

Managing Data Center Functions with Open Source Tools



By Jascha Wanger (jaschawanger@bse-inc.com)
(jascha@localareasecurity.com)

Outline

- ➔ Firewalls
- ➔ IDS (Intrusion Detection)
- ➔ Monitoring/Administration
- ➔ Auditing
- ➔ Spam/Virus/Worm Scanning
- ➔ Proprietary Solution Cost Comparison
- ➔ Conclusion

Firewalls

⇒ **Netfilter/IPtables/Shorewall - Linux**



⇒ **PF (Packet Filter) - OpenBSD**



⇒ **IPFW - FreeBSD**



Netfilter/IPtables

- ➔ **Netfilter** - a set of hooks inside the Linux 2.4.x kernel's network stack which allows kernel modules to register callback functions called every time a network packet traverses one of those hooks.
- ➔ **IPtables** - a generic table structure for the definition of rule sets Each rule within an IP table consists out of a number of classifiers (matches) and one connected action (target).

Netfilter/IPtables Features

- ⇒ packet filtering (stateless or stateful)
- ⇒ NAT (Network Address Translation)
- ⇒ advanced packet processing
- ⇒ packet mangling
- ⇒ framework inside the Linux 2.4.x kernel
- ⇒ additional features as patches

Shorewall

- ➔ **Shorewall** - high-level tool for configuring Netfilter/IPtables
 - Router/firewall/gateway applications
 - IP Masquerading
 - Source Network Address Translation (SNAT)
 - Blacklisting
 - VPN support
 - Traffic Control/Shaping
 - MAC Verification
 - Traffic accounting

PF (Packet Filter) on OpenBSD

- ➔ OpenBSD's system for filtering TCP/IP traffic
- ➔ NAT (Network Address Translation)
- ➔ Packet “Scrubbing”
- ➔ Queuing (Class Based, Priority, etc.)
- ➔ Traffic Redirection
- ➔ Part of the OpenBSD kernel
- ➔ Security hardened host Operating System
- ➔ OpenBSD security track record

IPFW on FreeBSD

- ⇒ Packet filtering
- ⇒ IP accounting
- ⇒ NAT (Network Address Translation)
- ⇒ Part of FreeBSD kernel
- ⇒ FreeBSD rock solid stability/reliability

IDS (Intrusion Detection System)

- ⇒ Snort Intrusion Detection System
- ⇒ Prelude Hybrid Intrusion Detection System
- ⇒ AIDE (Advanced Intrusion Detection Environment)
- ⇒ Samhain File Integrity / Intrusion Detection System

Snort

➔ **Snort** - an open source network intrusion detection system.



- Real-time traffic analysis/alerts
- Packet logging
- Protocol analysis
- Content searching/matching
- Detect attacks/probes
- Flexible rules language
- Modular plug-in architecture
- Mature (current version 2.0.4)
- Released under GPL

Snort / A.C.I.D. / Snort Center

- ➔ **A.C.I.D.** - Analysis Console for Intrusion Databases
- ➔ **Snort Center** - Snort IDS Rule and Sensor Management
 - Centralized management of multiple Snort IDS nodes across a network from one console
 - Centralized logging to SQL database
 - Secured distribution of rule sets (via SSH)

Snort Center Sensor Overview

The screenshot displays the SnortCenter v0.9b4 web interface. At the top, there are navigation buttons for 'Sensor Console', 'Rules', 'Config Types', 'Admin', 'Alert Console', and 'Logout'. The main content area is divided into two sections. The upper section, titled 'Sensor Overview', lists four sensors: 'Internet -> eth0' (running), 'Lamp Development -> eth0' (disabled), 'Mail Servers -> eth0' (not running), and 'Web Servers -> eth0' (running). Each sensor entry includes a 'Sensor Control' section with buttons for 'Stop' and 'Restart' (or 'Start'), and a 'Snort Configuration File' section with buttons for 'Push', 'Preview', 'Download', and 'Test'. The lower section, titled 'SnortCenter Sensor Agent Version 0.1.2 Running Snort Version 1.8.7 (Build 128)', provides system status information, including system uptime, memory usage (7.1% used), swap usage (4% used), and a table of filesystem usage.

Sensor	Status	Sensor Control	Snort Configuration File
Internet -> eth0	Snort is running Fic# 7103	Stop - Restart	Push - Preview - Download - Test: System Status
Lamp Development -> eth0	Sensor Disabled		
Mail Servers -> eth0	Snort is not Running	Start - Restart	Push - Preview - Download - Test: System Status
Web Servers -> eth0	Snort is running Fic# 9173	Stop - Restart	Push - Preview - Download - Test: System Status

SnortCenter Sensor Agent Version 0.1.2
Running
Snort Version 1.8.7 (Build 128)

System status: 3:42pm up 1 day, 2:32, 3 users, load average: 0.36, 0.24, 0.16

Memory Usage: 7.1% memory used, 74684 K free
(shared: 0, buffers: 2592, cached: 67356)

Swap Usage: 4% used, 253008 K free

Filesystem	Size	Used	Avail	Use%	Mounted
/dev/rdah	4M	291K	3.7G	7%	/
/dev/rdas7	5.1G	373M	4.4G	8%	/home
none	125M	0	124M	0%	/dev/shm

Sensor Message

SnortCenter v0.9b4 Copyright © 2001, 2002 Stefan Dens

Snort Center Rule Set Control

The screenshot displays the Snort Center v1.0 web interface. At the top, there are navigation tabs for "Sensor Console", "Sensor Config", "Resources", "Admin", "Alert Console", and "Logout". Below the navigation, there are buttons for "Hide Policy Rules", "Show Only Rule Changes", "Hide Activated Rules", and "Hide Disabled Rules". A search bar labeled "Find" contains the text "sid".

On the left side, there are two panels:

- Sensor Group:** A dropdown menu showing "Internet Security Sensors".
- Rule Policy Templates:** A list of three templates with status indicators:
 - ✓ Activate Default Inactive Snort Rules.
 - ✗ Activate Default Inactive Snort Rules.
 - ✗ Activate Inactive Deleted Rules.

The main content area is titled "Category Search" and shows a dropdown menu with "http.misc" selected. Below this, there is a "Rule Category Overview" section. The main list of rules is displayed in a table-like format with the following columns: status, action, rule ID, rule name, and rule details.

