Its content is displayed prior to the login prompt on the system's console and serial devices.

/etc/motd

To fulfill legal requirements and enable prosecution of trespassers on the computer system we put a warning banner "Authorized use only. Activities are monitored and reported." in the /etc/motd. Its content is generally displayed after all successful logins, no matter where the user is logging in from.

/etc/rc2.d/S00set-tmp-permissions and /etc/rc2.d/S07set-tmp-permissions

S00set-tmp-permissions and S07set-tmp-permissions are links to /etc/init.d/set-tmp-permission, which sets the correct permissions on the /tmp and /var/tmp directories when the system is rebooted. If an inconsistency is found, it will be displayed to standard output and logged via SYSLOG. The reason why this script is executed twice (S00 and S07) is to permit the check to be performed both before and after the "mountall" command is run (from S01MOUNTFSYS). That way, both the mount point and the mounted filesystem will be sure to have the correct permissions and ownership.

/etc/rc2.d/S70nddconfig

The /etc/rc2.d/S70nddconfig is a link to /etc/init.d/nddconfig [C.2.2] which sets network driver parameters to prevent some network attacks. The script and its settings were developed and documented very nicely by A. Noordergraaf and K. Watson [16].

arp_cleanup_interval=60000: With this option we can modify the ARP cache refresh timer. ARP flooding attacks may be effective with the default interval, which is 5 minutes. Therefore we shorten the timeout interval to reduce the effectiveness of such an attack.

ip_forward_directed_broadcasts=0: With this option we define if we forward broadcast packets directed to a directly-connected net or subnet. The system can like this be exploited to generate a great deal of broadcast network traffic. We therefore turn it off. Default value is 1 (true).

ip_forward_src_routed=0: With this option we allow to forward packets source routed packets. Source routing is an IPv4 option, which only causes problems. Packet should be forwarded as defined by the routing protocol. Therefore we turn it off. Default value is 1 (true).

ip_ignore_redirect=1: With this option we allow to forward ICMP redirect packets. An attacker may send redirect messages to alter routing tables as part of a man in the middle attack or a simple denial of service. Default value is 0 (false).

ip_respond_to_address_mask_broadcast=0: With this option we define if we respond to ICMP netmask requests, which are typically sent by diskless clients when booting. An attacker may use the netmask information for determining network topology or the broadcast address for the subnet. Default value is 0 (false).

ip_respond_to_echo_broadcast=0: With this option we define if we respond to an ICMP broadcast echo requests packet (ping). An attacker may use this to fingerprint us or for a DOS attack. Default value is 1 (true).

ip_respond_to_timestamp=0: With this option we define if we answer ICMP timestamp requests. An attacker may use it to fingerprint us and for other nasty things. Default value is 1 (true).

ip_respond_to_timestamp_broadcast=0: With this option we define if we answer ICMP broadcast timestamp requests, which are used to discover the time on all systems in the broadcast range. It is obvious that this has an even worse DOS effect than the command from before. Default value is 1 (true).

ip_send_redirects=0: With this option we allow ICMP redirect messages, which can alter the remote's system routing table. Its only use is for Mobile-IP, but most router block it anyway. Since our firewall node marks the security perimeter we do not allow them. Default value is 1 (true).

ip_strict_dst_multihoming=1: This option determines whether to enable strict destination multi-homing. If this option is set to 1 and ip_forwarding is set to 0, then a packet sent to an interface from, which it did not arrive will be dropped. This setting prevents an attacker from passing packets across a machine with multiple interfaces that is not acting as a router. Default value is 0 (false).

ip_def_ttl=255: With this option we could change the default IP packet TTL value. To change it could sometimes prevent from fingerprinting tools. We leave the default value of 255.

tcp_conn_req_max_q0=4096: With this option we could change the queue size of unestablished connections. By increasing it we can improve the protection against SYN flood attacks. Even so the default queue size is adequate for most systems we increase it from its default value of 1024.

tcp_conn_req_max_q=1024: With this option we set the number of maximum fully established TCP connections. Increasing the size of this queue provides some limited protection against resource consumption attacks. Even so the default value is adequate for most systems we increase it from its default value of 128.

tcp_smallest_anon_port=32768, tcp_largest_anon_port=65535, udp_smallest_anon_port=32768, udp_largest_anon_port=65535: With these options we define the upper and lower bounds on ephemeral ports. Ephemeral (means short-lived) ports are used when establishing outbound network connections. Defaults values are tcp_smallest_anon_port=32768, tcp_largest_anon_port=65535, udp_smallest_anon_port=32768, udp_largest_anon_port=65535

tcp_smallest_nonpriv_port=1024, udp_smallest_nonpriv_port=1024: With these options we define the start of non-privileged TCP and UDP ports. Normally the non-privileged port range starts at 1024. Any program that attempts to bind a non-privileged port does not have to run as root. Defaults values are:

tcp_smallest_nonpriv_port=1024,
udp_smallest_nonpriv_port=1024

ip_ire_arp_interval=60000: With this option we define the period of time at which a specific route will be kept in the routing table. If we reduce the time we are more secure against poisoned roués. Default interval is 1200000 milliseconds (20 minutes).

tcp_extra_priv_ports_add="6112": With this option we define additional TCP and UDP privileged ports outside of the 1-1023 range. Any program that attempts to bind the ports listed here must run as root. This prevents normal users from starting server processes on specific ports. Multiple ports can be specified by quoting and separating them with spaces. We add tcp/6112 the port for the CDE Subprocess Control Server daemon (dtspcd). Defaults values are: tcp_extra_priv_ports: 2049 (nfsd) 4045 (lockd), udp_extra_priv_ports = '2049 4045 '

tcp_rev_src_routes=0: With this option we define if the specified route in a source routed packet will be used in returned packets. TCP source routed packets may be used for spoofing attacks. Default value is 0 (false)

And the same for IPv6: ip6_forward_src_routed = '0', ip6_respond_to_echo_multicast = '0', ip6_send_redirects = '0', ip6_ignore_redirect = '1', ip6_strict_dst_multihoming = '1'.

After driver.run scripts has copied the JASS_FILES it executes the JASS_SCRIPTS, which are summarized in Table 8.

A detailed documentation can be found in the JASS manuals [4].

JASS finish script	Description
disable-keyboard-abort.fin	Disabling system suspend via STOP-A.
disable-keyserv-uid-nobody.fin	Disabling 'nobody' access to SecureRPC information.
disable-ldap-client.fin	Disabling LDAP client by renaming K41ldap.client in rcS, 0, 1.d and S71ldap.client in /etc/rc2.d
disable-lp.fin	Disabling the line printer (LP) service and removes the user's access to the cron service. This is done independent of the update-cron-deny.fin script in the event that the lp service is required.
disable-nfs-client.fin	Disabling the NFS client service by renaming K41nfs.client and K75nfs.client in /etc/rc0.d and K80nfs.client in rc1.d and S73nfs.client in rc2.d
disable-nfs-server.fin	Disabling the NFS server service.
disable-nscd-caching.fin	Disabling the caching of entries by the Name Service Cache Daemon (NSCD) for passwd, group, hosts and ipnodes entries by setting the ttl value to zero in /etc/nscd.conf configuration file. Like this we reduce the chance of spoofing attacks due to the reduction of the name service cache.
disable-preserve.fin	Disabling the file PRESERVE functionality for lost vi buffers by renaming S80PRESERV and S89PRESERV in /etc/rc2.d.
disable-power-mgmt.fin	Disabling the Power Management service by renaming the K37power in /etc/rcS, 0, 1.d, K85power in /etc/rc1.d and S85power in/etc/rc2.d. The script further creates a /noautoshutdown file to prevent the system from prompting for the power management status at the first installation reboot.
disable-rhosts.fin	Disabling the ability to use 'rhosts' authentication by commenting out in /etc/pam.conf.

JASS finish script	Description
disable-rpc.fin	Disabling the Remote Procedure Call (RPC) service.
disable-sendmail.fin	Disabling the sendmail service (for mail receipt) by renaming the K36sendmail in /etc/rcS, 0, 1.d, K57sendmail in rc0 and 1.d, S88sendmail in rc2.d. It further adds an entry to root's crontab file to purge the outgoing mail queue hourly. Outbound mail only will be processed.
disable-syslogd-listen.fin	Preventing the SYSLOG service from logging remote connections. The service will no longer accept log messages from other systems by setting the LOG_FROM_REMOTE variable to NO in /etc/default/syslogd.
disable-system-accounts.fin	Disabling system accounts by setting the shell of daemon, bin, adm, lp, uucp, nuucp, nobody, listen, noacces, nobody4 to /sbin/noshell script which denies access to the account and logs access attempts to SYSLOG.
disable-uucp.fin	Disabling the Unix-to-Unix Copy by removing nuucp account by renaming S70uucp in rc2.d. It further removes the nuucp system account and the uucp crontab entries in the /var/spool/cron/crontabs directory.
disable-picld.fin	Disabling the Platform Information and Control Library (PICL) service.
enable-coreadm.fin	Configuring coreadm to use pattern matching, because the contents the file could include sensitive or confidential data.
install-at-allow.fin	Updating 'at' facility access controls (at.allow).
install-newaliases.fin	Creating the 'newaliases' link to the 'sendmail' program. This link is necessary for minimized installations for proper local mail handling.
remove-unneeded-accounts.fin	Removing non-essential system accounts, such as listen and nobody4.
set-login-retries.fin	Setting RETRIES to 3 in /etc/default/login.
set-root-group.fin	Changing the root user's primary group to avoid sharing the same group identifier with other users by setting root users primary group to 0 in /etc/passwd.
set-tmpfs-limit.fin	Setting limits on the usable size of the /tmp filesystem.
set-user-password-reqs.fin	Setting user password requirements. In particular MINWEEKS=1, MAXWEEKS=8, WARNWEEKS=1,PASSLENGTH=6-8.
set-user-umask.fin	Sets the user default file creation mask parameter to UMASK=022.
update-at-deny.fin	Updating 'at' facility access controls (at.deny)
update-cron-allow.fin	Updating cron facility access controls (cron.allow) by populating the /etc/cron.d/cron.allow with the root user accounts. As a security best practice it is common to deny all access to system facilities and explicitly allow access to those who require it. As a result root will be the only account authorized to use the cron facility
update-cron-deny.fin	Updating cron facility access controls (cron.deny).
update-cron-log-size.fin	Setting maximum size limits for the CRON facility log file.
install-md5.fin	Install MD5 package to create signatures.
install-fix-modes.fin	Install and run fixmodes package to tighten file permissions on Solari
print-rhosts.fin	Searches for .rhost and hosts.equiv files on the system.
print-sgid-files.fin	Searches for all Set-GID files on the system.
print-suid-files.fin	Searches for all Set-UID files on the system.
print-unowned-objects.fin	Searches for all objects belonging to nouser or nogroup.
print-world-writables-objects.fin	Searches for all objects with 0002 file permission.

Table 8: JASS Finish Scripts

3.2 Automated Installation

Automated installation is more a marketing word than reality. In real life we have to proceed with the following steps to install our firewall node:

- Gather pre-requisites
- Let jumpstart install server do its part
- Wrap-up with the manual tasks

3.2.1 Pre-Requisites

We need the following information before you can start the installation.

1. The hardware including the NICs. We recommend using the same hardware (CPU, HD, RAM) on all nodes. Check it by running: `/usr/platform/sun4u/sbin/prtdiag`
2. Checkpoint Firewall-1 license bound to the management stations IP address.
3. Stonebeat FullCluster license bound to the nodes control IP address.
4. Dragon/squire HIDS license for Solaris.
5. Your network topology.
6. Your cabling plan.

3.2.2 Jumpstart Server Installation Part

7. You do not want to connect your unprotected system to an active network nor the Internet; exposing the system could cause a compromise. Therefore install the hardware in a staging environment, which is a separated LAN segment.
8. Connect eri0 to the staging LAN.
9. Get the MAC-@ in ok prompt.
   `ok> .enet-addr`
10. Login to jumpstart server: `/export/install/jumpstart` is the jumpstart working directory
11. Run `JSSrv# ./setup-client`
12. Enter the cluster name. e.g. `c13f1`
13. Enter the MAC-@, which you got in step 9.
14. Enter the OS Version: e.g. Solaris_8_HW_02.02
15. Enter the installation profile:
    NG-fw-node-280R: For SunFire-280R with 2x 36 GB FC disks
    NG-fw-node-Ultra-60: For Ultra-60 with 2x 18 GB disks
    NG-fw-node-Ultra-80: For Ultra-80 with 2x 18 GB disks
16. Enter JS Server IP-@: e.g. IP-1
17. Confirm configuration
18. On the ok prompt of the firewall node enter:
    `ok> boot net – install`
19. If the installation is finished you should see a standard console login. Make sure that the first thing you do is changing the super-user password.
20. Check if the jumpstart server installation had no errors.
    `Node# grep ERROR /var/sadm/system/logs/*`

Errors with return code 8 result if the patch applies to a package that is not installed on the system and therefore can be ignored. Also return code 2, which indicates that the patch was already installed from the OS CD. A nice overview of Solaris Patch return codes can be found at [17].

If you run into problems with the jumpstart install server check:

- /etc/hosts
- /etc/ethers
- /etc/bootparams
- /export/install/jumpstart/rules.ok
- /export/install/jumpstart/rules

3.2.3 Manual Post-Installation

As said not everything is automated, therefore you have to do a few things by hand.

3.2.3.1 NIC Configuration

It is important that you have the network topology ready as recommended in 3.2.1. If you are not happy with the current interface configuration you need to adapt it:

21. Adapt `/etc/netmasks` file.
22. Adapt `/etc/hosts` file
23. Reconfigure the interfaces by using the ifconfig.
    - Change IP.
    - Remove unused interfaces ones.
24. Remove all `/etc/hostname.*`, which are unused.
25. Reboot the firewall node.
26. Check with 'ifconfig –a' that all interfaces are correct now.

3.2.3.2 Attach the Sub-Mirrors

Attaching the sub-mirrors during jumpstart installation does not work on the Sun Fire 280R platform as explained in 3.1.2.6. You therefore have to attach them manually one after each other.

27. Attach root sub-mirror
    ```
    node# /usr/sbin/metattach /dev/md/dsk/d0
    /dev/md/dsk/d20
    ```
28. Check if resync has finished after attaching
    ```
    /usr/sbin/metastat | grep sync
    ```
29. Attach var sub-mirror
    ```
    /usr/sbin/metattach /dev/md/dsk/d3 /dev/md/dsk/d23
    ```
30. Check if resync has finished after attaching
    ```
    /usr/sbin/metastat | grep sync
    ```

31. Attach swap sub-mirror
```
/usr/sbin/metattach /dev/md/dsk/d1 /dev/md/dsk/d21
```
32. Check if resync has finished after attaching
```
/usr/sbin/metastat | grep sync
```
33. Attach opt sub-mirror
```
/usr/sbin/metattach /dev/md/dsk/d5 /dev/md/dsk/d25
```
34. Check if resync has finished after attaching
```
/usr/sbin/metastat | grep sync
```

3.2.3.3 Bootable 2nd Disk

This step has to be taken to adjust the boot order of the system and to enable the "boot disk1" command.

35. Make the 2nd disk (disk1) bootable:
node# installboot /usr/platform/`uname –i`/lib/fs/ufs/bootblk /dev/rdsk/c1t1d0s0
36. Change the boot order in the boot prompt. Check if the correct nvalias exists for disk0 and disk1:
```
ok> devalias
```
37. If a correct alias does not exists for disk0 or disk1 create them:
```
ok> show-disks
    > select the disk
ok> nvalias disk0 ^y@0,0
ok> nvalias disk1 ^y@1,0
```
38. Check the boot device-order:
```
ok> printenv
```
39. Set if not already done the boot-device order:
```
ok> setenv boot-device disk0 disk1
```

3.2.3.4 SSH Configuration

40. Change in `/etc/sshd_config`
```
ListenAddr <Node Mgmt IP>
```

3.2.3.5 Checkpoint Installation (part 1)

The following configuration needs to be done by hand:

41. run cpconfig
Below is the short configuration input sequence. Detailed configuration steps can be found in C.3.4.
    - Do you accept all the terms of this license agreement? **Y**
    - (**2**) Distributed - select components of the Enterprise Product.
    - (**1**) Enforcement Module.
    - Install a Check Point clustering product? **N**
    - Enable SecureXL acceleration feature? **N**

- Do you want to add licenses? **N**
- Configuring Random Pool...
- Configuring Secure Internal Communication...
- Do you want to reboot? **Y**

3.2.3.6 Stonebeat Configuration

The Stonebeat configuration differentiates between the master node (Node-1) and the slave nodes (Node-2 till Node-X). Therefore this configuration steps are differed in 0 and 0.

Master Node-1 Configuration

On the node-1 execute:

42. Run sbfcconfig
    Below is the short configuration input sequence. Detailed configuration steps can be found in C.3.2.
    - 1. Generate Keys & Certificates
      - **1**. Create CA Key
      - **2**. Create CA Certificate
      - **3**. Create Module Key
      - **4**. Create Module Certificate
      - **5**. Create Client Key
      - **6**. Create Client Certificate
    - 2. Configure This Node
      - 1. Set Node ID [**1**]
      - 2. Set Cluster ID [i.e. **13**]
      - 3. Set Capacity [(1) **auto**]
      - 4. Set Load Measurement Interval [**15**]
      - 5. Set Start Up Mode [(2) **standby**]
      - 6. Set Clustering Mode [(1) **balancing**]
      - 7. Set Control IP address [**Node1-IP**]
      - 8. Set Control Port [**tcp/3002**]
      - 9. Configure Interfaces
        - Configure primary heartbeat (**HB-1**)
        - Configure secondary heartbeat (**HB-2**) and control interface.
        - Configure Operational Interfaces.
    - **3**. Set Passphrase
    - **4**. Install License

- Deploy certificates: Copy the files below from $SBFCHOME/etc/cert/ to $SBFCHOME/etc
    - Modulecert.pem
    - Modulekey.pem
    - Dhparams.pem
    - Cacert.pem
    - Clients
    - Sbfccert.pem
    - Sbfckey.pem
- Run sbfcpassphrase
- Clean up unused interfaces. In /etc do
  `rm `ls | grep hostn | grep –v __ | grep –v sbif`

Slave Node Configuration

On all the remaining nodes in the cluster execute:

43. Run sbfcconfig on the node
    Below is the short configuration input sequence. Detailed configuration steps can be found in C.3.3.

- 2. Configure This Node
    - 1. Set Node ID [**2**]
    - 2. Set Cluster ID [i.e. **13**]
    - 3. Set Capacity [(1) **auto**]
    - 4. Set Load Measurement Interval [**15**]
    - 5. Set Start Up Mode [(2) **standby**]
    - 6. Set Clustering Mode [(1) **balancing**]
    - 7. Set Control IP address [**Node2-IP**]
    - 8. Set Control Port [**tcp/3002**]
    - 9. Configure Interfaces
        - Configure primary heartbeat (**HB-1**)
        - Configure secondary heartbeat (**HB-2**) and control interface.
        - Configure Operational Interfaces.
- 3. Set Passphrase
- 4. Install License
- Deploy certificates: Copy from **Master Node-1** the files below from $SBFCHOME/etc/cert/ to $SBFCHOME/etc on the new installed slave Node.
    - Modulecert.pem

- Modulekey.pem
- Dhparams.pem
- Cacert.pem
- Clients
- Sbfccert.pem
- Sbfckey.pem

- Run sbfcpassphrase
- Clean up unused interfaces. In /etc do
  `rm `ls | grep hostn | grep –v __ | grep –v sbif`

3.2.3.7 Stonebeat Modifications

SBFC Test System (checklist)

Check the $SBFCHOME/etc/checklist that all operational interfaces are controlled and that the correct heartbeat interfaces are used.

Sbif mappings

If you need to change the sbif to qfe mapping you may do this by editing /etc/init.d/sbfcifconfig file.

