Innovation with ExtremeXOS

Abstract

As today's enterprise networks continue to converge, there is an increasing need for a smarter network operating system that can deliver plug-and-play connectivity, resilient connectivity, and secured connectivity. Historically, network switches and routers have operated on a monolithic operating system.

However, today's converged applications such as voice, video, storage and data require capabilities from a network infrastructure that are beyond the reach of a monolithic operating system.

This informative white paper describes how ExtremeXOS®, Extreme Networks® modular operating system is a key component in building networks that run today's business applications. This paper provides an overview of ExtremeXOS, including the key benefits of a modular operating system.

Make Your Network Mobile

Introduction

Historically, network switches and routers have operated on a monolithic Operating System (OS). However, today's converged applications such as voice, video, storage and data require capabilities from a network infrastructure that are beyond the reach of a monolithic OS. Convergence requires a network OS that can deliver the highest levels of availability and is capable of supporting rapidly changing business needs. Convergence requires a modular and open-standards network OS. Extreme Networks was the first switching vendor to respond to this requirement with a modular operating system – the ExtremeXOS operating system. Individual network components have become the building blocks in delivering a streamlined and effective network capable of meeting the demands of current and future business convergence initiatives.

Limitations of Monolithic Operating Systems

A monolithic OS is a static-compiled software image that uses a single memory address space. A software module (or network feature) cannot be dynamically loaded or unloaded because the software functions are bound together in a monolithic OS. Dynamically adding, changing or removing a network feature is not possible without a reboot of the OS and consequent downtime. A crash in one of the network features will also cause the entire OS to stop or become unstable since the network features and the use of resources are bound to each other. Monolithic OSs have millions of lines of programming code that have to be edited as new software features are required. In fact, their non-modular design adds to the cycle time of adding new software features. Thus, a monolithic OS may not achieve the highest levels of network availability and its operation is unpredictable under failure situations (e.g. software exception errors). It also lacks the ability to quickly incorporate new features to support the required functionality of a converged network infrastructure.

Removing the Barriers

ExtremeXOS removes the bond between the software functions by sectionalizing the many lines of code into multiple layers. As a result, this modular approach delivers a resilient, multi-threaded OS that increases network uptime.

- Predictable operation: ExtremeXOS is designed to be hardware independent and deliver consistent functionality across Extreme Networks hardware platforms.
- Consistent availability: Resilient with software and hardware failover, with the ability to dynamically stop/restart and load/unload software modules without impacting network operation.
- Secure networking: Memory-protected software modules and secure transport protocols guard the network infrastructure through authentication, encryption, integrity and Denial of Service (DoS) protection.
- Scalable operation: New feature-specific software modules can be added to and removed from the main OS. The software infrastructure scales as the network requirements change or grow.
- Extensible functionality: Open, yet secure, to extend network capabilities and manageability to interface with best-of-breed solutions or customer-defined solutions using Extensible Markup Language (XML) and Application Program Interface (API). Native XML infrastructure allows for future extensibility. ExtremeXOS is Portable Operating System Interface (POSIX) compliant.
- Simple to manage: Auto-provision connected users or devices and improve application fluency with the Universal Port framework. Flexible policy management using file-based Access Control Lists (ACLs). Common ExtremeXOS interface that is simple and user-customizable through simple Tool Command Language (Tcl) extensions. Tcl infrastructure allows changing the look and feel of the Command Line Interface (CLI) management.

The following sections summarize the architecture and the features and benefits of ExtremeXOS.

The Architecture

The ExtremeXOS software consists of a hardware abstraction layer, a kernel, kernel-loadable modules, application modules and a configuration management module.

Hardware Abstraction Layer

The Hardware Abstraction Layer (HAL) decouples all the hardware/ASIC control code from the underlying hardware through a message layer. A HAL allows an OS to interact with the hardware at a general level rather than at a detailed, hardware-specific level. To use an analogy from the PC world, Windows NT, BSD, or Linux are several OSs that include a HAL. These OSs have different subsystems for particular functions (e.g. sound and video) that run on a general hardware platform. The HAL makes it possible to add support for new devices and new ways of connecting devices to the computer, without modifying every application that uses the device.

The same concept applies to the HAL in ExtremeXOS. The HAL allows ExtremeXOS to run on different Extreme Networks hardware switching platforms, CPUs, and ASICs.