Status	Action	Rule ID	Rule Name	Rule Details
✓	✓	sid: 1941	HTTP file name overflow attempt	content: "UUU"; offset: 0; depth: 2; content: "UUU"; offset: 0; reference: CVE-2007-11-110; category: http.misc; action: deny;
✓	✓	sid: 1209	HTTP GET Admin.dll	content: "UUU"; offset: 0; depth: 2; content: "Admin.dll"; offset: 0; reference: CVE-2006-0820; category: http.misc; action: deny;
✓	✓	sid: 1441	HTTP GET /wp/	content: "wp/"; offset: 0; depth: 2; content: "wp/"; offset: 0; reference: CVE-2006-0820; category: http.misc; action: deny;
✓	✓	sid: 1442	HTTP GET shadow	content: "shadow"; offset: 0; depth: 2; content: "shadow"; offset: 0; reference: CVE-2006-0820; category: http.misc; action: deny;
✓	✓	sid: 1443	HTTP GET passwd	content: "passwd"; offset: 0; depth: 2; content: "passwd"; offset: 0; reference: CVE-2006-0820; category: http.misc; action: deny;
✓	✓	sid: 519	HTTP parent directory traversal	content: "parent directory traversal"; offset: 0; reference: CVE-2002-1208; category: http.misc; action: deny;
✓	✓	sid: 520	HTTP root directory	content: "root directory"; offset: 0; reference: CVE-2002-1208; category: http.misc; action: deny;
✓	✓	sid: 518	HTTP Path	content: "Path"; offset: 0; depth: 2; reference: CVE-2002-1208; category: http.misc; action: deny;
✓	✓	sid: 1444	HTTP GET	content: "GET"; offset: 0; depth: 2; category: http.misc; action: deny;

At the bottom of the interface, there is a footer that reads "SnortCenter v1.0 Copyright © 2001-2003 Stefan Dawid".

Snort Center ACID Plugin

The screenshot displays the Snort Center ACID Plugin interface. At the top, there is a navigation bar with buttons for "Alert Home", "Search", "Snapshots", "Graphs", "Admin", and "Sensor Console". Below this, the main content area is titled "Analysis C" and "Intrusion Databases". A dropdown menu is open, listing various analysis options: "Most recent alerts", "Today's alerts", "Last 24 Hours alerts", "Last 72 Hours alerts", "Last Source Ports", "Last Destination Ports", "Most Frequent 5 Alerts", "Most Frequent Source Ports", "Most Frequent Destination Ports", and "Most Frequent 15 Addresses". A sub-menu is also visible, listing "Unique", "Listing", "Source IP", and "Destination IP".

On the left side, there is a summary of alert statistics:

- Added 0 alert(s) to the Alert cache
- Queried on: Fri, June 21, 2002 15:53
- Time window: [2001-07-01 09:47:03
- Sensors: 5
- Unique Alerts: 41 (8 categories)
- Total Number of Alerts: 22226

Below the statistics, there are two columns of data:

- Source IP addresses: 1670
- Dest. IP addresses: 57
- Unique IP links: 4564
- Source Ports: 4479
 - TCP (4454) UDP (313)
- Dest. Ports: 19
 - TCP (19) UDP (1)

On the right side, there is a horizontal bar chart showing the distribution of traffic by protocol:

- UDP (1%)
- ICMP (16%)
- Portscan Traffic (0%)

At the bottom of the interface, there is a footer with the text: "ACID v3.9.6621 (by Roman Danyliw as part of the AircERT project)" and "SnortCenter Copyright © 2001, 2002 Stefan Dena".

Prelude Hybrid IDS

➔ **Prelude** - Hybrid Intrusion Detection system

Five Main Parts:

- **Sensors** – detect at strategic network points
- **Managers** – centralized data processors (can be distributed)
- **Counter Measure Agents** – perform action to stop/thwart anomaly reported by a Manager
- **The Prelude Library** – API for modules

Prelude Overview

- ⇒ Host and/or network based
- ⇒ Modular design
- ⇒ Distributed managers and sensors
- ⇒ Countermeasure ability built in

AIDE (Advanced Intrusion Detection Environment)

- ➔ Host based
- ➔ Monitors file integrity
- ➔ Creates database of known good hashes
- ➔ Uses multiple algorithms
(md5,sha1,rmd160,tiger,haval,etc)
- ➔ Intended as GPL replacement of TripWire

Samhain File Integrity / IDS

- ⇒ Host based
- ⇒ File checksums detect modifications
- ⇒ Searches for rouge SUID executables
- ⇒ Detects root kits (under Linux and FreeBSD)
- ⇒ Able to log to SQL database
- ⇒ 'Complete' integrity checking
- ⇒ Released under GPL

Monitoring / Administration

- ➔ OpenNMS
- ➔ MRTG
- ➔ NeDi
- ➔ Nagios
- ➔ Webmin
- ➔ LogWatch / SWATCH
- ➔ SEC
- ➔ openMosix / openMosixview
- ➔ Etherape

OpenNMS

- ➔ Network Management System framework
- ➔ Easily customized to fit needs
- ➔ Modular design
- ➔ Supports plug-ins



OpenNMS

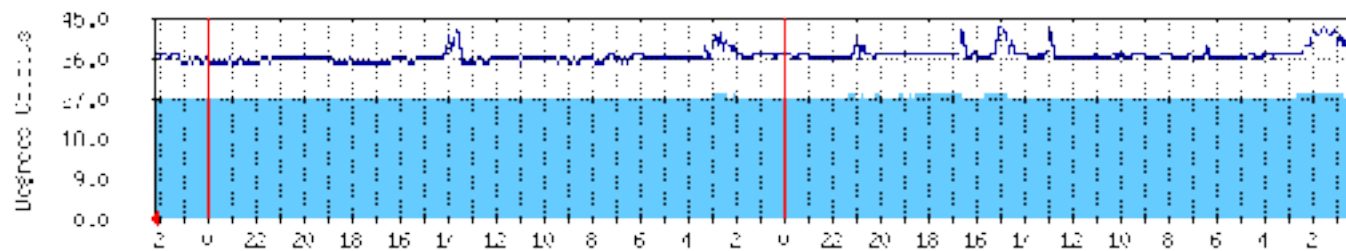
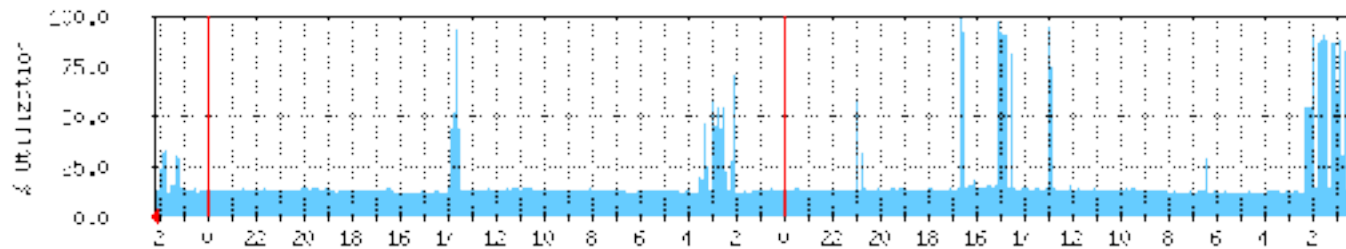
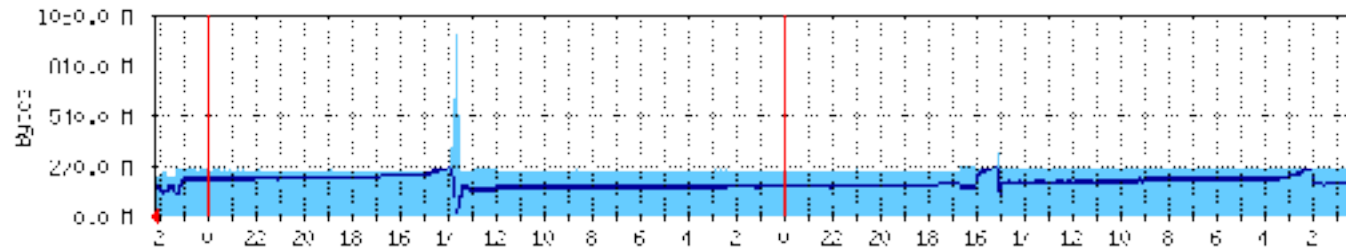
Concurrent management tasks:

- ➔ Action daemon - *automated action (work flow)*
- ➔ Collection daemon - *collects data*
- ➔ Capability daemon - *capability check on nodes*
- ➔ DHCP daemon - *DHCP client for OpenNMS*
- ➔ Discovery daemon - *initial and ongoing discovery*
- ➔ Events manager daemon – *manages/stores events*
- ➔ Notification daemon - *external notification of users*
- ➔ Outage manager daemon - *consolidates events*
- ➔ Poller daemon - *polls managed nodes/services*
- ➔ RTC manager daemon - *real time availability information*
- ➔ SNMP trap daemon – *handles SNMP traps*
- ➔ Threshold daemon – *monitor for threshold values*

MRTG (Multi Router Traffic Grapher)

- ➔ A tool to monitor the traffic load on network
- ➔ Generates HTML with PNG reports
- ➔ Provides a LIVE visual representation of traffic
- ➔ Allows monitoring and analysis of many data center functions (router, server, latency, utilization, temperature, etc.)
- ➔ Countless ways to utilize for data visualization
- ➔ Released under GPL

MRTG Example Output



NeDi

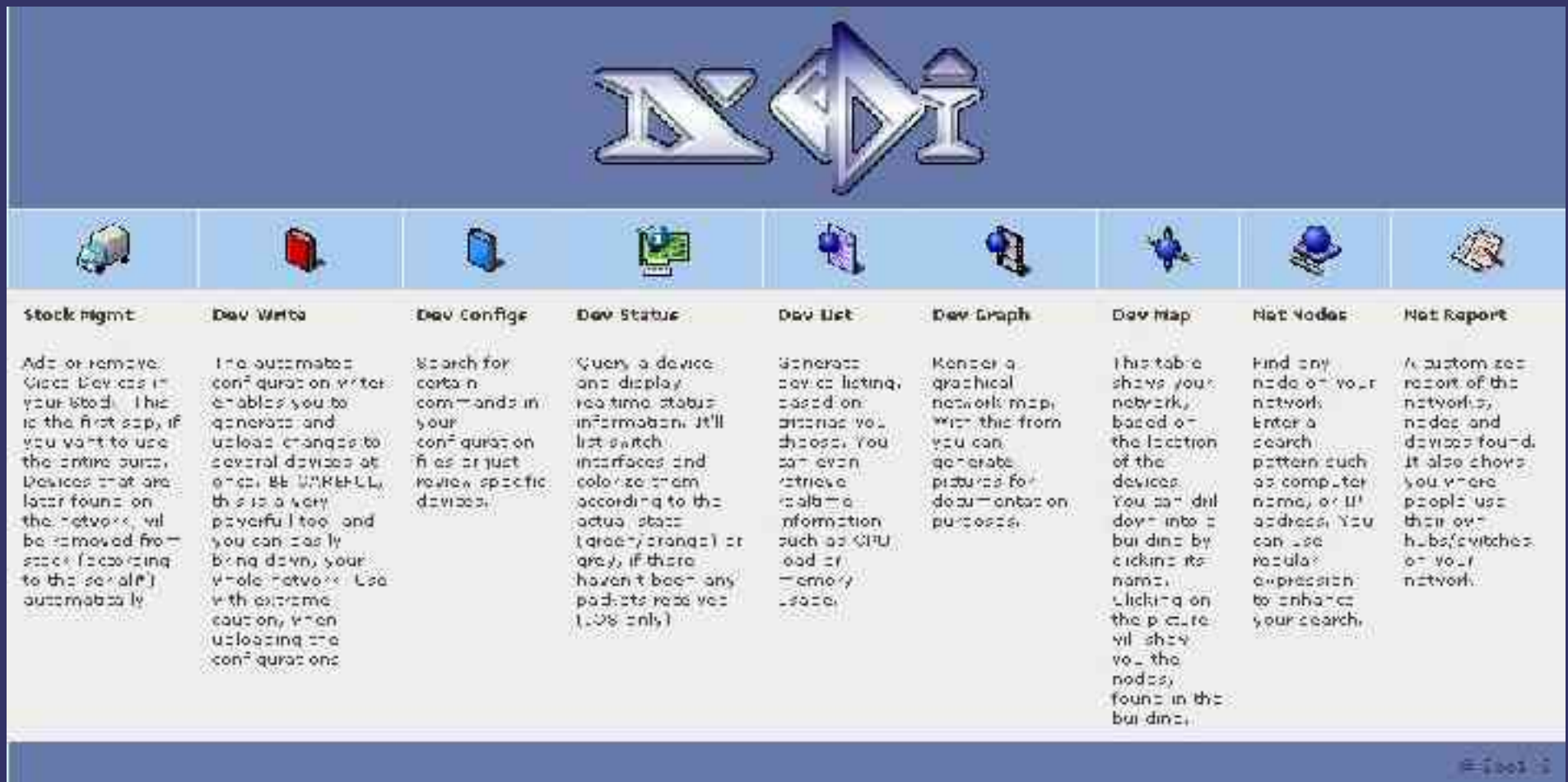
➔ **NeDi** - a perl based network discovery & management system for Cisco components.

Features

- Centralized management
- Manage device configurations
- Push out new configurations
- Intuitive console
- Accessed via web browser
- Graph devices
- Lists devices
- Supports IOS and CatOS

NeDi Example Screen shots

Main Page



The screenshot shows the NeDi main page with a blue header containing the NeDi logo. Below the header is a navigation bar with nine icons and labels: Stock Mgmt, Dev Write, Dev Config, Dev Status, Dev List, Dev Graph, Dev Map, Net Nodes, and Net Report. Each icon is a small 3D-style graphic representing its function.

Stock Mgmt	Dev Write	Dev Config	Dev Status	Dev List	Dev Graph	Dev Map	Net Nodes	Net Report
<p>Add or remove Cisco devices in your stock. This is the first step, if you want to use the entire built. Devices that are later found on the network will be removed from stock (according to the serial#) automatically.</p>	<p>The automated configuration writer enables you to generate and upload changes to several devices at once. BE CAREFUL, this is a very powerful tool and you can easily bring down your whole network. Use with extreme caution, when uploading the configuration.</p>	<p>Search for certain commands in your configuration files or just review specific devices.</p>	<p>Query a device and display real-time status information. It'll list switch interfaces and colorize them according to the actual state (green/orange) or grey, if there haven't been any packets received (LOS only).</p>	<p>Generate device listing, based on address you choose. You can even retrieve real-time information such as CPU load or memory usage.</p>	<p>Render a graphical network map, with this from you can generate pictures for documentation purposes.</p>	<p>This table shows your network, based on the location of the devices. You can drill down into a building by clicking its name. Clicking on the picture will show you the nodes found in the building.</p>	<p>Find any node on your network. Enter a search pattern such as computer name, or IP address. You can use regular expression to enhance your search.</p>	<p>A customised report of the network, nodes and devices found. It also shows you where people use their own hubs/switches on your network.</p>

#0001

NeDi Example Screen shots

NeDi Device Report

The screenshot displays the NeDi Network Report interface. At the top is a navigation bar with icons for Home, Home, Wlfc, Config, Status, List, Net, Tools, Tools, and Report. Below this is the 'Network Report' section, which includes a 'Take the Reports and wish to generate' button, a 'Device Summary' dropdown menu (set to 'Most used switch used in switches'), a 'Node Summary' dropdown menu (set to 'Switch - sum 12 - YL 2016 MAU per port IIG vendor chart'), and a 'Subnet Population' dropdown menu (set to 'Wireless Devices - cooler nodes'). A 'Generate' button is located to the right of these dropdowns.