3.2.3.8 Checkpoint Modifications (part 2)

Object Creation

44. Create the 2 node objects in the Checkpoint management station.
45. Establish SIC.
46. Import Topology and define anti-spoofing settings.
47. Create the cluster object and add the 2 nodes to it.
48. Create the sync networks. Select synchro tab, use state sync and define sync networks.
49. Create rule-set.
50. Ensure that cluster XL is NOT selected in the object.
51. On each firewall node do: cpconfig > [7] Enable Check Point Cluster XL and State Synchronization > y > y

Magic MAC address

If you have more than one firewall cluster connected to the same LAN segement you have to change the magic MAC address. Otherwise your switch sees duplicated source MAC addresses and starts to flood frames, because Checkpoint is using a special MAC address as the sender address for the clustering, which is called Magic MAC address. See the Checkpoint documentation for a detailed description. To prevent this we always recommend you to change the magic MAC address from the default values

'fwha_mac_magic=0xfe' and 'fwha_mac_forward_magic=0xfd' to a specific setting. Add the following line in /etc/system on all the nodes.

* Begin FW-1 set magic MAC address
Set fw:fwha_mac_magic=0x(control IP address node-1 in hex), e.g. 0x33
Set fw:fwha_mac_forward_magic=0x(control IP address node-2 in hex), e.g. 0x34
* End FW-1 set magic MAC address

Control if the settings are correct:

```
fw ctl get int fwha_mac_magic
fw ctl get int fwha_mac_forward_magic
```

3.2.3.9 ELA logging

To enable sending Stonebeat FullCluster logs automatically to the Checkpoint log station we have to enable ELA logging.

- On the Checkpoint Dashboard create the ELA objects.
  [Mgmt]: Manage > OPSEC Applications
  [Mgmt]: New OPSEC Applications
  Choose your host and select as the only Client Entity: ELA
- Then press on: Communication…
  Set a passphrase and press CLOSE (Status: Initialized but trust not established)
- Press OK and SAVE.
- No install the security policy on all nodes.
- Back on the firewall node:
  - Edit $SBFCHOME/etc/stonebeat_ela.conf
  - Change $opsec_application_sic_name to node CN (Example: CN=c13f1-ela,O=fwadmin..zrofys)
  - Chang $ela_server_sic_name to mgmt CN (Example: ch=cp_mgmt,o=fwadmin..zrofys)
  - Execute: opsec_pull_cert –h <CP Mgmt IP> –n <fw1-ela-object> -p <password>
- Reboot node – otherwise ela logging does not work. We had a cause open they do not why, but it works.
- Check in the Smart Tracker for a StoneBeat FullCluster entry.
- Increase the log level to 3 on all nodes. Like this we get some more useful information:
  sbfc setlog 3 node <node-id>

3.2.3.10 NTP Configuration

Node

Check that ntp is working: `ntpq -p`

NTP Source Stations

On all the stations you have defined as time-source you have allow the node to query for the time and set the authentication key.

- Edit /etc/inet/ntp.conf on the NTP source station. Add the following line:
  restrict <node-ip-@> notrust nomodify notrap
- Restart daemon
  /etc/init.d/xntp stop
  /etc/init.d/xntp start
- Check that ntp is running fine
  ntpq –p

3.2.3.11 Start Scripts

If you have to add starting scripts, such as the staticroutes.

3.2.3.12 EEPROM Security Mode

- Set the eeprom password. This setting does not ask for the password during normal multi-user boot.
  Execute: ‘eeprom security-mode=command’
  Do not loose the password! If you did you can simply run
  ’eeprom security-mode=none’ to erase the forgotten password.

3.2.3.13 Final Steps

- Label the interfaces (hme0, qfe0 – qfeX)
- Label the box.
- Update your documentation.

4 ONGOING MAINTENANCE PROCEDURES

In the risk mitigation planning section 2.1.2 we defined several actions and besides the system hardening also the following maintenance tasks:

1. Automated system monitoring
2. Change request process
3. Harden security policy
4. System integrity monitoring
5. Log file monitoring
6. Log file rotation
7. Regular software and OS patching
8. Regular system audits
9. Fulfill legal requirements
10. Use strong passwords

We achieve these tasks by implementing the following maintenance procedures:

- **Change Control and Configuration Management**: covers 2 and 3.
- **Software and OS patches:** covers 7.
- **System Monitoring:** covers 1.
- **Log Maintenance:** covers 5 and 6.
- **Intrusion Detection:** covers 4.
- **System Audits:** covers 8, 10.
- **Backups:** covers 9.

4.1 Change Control and Configuration Management

We differentiate between two kinds of changes:

- Firewall policy changes
- System configuration changes

4.1.1 Firewall Policy Change

In a firewall policy change request different roles are involved. We will shortly introduce them:

- **Requester**: In most cases this is a project manager or a system administrator.
- **Approver:** In most cases this is the Chief Security Officer.

- **Implementer:** In most cases this is done by the firewall system administrator.

Since there are those many people involved, it is necessary that a clear process is defined. If not it will happen that not all parties are involved and as a result rules are implemented without approval or documentation. In larger environment it is useful to integrate this process directly into the trouble ticketing system, such as ARS Remedy [18]. We achieve with such a process that:

- Changes are communicated.
- Rule requester can actively monitor the status of its change request.
- Changes are documented.

4.1.2 System Configuration Changes

Also for the system configuration changes it is important to follow a process. We respect the following rules:

- Prepare your change. E.g. get the software, built the software, test the new software and configuration change in the lab.
- Carefully plan the change. Have a backup strategy ready and a testing scenario.
- Schedule and communicate the change with the change-manager. Like this you can reduce the risk that somebody else is changing something, which interfere with your change. You are also certain that all required people are informed.
- Implement your change and verify it with the test procedures.
- Communicate the success or failure of the change and wrap-up the documentation.

It proved to us that it is helpful to directly document into a log-file all system changes. This log-file is kept on the system. If we are troubleshooting we always have a quick look on what was recently changed on the system. It also helps to understand and justifying your decision on making the change in this way.

4.2 Software and OS Patching

As we saw in the mitigation planning section 2.1.2 software bugs and vulnerabilities are the ones were risk mitigation is very difficult to achieve. Therefore it is vital to keep track of the latest vulnerabilities. To achieve this there are different strategies – explained below. The first step you should do is to be at least subscribed to the vendor incident response mailing lists of the products you are using. Such as:

- Sun mailing list
  To subscribe, send an e-mail to security-alert@sun.com with the following in the subject line: subscribe CWS <your-mail-address>

- Checkpoint FW-1 mailing list
  To subscribe, send an e-mail to Listserv@amadeus.us.checkpoint.com with the following in the message body: SUBSCRIBE fw-1-mailinglist <your name>
- Stonesoft mailing list
  To subscribe, send an e-mail to subscribe-security-alert@stonesoft.com.
- Enterasys mailing list
  To subscribe http://www.lcoim.com/Enterasys/EnteraNews/Subscriber.php

In addition to this you should track the newest vulnerabilities. This can be done by reading different forums or mailing-lists. An overview can be found at [19]. Here we list three well-known ones:

- **Bugtraq**
  To subscribe, send an e-mail to LISTSERV@SECURITYFOCUS.COM with following in the message body: SUBSCRIBE BUGTRAQ Lastname, Firstname
- **CIAC**
  To subscribe, send an e-mail to ciac-listproc@llnl.gov with following in the message body: subscribe ciac-bulletin LastName, FirstName PhoneNumber
- **Best of Security**
  To subscribe, send an e-mail to best-of-security-request@suburbia.net with following in the message body: subscribe best-of-security

Another strategy is to let others make the work for you and only get very specific security bulletins. Such a service you may get from [20].

Always make sure to read and understand the bulletin. In many cases your system is not affected, because it is an issue of a specific feature, which you do not use. In such a case an upgrade would represent a higher risk than not upgrading.

If a real threat was identified the patch should be downloaded and md5 checked. Adhere to the change process outlined in section 4.1.2. Then built the software on your own if possible and use your secure built-server. The next step is installing the patch on your test environment, which you can stress test and monitor to be sure that the system is running stable and the patch does not have a bad impact on your system. Then carefully plan and communicate your upgrade. We further recommend documenting your modifications.

Most patches come with their install scripts – we therefore do not further discuss this issue. It is important that you are in single-user mode when you install system patches. Also run the JASS scripts after each system modification. Execute: /opt/SUNWjass/jass-execute -d undoable-hardening.driver

4.3 System Monitoring

The health of the firewall cluster should be monitored by a network management centre (NMC) and by the cluster itself. As already described in section 3.1.2.10 the system monitors different parameters such as its log space, CPU load, interfaces and if the firewall module is running and has a policy installed. If it detects that something is wrong, it sends a trap to the NMC and takes in certain cases actions as defined in alert.sh [C.3.6]. On the other hand the NMC reads every 30 seconds the SBFC node status (MIB: .1.3.6.1.4.1.1369.2.2). The system health can also be checked on the node itself by executing: 'sbfc status'. If you need to increase the log level execute 'sbfc setlog <log-level> node <node-id>'. <log-level> is a number between 0 and 5, which is the most verbose level. Default value is 3.

4.4 Log Maintenance

Several log files are produced on the system, such as:

1. /var/adm/messages – the system logs
2. /var/log/syslog – the syslog logs
3. /var/log/snmpd – the logs of the snmpd
4. /var/cron/log – the cron logs
5. /var/adm/ssh.log – the ssh authentication logs
6. /var/opt/CPfw1-54 – the Checkpoint firewall logs

Log File Rotation

We rotate these logs regularly by cron jobs:

10 3 * * 0,4 /etc/cron.d/logchecker – for the /var/cron/log

10 3 * * 0 /opt/scripts/rotate-logs.sh – for the /var/adm/messages, /var/adm/ssh.log, /var/log/syslog

The rotate-logs.sh script can be found in appendix C.6.2.

Log File Monitoring

Besides rotating them they also need to be analyzed. This task should be automated with a tool such as LogCheck, swatch, sac or logsurfer.

As defined in 2.2.3 we are using the LogCheck tool to analyse the log files.

The installation of LogCheck was already explained in section 3.1.2.12. Here we outline the configuration. LogCheck requires 6 files:

- **logcheck.sh**: The main script, which we execute every 15 minutes by a cronjob. It controls all the processing and parses the log files with grep commands.
- **logtail:** Which is similar to tail –f on each logfile. It remembers what was already processed in the logfile by LogCheck.

- **logcheck.hacking**: In this file we defined keywords which clearly indicate an attack on the system.
- **logcheck.violations**: In this file we define keywords of system events that are usually seen as negative such as "denied" and "refused". Positive words such are also placed in this list. Violations are reported as "Security Violations" in the report.
- **logcheck.violations.ignore**: In this file we define the words that are reverse searched against the logcheck.violations file, essentially nullifying them.
- **logcheck.ignore:** In this file we define the keywords which are never reported by LogCheck.

Summarizing the process, every 15 minutes logcheck.sh is executed and calls logtail on all log files. Logtail parses off any text from the last time it was run. LogCheck executes a grep on this text with the keywords specified in logcheck.hacking to identify a possible system attack message. Then LogCheck greps keywords in logcheck.violations for any violations. It greps for ignore violations and at the end for all messages to ignore from logcheck.ignore. Any messages found are mailed to system admin.

The best way to start with the keyword files is by training them step-by-step. The ones provided in the package are a good start.

The configuration of the logcheck.sh is straight forward and can be found in appendix C.6.2. We had to add the log logfiles, which LogCheck is going to analyze.

4.5 Intrusion Detection

As stated in the mitigation actions the system integrity should be monitored. This can be done with tools such as tripwire or AIDE. In this setup we decided to use Dragon from Enterasys, a host-based IDS, which provides besides file integrity control the following features:

- Log file analysis against a signature policy
- Honeypot port detection
- MD5 File integrity analysis

The Dragon installation was already outlined in section 3.1.2.13. We explain here the configuration.

Signature Based Log File Analysis

The signatures explained in detail in appendix C.5.4 are compared against different log entries from the messages file such as failed attempts, failed ‘su’ command, system reboots, failed ssh attempts, ssh port forwarding, etc.

MD5 File Integrity Analysis

To verify that nobody compromised the system and modified important system files we run a file integrity check each day. The dragon dsquire.net file defines each file and its attribute. To list it here would be a waste of

paper. We therefore decided to include instead a list of files, which should be checked and its attributes in tripwire convention. This file is based on the sample file from [21] and is slightly adapted to support our specific software (Checkpoint, Stonesoft and Enterasys).

The tripwire has more capabilities than dragon. Dragon does not know the 'a' (access time) flag, but that doesn't matter, because the 'a' flag is not used by the R-L-E-N tripwire representation. Furthermore dragon only supports MD5. Below we list the dragon representation of R-L-E-N:

R: +pinugsmc1 – read-only
L: +pinug-sm1 – logfile
E: -pinugsmc1 – ignore everything

The detailed list of controlled files is listed in appendix C.5.3

Honeypot Port Detection

Dragon allows us to open unused service ports and as soon as someone tries to connect alerts us. We are using this for the following services: telnet, http and ftp. To use more does not make sense since it just generates more false-positives.

IDS alerts should be constantly monitored by the Network Management Centre.

4.6 System Audits

To reduce the risk of configuration errors we defined a process within the company to conduct the following system audits:

- Monthly automated vulnerability scans from external (Internet) and internal (tier-1 LAN, Intranet). We do this with the QualysGuard scanner from Qualys [10]. A sample scan can be found in section 5.2.6.
- Once a year external security experts come on-site and inspect the firewall.
- Change all system passwords and ssh keys each 3 months.

4.7 Backups

The backup of the firewall is very slim. We only backup the log files in /var/adm, the /etc/init.d and the $SBFCHOME/etc files. This is executed daily by the following cron job.

55 23 * * * /opt/scripts/backup.sh

The backup.sh [C.10] script compresses and md5 hashes the following directories:

- /etc/init.d – the init scripts
- /var/adm – the system log files
- /opt/fullcluster/etc – the stonebeat configuration files

The backup file and md5-checksum file is pulled daily by a central backup station via ssh from the firewall nodes.

A full node restore can be done within 3 hours, due to the automated installation.

5 TEST AND VERIFY THE SETUP

In order to verify that the setup was successful and that the system is now hardened we need to test it. We split the test into two groups:

- **Functional Tests:** We test if the installation is working as expected and no configuration errors occurred.
- **Security Tests:** We test that the system is hardened and achieves the required mitigation actions.

With this testing chapter we implement the following risk mitigation actions:

- Regular system audits
- Testing process
- Use strong passwords

5.1 Functional Tests

Verify with the following tests that the installation fully functional and that no errors occurred:

1. Verify that the correct boot prompt parameters are set.
   - ```
     I: devalias
     O: boot-device  disk0 disk1
     ```
   - ```
     I: printenv
     O: disk0  /pci@1f,400/scsi@3/disk@0,0
        disk1  /pci@1f,400/scsi@3/disk@1,0
     ```
2. Check if the jumpstart server installation had no errors.
   ```
   Node# grep ERROR /var/sadm/system/logs/*
   ```
   - Errors with return code 8 result if the patch applies to a package that is not installed on the system and therefore can be ignored. Also return code 2, which indicates that the patch was already installed from the OS CD. A nice overview of Solaris Patch return codes can be found at [17].
3. Check that the disk mirroring is working. Execute:
   - I: metadb – the 'luo' attributes indicate that the mirroring works fine. In particular it means that the system could read from the locator (l), the replica is up to date (u) and replica was active prior to last mdd change. If you would see a 'W', which stands for replica has device write errors, you have to repair the databases.

     ```
     O:      flags          first blk      block count
        a m  p  luo         16             1034          /dev/dsk/c1t0d0s4
        a    p  luo         1050           1034          /dev/dsk/c1t0d0s4
        ….
        a    p    luo       16             1034          /dev/dsk/c1t1d0s6
     ```

- I: metastat – if the states are okay everything is fine.
  O: d5: Mirror
  Submirror 0: d15
  State: Okay
  Submirror 1: d25
  State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 16559748 blocks

  d15: Submirror of d5
  State: Okay
  Size: 16559748 blocks
  Stripe 0:

Device	Start Block	Dbase	State	Hot Spare
c1t0d0s5	0	No	Okay	

  d25: Submirror of d5
  State: Okay
  Size: 16559748 blocks
  Stripe 0:

Device	Start Block	Dbase	State	Hot Spare
c1t1d0s5	0	No	Okay	

4. Check that you can log in as root from the console port.
5. Run cpconfig and check that it was executed and configured.
6. Check that the magic MAC addresses are correctly set:
   - I: fw ctl get int fwha_mac_magic
     O: fwha_mac_magic = 31
   - I: fw ctl get int fwha_mac_forward_magic
     O: fwha_mac_forward_magic = 32
7. Check that Stonebeat is running correctly.
   - Execute 'sbfc status'
   - Verify that /etc/init.d/sbfcifconfig is correctly setup. This file contains the interface configuration, in particular the sbif to qfe mappings.
   - Check $SBFCHOME/etc/node.conf if your configuration is correct. This configuration file contains the mapping of virtual and physical IP address to the interfaces.
8. Check that ela logging is working, as mentioned above ela reports the Stonebeat logs into the Checkpoint log server.
   - Make a connection through the firewall, which is permitted by your firewall policy. Then take one node offline and back online. The connection should be still alive and you should have an entry in the CP log viewer.

9. Check that NTP is able to synchronize with your time sources.
   - I: ntpq- p

```
remote           refid      st t when poll reach   delay   offset  disp
==============================================================================
*time-src1       <IP-1>      3 u  246 1024  377    6.06    2.619   2.67
+time-src2       <IP-2>      4 u  380 1024  377    3.54    0.366   0.41
```

10. Check that the routing table is correct.
    - 'netstat –nr' shows you the routing table of the system. If necessary adapt /etc/rc3.d/S99staticroutes and /etc/defaultrouter.
11. If you are NATing Stonebeat needs to be adapted.
    - Check that $SBFCHOME/etc/filter.conf is correctly set.
12. Check that all cron jobs are running:
    - crontab –l

```
10 3 * * 0,4 /etc/cron.d/logchecker
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
0 0,8,16 * * * /usr/bin/logger -p auth.info `/usr/sbin/eeprom security-#badlogins`
0,20,40 * * * * /usr/lib/sa/sa1
45 23 * * * /usr/lib/sa/sa2 –s 00:00 –e 23:59 –I 1200 –A
55 23 * * * /opt/scripts/backup.sh
10 3 * * 0   /opt/scripts/rotate-logs.sh
00,15,30,45 * * * * /opt/logcheck/bin/logcheck.sh
```

13. Check that the latest patches are installed.
    - showrev –p for Solaris
    - sbfc status for Stonebeat FullCluster
    - fw ver –k for Checkpoint

5.2 Security Tests

With the security tests below we check that hardening was effective during installation. One single mistake can result in a big security hole.

5.2.1 Network Access

14. Check that it is possible to log in as user 'sans' via ssh.
    Execute ssh sans@c13f1
    Using username "sans".
    Authenticating with public key "sysadmin-1@sans.org" from agent
    Last login: Fri Sep 17 20:09:33 2004 from <IP-@>
    <banner>
    c13f1$
15. Check that you may get root privileges.
    Execute: ssh root@localhost
    Last login: Fri Sep 17 13:17:54 2004 from 127.0.0.1
    <banner>
    c13f1#

16. Check that it is not possible to log in as root via ssh from remote.
    Execute: ssh root@c13f1
    Permission denied (publickey).

5.2.2 Firewall Policy

17. Check the firewall policy installed on the node. Scan with nmap a server, which is standing behind the firewall. We use stealth scanning to further check if the Checkpoint firewall is able to detect it and reports it in its log. We further use as source port 80 that it looks like http return traffic. We scan the full range of all 65'000 ports. This will take around 50 minutes.

    I: nmap -v –g80 -sS -sR -P0 -O -p1-65000 -o results.out <target IP>

    O: cat results.out

    (The 64990 ports scanned but not shown below are in state: filtered)

Port	State	Service (RPC)
21/tcp	open	ftp
22/tcp	closed	ssh
23/tcp	closed	telnet
53/tcp	closed	domain
80/tcp	open	http
111/tcp	closed	sunrpc
443/tcp	open	https
512/tcp	closed	exec
513/tcp	closed	login
514/tcp	closed	shell

    Now we control if the security policy should really allow these 10 ports to be open.

5.2.3 Restricted User Accounts

18. Check that only the required users are installed.
    - cat /etc/passwd

5.2.4 Minimized System

19. Check that only necessary services are running.
    - ps –ef compare it to the definition in section 2.2.5.
20. Check that only the necessary network services are open.
    - lsof -i result is in Table 9 compare it with the defined ports in section 2.2.4.
21. Check that only the required packages are installed.
    - pkginfo compare it with the list in B.2.

	Binding	Service	Description
TCP	other-fw-node.3002	sbfcd	Stonebeat control port.
	127.0.0.1.22	sshd	SSH for root login.
	this-fw-node.22	sshd	SSH for user login.
	*.18208	cprid	Check Point Remote Installation Protocol. - Protocol used from MM to FWM when installing Updates.
	127.0.0.1.8989	cpd	Check Point unknown internal service.
	*.18191	cpd	Check Point Daemon Protocol. - Download of rulebase from MM to FWM - Fetching rulebase, from FWM to MM when starting FWM - Download of rulebase from MDS/CMA to FWM - Fetching rulebase, from FWM to CMA when starting FWM
	127.0.0.1.18196	cpd	Check Point unknown internal service.
	*.18192	cpd_amon	Check Point Internal Application Monitoring. - Protocol for getting System Status, from MM or MDS/CMA to FWM
	*.257	fw1_log	Check Point VPN-1 & FireWall-1 Logs. - Protocol used for delivering logs from FWM to MM - Protocol used for delivering logs from FWM to CMA or CLM
	127.0.0.1.1024	fw	Check Point unknown internal service.
	127.0.0.1.65535	fw	Check Point unknown internal service.
	*.256	FW1	Check Point VPN-1 & FireWall-1 Service - Download of rulebase from MM to FWM (4.x) - Fetching rulebase from FWM to MM when starting (4.x) - Get topology information from MM or CMA to FWM (also for NG) - Full synchronisation for HA configuration (also for NG)
	127.0.0.1.18281	cp_local	Check Point unknown internal service.
	*.32775	fw	Check Point Secure FTP - undocumented.
	*.32776	fw	Check Point unknown service.
	*.32777	fw	Check Point authenticated Rlogin - undocumented.
	*.32778	fw	Check Point secure ESMTP server - undocumented.
	*.32779	fw	Check Point authenticated telnet - undocumented.
	*.259	clntauth clntauth_telnet	Check Point VPN-1 & FireWall-1 Client Authentication (Telnet). - Protocol for performing Client-Authentication at FWM using telnet
	*.32780	fw	Check Point unknown service.
	*.900	clntauth clntauth_http	Check Point VPN-1 & FireWall-1 Client Authentication (HTTP). - Protocol for performing Client-Authentication at FWM using HTTP
	*.18183	sam	Check Point OPSEC Suspicious Activity Monitor API. - Protocol e.g. for Block Intruder between MM (or CMA) and FWM

	Binding	Service	Description
	*.18187	ela	Check Point OPSEC Event Logging API - Protocol for applications logging to the Firewall log at MM
	*.18184	lea	Check Point OPSEC Log Export API. - Protocol for exporting logs from MM
	127.0.0.1.4532	fwssd	Check Point unknown internal service.
	127.0.0.1.262	fwssd	Check Point unknown internal service.
UDP	*.165	snmpd	SNMP daemon listens for NMS reads.
	*.123	xntpd	NTP daemon
	127.0.0.1.123	xntpd	NTP daemon
	time-source-1.123	xntpd	NTP daemon
	time-source-2.123	xntpd	NTP daemon

Table 9: Listening Services

5.2.5 System Benchmarking – CISscan

We are benchmarking the installation with the Solaris version of the CIS scoring tool. To install it we:

1. Download it from [22] into /tmp
2. c13f2# uncompress cis_score_tool_solaris_v1.5.0.sh.Z
3. c13f2# chmod u+x cis_score_tool_solaris_v1.5.0.sh
4. c13f2# ./cis_score_tool_solaris_v1.5.0.sh
5. c13f2# pkgadd -d CISscan all
6. Since the CISscan may modify some of the files we make a backup.
   c13f2# ./do-backup.sh

To run the benchmark execute: /opt/CISscan all the result is located in /opt/CIS/cis-ruler-log.[date-timestamp]. To filter for the positive and negative results execute:

node# grep "^Positive" /opt/CIS/ cis-ruler-log.20040918-10:15:41.802
node# grep "^Negative" /opt/CIS/cis-ruler-log.20040918-10:15:41.802

Let us have a look on the negative benchmark results of our installation.

node# grep "^Negative" /opt/CIS/cis-ruler-log.20040918-10:15:41.802

Negative: 4.5 ip_forwarding not deactivated.
This is has to be like this on all systems who forward packets, such as our firewall. The ip_forwarding is also controlled by the Checkpoint daemon. If we stop the daemon ip_forwarding is turned off.

Negative: 5.8 kernel-level auditing isn't enabled.
The BSM module produces to heavy output and therefore, we have not installed it.

Negative: 7.6 /etc/dt/config/Xconfig doesn't exist, thus permits xdmcp port listening.
We do not need it since X is not installed on our system.

Negative: 6.5 Non-standard world-writable file: /var/opt/CPshrd-54/registry/HKLM_registry.data.old
Negative: 6.5 Non-standard world-writable file: /var/opt/CPshrd-54/registry/HKLM_registry.data
This is not nice but we can not change it. Checkpoint requires it. If we change it, we would break the application.

The result looks fine and also the score. Ten is the most secure and zero is the least. The rating from an unhardened system is: **3.16** / 10.00

The CIS Benchmark Results of our hardened system is: **9.38** / 10.00

5.2.6 QualysGuard Scan

As mentioned before we also test our firewall with the QualysGuard [10]. Figure 7 shows the QualysGuard at scanner at work.

Figure 7: QualysGuard Scan Process

After a few minutes the QualysGuard scanner reports the result shown in , which is very satisfactorily and proves that the system is secure.

Figure 8: QualysGuard Scanning Results

BIBLIOGRAPHY

[1] D. Chapman, E. Zwicky. *Building Internet Firewalls.* Sebastopol: O'Reilly & Associates, Inc, 1995. 17 – 19.

[2] FMEA. *"FMEA Info Centre".* http://www.fmeainfocentre.com (15 Sep. 2004).

[3] R. McDermott, R. Mikulak, M. Beauregard. *The Basics of FMEA.* New York: Productivity Press, 1996.

[4] Sun. *"Get Solaris OS Software".* http://wwws.sun.com/software/solaris/get.html (15 Sep. 2004).

[5] Sun. *"Solaris Security Toolkit (JASS)".* http://wwws.sun.com/software/security/jass/ (15 Sep. 2004).

[6] Spitzner, Lance. *"Building Your Firewall Rule Base".* http://www.spitzner.net/rules.html (15 Sep. 2004).

[7] Dragon Enterasys. *"Dragon Intrusion Defense".* https://dragon.enterasys.com/ (15 Sep. 2004).

[8] Stonesoft. *"Product Downloads".* https://my.stonesoft.com/download/ (15 Sep. 2004).

[9] Rowland, Craig. *"Project: LogCheck/LogSentry".* http://sourceforge.net/projects/sentrytools (15 Sep. 2004).

[10] Qualys. *"QualysGuard Scanner".* http://www.qualys.com (15 Sep. 2004).

[11] Pomeranz, Hal. *"Solaris Jumpstart Ressources".* http://www.deer-run.com/~hal/jumpstart/ (15 Sep. 2004).

[12] Duke University. *"Duke Sysadmin Talk".* http://www.cs.duke.edu/~braun/jumpstart/ (15 Sep. 2004).

[13] Spitzner, Lance. *"Armoring Solaris: II".* http://www.spitzner.net/armoring2.html (15 Sep. 2004).

[14] Gev, Yechiel. *"Minimum Solaris 8 Packages Required for VPN-1/FW-1 NG".* http://secureknowledge.checkpoint.com/pub/sk/docs/public/os/solaris/pdf/solaris8_pkgs_fp3_rev2.pdf (15 Sep. 2004).

[15] Belovince, S. *"RFC-1958: Defending Against Sequence Number Attacks".* http://rfc.sunsite.dk/rfc/rfc1948.html (15 Sep. 2004).

[16] Sun. *"Solaris Operating Environment Network Settings 1.0".* http://www.sun.com/security/blueprints (15 Sep. 2004).

[17] Sinclair InterNetworking Services. *"Solaris Patch Return Codes".* http://www.sins.com.au/unix/solaris_patch_return_codes.html (15 Sep. 2004).

[18] Remedy. *"AR System Product Suite".* http://www.remedy.com/solutions/coretech/index.html (15 Sep. 2004).

[19] INFOSYSSEC. *"Security Mailing List".* http://www.infosyssec.net/infosyssec/secmail1.htm (15 Sep. 2004).

[20] SECUNIA. *"Vulnerability Tracking Service".* http://secunia.com/vulnerability_tracking_service/?menu=prod (15 Sep. 2004).

[21] Pomeranz, Hal. *"Tripwire configuration file".* http://www.deer-run.com/~hal/tw.config (15 Sep. 2004).

[22] Center for Internet Security. *"Solaris – Level 1"* http://www.cisecurity.org/bench_solaris.html (17 Sep. 2004)

[23] Checkpoint. "Downloads". http://www.checkpoint.com/techsupport/downloads.jsp (15 Sep. 2004).

[24] OpenSSH. *"Portable OpenSSH".* http://www.openssh.org/portable.html (15 Sep. 2004).

[25] Sun. *"lsof for Solaris 8".* http://cgi.sun.com/freeware/package?id=4290 (15 Sep. 2004).

[26] Sun. *"FixModes Scripts".* http://www.sun.com/blueprints/tools/ (15 Sep. 2004).

[27] Sun. *"MD5 Scripts".* http://www.sun.com/blueprints/tools/ (15 Sep. 2004).

[28] DSquire. *"Enterasys Dragon Host Sensor".* http://www.enterasys.com/products/ids/DSHSS-xxx/ (15 Sep. 2004).

[29] Behrens, Matt. *"Solaris Package System".* http://developer.berlios.de/projects/solpkg/ (15 Sep. 2004).

APPENDIX A – FEMA TABLES

A.1 SEV: Severity Table

Ranking	Effect	Criteria: Severity of Effect (SEV)
1	None	No effect
2	Very Minor	Very minor effect on service performance.
3	Minor	Minor effect on service performance.
4	Low	Small effect on service performance. The service does not require repair.
5	Moderate	Moderate effect on service performance. The service requires repair.
6	Significant	Service performance is degraded. Comfort or convenience functions may not operate.
7	Major	Service performance is severely affected but functions. The system may not be operable.
8	Extreme	Service is inoperable with loss of primary function. The system is inoperable.
9	Serious	Failure involves hazardous outcomes and/or noncompliance with standards or regulations.
10	Hazardous	Failure is hazardous, and occurs without warning. It suspends operation of the system and/or involves noncompliance with regulations.

A.2 OCC: Likelihood of Occurrence Table

Ranking	Possible Failure Rate	Likelihood of Occurrence (**OCC**)	Example
1	≤ 2.7 x 10E-5	Nearly Impossible	each 100 years
2	5.4 x 10E-5	Remote	each 50 years
3	1.3 x 10E-4	Low	each 20 years
4	5.4 x 10E-4	Relatively Low	each 5 years
5	2.7 x 10E-3	Moderate	each year
6	3.7 x 10E-3	Moderately High	each 9 months
7	5.5 x 10E-3	High	each 6 months
8	1.1 x 10E-2	Repeated Failures	each 3 months
9	3.3 x 10E-2	Very High	each month
10	⩾ 0.14	Extremely High. Failure Almost Inevitable	each week

A.3 DET: Likelihood of Detection Table

Ranking	Likelihood of Detection (**DET**)
1	Almost Certain Detection
2	Very High Chance of Detection
3	High Probability of Detection
4	Moderately High Chance of Detection
5	Moderate Chance of Detection
6	Low Probability of Detection
7	Very Low Probability of Detection
8	Remote Chance of Detection
9	Very Remote Chance of Detection
10	Absolute Uncertainty - No Control

APPENDIX B – JUMPSTART SERVER

B.1 Directory Tree

```
/
└ export
    └ install
        └ jumpstart
              └ OS
                 └ Solaris-8_2-02            # Solaris Distribution
              └ drivers                      # JS Drivers
              └ files                        # add-on software dir
                 └ .ssh                      # root ssh dir[3]
                 └ Patches                   # Solaris Patchcluster
                     └ 8_Recommended
                 └ etc                       # See footnote 3
                     └ default
                     └ inet
                     └ init.d
                 └ opt
                     └ scripts
                 └ home                       # See footnote 3
                     └ sans
                         └ .ssh               # user ssh dir
                 └ kernel                     # See footnote 3
                     └ drv
                 └ opt                        # See footnote 3
                     └ fullcluster
                         └ etc
                     └ dragon
                 └ software                   # 3rd party software
                     └ Myopenssh-current      # openSSH pkg dir
                     └ Checkpoint-current     # CP pkg dir
                     └ DiskSuite-current      # DiskSuite pkg dir
                     └ SBFC-current           # SBFC pkg dir
                     └ SBFCsnmpd-current      # pkg dir
                     └ SUNWjass-current.pkg   # Jass pkg dir
                     └ lsof-current           # lsof pkg dir
                     └ tools-current          # logchecker pkg dir
                     └ Dragon-current         # Dragon pkg dir
                     └ tester-conf            # SBFC special files
                 └ var                        # See footnote 3
                     └ empty
                     └ sadm
                         └ system
                             └ logs           # fin script log
                                              # files
                     └ tmp
                         └ ODS
              └ finish                        # JS finish scripts
              └ profiles                      # JS profiles
```

[3] This directory is installed on the node by NG-cp-rm-files.fin script.

B.2 Installed Packages

```
application Myopenssh      OpenSSH for Solaris
application CPfw1-54       Check Point VPN-1/FW-1 NG with Appl Int
application CPshrd-54      Check Point SVN Foundation NG with Appl Int
application ESdsquire      Enterasys Dragon HIDS
system      SBFCbase       StoneBeat FullCluster For FW-1 Mngmt Tools
system      SBFCdrv        StoneBeat FullCluster For FW-1 STREAMS Driver
system      SBFCmod        StoneBeat FullCluster For FireWall-1 Module
system      SBFCsnmp       SNMP Agent and Tools for Stonesoft products
application SMClsof        lsof 4.68
application SUNBEfixm      Fix Modes 1.42
application SUNBEmd5       MD5 1.4
system      SUNWaccr       System Accounting, (Root)
system      SUNWaccu       System Accounting, (User)
system      SUNWadmc       System administration core libraries
system      SUNWadmfw      System & Network Administration Framework
system      SUNWbash       GNU Bourne-Again shell (bash)
system      SUNWbzip       The bzip compression utility
system      SUNWcar        Core Architecture, (Root)
system      SUNWcarx       Core Architecture, (Root) (64-bit)
system      SUNWced        Sun GigaSwift Ethernet Adapter (32-bit Driver)
system      SUNWcedx       Sun GigaSwift Ethernet Adapter (64-bit Driver)
system      SUNWcsd        Core Solaris Devices
system      SUNWcsl        Core Solaris, (Shared Libs)
system      SUNWcslx       Core Solaris Libraries (64-bit)
system      SUNWcsr        Core Solaris, (Root)
system      SUNWcsu        Core Solaris, (Usr)
system      SUNWcsxu       Core Solaris (Usr) (64-bit)
system      SUNWeridx      Sun RIO 10/100 Mb Ethernet Drivers (64-bit)
system      SUNWesu        Extended System Utilities
system      SUNWfcip       Sun FCIP IP/ARP over FibreChannel Dev Driver
system      SUNWfcipx      Sun FCIP IP/ARP over FC Dev Driver (64 bit)
system      SUNWfcp        Sun FCP SCSI Device Driver
system      SUNWfcpx       Sun FCP SCSI Device Driver (64-bit)
system      SUNWfctl       Sun Fibre Channel Transport layer
system      SUNWfctlx      Sun Fibre Channel Transport layer (64-bit)
system      SUNWfns        Federated Naming System
system      SUNWfnsx       Federated Naming System (64-bit)
system      SUNWgzip       The GNU Zip (gzip) compression utility
system      SUNWhmd        SunSwift SBus Adapter Drivers
system      SUNWhmdx       SunSwift SBus Adapter Drivers (64-bit)
Application SUNWjass       Solaris Security Toolkit 4.0.1
system      SUNWkvm        Core Architecture, (Kvm)
system      SUNWkvmx       Core Architecture (Kvm) (64-bit)
system      SUNWlibC       Sun Workshop Compilers Bundled libC
system      SUNWlibCx      Sun WorkShop Bundled 64-bit libC
system      SUNWlibms      Sun WorkShop Bundled shared libm
system      SUNWlmsx       Sun WorkShop Bundled 64-bit shared libm
system      SUNWloc        System Localization
system      SUNWlocx       System Localization (64-bit)
system      SUNWmdr        Solstice DiskSuite Drivers
system      SUNWmdu        Solstice DiskSuite Commands
system      SUNWmdx        Solstice DiskSuite Drivers(64-bit)
system      SUNWnamos      Northern America OS Support
system      SUNWntpr       NTP, (Root)
system      SUNWntpu       NTP, (Usr)
system      SUNWpd         PCI Drivers
system      SUNWpdx        PCI Drivers (64-bit)
system      SUNWpiclr      PICL Framework (Root)
system      SUNWpiclu      PICL Libraries, and Plugin Modules (Usr)
```

```
system      SUNWpiclx      PICL Libraries (64-bit)
system      SUNWqfed       Sun Quad FastEthernet Adapter Driver
system      SUNWqfedx      Sun Quad FastEthernet Adapter Driver (64-bit)
system      SUNWqlc        Qlogic ISP 2200/2202 Fibre Channel Dev Driver
system      SUNWqlcx       Qlogic ISP 2200/2202 FC Dev Driver (64 bit)
system      SUNWses        SCSI Enclosure Services Device Driver
system      SUNWsesx       SCSI Enclosure Services Device Driver (64-bit)
system      SUNWssad       SPARCstorage Array Drivers
system      SUNWssadx      SPARCstorage Array Drivers (64-bit)
system      SUNWssaop      SPARCstorage Array Utility
system      SUNWswmt       Install and Patch Utilities
system      SUNWter        Terminal Information
system      SUNWzlib       The Zip compression library
system      SUNWzlibx      The Info-Zip compression library (64-bit)
```