The 'Device Summary' section shows the 'Last Update of Devices' as 'Wed Jul 2 10:20:03 2008'. Below this is a table listing various device models and their counts.

Device Model	Count
Cat2950-32G	10
Cat2950-24G	10
Cat2950-48G	11
Cat3508gd	3
Cat3524d	24
Cat3540	53
Cat3750-32T	3
Cat6506	6
Cat6509	2

Nagios

➔ **Nagios** – a host and service monitor

- Accessed via web browser
- Services (POP, PING, HTTP, etc)
- Host resources
- Environmental factors
- Option of distributed monitoring
- Acknowledge issues via web interface
- Notification / event handlers
- Modular, allows for plug-ins
- Released under GPL

Nagios Example Screen Shots

Tactical Overview

The screenshot displays the Nagios web interface with a tactical overview. The interface is organized into several sections:

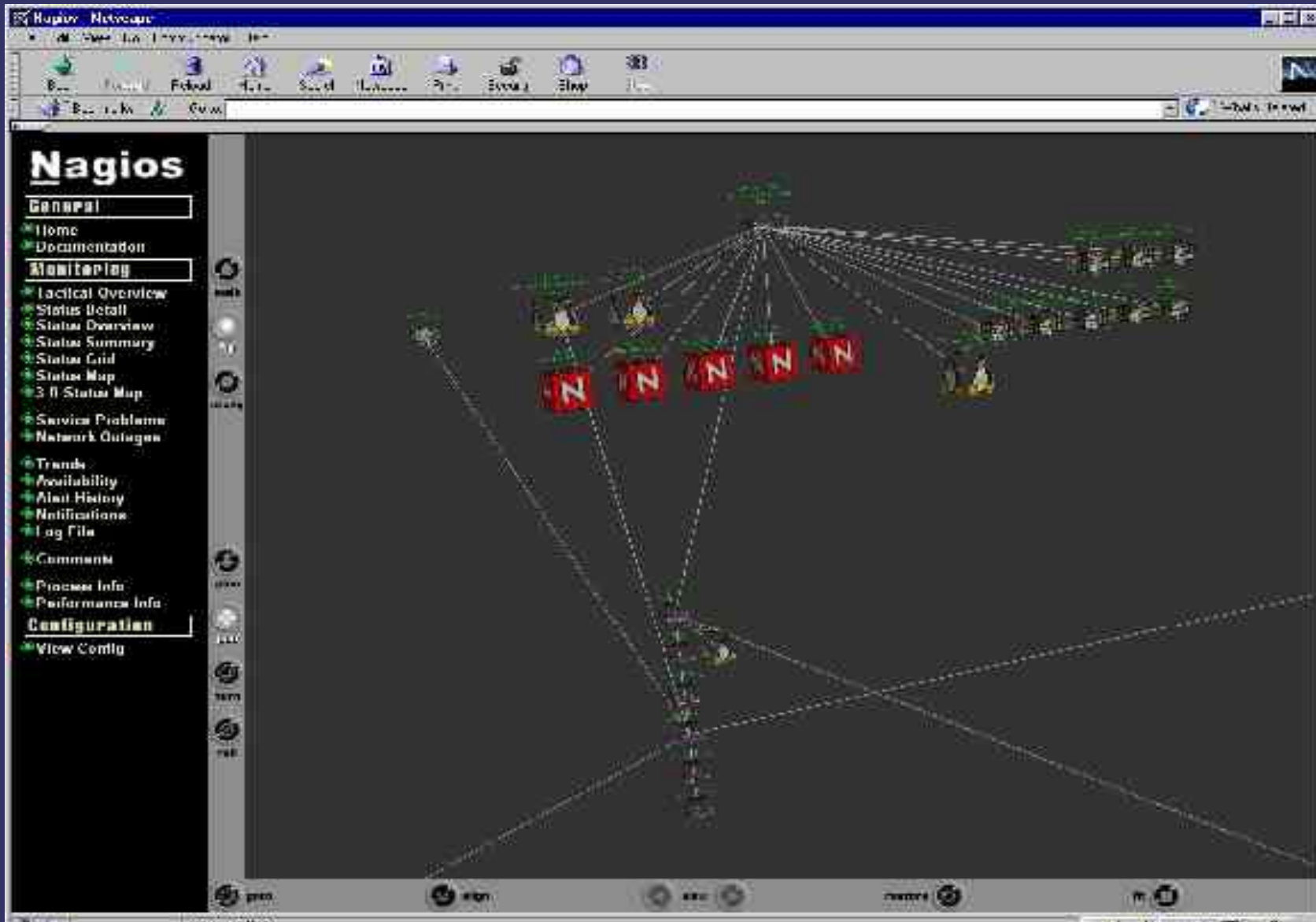
- General:** Includes links for Home, Documentation, Monitoring, System Changes, Hosts, Services, Monitoring Features, and Configuration.
- Monitoring:** Contains sub-sections for Logical Overview, Status Detail, Status Overview, Status Summary, Status List, Status Map, 3-D Status Map, Service Problems, and Network Outages.
- Trends:** Includes links for Availability, Alert History, Notifications, Log File, Comments, and Notifications.
- Performance:** Includes links for Process Info and Performance Info.
- Configuration:** Includes a link for View Config.

The main content area provides a summary of system status:

- Latest Monitoring Overview:** A box containing the latest monitoring data, including the current time and date.
- System Changes:** A box showing 2 changes.
- Hosts:** A table showing the status of hosts: 3 Down, 1 Unreachable, 28 Up, and 0 Pending.
- Services:** A table showing the status of services: 14 Critical, 2 Warning, 6 Unknown, 103 OK, and 11 Pending.
- Monitoring Features:** A table showing the status of monitoring features: Flip Detection, Notifications, Event Handlers, Active Checks, and Passive Checks.
- Monitoring Performance:** A box showing performance metrics: Check Execution Time: 0:50:35.42 sec, Check Latency: 0:1:17.70 sec, Active Checks: 137, and Passive Checks: 0.
- Network Health:** A box showing Host Health and Service Health, both represented by yellow bars.