B.3 Profile Scripts

B.3.1 NG-fw-node-280R.profile

```
# ------------------------------------------------------ #
# Install_type MUST be first                             #
# ------------------------------------------------------ #
install_type    initial_install

# ------------------------------------------------------ #
# start with the minimal required number of packages     #
# ------------------------------------------------------ #
cluster         SUNWCreq

# ------------------------------------------------------ #
# Since this is intended to be a minimal server, the     #
# Solaris Core software cluster was used.  However, in   #
# many cases, a few additional packages are required to  #
# make the system maintainable.                          #
# ------------------------------------------------------ #
# ------------------------------------------------------ #
# Required for CheckPoint FW-1 NG                        #
# ------------------------------------------------------ #
package         SUNWter         add
package         SUNWlibC        add
package         SUNWlibCx       add
package         SUNWadmc        add
package         SUNWadmfw       add

# ------------------------------------------------------ #
# To support the Network Time Protocol                   #
# ------------------------------------------------------ #
package         SUNWntpr        add
package         SUNWntpu        add

# ------------------------------------------------------ #
# To support Compression                                 #
# ------------------------------------------------------ #
package         SUNWgzip        add

# ------------------------------------------------------ #
# Support Bash Shell                                     #
# ------------------------------------------------------ #
package         SUNWbash        add

# ------------------------------------------------------ #
# To support snoop                                       #
# ------------------------------------------------------ #
```

```
package         SUNWfns         add
package         SUNWfnsx        add

# ---------------------------------------------------- #
# To support system accounting                         #
# ---------------------------------------------------- #
package         SUNWaccr        add
package         SUNWaccu        add

# ---------------------------------------------------- #
# To support Secure Shell (Solaris 7+)                 #
# ---------------------------------------------------- #
package                 SUNWzlib        add
package                 SUNWzlibx       add


# ---------------------------------------------------- #
# want to define how the disk is used - not use defaults  #
#                                                      #
# Firewall Server.  This box is fully minimized        #
# We allocate a great deal of space to /var as FW-1    #
# uses this for installation and logging.  Built       #
# for 36GB hard drive.                                 #
# ---------------------------------------------------- #

partitioning    explicit
usedisk         c1t0d0
filesys         c1t0d0s0     8192    /
filesys         c1t0d0s1     2048     swap
filesys         c1t0d0s3     16384   /var

# Two slices per disk are reserved for the metadbs
filesys         c1t0d0s4    10
filesys         c1t0d0s6     6
filesys         c1t0d0s5     free     /opt

# ---------------------------------------------------- #
# install system as standalone                         #
# ---------------------------------------------------- #
system_type     standalone
```

B.4 Drivers Scripts

B.4.1 NG-fw-node-280R.driver

```
#!/bin/sh
# ---------------------------------------------------- #
# Change just the SCRIPT_LIST variable                 #
# ---------------------------------------------------- #

SCRIPT_LIST="NG-cp-rm-files.fin
NG-delete-unneeded-FCarch.fin
NG-ods421base.fin
NG-openssh.fin
NG-install-patches.fin
NG-config-nics.fin
NG-sbfc.fin
NG-fw1-node.fin
NG-tools.fin
NG-dragon-hids.fin
NG-jass.fin"
```

```
# ------------------------------------------------------ #
# Do not change anything below here                      #
# ------------------------------------------------------ #

BASEDIR=/a
GENERIC=${SI_CONFIG_DIR}/files
PATCHFILES=${SI_CONFIG_DIR}/files/patches
PACKAGEASK=/tmp/pkg.ask
export BASEDIR PACKAGEASK PATCHFILES GENERIC

FINISH_DIR=${SI_CONFIG_DIR}/finish

for script in ${SCRIPT_LIST}
do
  if [ -f "${FINISH_DIR}/${script}" ]; then
    echo "starting finish script: ${script}"
    echo ""
    echo "***** $script ******" >>/a/tmp/debug.fins
    . ${FINISH_DIR}/${script} >>/a/tmp/debug.fins
  else
    echo "ERROR: file not found: ${script}"
  fi
done
```

B.5 Finish Scripts

B.5.1 NG-cp-rm-files.fin

```
LOG=/a/var/sadm/system/logs/NG-copyfiles.log

echo "Copying files in /etc/ /kernel/ /opt/ /usr/ and /var/"
echo "Copying files in /etc/ /kernel/ /opt/ /usr/ and /var/" >$LOG

echo ${GENERIC}/etc/* ${BASEDIR}/etc
cp -pr ${GENERIC}/etc/* ${BASEDIR}/etc
# adapt /etc/system
cat ${BASEDIR}/etc/system.add-on >> ${BASEDIR}/etc/system
rm ${BASEDIR}/etc/system.add-on
cp -pr ${GENERIC}/kernel/* ${BASEDIR}/kernel
cp -pr ${GENERIC}/opt/* ${BASEDIR}/opt
cp -pr ${GENERIC}/usr/* ${BASEDIR}/usr
cp -pr ${GENERIC}/var/* ${BASEDIR}/var
cp -pr ${GENERIC}/home/* ${BASEDIR}/home
cp -pr ${GENERIC}/.ssh/* ${BASEDIR}/.ssh
cp -p ${GENERIC}/.* ${BASEDIR}/
# disable serial login prompt
pmadm -d -p zsmon -s ttya
pmadm -d -p zsmon -s ttyb

rm -f ${BASEDIR}/etc/inet/inetd.conf
rm -f ${BASEDIR}/etc/rc2.d/S72inetsvc
ln -s ${BASEDIR}/etc/init.d/newinetsvc ${BASEDIR}/etc/rc2.d/S72inetsvc
rm -f ${BASEDIR}/etc/dfs/dfstab
mv ${BASEDIR}/etc/rc2.d/S73cachefs.daemon \
        ${BASEDIR}/etc/rc2.d/_S73cachefs.daemon
mv ${BASEDIR}/etc/rc2.d/S93cacheos.finish \
        ${BASEDIR}/etc/rc2.d/_S93cacheos.finish
mkdir -p ${BASEDIR}/var/core
chown root:root ${BASEDIR}/var/core
chmod 700 ${BASEDIR}/var/core
cd ${BASEDIR}/etc/default
awk ´/SYSLOG_FAILED_LOGINS=/ \
```

```
        { $1 = "SYSLOG_FAILED_LOGINS=0" }; \
        { print }´ login >login.new
mv ${BASEDIR}/etc/default/login.new ${BASEDIR}/etc/default/login
chown root:sys ${BASEDIR}/etc/default/login
chmod 444 ${BASEDIR}/etc/default/login
chmod 400 ${BASEDIR}/var/spool/cron/crontabs/*
eeprom oem-banner="Authorized uses only”

echo "NG-cp-rm-files.fin Finished." >>$LOG
```

B.5.2 NG-delete-unneeded-FCarch.fin

```
#!/bin/sh
LOG=/a/var/sadm/system/logs/NG-delete-unneeded-FCarch.log
NOASK=${SI_CONFIG_DIR}/files/software/noask

PACKAGES="SUNWadmr SUNWmdi SUNWmdix SUNWnamow SUNWluxdx SUNWluxop
SUNWluxox SUNWpcelx SUNWpcmci SUNWpcmcu SUNWpcmcx SUNWpcmem SUNWpcser
SUNWpsdpr SUNWnisr SUNWnisu SUNWcg6 SUNWcg6x SUNWdfb SUNWauda SUNWaudd
SUNWauddx SUNWm64 SUNWm64x SUNWrmodu SUNWsndmr SUNWsndmu SUNWtleux
SUNWwsr2 SUNWuaud SUNWuaudx SUNWudf SUNWudfr SUNWudfrx SUNWusb SUNWusbx
SUNWatfsr SUNWatfsu SUNWpl5u SUNWsolnm SUNWxwdv SUNWxwdvx SUNWxwmod
SUNWxwmox SUNWftpr SUNWftpu SUNWi15cs SUNWi1cs SUNWkey SUNWdtcor SUNWged"

echo "NG-delete-unneeded-FCarch.fin Started." >$LOG

for i in $PACKAGES
do
      echo "Removing unnecessary package $i"
      echo "Removing unnecessary package $i" >>$LOG
        echo y | pkgrm -a ${NOASK} -R ${BASEDIR} $i
done

echo "NG-delete-unneeded-FCarch.fin Finished." >$LOG
```

B.5.3 NG-ods421base.fin

```
#!/bin/sh
LOG=/a/var/sadm/system/logs/NG-ods421base.log
ODSPRODUCT=${SI_CONFIG_DIR}/files/software/DiskSuite-current
PATCHNUM=108693-14
ODSPATCH=${SI_CONFIG_DIR}/files/software/DiskSuite_4.2.1/${PATCHNUM}

echo "Installing SUNWmdr..."
echo "Installing SUNWmdr..." >$LOG
pkgadd -a ${ODSPRODUCT}/../noask \
-d ${ODSPRODUCT} \
-R ${BASEDIR} SUNWmdr

echo "Installing SUNWmdu..." >>$LOG
pkgadd -a ${ODSPRODUCT}/../noask \
-d ${ODSPRODUCT} \
-R ${BASEDIR} SUNWmdu

echo "Installing SUNWmdx..." >>$LOG
pkgadd -a ${ODSPRODUCT}/../noask \
-d ${ODSPRODUCT} \
-R ${BASEDIR} SUNWmdx

echo "Patching DiskSuite" >>$LOG
/usr/sbin/patchadd -R ${BASEDIR} ${ODSPATCH} >>$LOG

# ------------------------------------------------------ #
# Copy the automatic mirroring script to /etc/rc3.d so   #
# that it runs after reboot                              #
# ------------------------------------------------------ #
```

```
echo "Adding mirror script to perform mirroring at reboot" >>$LOG
cp -p ${ODSPRODUCT}/S99Mirror ${BASEDIR}/etc/rc3.d >>$LOG
echo "NG-ods421base.fin Finished. See dometa.log for more details."
>>$LOG

echo "NG-ods421base.fin Finished. See dometa.log for more details."
```

B.5.4 NG-openssh.fin

```
LOG=/a/var/sadm/system/logs/NG-openssh.log
BASEDIR="/a"
OPENSSH=${SI_CONFIG_DIR}/files/software/openssh-current
echo "Installing Myopenssh Package" >$LOG
echo "Installing Myopenssh Package"
echo all | /usr/sbin/pkgadd -v -a ${OPENSSH}/../noask \
-R ${BASEDIR} \
-d ${OPENSSH}/Myopenssh.pkg >>$LOG
echo "Installing patches..." >>$LOG
/usr/sbin/patchadd -R ${BASEDIR}  ${OPENSSH}/112438-02 >>$LOG

echo "Adding user sshd for privilege separation"
mkdir -p $BASEDIR/var/empty
chown root:sys $BASEDIR/var/empty
chmod 755 $BASEDIR/var/empty
echo "sshd:x:5432:sshd" >>$BASEDIR/etc/group
echo "sshd:x:5432:5432:ssh privsep:/var/empty:/sbin/noshell" \
       >>$BASEDIR/etc/passwd
echo " sshd:NP:6445::::::" >>$BASEDIR/etc/shadow

echo "Adding user sans for administration"
echo "sans:x:5433:1:Sans User:/home/sans:/bin/bash" >>$BASEDIR/etc/passwd

echo "Making sym link for ssh and scp" >>$LOG
/usr/bin/ln -s /usr/local/bin/ssh /usr/bin/ssh
/usr/bin/ln -s /usr/local/bin/scp /usr/bin/scp

echo "Myopenssh done\n\n"
echo "NG-openssh.fin Finished." >>$LOG
```

B.5.5 install-patches.fin

```
#!/bin/sh -x
#
# This script is responsible for installing a Sun Recommended
# and Security Patch Cluster from ${BASEDIR}/${PATCH_DIR}.

errorCondition=0

BASEDIR="/a"
PATCH_SERV_DIR=""
PATCH_DIR="/mnt"
OE_VER="`uname -r`"

mount -F nfs -o ro <JS-IP>:/export/install/jumpstart/files
${BASEDIR}/${PATCH_DIR}

case ${OE_VER} in
   5.8)
      PATCH_SERV_DIR="Patches/8_Recommended"
      ;;

   5.7)
      PATCH_SERV_DIR="Patches/7_Recommended"
      ;;
```

```
   *)
      errorCondition=1
      ;;
esac

if [ ${errorCondition} = 0 ]; then
   if [ ! -d ${BASEDIR}/${PATCH_DIR} ]; then
      echo "The directory, ${PATCH_DIR}, does not exist."
   else
        /usr/sbin/patchadd -d -R ${BASEDIR} -M \
        ${BASEDIR}/${PATCH_DIR}/${PATCH_SERV_DIR} patch_order
   fi
fi
```

B.5.6 NG-config-nics.fin

```
# !/bin/sh
DIR="/a/etc"
LOG=/a/var/sadm/system/logs/NG-config-nics.log
hostname=`hostname`

C=`echo $hostname | /usr/bin/sed -ne 's/c//' -e 's/f/ /p' | \
/usr/bin/awk '{print $1}'` # Clusternummer

F=`echo $hostname | /usr/bin/sed -ne 's/c//' -e 's/f/ /p' | \
/usr/bin/awk '{print $2}'` # Firewall

echo "Clusternr: $C   Nodenr: $F" >$LOG

# ------------------------------------------------------ #
# Define number of qfe's                                 #
# ------------------------------------------------------ #

qfenr=`sysdef -d | grep qfe | wc -l`
qfenr=`expr $qfenr - 1`
echo "Number of qfe on this box is: $qfenr" >>$LOG

# ------------------------------------------------------ #
# Netmasks file
# ------------------------------------------------------ #

rm -rf $DIR/netmasks.add 2> /dev/null
touch $DIR/netmasks.add
N=255.255.255.0

# ------------------------------------------------------ #
# Hosts file                                             #
# ------------------------------------------------------ #

rm -rf $DIR/hosts.add 2> /dev/null
touch $DIR/hosts.add
echo "# Hostsfile fuer `hostname`" > $DIR/hosts.add


# ------------------------------------------------------ #
# eri Configuration                                      #
# ------------------------------------------------------ #

IPn=<Mgmt IP-@>
IPe=`expr 10 \* $C + $F`
IP=$IPn.$IPe
ifconfig eri0 plumb up
ifconfig eri0 $IP netmask $N

# ------------------------------------------------------ #
# qfe Configuration                                      #
# ------------------------------------------------------ #
```

```
X=0
while [ $X -le $qfenr ]
do
        echo "$hostname-qe$X" > $DIR/hostname.qfe$X
        echo "c$C-qe$X" > $DIR/hostname.qfe$X:1
        IPa=192
        IPb=168
        IPc=`expr 253 - $X`
        IPd=`expr 20 \* $C + $F - 20 + 1`
        IPe=`expr 20 \* $C - 20 + 1`
        IPn=$IPa.$IPb.$IPc
        IP=$IPn.$IPd
        IP1=$IPn.$IPe
        echo qfe$X $IP
        echo qfe$X:1 $IP1

#       Interfaceconfiguration
        ifconfig qfe$X plumb up
        ifconfig qfe$X $IP netmask 255.255.255.0
        ifconfig qfe$X:1 plumb up
        ifconfig qfe$X:1 $IP1 netmask 255.255.255.0

#       File generation
        echo "$IPn.0\t$N\t#c$C-qe$X" >> $DIR/netmasks.add
        Y=`expr $X + 1`
        echo "## DMZ\t\tsbif$Y" >> $DIR/hosts.add
        IPe=`expr 20 \* $C - 20 + 1`
        IP=$IPn.$IPe

        echo "$IP\t\tc$C-qe$X  \t\tc$C-qe$X.sans.net" >> \
        $DIR/hosts.add

        IPe=`expr 20 \* $C - 20 + 2`
        IP=$IPn.$IPe

        echo "$IP\t\tc$C""f1-qe$X\t\tc$C""f1-qe$X.sans.net" >> \
        $DIR/hosts.add

        IPe=`expr 20 \* $C - 20 + 3`
        IP=$IPn.$IPe

        echo "$IP\t\tc$C""f2-qe$X\t\tc$C""f2-qe$X.sans.net" >> \
        $DIR/hosts.add

        echo  >> $DIR/hosts.add
        X=`expr $X + 1`
done

cat $DIR/netmasks.add >> /$DIR/netmasks
cat $DIR/hosts.add >> $DIR/hosts
rm -rf $DIR/netmasks.add 2> /dev/null
rm -rf $DIR/hosts.add 2> /dev/null
echo "All NICs and hostfiles are installed." >>$LOG
echo "NG-config-nics.fin Finished." >>$LOG
```

B.5.7 NG-sbfc.fin

```
#!/bin/sh
LOG=/a/var/sadm/system/logs/NG-sbfc.log
SBFCPRODUCT=${SI_CONFIG_DIR}/files/software
SBFCINST=/opt/fullcluster

# ------------------------------------------------------ #
# Copy the script to /etc/rc3.d directory, where it will  #
# run on reboot                                           #
# ------------------------------------------------------ #
```

```
echo "Adding script to install StoneBeat Fullcluster at reboot"
echo "Adding script to install StoneBeat Fullcluster at reboot" >$LOG

cp -p ${SBFCPRODUCT}/S90SBFCInstall ${BASEDIR}/etc/rc3.d

echo "NG-sbfc.fin Finished. See S90SBFCInstall.log for more details."
>>$LOG

echo "NG-sbfc.fin Finished. See S90SBFCInstall.log for more details."
```

B.5.8 NG-fw1-node.fin

```
#!/bin/sh
LOG=/a/var/sadm/system/logs/NG-fw1-node.fin
CHKPRODUCT=${SI_CONFIG_DIR}/files/software/Checkpoint-current

# ------------------------------------------------------ #
# Copy the script to /etc/rc3.d directory and run it     #
# after reboot.                                          #
# ------------------------------------------------------ #

echo "Adding script to install Checkpoint1 Firewall Base at reboot"
echo "Adding script to install Checkpoint1 Firewall Base at reboot" >$LOG

cp -p ${CHKPRODUCT}/S89FirewallInstall ${BASEDIR}/etc/rc3.d >$LOG

echo "NG-fw1-node.fin Finished. See further S89FirewallInstall.log."
>>$LOG
```

B.5.9 NG-tools.fin

```
#!/bin/sh
LOG=/a/var/sadm/system/logs/NG-tools.fin
TOOLSPRODUCT=${SI_CONFIG_DIR}/files/software/Tools-current

# ------------------------------------------------------ #
# Copy the script to /etc/rc3.d directory and run it     #
# after reboot.                                          #
# ------------------------------------------------------ #

echo "Adding script to install Tools Base at reboot"
echo "Adding script to install Tools Base at reboot" >$LOG

cp -p ${TOOLSRODUCT}/S91ToolsInstall ${BASEDIR}/etc/rc3.d >$LOG

echo "NG-tools.fin Finished. See further S91ToolsInstall.log." >>$LOG
```

B.5.10 NG-dragon-hids.fin

```
#!/bin/sh
LOG=/a/var/sadm/system/logs/NG-dragon-hids.fin
HIDSPRODUCT=${SI_CONFIG_DIR}/files/software/Dragon-current

# ------------------------------------------------------ #
# Copy the script to /etc/rc3.d directory and run it     #
# after reboot.                                          #
# ------------------------------------------------------ #

echo "Adding script to install Dragon Base at reboot"
echo "Adding script to install Dragon Base at reboot" >$LOG

cp -p ${HIDSPRODUCT}/S92DragonHIDSInstall ${BASEDIR}/etc/rc3.d >$LOG

echo "NG-dragon-hids.fin Finished. See further S92DragonHIDSInstall.log."
>>$LOG
```

B.5.11 NG-jass.fin

```
BASEDIR="/a"
SOFTWARE=${SI_CONFIG_DIR}/files/software
LOG=/a/var/sadm/system/logs/NG-jass.log

echo jass.fin > $LOG
echo "Installing SUNWjass Package"
echo "Installing SUNWjass Package" >>$LOG

# create dir manually, so pkgadd will not ask
mkdir -p $BASEDIR/opt/SUNWjass

# install package
echo all | /usr/sbin/pkgadd -v -a ${SOFTWARE}/noask \
-R ${BASEDIR} \
-d ${SOFTWARE}/SUNWjass-current.pkg >>$LOG
echo "SUNWjass done\n\n"
echo "SUNWjass done\n\n" >>$LOG

# copy MD5 and fixmodes package
cp -pr ${SOFTWARE}/SUNBEmd5.pkg ${BASEDIR}/opt/SUNWjass
cp -pr ${SOFTWARE}/SUNBEfixm.pkg ${BASEDIR}/opt/SUNWjass

# run jass
echo "Start jass"
echo "Start jass" >>$LOG
$BASEDIR/opt/SUNWjass/jass-execute -d undoable-hardening.driver >>$LOG
echo "Finish jass"
echo "Finish jass" >>$LOG

# undo disable-keyboard-abort.fin
cp $BASEDIR/etc/default/kbd.JASS* $BASEDIR/etc/default/kbd
echo "NG-jass.fin Finished." >>$LOG
```

B.6 Jumpstart GUI

B.6.1 /export/install/jumpstart/setup-client

```
#!/bin/ksh
# ------------------------------------------------------ #
# Define the root of the JASS installation               #
# ------------------------------------------------------ #

JASS_HOME_DIR=/export/install/jumpstart
JUMPSTART_SERVER=<IP>

# ------------------------------------------------------ #
# Read a non blank value from the user                   #
# ------------------------------------------------------ #

function read_non_blank {
        PROMPT="$1"
        REPLY=""
        while [ -z "$REPLY" ]
        do
                print "$PROMPT : \c"
                read REPLY
        done
}

# ------------------------------------------------------ #
# Read a defaulted value from the user                   #
# ------------------------------------------------------ #
```

```
function read_default {
        PROMPT="$1"
        DEFAULT="$2"
        REPLY=""
        print "$PROMPT [$DEFAULT]: \c"
        read REPLY
        [ -z "$REPLY" ] && REPLY="$DEFAULT"
}

# ------------------------------------------------------ #
# Get the Solaris version. Select from the available ones #
# ------------------------------------------------------ #

function read_solaris_version {
        OS_VERSION=""
        while [ -z "$OS_VERSION" ] || [ ! -d \
           "$JASS_HOME_DIR/../$OS_VERSION" ]
        do
                print "Available Operating Systems:"
                print
                for FILE in `ls $JASS_HOME_DIR/..`
                do
                        basename $FILE
                done
                print
                print "Enter OS Version: \c"
                read OS_VERSION
        done
}

# ------------------------------------------------------ #
# Get the system arch. Select from the available ones.    #
# ------------------------------------------------------ #

function read_architecture {
        # Single Architecture at the moment
        ARCH="sun4u"
}

# ------------------------------------------------------ #
# Get the profile. Select from the available ones         #
# ------------------------------------------------------ #

function read_profile {
        PROFILE=""
        while [ -z "$PROFILE" ] || [ ! -f \
           "$JASS_HOME_DIR/profiles/$PROFILE.profile" ]
        do
                print "Available profiles:"
                print
                for FILE in `ls $JASS_HOME_DIR/profiles/*.profile \
                   | cut -d. -f1`
                do
                        basename $FILE
                done
                print
                print "Enter profile: \c"
                read PROFILE
        done
}

# ------------------------------------------------------ #
# Ask a yes no question                                   #
# ------------------------------------------------------ #
```

```
function yesno {
        PROMPT="$1"
        REPLY=""
        while [ "$REPLY" != 'Y' ] && [ "$REPLY" != 'N' ]
        do
                print "$PROMPT [Y/N]: \c"
                read $REPLY
                REPLY=`echo $REPLY | tr '[:lower:]' '[:upper:]'\n`
        done
}
# ------------------------------------------------------ #
# Jumpstart server configuration                         #
#                                                        #
# Prompt for the client values                           #
# We want: hostname, ip address, ethernet address, os    #
# version, architecture and system type.                 #
# ------------------------------------------------------ #

clear
read_non_blank "Hostname"
HOSTNAME=$REPLY
#read_non_blank "IP address"
#IP_ADDRESS=$REPLY
C=`echo $HOSTNAME | sed -ne 's/c//' -e 's/f/ /p' | awk '{print $1}'`
#Clusternumber
F=`echo $HOSTNAME | sed -ne 's/c//' -e 's/f/ /p' | awk '{print $2}'`
# Firewallnodenumber
IPn=<Mgmt IP-@>
IPe=`expr 10 \* $C + $F`
IP_ADDRESS=$IPn.$IPe
print "IP address : $IP_ADDRESS"
read_non_blank "Ethernet Address"
ETHERNET=$REPLY

read_solaris_version
read_architecture
read_profile

read_default "Jumpstart Server" "$JUMPSTART_SERVER"
JUMPSTART_SERVER=$REPLY

# ------------------------------------------------------ #
# Confirm the details                                    #
# ------------------------------------------------------ #

print
print "Jumpstart host configuration"
print
print "Hostname     : $HOSTNAME"
print "IP address   : $IP_ADDRESS"
print "Ethernet     : $ETHERNET"
print "OS Version   : $OS_VERSION"
print "Architecture : $ARCH"
print "Profile      : $PROFILE"
print "JS Server    : $JUMPSTART_SERVER"
print
yesno "Confirm configuration"

if [ $REPLY = "N" ] ; then
        exit
fi

# ------------------------------------------------------ #
# All values present and correct update the files.       #
```

```
#                                                              #
# Check to see if the client already exists. If it does        #
# remove it first.                                             #
# ------------------------------------------------------------ #

if [ -f /etc/bootparams ] ; then
        if [ -n "`grep $HOSTNAME /etc/bootparams`" ] ; then
                $JASS_HOME_DIR/rm-client $HOSTNAME
        fi
fi

# ------------------------------------------------------------ #
# Edit the hosts file                                          #
# ------------------------------------------------------------ #

cp -p /etc/hosts /etc/hosts.bak
if [ -n "`grep $HOSTNAME /etc/hosts`" ] ; then
        grep -v $HOSTNAME /etc/hosts > /tmp/hosts.tmp
        cp /tmp/hosts.tmp /etc/hosts
        rm /tmp/hosts.tmp
fi
echo "$IP_ADDRESS       $HOSTNAME" >> /etc/hosts

# ------------------------------------------------------------ #
# Edit the ethers file                                         #
# ------------------------------------------------------------ #
#[ -f /etc/ethers ] && cp -p /etc/ethers /etc/ethers.bak
#if [ -n "`grep $HOSTNAME /etc/ethers`" ] ; then
#        grep -v $HOSTNAME /etc/ethers > /tmp/ethers.tmp
#        cp /tmp/ethers.tmp /etc/ethers
#        rm /tmp/ethers.tmp
#fi
#echo "$ETHERNET $HOSTNAME" >> /etc/ethers

# ------------------------------------------------------------ #
# Update the rules file and check it                           #
# ------------------------------------------------------------ #

[ -f $JASS_HOME_DIR/rules ] && cp -p $JASS_HOME_DIR/rules \
$JASS_HOME_DIR/rules.bak

if [ -n "`grep $HOSTNAME $JASS_HOME_DIR/rules`" ] ; then
        grep -v $HOSTNAME $JASS_HOME_DIR/rules > /tmp/rules.tmp
        cp /tmp/rules.tmp $JASS_HOME_DIR/rules
        rm /tmp/rules.tmp
fi

echo "hostname $HOSTNAME        -       profiles/${PROFILE}.profile \
drivers/${PROFILE}.driver" >> $JASS_HOME_DIR/rules
$JASS_HOME_DIR/check


# ------------------------------------------------------------ #
# Add the install client                                       #
# ------------------------------------------------------------ #

SOL8HOME=/export/install/Solaris_8_HW_02.02
cd $SOL8HOME/Solaris_8/Tools
./add_install_client -e $ETHERNET -s $JUMPSTART_SERVER:$SOL8HOME \
-c $JUMPSTART_SERVER:$JASS_HOME_DIR -p
$JUMPSTART_SERVER:$JASS_HOME_DIR/sysidcfg/Solaris_8_HW_02.02 \
$HOSTNAME $ARCH
```

APPENDIX C – CONFIG FILES

C.1 Disk Mirroring

C.1.1 /export/install/jumpstart/files/tmp/ODS/dometa-ODS421-SUNW,Sun-Fire-280R

```
#!/bin/ksh
DATE=$(date '+%Y%m%d')
LOG=/var/sadm/system/logs/dometa.log

# copy VTOC from primary to secondary device
prtvtoc /dev/dsk/c1t0d0s0 | fmthard -s - /dev/rdsk/c1t1d0s0 >$LOG

# ------------------------------------------------------ #
# Modify /etc/lvm/md.tab                                 #
# ------------------------------------------------------ #

cp /etc/lvm/md.tab /etc/lvm/md.tab.${DATE}
cat >> /etc/lvm/md.tab <<-EOF_FOE

        #
        #
        # -------------
        # MetaDB
        mddb01 -c 3 /dev/dsk/c1t0d0s4 /dev/dsk/c1t1d0s4
        mddb02 -c 1 /dev/dsk/c1t1d0s6

        # mirror for root
        d10 1 1 /dev/dsk/c1t0d0s0
        d0 -m d10
        d20 1 1 /dev/dsk/c1t1d0s0

        # mirror for swap
        d11 1 1 /dev/dsk/c1t0d0s1
        d1 -m d11
        d21 1 1 /dev/dsk/c1t1d0s1

        # mirror for var
        d13 1 1 /dev/dsk/c1t0d0s3
        d3 -m d13
        d23 1 1 /dev/dsk/c1t1d0s3

        # mirror for opt
        d15 1 1 /dev/dsk/c1t0d0s5
        d5 -m d15
        d25 1 1 /dev/dsk/c1t1d0s5

EOF_FOE

# ------------------------------------------------------ #
# Make all the Disksuite Commands
# ------------------------------------------------------ #

# MetaDB
/usr/sbin/metadb -a -f mddb01 >>$LOG
/usr/sbin/metadb -a -f mddb02 >>$LOG

# /
/usr/sbin/metainit /dev/md/dsk/d20 >>$LOG
/usr/sbin/metainit -f /dev/md/dsk/d10 >>$LOG
/usr/sbin/metainit /dev/md/dsk/d0 >>$LOG
/usr/sbin/metaroot /dev/md/dsk/d0 >>$LOG
```

```
#
# /var
/usr/sbin/metainit /dev/md/dsk/d23 >>$LOG
/usr/sbin/metainit -f /dev/md/dsk/d13 >>$LOG
/usr/sbin/metainit /dev/md/dsk/d3 >>$LOG
#
#
# swap
/usr/sbin/metainit /dev/md/dsk/d21 >>$LOG
/usr/sbin/metainit -f /dev/md/dsk/d11 >>$LOG
/usr/sbin/metainit /dev/md/dsk/d1 >>$LOG

#
# /opt
/usr/sbin/metainit /dev/md/dsk/d25 >>$LOG
/usr/sbin/metainit -f /dev/md/dsk/d15 >>$LOG
/usr/sbin/metainit /dev/md/dsk/d5 >>$LOG

# Working with /etc/vstab
cp /etc/vfstab /etc/vfstab.${DATE}
cat > /etc/vfstab <<-EOF_FOE
        #device        device    mount         FS      fsck     mount    mount
        #to mount      to fsck   point         type    pass     at boot options
        #
        fd             -         /dev/fd       fd      -        no       -

        /proc          -         /proc         proc    -        no       -

        /dev/md/dsk/d1 -         -             swap    -        no       -
        /dev/md/dsk/d0 /dev/md/rdsk/d0 /       ufs     1        no       logging
        /dev/md/dsk/d3 /dev/md/rdsk/d3 /var    ufs     1        no       logging
        /dev/md/dsk/d5 /dev/md/rdsk/d5 /opt    ufs     2        yes      logging
        swap           -         /tmp          tmpfs   -        yes      -

EOF_FOE
```

C.1.2 /export/install/jumpstart/files/software/DiskSuite-current/S99Mirror

```
#!/bin/sh
HARDIMP=`uname -i`

if [ ! -x /var/tmp/ODS/dometa-ODS421-$HARDIMP ]; then
        echo "Mirroring script does not exist. It was somehow not \
                copied during NG-cp-rm-files.fin."
        exit 1
else
        echo "Setting up Disk Mirroring......."
        /var/tmp/ODS/dometa-ODS421-$HARDIMP
        rm -rf /var/tmp/ODS
fi

# ------------------------------------------------------ #
# We have setup the 1st submirror; nuke ourselves        #
# ------------------------------------------------------ #
rm -f /etc/rc3.d/S99Mirror
```

C.2 JASS

C.2.1 undoable-hardening.driver

Details on the finish scripts can be found in [2].

```
#!/bin/sh
DIR="`/bin/dirname $0`"
export DIR

. ${DIR}/driver.init

JASS_FILES="
                /etc/dt/config/Xaccess
                /etc/init.d/inetsvc
                /etc/init.d/nddconfig
                /etc/init.d/set-tmp-permissions
                /etc/issue
                /etc/motd
                /etc/rc2.d/S00set-tmp-permissions
                /etc/rc2.d/S07set-tmp-permissions
                /etc/rc2.d/S70nddconfig
"

JASS_SCRIPTS="
                disable-keyboard-abort.fin
                disable-keyserv-uid-nobody.fin
                disable-ldap-client.fin
                disable-lp.fin
                disable-nfs-client.fin
                disable-nfs-server.fin
                disable-nscd-caching.fin
                disable-preserve.fin
                disable-power-mgmt.fin
                disable-rhosts.fin
                disable-rpc.fin
                disable-sendmail.fin
                disable-syslogd-listen.fin
                disable-system-accounts.fin
                disable-uucp.fin
                enable-coreadm.fin
                enable-ftp-syslog.fin
                install-at-allow.fin
                install-newaliases.fin
                install-sadmind-options.fin
                remove-unneeded-accounts.fin
                set-login-retries.fin
                set-root-group.fin
                set-tmpfs-limit.fin
                set-user-password-reqs.fin
                set-user-umask.fin
                update-at-deny.fin
                update-cron-allow.fin
                update-cron-deny.fin
                update-cron-log-size.fin
                install-md5.fin
                install-fix-modes.fin
                install-strong-permission.fin
                print-rhosts.fin
                print-sgid-files.fin
                print-suid-files.fin
                print-unowned-objects.fin
                print-world-writables-objects.fin
"

. ${DIR}/driver.run
```

C.2.2 /etc/init.d/nddconfig

```
#  The latest version of this script is available from the Blueprints
#  Online tools area at:
#
#       http://www.sun.com/blueprints/tools/
```

C.3 Stonebeat

C.3.1 /export/install/jumpstart/files/software/S90SBFCInstall

```
#!/bin/sh
MNT_DIR="/mnt"
OE_VER="`uname -r`"
PKGADD=/usr/sbin/pkgadd
LOG=/var/sadm/system/logs/S90SBFCInstall.log

# ------------------------------------------------------ #
# Mount up the patch directory                           #
# ------------------------------------------------------ #

mount -F nfs -o ro <JS IP-@>:/export/install/jumpstart/files ${MNT_DIR}

# ------------------------------------------------------ #
# Install StoneBeat Fullcluster packages                 #
# ------------------------------------------------------ #

echo "Installing StoneBeat Fullcluster..."
echo "Installing StoneBeat Fullcluster..." >$LOG
echo y | $PKGADD -d /mnt/software/SBFC-current -a /mnt/software/noask
SBFCbase SBFCdrv SBFCmod >> $LOG 2>&1

echo "Installing StoneBeat SNMPD..."
echo y | $PKGADD -d /mnt/software/SBFCsnmpd-current \
-a /mnt/software/noask SBFCsnmp >> $LOG 2>&1

sleep 5

# ------------------------------------------------------ #
# copy snmpd.conf                                        #
# ------------------------------------------------------ #

cp /mnt/software/tester-conf/snmpd.conf /opt/stonebeat/snmp/etc
echo "Copied snmpd.conf..."
echo "Copied snmpd.conf..." >>$LOG

# ------------------------------------------------------ #
# rewrite checklist file                                 #
# if the line contains the string SBIFID                 #
# rewrite the line for every qfe interface found         #
# copy files from jumphost tester-conf-directory to      #
# /opt/fullcluster/etc:                                  #
# ------------------------------------------------------ #

ETC=/opt/fullcluster/etc
for i in alert.sh online.sh offline.sh checklist; do
    cp /mnt/software/tester-conf/$i $ETC >>$LOG 2>&1
    echo "Copied $i ..."
done

# ------------------------------------------------------ #
# backup original checklist file                         #
# ------------------------------------------------------ #

mv $ETC/checklist $ETC/checklist.orig >>$LOG
```

```
# ------------------------------------------------------ #
# check how many qfe interfaces we have                  #
# ------------------------------------------------------ #

QFENR=`sysdef -d |grep qfe |wc -l`
QFENR=`expr $QFENR + 0`

# ------------------------------------------------------ #
# read checklist list file again from jumphost, line for line
# ------------------------------------------------------ #

cat /mnt/software/tester-conf/checklist |while read line; do
    # if the line contains the string SBIFID repeat the line
    # for every qfe

    if [ "`echo $line |grep SBIFID`" ]; then
        if [ "$QFENR" -gt 0 ]; then
            # start with qfe2 instead of qfe1, because qfe0
            # & qfe1 are hearbeat intf's
            i=2 ; while [ $i -le "$QFENR" ]; do
                echo $line |sed 's/SBIFID/sbif'$i'/g' >>$ETC/checklist
                echo $line |sed 's/SBIFID/sbif'$i'/g' >>$LOG
                i=`expr $i + 1`
            done
        fi
    else
    # other lines just reprint unchanged
        echo $line >>$ETC/checklist
        echo $line >>$LOG
    fi
done

# ------------------------------------------------------ #
# Umount the patch directory                             #
# ------------------------------------------------------ #

umount /mnt

# ------------------------------------------------------ #
# and nuke ourselves!                                    #
# ------------------------------------------------------ #

rm -f /etc/rc3.d/S90SBFCInstall
echo "S90SBFCInstall Finished." >>$LOG
```


C.3.2 Node-1 sbfcconfig Detailed Installation Sequence

1. Generate Keys & Certificates
2. Configure This Node
3. Set Passphrase
4. Install License
5. Exit

Select Option: **1**

1. Create CA Key
2. Create CA Certificate
3. Create Module Key
4. Create Module Certificate
5. Create Client Key
6. Create Client Certificate
7. Back

Select Option: **1**

Generating random data
Please type random text.
.....
Thank you.

Generating DSA parameters
This may take some time
.....

Making key for CA

Enter the pass phrase for the CA key

Enter PEM pass phrase: **<passwd>**

Verifying password - Enter PEM pass phrase: **<passwd>**

CA key created successfully

1. Create CA Key
2. Create CA Certificate
3. Create Module Key
4. Create Module Certificate
5. Create Client Key
6. Create Client Certificate
7. Back

Select Option: **2**

Making certificate for CA

Enter the number of years to certify the certificate: **5**

Enter the pass phrase of the CA key

Enter PEM pass phrase: **<passwd>**

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]: **<input>**

State or Province Name (full name) []:**<input>**

Locality Name (eg, city) []:**<input>**

Organization Name (eg, company) []:**<input>**

Organizational Unit Name (eg, section) []:**<input>**

Common Name (eg, YOUR name) []:**<input>**

Email Address []:**<input>**

CA certificate created successfully

1. Create CA Key
2. Create CA Certificate
3. Create Module Key
4. Create Module Certificate
5. Create Client Key

6. Create Client Certificate
7. Back

Select Option: **3**

Generating Diffie-Hellman parameters
.....