Nagios Example Screen Shots

3D Overview



Webmin

- ➔ **Webmin** - a web-based interface for system administration for Unix
 - Administrate multiple servers via a web browser
 - Configure/monitor Shorewall firewalls
 - Manage http, DNS, Clusters, DHCP, POP, SMTP, etc.
 - Modular design using plug-ins
 - Released under GPL

Webmin Example Screen Shots

The screenshot shows a Netscape browser window titled "Apache Webserver - Netscape" with the address bar containing "http://fudu:10000/rapid/". The browser's bookmark bar shows "Webmin on fud...", "Webmin on flori...", "Webmin on lenL", and "User Mailboxes".

Global Configuration

The Global Configuration section contains ten icons with corresponding links:

- [Process and Units](#)
- [Networking and Addresses](#)
- [Apache Modules](#)
- [MIME Types](#)
- [Miscellaneous](#)
- [CGI Programs](#)
- [Per-Directory Options Files](#)
- [Re-Configure Known Modules](#)
- [User Defined Parameters](#)
- [Edit Config Files](#)

Virtual Servers

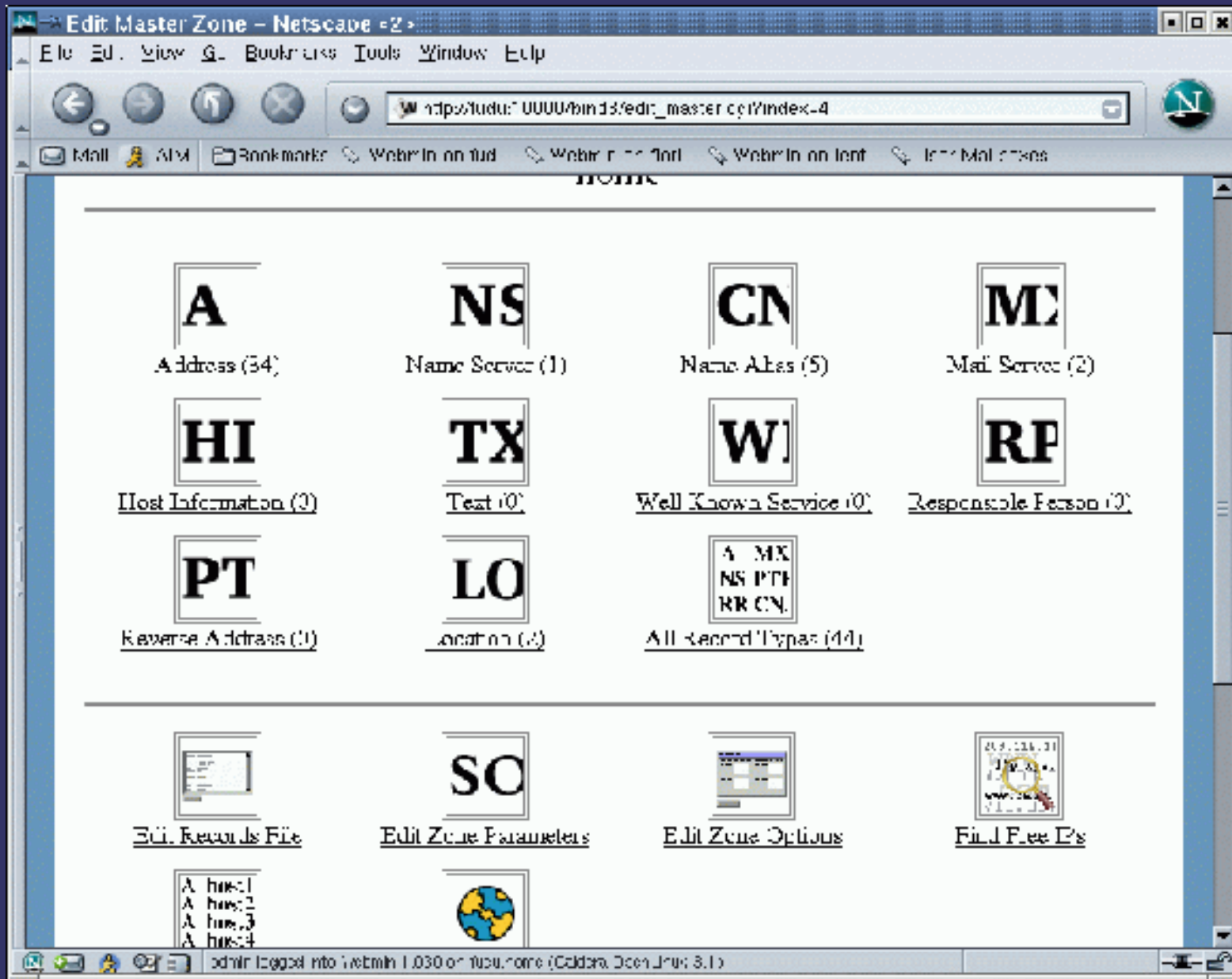
The Virtual Servers section shows the configuration for the "Default Server":

- Default Server**
- Address Any**
- Port Any**
- Server Name** `Automatic`
- Document Root** `/home/httpd/ahnl`

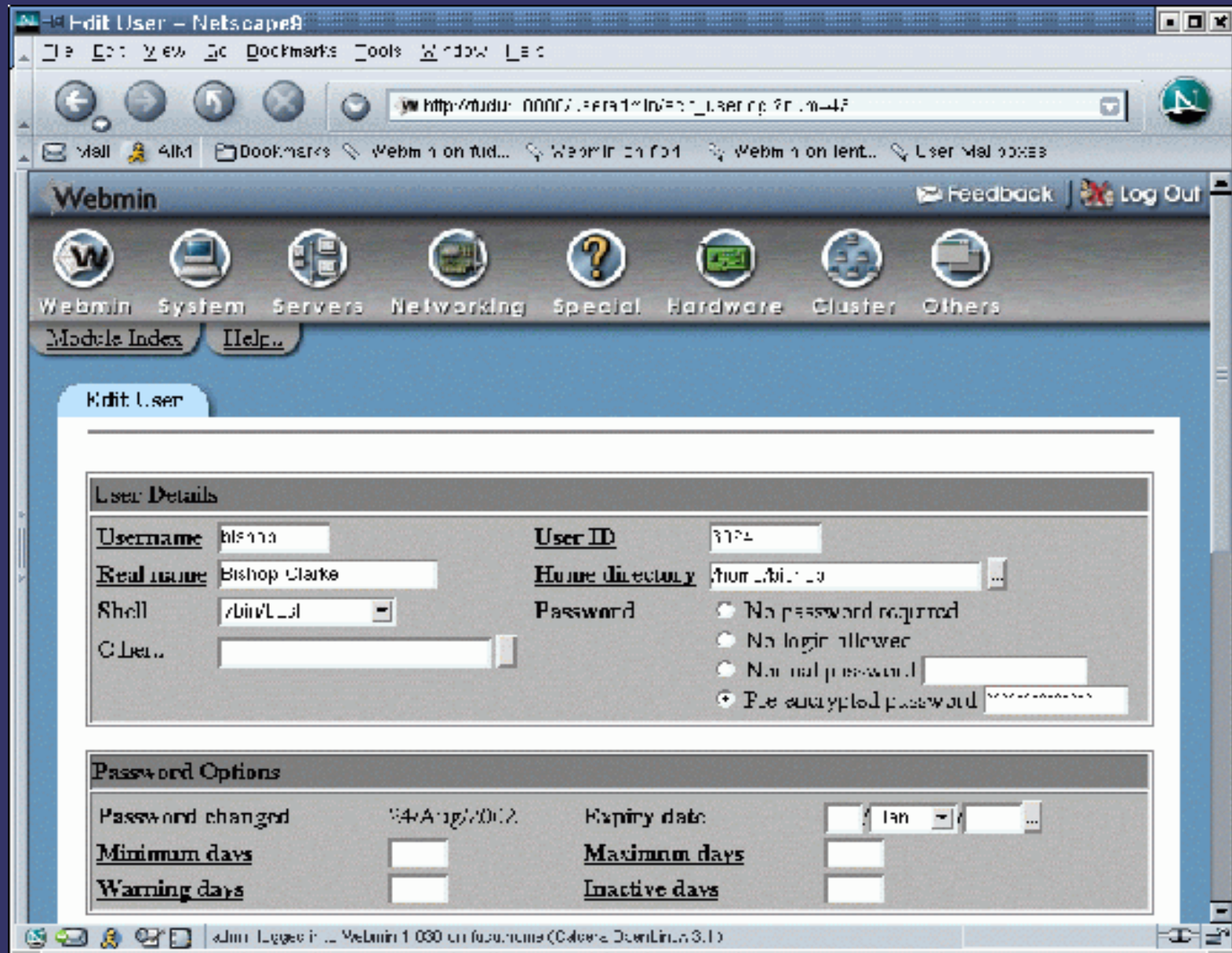
Below this, there is a partially visible entry for another virtual server: "Handles the main based server fudu.courtybracket.com on all addresses".