Making key for module

Enter the pass phrase for the module key

Enter PEM pass phrase: **<passwd>**

Verifying password - Enter PEM pass phrase: **<passwd>**

Module key created successfully

1. Create CA Key
2. Create CA Certificate
3. Create Module Key
4. Create Module Certificate
5. Create Client Key
6. Create Client Certificate
7. Back

Select Option: **4**

Making certificate for module

Enter the number of years to certify the certificate: 5

Enter the pass phrase of the module key

Enter PEM pass phrase: **<passwd>**

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]: **<input>**

State or Province Name (full name) []:**<input>**

Locality Name (eg, city) []:**<input>**

Organization Name (eg, company) []:**<input>**

Organizational Unit Name (eg, section) []:**<input>**

Common Name (eg, YOUR name) []: **Node**

Email Address []:**<input>**

Enter the pass phrase of the CA key
Enter PEM pass phrase: **<passwd>**

Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName :PRINTABLE:"

stateOrProvinceName :PRINTABLE:''
localityName :PRINTABLE:''
organizationName :PRINTABLE:''
organizationalUnitName:PRINTABLE:''
commonName :PRINTABLE:'Node'
emailAddress :IA5STRING:''

Certificate is to be certified until Apr 14 17:35:20 2009 GMT (10950 days)
Sign the certificate? [y/n]:**y**

1 out of 1 certificate requests certified, commit? [y/n]**y**

Certificate created successfully

1. Create CA Key
2. Create CA Certificate
3. Create Module Key
4. Create Module Certificate
5. Create Client Key
6. Create Client Certificate
7. Back

Select Option: **5**

1. The client is GUI-based
2. The client is Command Line-based
3. Back

Select Option: **2**

Making key for sbfc

Enter the pass phrase for the sbfc key

Enter PEM pass phrase: **<passwd>**

Verifying password - Enter PEM pass phrase: **<passwd>**

sbfc key created successfully

1. Create CA Key
2. Create CA Certificate
3. Create Module Key
4. Create Module Certificate
5. Create Client Key
6. Create Client Certificate
7. Back

Select Option: **6**

Making certificate for sbfc

Enter the number of years to certify the certificate: **5**

Enter the pass phrase of the sbfc key
Enter PEM pass phrase: **<passwd>**

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]: **<input>**

State or Province Name (full name) []:**<input>**

Locality Name (eg, city) []:**<input>**

Organization Name (eg, company) []:**<input>**

Organizational Unit Name (eg, section) []:**<input>**

Common Name (eg, YOUR name) []: **CLI**

Email Address []:**<input>**

Enter the pass phrase of the CA key

Enter PEM pass phrase: **<passwd>**

Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName :PRINTABLE:''
stateOrProvinceName :PRINTABLE:''
localityName :PRINTABLE:''
organizationName :PRINTABLE:''
organizationalUnitName:PRINTABLE:''
commonName :PRINTABLE:'CLI'
emailAddress :IA5STRING:''
Certificate is to be certified until Apr 14 17:49:16 2009 GMT (10950 days)
Sign the certificate? [y/n]:**y**

1 out of 1 certificate requests certified, commit? [y/n]**y**

Certificate created successfully

1. Create CA Key
2. Create CA Certificate
3. Create Module Key
4. Create Module Certificate
5. Create Client Key
6. Create Client Certificate
7. Back

Select Option: **7**

1. Generate Keys & Certificates
2. Configure This Node
3. Set Passphrase
4. Install License
5. Exit

Select Option: **2**

1. Set Node ID [**1**]
2. Set Cluster ID [i.e. **13**]
3. Set Capacity [**(1) auto**]
4. Set Load Measurement Interval [**15**]
5. Set Start Up Mode [**(2) standby**]
6. Set Clustering Mode [**(1) balancing**]
7. Set Control IP address [**e.g. 192.168.1.2**]
8. Set Control Port [**tcp/3002**]
9. Configure Interfaces
10. Back

Select Option: **1, 2, 3, 4, 5, 6, 7, 8, 9**

- Configure primary heartbeat (HB-1)

1. Define Heartbeat Protocol Interface(s)
2. Define Operational Interface(s)
3. Back

Select Option: **1**

Currently available interfaces: (i.e. for c13f1)

1. eri0 NOT configured. Assigned to IP-1
2. qfe0 NOT configured. Assigned to IP-2
3. qfe1 NOT configured. Assigned to IP-3
4. qfe2 NOT configured. Assigned to IP-4
5. qfe3 NOT configured. Assigned to IP-5
6. qfe4 NOT configured. Assigned to IP-6
7. qfe5 NOT configured. Assigned to IP-7

Select interface: **2**

qfe0: MAC address is [0:3:ba:x:x:x] and IP address is [IP-1]
Do you want to update or remove this interface (y/N)? **N**
Enter a multicast MAC address: 1:0:x:x:x:x (1:0:plus IP-@ of node-1)
Do you want to assign control ip and port (y/N)? **n**

- Configure secondary heartbeat (HB-2) and control interface.

1. Define Heartbeat Protocol Interface(s)
2. Define Operational Interface(s)
3. Back

Select Option: **1**

Currently available interfaces:

1. eri0 NOT configured. Assigned to IP-1
2. qfe0 NOT configured. Assigned to IP-2
3. qfe1 NOT configured. Assigned to IP-3
4. qfe2 NOT configured. Assigned to IP-4
5. qfe3 NOT configured. Assigned to IP-5
6. qfe4 NOT configured. Assigned to IP-6
7. qfe5 NOT configured. Assigned to IP-7

Select interface: **1**

eri0: MAC address is [0:3:ba:x:x:x] and IP address is [IP-2]
Do you want to update or remove this interface (y/N)? **N**
Enter a multicast MAC address: 1:0:x:x:x:x (1:0:plus IP-@ of node-1)
Do you want to assign control ip and port (y/N)? **y**
Enter this host's control IP address: **IP-2**
Enter a control port number (1025 - 65535): **3002**

- Configure Operational Interfaces.

1. Define Heartbeat Protocol Interface(s)
2. Define Operational Interface(s)
3. Back

Select Option: **2**

Currently available interfaces:

1. eri0 NOT configured. Assigned to IP-1
2. qfe0 NOT configured. Assigned to IP-2
3. qfe1 NOT configured. Assigned to IP-3
4. qfe2 NOT configured. Assigned to IP-4
5. qfe3 NOT configured. Assigned to IP-5
6. qfe4 NOT configured. Assigned to IP-6
7. qfe5 NOT configured. Assigned to IP-7

Select interface: **3**

qfe1: MAC address is [0:3:ba:x:x:x] and IP address is [IP-3]
Do you want to update or remove this interface (y/N)? **n**
Do you want to use multicast support for this interface (y/N)? **y**
Enter a multicast IP or MAC address: 1:0:x:x:x:x (1:0:plus Cluster IP-@)
Enter the unicast cluster IP address(es): (Cluster IP-@)
using the following cluster address(es): IP-Cluster
Do you want to add more addresses (y/N)? **N**

- Install the other operational interfaces in the same way.
- Go back: **10**

1. Generate Keys & Certificates
2. Configure This Node
3. Set Passphrase
4. Install License
5. Exit

Select Option: **3**

Select Option: **4**

Select Option: **5**

- Deploy certificates: Copy the files below from $SBFCHOME/etc/cert/ to $SBFCHOME/etc
    - Modulecert.pem
    - Modulekey.pem
    - Dhparams.pem
    - Cacert.pem
    - Clients
    - Sbfccert.pem
    - Sbfckey.pem
- Run `sbfcpassphrase`
- Clean up unused interfaces. In /etc do
  `rm `ls | grep hostn | grep –v __ | grep –v sbif``

C.3.3 Slave Node sbfcconfig Detailed Installation Sequence

1. Generate Keys & Certificates
2. Configure This Node
3. Set Passphrase
4. Install License
5. Exit

Select Option: **2**

1. Set Node ID [**2**]
2. Set Cluster ID [i.e. **13**]
3. Set Capacity [**(1) auto**]
4. Set Load Measurement Interval [**15**]
5. Set Start Up Mode [**(2) standby**]
6. Set Clustering Mode [**(1) balancing**]
7. Set Control IP address [i.e. Node-IP]
8. Set Control Port [**tcp/3002**]
9. Configure Interfaces
10. Back

Select Option: **1, 2, 3, 4, 5, 6, 7, 8, 9**

- Configure primary heartbeat (HB-1)

1. Define Heartbeat Protocol Interface(s)
2. Define Operational Interface(s)
3. Back

Select Option: **1**

Currently available interfaces: (i.e. for c13f2)

1. eri0 NOT configured. Assigned to IP-1
2. qfe0 NOT configured. Assigned to IP-2
3. qfe1 NOT configured. Assigned to IP-3
4. qfe2 NOT configured. Assigned to IP-4
5. qfe3 NOT configured. Assigned to IP-5
6. qfe4 NOT configured. Assigned to IP-6
7. qfe5 NOT configured. Assigned to IP-7

Select interface: 2

qfe0: MAC address is [0:3:ba:x:x:x] and IP address is [IP-2]
Do you want to update or remove this interface (y/N)? N
Enter a multicast MAC address: 1:0:x:x:x:x (1:0:plus IP-@ of node-1)
Do you want to assign control ip and port (y/N)? **n**

- Configure secondary heartbeat (HB-2) and control interface.

1. Define Heartbeat Protocol Interface(s)
2. Define Operational Interface(s)
3. Back

Select Option: **1**

Currently available interfaces:

1. eri0 NOT configured. Assigned to IP-1
2. qfe0 NOT configured. Assigned to IP-2
3. qfe1 NOT configured. Assigned to IP-3
4. qfe2 NOT configured. Assigned to IP-4
5. qfe3 NOT configured. Assigned to IP-5
6. qfe4 NOT configured. Assigned to IP-6
7. qfe5 NOT configured. Assigned to IP-7

Select interface: **1**

eri0: MAC address is [0:3:x:x:x:x] and IP address is [IP-1]
Do you want to update or remove this interface (y/N)? **N**
Enter a multicast MAC address: 1:0:x:x:x:x (1:0:plus IP-@ of node-1)
Do you want to assign control ip and port (y/N)? **y**

Enter this host's control IP address: Node-IP
Enter a control port number (1025 - 65535): **3002**

- Configure Operational Interfaces.

1. Define Heartbeat Protocol Interface(s)
2. Define Operational Interface(s)
3. Back

Select Option: **2**

Currently available interfaces:

1. eri0 NOT configured. Assigned to IP-1
2. qfe0 NOT configured. Assigned to IP-2
3. qfe1 NOT configured. Assigned to IP-3
4. qfe2 NOT configured. Assigned to IP-4
5. qfe3 NOT configured. Assigned to IP-5
6. qfe4 NOT configured. Assigned to IP-6
7. qfe5 NOT configured. Assigned to IP-7

Select interface: **3**

qfe1: MAC address is [0:3:ba:x:x:x] and IP address is [IP-3]
Do you want to update or remove this interface (y/N)? **n**
Do you want to use multicast support for this interface (y/N)? **y**
Enter a multicast IP or MAC address: 1:0:x:x:x:x (1:0:plus Cluster IP-@)
Enter the unicast cluster IP address(es): (Cluster IP-@)
using the following cluster address(es): Cluster-IP
Do you want to add more addresses (y/N)? **N**

- Install the other operational interfaces in the same way.
- Go back: **10**

1. Generate Keys & Certificates
2. Configure This Node
3. Set Passphrase
4. Install License
5. Exit

Select Option: **3**

Select Option: **4**

Select Option: **5**

- Deploy certificates: Copy from Master Node-1 the files below from $SBFCHOME/etc/cert/ to $SBFCHOME/etc on the new installed slave Node.
    - Modulecert.pem
    - Modulekey.pem
    - Dhparams.pem
    - Cacert.pem
    - Clients
    - Sbfccert.pem
    - Sbfckey.pem
- Run sbfcpassphrase

```
Clean up unused interfaces. In /etc do
rm `ls | grep hostn | grep -v __ | grep -v sbif`
```

C.3.4 /opt/fullcluster/etc/metatest.sh

This script checks if the metadbs are in good health.

```
#!/usr/bin/bash
if /usr/sbin/metadb | grep W &> /dev/null
then exit 1 # harddisk crashed
else exit 0 # harddisk is running fine
fi
```

C.3.5 /opt/fullcluster/etc/stonebeat_ela.conf

```
ela_server     ip       <CP Mgmt Station IP-@>
ela_server     auth_port      18187

# For FireWall-1 NG FP1,FP2,FP3 and later
#
ela_server auth_type sslca

# $opsec_application_sic_name is the CN=node-ela...
opsec_sic_name "$opsec_application_sic_name"

# $ela_server_sic_name is the CN=mgmt,..
ela_server opsec_entity_sic_name "$ela_server_sic_name"
opsec_sslca_file "opsec.p12"
```

C.3.6 /opt/fullcluster/etc/alert.sh

```
#!/sbin/sh
SBFCHOME="/opt/fullcluster"
export SBFCHOME

SBFC_SNMP_HOME=/opt/stonebeat/snmp/
export SBFC_SNMP_HOME

MIBS=ALL
MIBDIRS=$SBFC_SNMP_HOME/etc/mibs
export MIBS MIBDIRS

PATH=${PATH}:$SBFC_SNMP_HOME/bin:$SBFCHOME/bin
export PATH

FULLCLUSTER_OID=.1.3.6.1.4.1.1369.2.2

## Management Station parameters
FW_MNGT_STATION=<Mgmt IP-@>

## Cluster Member parameters : to be set!
FC_NODE_IP=<Node IP-@>
FC_NODE_NAME=`hostname`
COMMUNITY="<mypassword>"
CLUSTER_MEMBER=1

# set_cluster_member_offline
isnumeric () {
   [ $# -eq 1 ] || return -1

  case $1 in
        *[!1-2]*|"") return -1;;
        *) return $SUCCESS;;
  esac
}

# set_cluster_member_offline ()
```

```
set_cluster_member_offline ()
{
        if [ ! "`$SBFCHOME/bin/sbfc status | grep ["$CLUSTER_MEMBER"] \
                | awk '{print $3}'`" = "offline" ]
        then
         $SBFCHOME/bin/sbfc offline $CLUSTER_MEMBER 2> /dev/null \
          >/dev/null
        fi
}
# Main loop
echo "`date`: $*" >>/tmp/alert.tmp

case $2 in
        *nic-HB-and-*)
        snmptrap -v1 $FW_MNGT_STATION $COMMUNITY $FULLCLUSTER_OID \
         $FC_NODE_IP 6 3 ''\

         system.sysName.0 s "$FC_NODE_NAME" 2> /dev/null > /dev/null
        set_cluster_member_offline
        exit 0;;

        *nic-HB-or-*)
        snmptrap -v1 $FW_MNGT_STATION $COMMUNITY $FULLCLUSTER_OID \
          $FC_NODE_IP 6 3 ''\

         system.sysName.0 s "$FC_NODE_NAME" 2> /dev/null > /dev/null
        #set_cluster_member_offline
        exit 0;;

        *ext-meta-test*)
        snmptrap -v1 $FW_MNGT_STATION $COMMUNITY $FULLCLUSTER_OID \
          $FC_NODE_IP 6 3 ''\

         system.sysName.0 s "$FC_NODE_NAME" 2> /dev/null > /dev/null
        set_cluster_member_offline
        exit 0;;

        *fw-module-running-test*)

        snmptrap -v1 $FW_MNGT_STATION $COMMUNITY $FULLCLUSTER_OID \
          $FC_NODE_IP 6 3 ''\

         system.sysName.0 s "$FC_NODE_NAME" 2> /dev/null > /dev/null
        set_cluster_member_offline
        exit 0;;

        *fw-policy-test*)
        snmptrap -v1 $FW_MNGT_STATION $COMMUNITY $FULLCLUSTER_OID \
          $FC_NODE_IP 6 4 ''\

         system.sysName.0 s "$FC_NODE_NAME" 2> /dev/null > /dev/null
       set_cluster_member_offline
        exit 0;;

        *loadaverage-test*)
        snmptrap -v1 $FW_MNGT_STATION $COMMUNITY $FULLCLUSTER_OID \
         $FC_NODE_IP 6 5 ''\

          system.sysName.0 s "$FC_NODE_NAME" 2> /dev/null > /dev/null
        exit 0;;

        *processorusage-test*)
        snmptrap -v1 $FW_MNGT_STATION $COMMUNITY $FULLCLUSTER_OID \
          $FC_NODE_IP6 6 ''\

         system.sysName.0 s "$FC_NODE_NAME" 2> /dev/null > /dev/null
        exit 0;;
```

```
        *fwlog-test*)
        snmptrap -v1 $FW_MNGT_STATION $COMMUNITY $FULLCLUSTER_OID \
          $FC_NODE_IP 6 8 ''\
         system.sysName.0 s "$FC_NODE_NAME" 2> /dev/null > /dev/null
        exit 0;;
        *systemlog-test*)
         snmptrap -v1 $FW_MNGT_STATION $COMMUNITY $FULLCLUSTER_OID \
          $FC_NODE_IP 6 7 ''\
         system.sysName.0 s "$FC_NODE_NAME" 2> /dev/null > /dev/null
        exit 0;;
        *nic-linkstatus*)
        # Extract the interface number from the $2 parameter
        interface=`echo $2 | sed 's/[^0-9]*//g'`
        snmptrap -v1 $FW_MNGT_STATION $COMMUNITY $FULLCLUSTER_OID \
         $FC_NODE_IP 6 9 ''\
         system.sysName.0 s "$FC_NODE_NAME"\
         ".1.3.6.1.2.1.2.2.1.8.$interface" i 0\
         2>> /dev/null > /dev/null
        set_cluster_member_offline
        exit 0;;
        *nic-up*)
        # Extract the interface number from the $2 parameter
        interface=`echo $2 | sed 's/[^0-9]*//g'`
        snmptrap -v1 $FW_MNGT_STATION $COMMUNITY $FULLCLUSTER_OID \
         $FC_NODE_IP 6 10 ''\
         system.sysName.0 s "$FC_NODE_NAME"\
         ".1.3.6.1.2.1.2.2.1.8.$interface" i 0\
         2> /dev/null > /dev/null
        set_cluster_member_offline
        exit 0;;
esac
```

C.3.7 /opt/fullcluster/etc/checklist

```
# General configuration parameters                          #
alert-interval 30
boot-delay 60
status-delay 5
restart-delay 5
distribute on each node
# ONLINE tests                                              #
# Testing Disk Devices
#######
# Testing if enough space for logging is available -> min 500 MByte
systemlog-test 900 online alert 1 1 systemlog /var/adm 500000
# Testing if enough space for fw logging is available -> min 500 MByte
fwlog-test 900 online alert 1 1 fwlog 500000
# Testing Network Interfaces
#######
```

```
# Test if heartbeat interfaces are up and running

nic-HB-and-linkstatus 60 online alert (1 1 networkinterace-linkstatus
sbif0 and 1 1 networkinterface-linkstatus sbif1)

nic-HB-or-linkstatus 60 online alert (1 1 networkinterace-linkstatus
sbif0 or 1 1 networkinterface-linkstatus sbif1)

nic-HB-and-up 60 online alert (1 1 networkinterface-up sbif0 and 1 1
networkinterface-up sbif1)

nic-HB-or-up 60 online alert (1 1 networkinterface-up sbif0 or 1 1
networkinterface-up sbif1)

# Test if operational interface are up and running

nic-up 60 online alert 1 1 networkinterface-up SBIFID

nic-linkstatus 60 online alert 1 1 networkinterface-linkstatus SBIFID

# Testing Operating System Resources

#######

# Test if load average is acceptable

loadaverage-test 300 online alert 1 1 loadaverage 3

# Test if processor usage is acceptable

processorusage-test 300 online alert 1 1 processorusage 40 30

# Testing FireWall-1

#######

# Test if a firewall policy is loaded

fw-policy-test 3600 online alert 1 1 fw-policy-loaded *

# Test if firewall module is running

fw-module-running-test 30 online alert 1 1 fw-module-running

# External Tests

#######

# Test if mirroring is working fine or if a harddisk crashed

ext-meta-test 3600 online alert 1 1 /opt/fullcluster/etc/metatest.sh


# OFFLINE tests                                              #

# Testing Disk Devices

#######

# Testing if enough space for logging is available -> min 500 MByte

systemlog-test 900 offline alert 1 1 systemlog /var/adm 500000

# Testing if enough space for fw logging is available -> min 500 MByte

fwlog-test 900 offline alert 1 1 fwlog 500000

# Testing Network Interfaces

#######

# Test if heartbeat interfaces are up and running

nic-HB-and-linkstatus 60 offline alert (1 1 networkinterace-linkstatus
sbif0 and 1 1 networkinterface-linkstatus sbif1)
```

```
nic-HB-or-linkstatus 60 offline alert (1 1 networkinterace-linkstatus
sbif0 or 1 1 networkinterface-linkstatus sbif1)

nic-HB-and-up 60 offline alert (1 1 networkinterface-up sbif0 and 1 1
networkinterface-up sbif1)

nic-HB-or-up 60 offline alert (1 1 networkinterface-up sbif0 or 1 1
networkinterface-up sbif1)

# Test if operational interface are up and running %%TO DO%%

nic-up 60 offline alert 1 1 networkinterface-up SBIFID

nic-linkstatus 60 offline alert 1 1 networkinterface-linkstatus SBIFID

# Testing Operating System Resources

#######

# Test if load average is acceptable

loadaverage-test 300 offline alert 1 1 loadaverage 3

# Test if processor usage is acceptable

processorusage-test 300 offline alert 1 1 processorusage 40 30

# Testing FireWall-1

#######

# Test if a firewall policy is loaded

fw-policy-test 3600 offline alert 1 1 fw-policy-loaded *

# Test if firewall module is running

fw-module-running-test 30 offline alert 1 1 fw-module-running

# External Tests

#######

# Test if mirroring is working fine or if a harddisk crashed

ext-meta-test 3600 offline alert 1 1 /opt/fullcluster/etc/metatest.sh
```