The bottom status bar of the browser indicates: "admin logged into webmin 1:030 on fudu.home (Cadera OpenLinux 3.1)".

Webmin Example Screen Shots



Webmin Example Screen Shots



LogWatch

LogWatch – analyzes and reports on system logs

- Customizable and pluggable log-monitoring
- Reports on given parameters over given period

SWATCH

- ➔ **Swatch** – (Simple WATCHer) of log files
 - Monitors virtually any log file
 - Reports on given parameters over given period
 - Easily customized

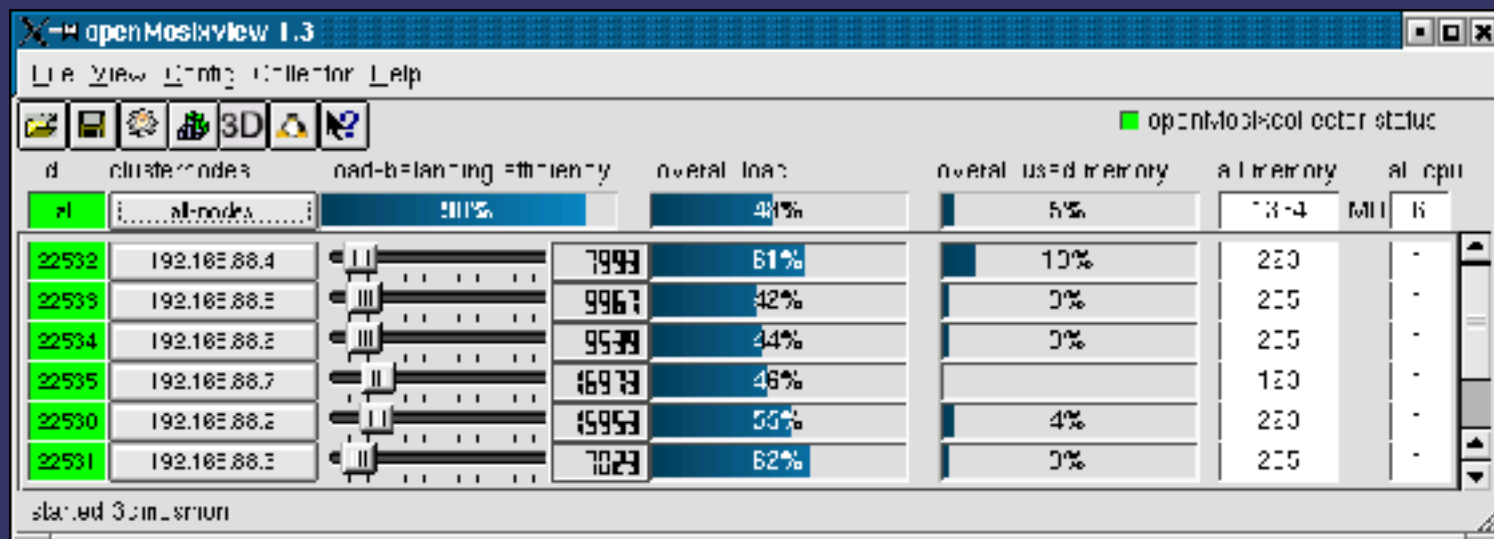
SEC – Simple Event Correlation

- ➔ Fill the gap between commercial and homegrown solutions
- ➔ Event correlation engine for HP OpenView NNM
- ➔ Event correlation engine for HP OpenView ITO management server and agents
- ➔ Event management for CiscoWorks
- ➔ Event consolidation and correlation for Snort IDS
- ➔ Logfile monitoring and analysis (used in place of Swatch and Logwatch)

openMosix / openMosixview

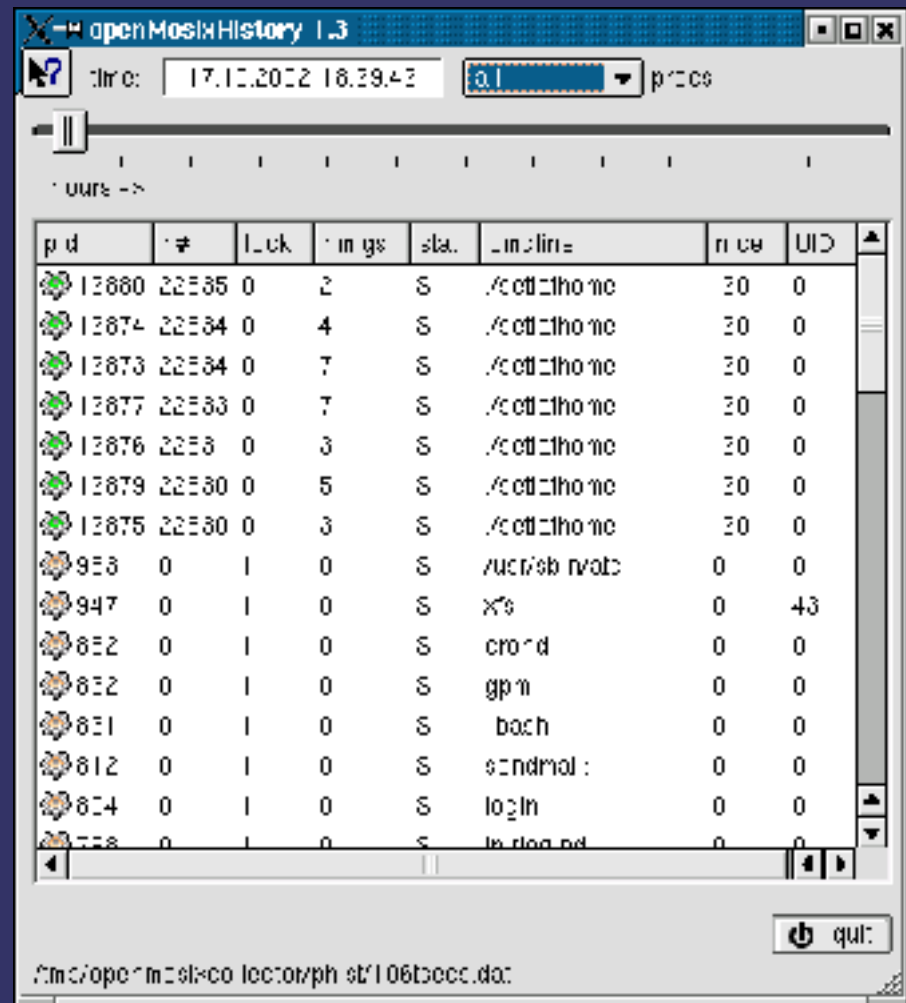
- ➔ **openMosix** - a Linux kernel extension for single-image clustering
- ➔ **openMosixview** – GUI for management of openMosix clusters

Main Window



OpenMosixview Screen Shots

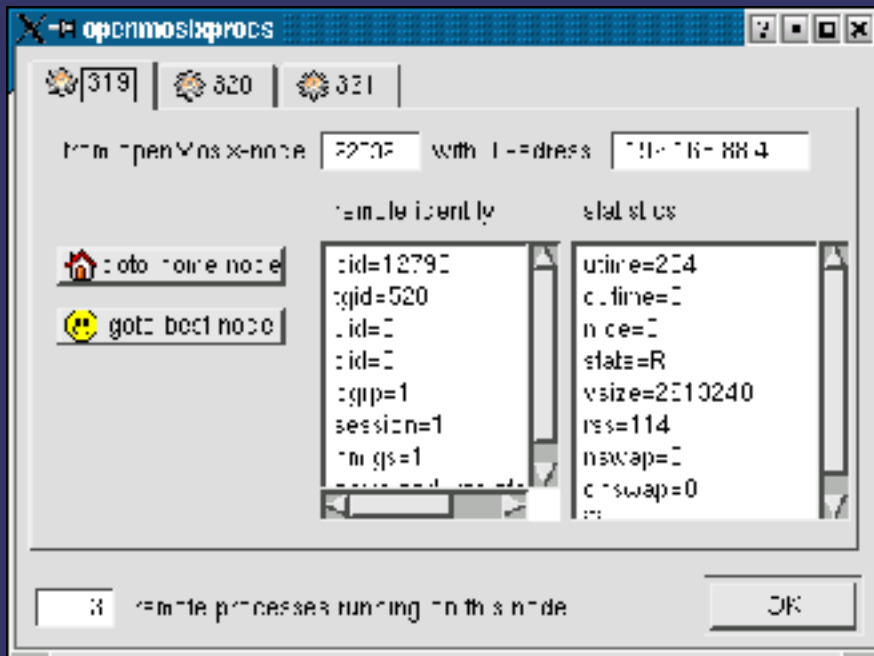
History



The screenshot shows the 'openMosixHistory 1.3' window. At the top, there is a search bar with 'dir:' set to '17.10.2012 16:29:43' and a dropdown menu set to 'all'. Below the search bar is a table with columns: pid, ppid, lock, nprocs, stat, cmdline, nprocs, and UID. The table lists various processes running on the system, including /etc/crontab, /usr/sbin/rsyncd, xfs, cron, gpm, bash, and sshd.

pid	ppid	lock	nprocs	stat	cmdline	nprocs	UID
12880	22535	0	2	S	/etc/crontab	20	0
12874	22534	0	4	S	/etc/crontab	20	0
12873	22534	0	7	S	/etc/crontab	20	0
12877	22533	0	7	S	/etc/crontab	20	0
12876	2253	0	3	S	/etc/crontab	20	0
12879	22530	0	5	S	/etc/crontab	20	0
12875	22530	0	3	S	/etc/crontab	20	0
953	0	1	0	S	/usr/sbin/rsyncd	0	0
947	0	1	0	S	xfs	0	43
852	0	1	0	S	cron	0	0
852	0	1	0	S	gpm	0	0
831	0	1	0	S	bash	0	0
812	0	1	0	S	sshd	0	0
814	0	1	0	S	login	0	0
728	0	1	0	S	sshd	0	0

Remote Processes



The screenshot shows the 'openMosixprocs' window. At the top, there are three icons labeled 319, 320, and 321. Below them, there is a search bar with 'from openMosix-node' set to '22732' and 'with IP-address' set to '192.168.1.104'. The main area is divided into two columns: 'remote idently' and 'stat status'. The 'remote idently' column contains fields like pid=12790, ppid=520, uid=0, gid=1, session=1, nprocs=1, and cswap=0. The 'stat status' column contains fields like utime=214, ctime=0, nprocs=0, stats=R, vsize=2110240, rss=114, nswap=0, and cswap=0. At the bottom, there is a checkbox for 'remote processes running on this node' and an 'OK' button.

Etherape

- ➔ **Etherape** - graphical network monitor for Unix modeled after etherman
- ➔ Network traffic is displayed graphically
- ➔ 'Top Talkers' represented most
- ➔ Select protocol stack of focus
- ➔ Network filters
- ➔ View internal traffic, end to end IP, or port to port TCP
- ➔ Can read saved tcpdump file
- ➔ Most protocols supported

Etherape Screen Shot

The screenshot displays the Etherape application interface. The main window shows a network diagram with nodes and connections. A red node is highlighted, and a tooltip is visible over it. The interface includes a menu bar (File, Capture, View, Help), a toolbar with Start, Pause, Stop, and Pref buttons, and a status bar at the bottom.

Network Diagram: The diagram shows a central red node (R001) connected to several other nodes. The nodes are labeled with IP addresses and names, such as 172.16.1.255, 172.16.1.256, 172.16.1.257, 172.16.1.258, 172.16.1.259, 172.16.1.260, 172.16.1.261, 172.16.1.262, 172.16.1.263, 172.16.1.264, 172.16.1.265, 172.16.1.266, 172.16.1.267, 172.16.1.268, 172.16.1.269, 172.16.1.270, 172.16.1.271, 172.16.1.272, 172.16.1.273, 172.16.1.274, 172.16.1.275, 172.16.1.276, 172.16.1.277, 172.16.1.278, 172.16.1.279, 172.16.1.280, 172.16.1.281, 172.16.1.282, 172.16.1.283, 172.16.1.284, 172.16.1.285, 172.16.1.286, 172.16.1.287, 172.16.1.288, 172.16.1.289, 172.16.1.290, 172.16.1.291, 172.16.1.292, 172.16.1.293, 172.16.1.294, 172.16.1.295, 172.16.1.296, 172.16.1.297, 172.16.1.298, 172.16.1.299, 172.16.1.300. The connections are represented by colored lines (red, yellow, blue, white) and arrows.