C.3.8 /etc/opt/fullcluster/etc/offline.sh

```
#!/bin/sh

SBFCHOME="/opt/fullcluster"

export SBFCHOME

SBFC_SNMP_HOME=/opt/stonebeat/snmp/

export SBFC_SNMP_HOME

MIBS=ALL

MIBDIRS=$SBFC_SNMP_HOME/etc/mibs

export MIBS MIBDIRS

PATH=${PATH}:$SBFC_SNMP_HOME/bin

export PATH

SNMP_MANAGEMENT=<Mgmt IP-@>

FC_NODE_IP=<Node IP-@>

FC_NODE_NAME=`hostname`

COMMUNITY="public"

FULLCLUSTER_OID=.1.3.6.1.4.1.1369.2.2
```

```
TRAP_ID=1
FULLCLUSTER_MODULE_OID=1.3.6.1.4.1.1369.2.2.4

#echo "Da Cluster Member 2 is OFFLINE indahouse!"
snmptrap -v1 $SNMP_MANAGEMENT $COMMUNITY $FULLCLUSTER_OID $FC_NODE_NAME \
 6 $TRAP_ID ''\
 system.sysName.0 s "$FC_NODE_NAME" 2> /dev/null > /dev/null
```

C.3.9 /opt/fullcluster/online.sh

```
#!/bin/sh
SBFCHOME="/opt/fullcluster"
export SBFCHOME
SBFC_SNMP_HOME=/opt/stonebeat/snmp/
export SBFC_SNMP_HOME
MIBS=ALL
MIBDIRS=$SBFC_SNMP_HOME/etc/mibs
export MIBS MIBDIRS
PATH=${PATH}:$SBFC_SNMP_HOME/bin
export PATH
PATH=/opt/stonebeat/snmp/bin:$SBFCHOME/bin:$PATH
SNMP_MANAGEMENT=<Mgmt IP-@>
FC_NODE_IP=<Node IP-@>
FC_NODE_NAME=`hostname`
COMMUNITY="public"
FULLCLUSTER_OID=.1.3.6.1.4.1.1369.2.2
TRAP_ID=2
#echo "Da Cluster Member 2 is ONLINE indahouse!"
#echo "snmptrap -v1 $SNMP_MANAGEMENT $COMMUNITY $FULLCLUSTER_OID
$FC_NODE_NAME \
# 6 $TRAP_ID ''\
# system.sysName.0 s $FC_NODE_NAME"
snmptrap -v1 $SNMP_MANAGEMENT $COMMUNITY $FULLCLUSTER_OID $FC_NODE_NAME \
 6 $TRAP_ID ''\
 system.sysName.0 s "$FC_NODE_NAME" 2> /dev/null > /dev/null
```

C.3.10 /opt/stonebeat/snmp/etc/snmpd.conf

```
####
# First, map the community name (COMMUNITY) into a security name
# (local and mynetwork, depending on where the request is coming
# from):
#       sec.name  source                community
```

```
com2sec local     localhost             <community-1>
com2sec mngt      <Mgmt IP-@>       <community-2>

####
# Second, map the security names into group names:
#               sec.model  sec.name
group MyRWGroup v1      local
group MyRWGroup v2c     local
group MyRWGroup usm     local
group MyROGroup v1      mngt

####
# Third, create a view for us to let the groups have rights to:
#           incl/excl subtree                          mask
view all        included  .1                           80
view sbfcall    included  .1.3.6.1.4.1.1369.2.2        80

####
# Finally, grant the 2 groups access to the 1 view with different
# write permissions:

#                context sec.model sec.level prefix read   write  notif
access MyROGroup ""      any       noauth    exact  sbfcall none   none
access MyRWGroup ""      any       noauth    exact   all    all    none
################################################################################
# load average checks
#
# load [1MAX=12.0] [5MAX=12.0] [15MAX=12.0]
#
# 1MAX:   If the 1 minute load average is above this limit at query
#         time, the errorFlag will be set.
# 5MAX:  Similar, but for 5 min average.
# 15MAX:  Similar, but for 15 min average.
# Check for loads:
load 12 14 14
```

C.4 Checkpoint

C.4.1 /export/install/jumpstart/files/software/Checkpoint-current/S89FirewallInstall

```
#!/bin/sh
PATCH_DIR="/mnt"
OE_VER="`uname -r`"
CP_DIR="software/Checkpoint-current"
PKGADD=/usr/sbin/pkgadd
LOG=/var/sadm/system/logs/S89FirewallInstall.log
# ------------------------------------------------------ #
# Mount up the patch directory                           #
# ------------------------------------------------------ #
mount -F nfs -o ro <JS IP-@>:/export/install/jumpstart/files ${PATCH_DIR}
echo "Mount suceeded..."
# ------------------------------------------------------ #
# Install Checkpoint FW1 packages                        #
# ------------------------------------------------------ #
echo "Installing Checkpoint SVN Foundation R54..."
echo "Installing Checkpoint SVN Foundation R54..." >$LOG
echo y | $PKGADD -d /mnt/$CP_DIR -a /mnt/$CP_DIR/../noask CPshrd-54
>>$LOG 2>&1
echo "Installing Checkpoint Firewall-1/VPN-1 R54..."
echo "Installing Checkpoint Firewall-1/VPN-1 R54..." >>$LOG
echo y | $PKGADD -d /mnt/$CP_DIR -a /mnt/$CP_DIR/../noask CPfw1-54 >>$LOG
2>&1
# ------------------------------------------------------ #
# Install Checkpoint Hotfixes                            #
# ------------------------------------------------------ #
. /.profile

echo "Installing Checkpoint SVN Foundation Hotfix HFA404..."
#/usr/bin/mkdir $CPDIR/HFA404 >> $LOG 2>&1
#/usr/bin/cp /mnt/$CP_DIR/Hotfixes/HFA404/* $CPDIR/HFA404/ >> $LOG 2>&1
sleep 5
# ------------------------------------------------------ #
# Umount the patch directory                             #
# ------------------------------------------------------ #
umount /mnt
# ------------------------------------------------------ #
# and nuke ourselves!                                    #
```

```
# ------------------------------------------------------ #
rm -f /etc/rc3.d/S89FirewallInstall
echo "S89FirewallInstall Finished." >>$LOG
```

C.4.2 cpconfig Detailed Installation Sequence

< license text >

Do you accept all the terms of this license agreement (y/n) ? **y**

Select installation type:
---
(1) Stand Alone - install VPN-1 / FireWall-1 Internet Gateway.
(2) Distributed - select components of the Enterprise Product.
Enter your selection (1-2/a-abort) [1]: **2**

Select installation type:
---
(1) Enforcement Module.
(2) Enterprise SmartCenter.
(3) Enterprise SmartCenter and Enforcement Module.
(4) Enterprise Log Server.
(5) Enforcement Module and Enterprise Log Server.

Enter your selection (1-5/a-abort) [1]: **1**

********** VPN-1 & FireWall-1 kernel module installation **********
Installing VPN-1 & FireWall-1 kernel module...
Apr 21 17:50:08 c13f1 fw: FW-1: driver installed
Apr 21 17:50:09 c13f1 vpn: VPN-1: driver installed
Done.

********** Interface Configuration **********
Scanning for unknown interfaces...
Would you like to install a Check Point clustering product (CPHA, CPLS or State Synchronization)? (y/n) [n] ? **n**

Would you like to enable SecureXL acceleration feature? (y/n) [y] ? **n**

IP forwarding disabled
Hardening OS Security: IP forwarding will be disabled during boot.
Generating default filter.
Default Filter installed
Hardening OS Security: Default Filter will be applied during boot.
This program will guide you through several steps where you
will define your VPN-1 & FireWall-1 configuration.
At any later time, you can reconfigure these parameters by
running cpconfig

Configuring Licenses...
==========
Host Expiration Signature Features

Note: The recommended way of managing licenses is using SmartUpdate.
cpconfig can be used to manage local licenses only on this machine.
Do you want to add licenses (y/n) [y] ? **n**

Configuring Random Pool...
==========
You are now asked to perform a short random keystroke session.

The random data collected in this session will be used in
various cryptographic operations.

Please enter random text containing at least six different
characters. You will see the '*' symbol after keystrokes that
are too fast or too similar to preceding keystrokes. These
keystrokes will be ignored.

Please keep typing until you hear the beep and the bar is full.

[.....................]

Thank you.

Configuring Secure Internal Communication...
==========
The Secure Internal Communication is used for authentication between
Check Point components

Trust State: Uninitialized
Enter Activation Key: **<init-key>**
Again Activation Key: **<init-key>**

The Secure Internal Communication was successfully initialized
initial_module:
Compiled OK.

Hardening OS Security: Initial policy will be applied
until the first policy is installed

In order to complete the installation
you must reboot the machine.

Do you want to reboot? (y/n) [y] ? **y**

********** Installation completed successfully **********

C.5 Dragon

C.5.1 S92DragonHIDSInstall

```
#!/bin/sh

PATCH_DIR="/mnt"

OE_VER="`uname -r`"

CP_DIR="software/Dragon-current"

PKGADD=/usr/sbin/pkgadd

LOG=/var/sadm/system/logs/S92DragonHIDSInstall.log

# Mount up the patch directory

mount -F nfs -o ro <JS IP-@>:/export/install/jumpstart/files ${PATCH_DIR}

echo "Mount suceeded..."

# Install Dragon HIDS packages

echo "Installing Dragon HIDS..."

echo "Installing Dragon HIDS..." >$LOG

echo y | $PKGADD -d /mnt/$CP_DIR -a /mnt/$CP_DIR/../noask ESdsquire
>>$LOG 2>&1
```

```
# Copy config files
cp -p /mnt/squire.cfg /opt/dragon/etc/

sleep 5

# Umount the patch directory

umount /mnt

# and nuke ourselves!

rm -f /etc/rc3.d/S92DragonHIDSInstall
echo "S92DragonHIDSInstall Finished." >>$LOG
```

C.5.2 squire.cfg

```
Squire {
   SensorName              fwnode-hids
   Debug                   FALSE
   Daemonize               FALSE
   LogFileName             ../logs/dsquire.log
   RB_BufferSize           6144   # maximum event size in the ring buffer
   RB_NumElements          100    # number of events in the ring buffer
   HeartBeatRate           1800   # number of seconds between Heartbeat

   EventDetectionEngine {
      # (EDE) Modules run in parallel to feed the EFE

      AddInModules {

         HoneyPot {
            Command       ./modules/modules-ets/squire-EDE-honeypot
            EventFormat          %T%N%E%S%D%G%H%B%A%X
            Enabled              TRUE
            ConfigFile           ./conf/dsquire.net
            Version              1
         }

         SysInfo {
            Command             ./modules/modules-ets/sysinfo.pl
            EventFormat          %T%N%E%S%D%G%H%B%A%X
            Enabled             FALSE
            MaxNumErrors        1
            ConfigFile          ./conf/dsquire.net
            Version             1
         }

         MD5Detection {
            Enabled                 TRUE
            Debug                   FALSE
            ModuleName  ./modules/modules-ets/squire-EDE-md5Detection.so
            ModuleType              so
            Version                 1
            InitializeFunctionName  prepEnvironment
            ProcessFunctionName     run
            TerminateFunctionName   cleanEnvironment
            Identifier              MD5Detection
            ConfigFile              ./conf/dsquire.net
            MaxNumErrors            1
         }

         FileDetection {
            Enabled                 TRUE
            Debug                   FALSE
            ModuleName  ./modules/modules-ets/squire-EDE-fileDetection.so
            ModuleType              so
```

```
   Version                 1
   InitializeFunctionName  prepEnvironment
   ProcessFunctionName     run
   TerminateFunctionName   cleanEnvironment
   Identifier              FileDetection
   ConfigFile              ./conf/dsquire.net
   MaxNumErrors            1
}

LogDetection {
   Enabled                 TRUE
   Debug                   FALSE
   ModuleName   ./modules/modules-ets/squire-EDE-logDetection.so
   ModuleType              so
   Version                 1
   InitializeFunctionName  prepEnvironment
   ProcessFunctionName     run
   TerminateFunctionName   cleanEnvironment
   Identifier              LogDetection
   ConfigFile              ./conf/dsquire.net
   AltConfigFile           ./conf/dsquire.sigs
   MaxNumErrors            1
}

snmpDetection {
   Enabled                 TRUE
   Debug                   FALSE
   ModuleName ./modules/modules-ets/squire-EDE-snmpDetection.so
   ModuleType              so
   Version                 1
   InitializeFunctionName  prepEnvironment
   ProcessFunctionName     run
   TerminateFunctionName   cleanEnvironment
   Identifier              snmpDetection
   ConfigFile              ./conf/dsquire.net
   AltConfigFile           ./conf/dsquire.sigs
   MaxNumErrors            1
}

ServiceDetection {
   Enabled                 TRUE
   Debug                   FALSE
   ModuleName ./modules/modules-ets/squire-EDE-serviceDetection.so
   ModuleType              so
   Version                 1
   InitializeFunctionName  prepEnvironment
   ProcessFunctionName     run
   TerminateFunctionName   cleanEnvironment
   Identifier              ServiceDetection
   ConfigFile              ./conf/dsquire.net
   MaxNumErrors            1
}

KernelDetection {
   Enabled                 FALSE
   Debug                   FALSE
   ModuleName ./modules/modules-ets/squire-EDE-kernelDetection.so
   ModuleType              so
   Version                 1
   InitializeFunctionName  prepEnvironment
   ProcessFunctionName     run
   TerminateFunctionName   cleanEnvironment
   Identifier              KernelDetection
```

```
            ConfigFile              ./conf/dsquire.net
            AltConfigFile       ./modules/modules-ets/sqm-kernel-2.2.o
            MaxNumErrors            1
         }
      }
   }

   EventFilterEngine {
      # (EFE) Modules run serially against events prior to the EAE.  These

      # modules are invoked from recordEvent().  Each EDE Module will

      # invoke the EFE modules in a seperate thread (as the result of calling

      # recordEvent callback function).  The EFE modules

      # are then run serially in the order presented below.

      AddInModules {
         setSIPDIP {
            Enabled                 TRUE
            Debug                   FALSE
            ModuleName     ./modules/modules-ets/squire-EFE-setSipDip.so
            ModuleType              so
            Identifier              setSIPDIP
            InitializeFunctionName  prepEnvironment
            ProcessFunctionName     filterEvent
            TerminateFunctionName   cleanEnvironment
            ConfigFile              ./conf/dsquire.net
            MaxNumErrors            1
            Version                 1
         }

         setVSensor {
            Enabled                 TRUE
            Debug                   FALSE
            ModuleName    ./modules/modules-ets/squire-EFE-setVSensor.so
            ModuleType              so
            Identifier              setVSensor
            InitializeFunctionName  prepEnvironment
            ProcessFunctionName     filterEvent
            TerminateFunctionName   cleanEnvironment
            ConfigFile              ./conf/dsquire.net
            MaxNumErrors            1
            Version                 1
         }
      }
   }

   EventAlertingEngine {
      # (EAE) Modules run in parallel to alert events

      AddInModules {
         HexDumpAlert {
            Enabled                 FALSE
            Debug                   FALSE
            ModuleName ./modules/modules-ets/squire-EAE-hexDumpAlert.so
            ModuleType              so
            Version                 1
            InitializeFunctionName  prepEnvironment
            ProcessFunctionName     receiveEvent
            TerminateFunctionName   cleanEnvironment
            Identifier              HexDumpAlert
            MaxNumErrors            1
         }
```

```
        SNMPTrapAlert {
           Enabled                  TRUE
           Debug                    FALSE
           ModuleName ./modules/modules-ets/squire-EAE-snmpTrapAlert.so
           ModuleType               so
           Version                  1
           InitializeFunctionName   prepEnvironment
           ProcessFunctionName      receiveEvent
           TerminateFunctionName    cleanEnvironment
           Identifier               SNMPTrapAlert
           ConfigFile               ./conf/dsquire.net
           MaxNumErrors             1
        }

        RingBufferAlert {
           Enabled                  TRUE
           Debug                    FALSE
           ModuleName ./modules/modules-ets/squire-EAE-ringbufferAlert.so
           ModuleType               so
           Version                  1
           InitializeFunctionName   prepEnvironment
           ProcessFunctionName      receiveEvent
           TerminateFunctionName    cleanEnvironment
           Identifier               RingBufferAlert
           ConfigFile               ../dragon.cfg
           MaxNumErrors             1
        }

        DragonDBALert {
           Enabled                  FALSE
           Debug                    FALSE
           ModuleName ./modules/modules-ets/squire-EAE-dragonDBAlert.so
           ModuleType               so
           Version                  1
           InitializeFunctionName   prepEnvironment
           ProcessFunctionName      receiveEvent
           TerminateFunctionName    cleanEnvironment
           Identifier               DragonDBAlert
           ConfigFile               ../dragon.cfg
           MaxNumErrors             1
        }
     }
  }
}
```

C.5.3 dsquire.net

```
=/                  L
/.profile           R
/.bashrc            R
/.cshrc             R
/.ssh               R
/.ssh/known_hosts   L
/.ssh/prng_seed     E
/.ssh/random_seed   E

/dev                L
/devices            L
=/devices/pseudo    L
/etc                R
/etc/dumpdates      L
```

```
/etc/motd               L
/etc/passwd             L
/etc/rmtab              L
=/etc/saf               L
/etc/shadow             L
/etc/ttydefs            L
/etc/.syslog_door       E
/etc/inet/ntp.drift     E
/etc/sshd.pid           E
/etc/ssh_random_seed    E
/etc/syslog.pid         E
/etc/utmppipe           E

=/var                   L
=/var/adm               L
/var/adm/utmp           L
/var/adm/utmpx          L
/var/adm/wtmp           L
/var/adm/wtmpx          L
/var/adm/sulog          L
/var/adm/sshd.log       L
=/var/adm/sa            L
=/var/spool             L

=/tmp                   L
=/var/tmp               L
=/proc                  L
=/usr                   R

/kernel                 R
/opt                    R
/usr/kernel             R
/sbin                   R
/usr/sbin               R
/usr/bin                R
/usr/local/bin          R
/usr/lib                R
/usr/xpg4/lib           R
/usr/ccs                R

# Sensitive programs
/usr/bin/sh             R
/usr/bin/csh            R
/usr/bin/ksh            R
/usr/bin/bash           R
/usr/bin/crontab        R
/usr/bin/diff           R
/usr/bin/df             R
/usr/bin/du             R
/usr/bin/find           R
/usr/bin/finger         R
/usr/bin/kill           R
/usr/bin/login          R
/usr/bin/ls             R
/usr/bin/netstat        R
/usr/bin/passwd         R
/usr/bin/ps             R
```

```
/usr/bin/su                 R
/usr/bin/sum                R
/usr/bin/w                  R
/usr/bin/who                R
/usr/ucb/df                 R
/usr/ucb/du                 R
/usr/ucb/ls                 R
/usr/ucb/ps                 R
/usr/ucb/sum                R
/usr/sbin/cron              R
/usr/sbin/ifconfig          R
/usr/sbin/inetd             R
/usr/sbin/in.ftpd           R
/usr/sbin/in.telnetd        R
/usr/sbin/in.rshd           R
/usr/sbin/in.rlogind        R
/usr/sbin/syslogd           R
/opt/openssh/sbin/sshd      R

# Checkpoint
/opt/CPshrd-54              R
!/opt/CPshrd-54/tmp         R
!/opt/CPshrd-54/log         R
!/opt/CPshrd-54/databaseR
/opt/CPshrd-54/database     L
/opt/CPshrd-54/log          L
/opt/CPfw1-54               R
!/opt/CPfw1-54/tmp          R
!/opt/CPfw1-54/log          R
!/opt/CPfw1-54/database     R
!/opt/CPfw1-54/spool        R
/opt/CPfw1-54/log           L
/opt/CPfw1-54/database      L

# Stonebeat
/opt/fullcluster            R

# Dragon
/opt/dragon                 R
!/opt/dragon/logs           R
!/opt/dragon/DB             R
/opt/dragon/logs            L
/opt/dragon/DB              L
```

C.5.4 Signature Files

Solaris-login

HOST:SOLARIS:LOGIN-FAILED messages S %1:ftpd,LOGIN/20FAILED
An ftp login attempt did not succeed.