Left Panel (Statistics):

Statistic	Value
Inst. Inbound	Accumulated
Inst. Outbound	Accumulated
Inst. Inbound	0 kbps
Inst. Outbound	0 kbps
Inst. Inbound	0 bytes
Inst. Outbound	0 bytes

Bottom Panel (Protocol Stack Level):

Protocol Stack Level	Node size estimate
...	...

Right Panel (Node Properties):

Property	Value
Name	R001
Numeric Name	R001 (with location/behavior)
Inst. Inbound	Accumulated
Inst. Outbound	Accumulated
Inst. Inbound	0 kbps
Inst. Outbound	0 kbps
Inst. Inbound	0 bytes
Inst. Outbound	0 bytes

Bottom Bar: The bottom bar shows the system tray with icons for network, volume, and other applications. The system clock displays 10:01 AM on 11 Apr 03.

Auditing

- ➔ Nessus
- ➔ NMAP
- ➔ Inprotect
- ➔ Ethereal

Nessus

- ➔ **Nessus** - easy to use remote security scanner
- ➔ Software to remotely audit a given network
- ➔ Detects service on non-standard ports
- ➔ Will try to exploit remote service vulnerabilities
- ➔ Utilizes plug-ins
- ➔ Very up to date
- ➔ NASL (Nessus Attack Scripting Language)
- ➔ Client-server architecture
- ➔ Can test multiple host simultaneously
- ➔ Exportable reports in multiple formats
- ➔ Smart service recognition



Nmap

- ➔ **Nmap** - utility for network exploration or security auditing
- ➔ Can rapidly scan large networks
- ➔ Detects application name and version
- ➔ Detects OS version
- ➔ Detects firewalls etc.
- ➔ Easy to use

Inprotect

- ➔ **Inprotect** - web interface for Nessus and Nmap security scanners
- ➔ Manual and scheduled security scans
- ➔ Easy to use web browser interface
- ➔ Administrate regular network scans easily

Ethereal

- ➔ **Ethereal** - network protocol analyzer (sniffer)
- ➔ Examine data from a live network
- ➔ Examine saved capture file
- ➔ Supports many capture formats
- ➔ Intuitive interface
- ➔ View reconstructed TCP sessions
- ➔ Easy to use filters

Ethereal Screen Shot

The screenshot shows the 'The Ethereal Network Analyzer' window. The top menu bar includes 'File', 'Edit', 'Capture', 'Display', 'Tools', and 'Help'. Below the menu is a table of captured packets. The table has columns for 'No.', 'Len', 'Time', 'Source', 'Destination', 'Protocol', and 'Info'. Packet 9 is highlighted in blue. Below the table, the packet details for packet 9 are shown in a tree view. The tree view shows the following structure:

- Frame 977 on wire (977 captured)
 - Ethernet II
 - Internet Protocol
 - User Datagram Protocol
 - DNS Query
 - Transaction ID: 66000
 - Flags: 0x0000 (Standard query)
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 0
 - Queries:
 - www.brunching.com: type A, class IN

The detailed view for the query shows:

- Name: www.brunching.com
- Type: host address
- Class: INET

At the bottom of the window, there is a 'Filter:' field and a file name 'file: dns.pcap'.

No.	Len	Time	Source	Destination	Protocol	Info
1	77	0.000000	24.94.100.11	pcw.zing.org	DNS (UDP)	Standard query
2	77	0.010000	pcw.zing.org	f.root-servers.net	DNS (UDP)	Standard query
3	164	0.060000	f.root-servers.net	pcw.zing.org	DNS (UDP)	Standard query response
4	70	0.070000	pcw.zing.org	f.root-servers.net	DNS (UDP)	Standard query
5	71	0.080000	pcw.zing.org	f.root-servers.net	DNS (UDP)	Standard query
6	161	0.120000	f.root-servers.net	pcw.zing.org	DNS (UDP)	Standard query response
7	158	0.130000	f.root-servers.net	pcw.zing.org	DNS (UDP)	Standard query response
8	77	9.890564	24.94.186.88	pcw.zing.org	DNS (UDP)	Standard query
9	77	9.890904	pcw.zing.org	1.2.1.inet	DNS (UDP)	Standard query
10	148	10.090564	1.2.1.inet	pcw.zing.org	DNS (UDP)	Standard query response
11	148	10.090904	pcw.zing.org	24.94.186.99	DNS (UDP)	Standard query response

SPAM/Virus/Worm Filtering

- ⇒ Spam Assassin
- ⇒ ClamAV
- ⇒ MailScanner

SPAM Assassin

- ➔ **Spam Assassin** – rule based mail filter to identify Spam
- ➔ Header analysis
- ➔ Text analysis
- ➔ Blacklists
- ➔ Razor - (collaborative Spam-tracking database)

ClamAV

- ⇒ **ClamAV** - anti-virus toolkit for UNIX
- ⇒ Command-line scanner
- ⇒ Fast, multi-threaded daemon
- ⇒ Simple database updater
- ⇒ Virus scanner C library
- ⇒ On-access scanning (Linux)
- ⇒ Detection of over 10000 viruses, worms and Trojans
- ⇒ Built-in support for RAR (2.0), Zip, Gzip, etc.
- ⇒ Up to date signature additions

Mail Scanner

- ➔ **Mail Scanner** - scans all e-mail for viruses, Spam and attacks against security vulnerabilities
- ➔ Protects over 5 billion e-mails every week*
- ➔ All in one solution for email scanning
- ➔ any combination of 14 different virus scanners including ClamAV
- ➔ Scalable and robust
- ➔ Automatic updates
- ➔ Multiple Spam identification techniques
- ➔ Released under GPL

Open Source Solution Cost Comparison

- ➔ **Example Network**
- ➔ 50 servers
- ➔ 4000 users
- ➔ 450 VPN users

Proprietary Solution Cost Breakdown

- ➔ **Checkpoint Suite** - VPN-1, FireWall-1, FloodGate-1, Meta IP, Account Management Module, UserAuthority, ConnectControl, Reporting Module, Real-Time Monitor, SecureUpdate and Visual Policy Editor.

Licensing by number of IP addresses = 4500 IPs

- \$200 per 10 IP addresses = \$90,000*
- \$50 per VPN client software (450 users) = \$22,500*
- plus cost of hardware / OS to run it on
- plus cost of renewing license every year
- plus deployment costs
- plus support and updates - "Licensing notes: Support and software updates are not included with CheckPoint products"*

Licensing Total = \$112,500*

Open Source Solution Cost Breakdown

➔ **Linux Firewall** – utilizing Shorewall and FreeS/WAN (for VPN)

IP addresses = 4500 IPs

- **NO** per IP licensing
- **NO** VPN client software licensing
- cost of hardware to run it on
- **NO** cost of renewing license every year
- deployment costs
- **NO** cost of support contracts and updates

Licensing Total = \$0.00

Conclusion

- ➔ Multiple options to choose from
- ➔ You get the source code
- ➔ Manage many data center functions from centralized console
- ➔ Easily create custom solutions to fit needs
- ➔ No licensing costs (per seat, IP, etc)
- ➔ Paid support available if needed
- ➔ Lower cost of ownership
- ➔ Better ROI than many proprietary offerings



LOCALAREASECURITY.COM

Sponsored by

lōgatren

a subsidiary of Barrow Systems Engineering, Inc.