Solaris-messages

HOST:SOLARIS:HALTED messages B %1:unix/3a/20halted
Solaris 2.x system halted by user. Sample message 'Mar 31 12:48:41 hostname unix: halted by userid'

HOST:SOLARIS:REBOOTING messages B %1:unix/3a/20rebooting
Solaris system reboot. Sample message 'Mar 31 12:48:41 ahost.domain.com unix: rebooting.'

HOST:SOLARIS:SU-FAILED messages B %1:su/20root,failed
Failed su command. Sample message 'Mar 31 12:37:43 hostname su: 'su root' failed for userid on /dev/pts/'

HOST:SOLARIS:SU-SUCCEEDED messages B %1:su/20root,succeeded
Successful su command, check for unexpected use. Sample message 'Mar 28 14:31:11 hostname su: 'su root' succeeded for userid on /dev/console'

HOST:SOLARIS:REPEATED-FAILURES messages S %1:login,/20REPEATED/20LOGIN/20FAILURES
A telnet login failed multiple times in succession

Solaris-sulog

HOST:SU-ROOT sulog B %1:/2droot
An attempt was made to "su" to the root account.

Solaris-ssh

HOST:SSH:ROOT-NOT-PERMITTED messages B %1:sshd,root/20logins/20are/20not/20permitted

HOST:SSH:FORWARD-TCPIP messages B %1:sshd,Remote/20TCP/2fIP/20forwarding/20request/20received/20from/20host

HOST:SSH:FORWARD-FAIL messages B %1:sshd,not/20root/2c/20tried/20to/20forward/20privileged/20port
A non root user attempted to set up SSH port forwarding. This could be an inexperienced user, a system administrator who forgot to 'SU' to root or a hacker trying to set up a backdoor.

HOST:SSH:FORWARD-SETUP messages B %1:sshd,Port/20#,/20set/20up/20for/20remote/20forwarding.

HOST:SSH:BAD-DNS messages B %1:sshd,Client/20gave/20us/20a/20hostname,which/20doesn't/20match/20the/20one/20we/20got/20from/20DNS
Secure Shell applications attempt to match the DNS name of a connecting host and the name provided by the connecting host. If there is a discrepancy, the host could be misconfigured. An extreme case of this signature occurs when an attacker is attempting to spoof or alter DNS records.

HOST:SSH:BAD-ALGORITHM messages B %1:sshd,Client's/20public/20key/20algorithms/20are/20not/20supported/20by/20us
This signature watches logs for failed SSH connection attempts. These messages may result from incompatible SSH clients, but can also occur when various exploits are used against the SSH daemon. The message may also occur when a user attempts to telnet to port 22 which can also be part of a network probe.

HOST:SSH:X11-REJECT messages B %1:sshd,X11/20connection/20rejected/20because/20of/20wrong/20authentication

HOST:SSH:CONNECTION-DENIED messages B %1:sshd,Received/20request/20to/20connect/20to,but/20the/20request/20was/20denied.
A connection attempt to the SSH daemon was denied.

HOST:SSH:LOGIN-NONROOT messages B %1:sshd,Attempt/20to/20write/20login/20records/20by/20non-root/20user

HOST:SSH:ROOT-LOGIN messages B %1:sshd,ROOT/20LOGIN/20as/20

HOST:SSH:SEQNR-WRAP messages B
%1:sshd,incoming/20seqnr/20wraps/20around

HOST:SSH:ROOT-LOGIN-REFUSED messages B
%1:sshd,ROOT/20LOGIN/20REFUSED/20FROM

HOST:SSH:FAKE-AUTHLOOP messages B
%1:sshd,Faking/20authloop/20for/20illegal/20user
OpenSSH will attempt to request a password for usernames which do not exist. This prevents a remote user from determining which accounts are active or inactive on a target machine. It also provides a good indication of remote probes for SSH.

HOST:SSH:LOGIN-NOT-ALLOWED messages B
%1:sshd,login/20to/20account,not/20allowed

HOST:SSH:FORWARD-PRIVILEGED messages B
%1:sshd,Privileged/20user,forwarding/20a/20privileged/20port.
A privileged user, not necessarily the root user, has set up a port forwarding SSH condition on a port below 1024. This could forward requests for services such as telnet, http or even SSH to other systems.

HOST:SSH:ROOT-LOGIN messages B
%1:sshd,User/20root/2c/20coming/20from/20

HOST:SSH:ROOT-LOGIN2 messages B
%1:sshd,User/20root/2c/20coming/20from,authenticated.

HOST:SSH:DENIED messages B %1:sshd,Denied/20connection/20for
An SSH login attempt was denied by the SSH server.

HOST:SSH:LOGIN-FAILED messages B %1:sshd,User/20authentication/20failed

HOST:SSH:KERBEROS-FAILED messages B
%1:sshd,Kerberos/20ticket/20authentication/20of/20user,failed

HOST:SSH:KERBEROS-REJECTED messages B
%1:sshd,Kerberos/20V5/20tgt/20rejected/20for/20user

HOST:SSH:CIPHER messages B
%1:sshd,fatal:/20no/20matching/20cipher/20found:
This signature looks for an SSH client trying to connect to a server that does not support the client's encryption algorithm. This can be due to misconfiguration or attempted misuse by an unauthorized user.

HOST:SSH:PASSWORD-FAILED messages B %1:sshd,Failed/20password/20for

HOST:SSH:RSA-KEY-GENERATED messages B
%1:sshd,RSA/20key/20generation/20complete

HOST:SSH:USERAUTH-FAILURES messages B
%1:sshd,too/20many/20failed/20userauth_requests

HOST:SSH:AUTH-FAILURES messages B
%1:sshd,Too/20many/20authentication/20failures
A user has exceeded the maximum number of authentication failures set by the AUTH_FAIL_LOG variable (defaults to 3).

HOST:SSH:CRC32-ATTACK messages B
%1:sshd,crc32/20compensation/20attack/3a/20network/20attack/20detected
Possible exploitation of a vulnerability in SSH1 CRC-32 compensation attack detector. It is recommended that you disable support for SSH protocol version 1.

The vulnerability has been addressed in OpenSSH 2.3.0 and Secure Shell 1.2.32. CVE-2001-0144

HOST:SSH:CRC32-ERROR messages B %1:sshd,Corrupted/20check/20bytes/20on/20input
Possible exploitation of a vulnerability in SSH1 CRC-32 compensation attack detector. It is recommended that you disable support for SSH protocol version 1. CVE-2001-0144

HOST:SSH:NO-IDENTIFICATION messages B %1:sshd,Did/20not/20receive/20identification/20string

Solaris-login

HOST:SOLARIS:ROOT-LOGIN-ENABLED login S %1:/23CONSOLE=/2fdev/2fconsole

C.6 Log Management

C.6.1 S91ToolsInstall

```
#!/bin/sh

PATCH_DIR="/mnt"

OE_VER="`uname -r`"

CP_DIR="software/tools-current"

PKGADD=/usr/sbin/pkgadd

LOG=/var/sadm/system/logs/S91ToolsInstall.log

# Mount up the directory

mount -F nfs -o ro <JS IP-@>:/export/install/jumpstart/files ${PATCH_DIR}

echo "Mount suceeded..."

# Install lsof package

echo "Installing lsof..."

echo "Installing lsof..." >$LOG

echo y | $PKGADD -d /mnt/$CP_DIR -a /mnt/$CP_DIR/../noask SMClsof >>$LOG
2>&1

# Copy logcheck files
cp -pr /mnt/logcheck/* /opt/logcheck/

sleep 5

# Umount the patch directory

umount /mnt

# and nuke ourselves!
rm -f /etc/rc3.d/S91ToolsInstall
echo "S91ToolsInstall Finished." >>$LOG
```

C.6.2 /opt/scripts/rotate-logs.sh

```
#! /bin/sh
LOG=messages
cd /var/adm
test -f $LOG.2 && mv $LOG.2 $LOG.3
test -f $LOG.1 && mv $LOG.1 $LOG.2
```

```
test -f $LOG.0 && mv $LOG.0 $LOG.1
mv $LOG   $LOG.0
cp /dev/null $LOG
chmod 600    $LOG
#
LOG=ssh.log
cd /var/adm
test -f $LOG.2 && mv $LOG.2 $LOG.3
test -f $LOG.1 && mv $LOG.1 $LOG.2
test -f $LOG.0 && mv $LOG.0 $LOG.1
mv $LOG   $LOG.0
cp /dev/null $LOG
chmod 600    $LOG
#
LOGDIR=/var/log
LOG=syslog
if test -d $LOGDIR
then
        cd $LOGDIR
        if test -s $LOG
        then
                test -f $LOG.6 && mv $LOG.6  $LOG.7
                test -f $LOG.5 && mv $LOG.5  $LOG.6
                test -f $LOG.4 && mv $LOG.4  $LOG.5
                test -f $LOG.3 && mv $LOG.3  $LOG.4
                test -f $LOG.2 && mv $LOG.2  $LOG.3
                test -f $LOG.1 && mv $LOG.1  $LOG.2
                test -f $LOG.0 && mv $LOG.0  $LOG.1
                mv $LOG    $LOG.0
                cp /dev/null $LOG
                chmod 644    $LOG
                sleep 40
        fi
fi
#
kill -HUP `cat /etc/syslog.pid`
```

C.6.3 /opt/logcheck/bin/logcheck.sh

```
#!/bin/sh
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/ucb:/usr/local/bin
SYSADMIN=root
LOGTAIL=/opt/logtail/bin
TMPDIR=/opt/logtail/etc/tmp
GREP=egrep
MAIL=mail
HACKING_FILE=/opt/logtail/etc/logcheck.hacking
VIOLATIONS_FILE=/opt/logtail/etc/logcheck.violations
VIOLATIONS_IGNORE_FILE=/opt/logtail/etc/logcheck.violations.ignore
IGNORE_FILE=/opt/logtail/etc/logcheck.ignore
# Shouldn't need to touch these...
HOSTNAME=`hostname`
DATE=`date +%m/%d/%y:%H.%M`
umask 077
rm -f $TMPDIR/check.$$ $TMPDIR/checkoutput.$$ $TMPDIR/checkreport.$$
if [ -f $TMPDIR/check.$$ -o -f $TMPDIR/checkoutput.$$ -o –f \
        $TMPDIR/checkreport.$$ ]; then
        echo "Log files exist in $TMPDIR directory that cannot be \
         removed. This may be an attempt to spoof the log checker." \
        | $MAIL -s "$HOSTNAME $DATE ACTIVE SYSTEM ATTACK!" $SYSADMIN
```

```
        exit 1
fi

# LOG FILE CONFIGURATION SECTION
# SunOS, Sun Solaris 2.5
$LOGTAIL /var/log/syslog > $TMPDIR/check.$$
$LOGTAIL /var/adm/messages >> $TMPDIR/check.$$
$LOGTAIL /var/adm/ssh.log >> $TMPDIR/check.$$
$LOGTAIL /var/log/snmpd.log >> $TMPDIR/check.$$

# END CONFIGURATION SECTION. YOU SHOULDN'T HAVE TO EDIT ANYTHING

# BELOW THIS LINE - is nothing change we left it threfore away
```

C.7 Hardening

C.7.1 /default/inetinit

```
# @(#)inetinit.dfl 1.2 97/05/08
#
# TCP_STRONG_ISS sets the TCP initial sequence number generat parameters.
# Set TCP_STRONG_ISS to be:
#       0 = Old-fashioned sequential initial sequence number generation.
#       1 = Improved sequential generation, with random var increment.
#       2 = RFC 1948 sequence number generation, unique-per-connect-ID.
#
TCP_STRONG_ISS=2
```

C.7.2 /etc/system-addon

```
set nfssrv:nfs_portmon=1
set noexec_user_stack=1
set noexec_user_stack_log=1
*
set tcp:tcp_conn_hash_size=16384
set rlim_fd_max=16384
*
* set interfaces to 100Mbit, full-duplex, no autonegotiation
set hme:hme_adv_autoneg_cap=0
set hme:hme_adv_100fdx_cap=1
set hme:hme_adv_100T4_cap=0
set hme:hme_adv_100hdx_cap=0
set hme:hme_adv_10fdx_cap=0
set hme:hme_adv_10hdx_cap=0
*
set qfe:qfe_adv_autoneg_cap=0
set qfe:qfe_adv_100T4_cap=0
set qfe:qfe_adv_100fdx_cap=1
set qfe:qfe_adv_100hdx_cap=0
set qfe:qfe_adv_10fdx_cap=0
set qfe:qfe_adv_10hdx_cap=0
*
set eri:adv_autoneg_cap=0
```

```
set eri:adv_100T4_cap=0
set eri:adv_100fdx_cap=1
set eri:adv_100hdx_cap=0
set eri:adv_10fdx_cap=0
set eri:adv_10hdx_cap=0
```

C.7.3 /etc/syslog.conf

```
*.err;kern.notice;auth.notice   /dev/console
*.alert                         root
*.emerg                         *
*.debug                         /var/adm/messages
local1.info                     /var/adm/sshd.log
```

C.7.4 /etc/inittab

```
ap::sysinit:/sbin/autopush -f /etc/iu.ap
ap::sysinit:/sbin/soconfig -f /etc/sock2path
fs::sysinit:/sbin/rcS sysinit          >/dev/msglog 2<>/dev/msglog </dev/console
is:3:initdefault:
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2<>/dev/msglog
sS:s:wait:/sbin/rcS                    >/dev/msglog 2<>/dev/msglog </dev/console
s0:0:wait:/sbin/rc0                    >/dev/msglog 2<>/dev/msglog </dev/console
s1:1:respawn:/sbin/rc1                 >/dev/msglog 2<>/dev/msglog </dev/console
s2:23:wait:/sbin/rc2                   >/dev/msglog 2<>/dev/msglog </dev/console
s3:3:wait:/sbin/rc3                    >/dev/msglog 2<>/dev/msglog </dev/console
s5:5:wait:/sbin/rc5                    >/dev/msglog 2<>/dev/msglog </dev/console
s6:6:wait:/sbin/rc6                    >/dev/msglog 2<>/dev/msglog </dev/console
fw:0:wait:/sbin/uadmin 2 0             >/dev/msglog 2<>/dev/msglog </dev/console
of:5:wait:/sbin/uadmin 2 6             >/dev/msglog 2<>/dev/msglog </dev/console
rb:6:wait:/sbin/uadmin 2 1             >/dev/msglog 2<>/dev/msglog </dev/console
co:234:respawn:/usr/lib/saf/ttymon -g -h -p "`uname -n` console login: " \
         -T vt100 -d /dev/console -l console -m ldterm,ttcompat
```

C.8 SSH

C.8.1 /etc/ssh/sshd_config

```
# This sshd was compiled with
PATH=/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin
Protocol 2
ListenAddress <FW node IP-@>
ListenAddress localhost
SyslogFacility LOCAL1
LogLevel VERBOSE
LoginGraceTime 1m
PermitRootLogin yes
DSAAuthentication yes
PasswordAuthentication no
ChallengeResponseAuthentication no
UsePAM no
```

```
UseDNS no
Subsystem       sftp    /usr/local/libexec/sftp-server
```

C.8.2 /etc/ssh/ssh_config

```
ForwardAgent yes
Protocol 2
Ciphers aes256-cbc
```

C.8.3 /.ssh/authorized_keys

This is an example authorized_keys file for root.

```
from="127.0.0.1" ssh-dss
AAAAB3NzaC1kc3MAAAEBAIuPEVFTsVgI8odPKueeXh1BQ2D9XbBS8K6xSNe5Dx+in8qP+6Stw
…
IyM/qRvvfzCrwrDrxCIAO97g08W1Wyp7CeVzjbJz+NIc5CHwNAgVoq0ihiVSrtvJZ+BA==
sysadmin-1@sans.org
```

C.8.4 /home/sans/.ssh/authorized_keys

This is an example authorized_keys file for the users.

```
ssh-dss
AAAAB3NzaC1kc3MAAAEBAIuPEVFTsVgI8odPKueeXh1BQ2D9XbBS8K6xSNe5Dx+in8qP+6Stw
…
IyM/qRvvfzCrwrDrxCIAO97g08W1Wyp7CeVzjbJz+NIc5CHwNAgVoq0ihiVSrtvJZ+BA==
sysadmin-1@sans.org
```

C.9 NTP

C.9.1 /etc/inet/ntp.conf

```
# Prohibit general access to this service.
restrict default ignore

# Permit time synchronization with our time source, but do not permit the
# source to query or modify the service on this system.
restrict <IP-@ time-source-1> noquery nomodify notrap
restrict <IP-@ time-source-2> noquery nomodify notrap

# Permit all access over the loopback interface. This could be tightened
# well, but to do so would effect some of the administrative functions.
restrict 127.0.0.1

# Enable authentication
enable auth

# Locate the keyring
keys /etc/inet/ntp.keys.node

# Define which keys are trusted
trustedkey 1

# Permit only authenticated access.
server <IP-@ time-source-1> prefer key 1
server <IP-@ time-source-2> key 1
```

C.9.2 /etc/inet/ntp.keys

```
1 M myNTPkey
```

C.10 Backup

C.10.1 /opt/scripts/backup.sh

```
# backup the following directories or files

list="
/etc/init.d
/var/adm
/opt/fullcluster/etc
"

# log
tag="backup-ng-node"

# day of the month
d=`date +'%d'`
dest=/backup/$d.tar
md5log=/backup/$d.md5
adm=sysadmin@sans.org
errlog=/var/adm/backup
date >>$errlog

# error handling
err() {
    err="$tag: $1. exit now."
    logger $err
    echo "`hostname` see $errlog for details" |mailx -s "$err" $adm
    exit 1
}

# /var/adm/messages entry
logger start backup

# delete old tar file
rm -f $dest*
if [ $? -ne 0 ]; then
    err "cannot remove $dest.Z"
fi

# delete old md5log file
rm -f $md5log
if [ $? -ne 0 ]; then
    err "cannot remove $md5log"
fi

# create tar file
tar cf $dest $list >>$errlog 2>&1

# md5 hash
md5 compress $dest >$md5log >>$errlog 2>&1
if [ $? -ne 0 ]; then
    err "cannot compress $dest"
fi

# compress
compress $dest >>$errlog 2>&1
if [ $? -ne 0 ]; then
    err "cannot compress $dest"
fi

# /var/adm/messages entry
logger backup finished successfully
```