ExtremeXOS Kernel

ExtremeXOS is built on a powerful kernel that provides the core foundation for modular and portable extensions. The ExtremeXOS kernel is responsible for process management, preemptive multitasking, fair scheduling between processes, dynamic process management, memory management and multi-threading support. In addition, the ExtremeXOS kernel provides file support system and inter-process communication. Components or layers above the kernel communicate directly with one another, using message passing.

Kernel-Loadable Modules

Kernel-loadable modules provide the modular function of the OS kernel. The use of kernel-loadable modules avoids direct modification of the ExtremeXOS kernel. As new functionality is needed in the kernel, a loadable module can be live-upgraded and appended to the ExtremeXOS kernel. A kernel-loadable module provides functionality that may include VLAN mappings, QoS mappings, multicast cache handling and ACL processing, for example. Kernel-loadable modules facilitate the process of adding more value to the kernel.

Application Modules

ExtremeXOS application modules are individual applications that have a specific task. For example, services such as virtualization, Network Login, 802.1x, EAPS, ESRP, VRRP, SSH2, and protocols associated with IPv4 and IPv6 are individual ExtremeXOS applications that operate as a separate entity. Using a PC analogy again, ExtremeXOS loadable application modules are similar to user applications such as Microsoft Excel, or Adobe Photoshop. Each PC application interfaces directly to the OS kernel and the user can load and unload either application without impacting the other.

The same is true for ExtremeXOS application modules. Each ExtremeXOS loadable application module has a distinct function and operates independently. For example, one can dynamically load and unload an SSH2 module without impact to the core kernel and other ExtremeXOS application modules, for example BGP operations. Each ExtremeXOS application module runs in its own memory protected process so that a malfunction in a single application module does not impact the rest of the system.

Configuration Management Modules

ExtremeXOS configuration management modules are responsible for interfacing with the users or management agents and relaying the configuration-related information to the ExtremeXOS applications (e.g. BGP, OSPFv3, etc.). ExtremeXOS configuration management includes the following major functionality:

- Multiple concurrent configuration access methods (Console CLI, XML, SNMP, etc.)
- Saving and retrieving configuration and statistics for the switching platform
- Management of multiple users and their associated security clearance

See Figure 1.

ExtremeXOS brings the latest operating system technology to the networking market and enables various features and benefits that are described in the next section.

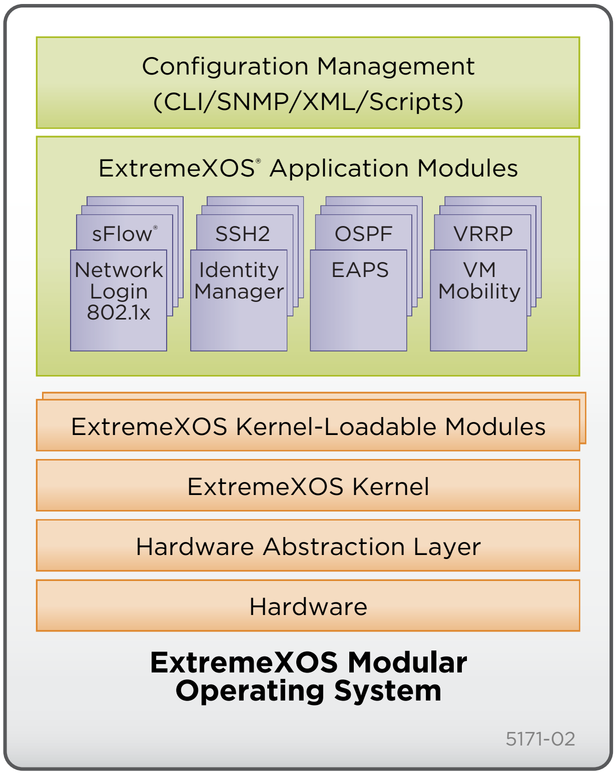


Figure 1

Features and Benefits

ExtremeXOS has significant attributes that make it the platform for emerging and demanding converged applications. They include:

- Modular architecture
- POSIX compliance
- Secure operation with authentication, encryption, integrity and device protection
- Flexible ACL policy language
- Extensible Markup Language (XML)
- Tool Command Language (Tcl)
- Universal Port
- XNV™
- ID Manager
- M-LAG (Multi-Switch Link Aggregation Groups)

Feature: Modular Architecture

Benefit = Available Networking and Scalable Operation

In the previous sections of this document, we have discussed how ExtremeXOS overcomes the restrictions of a monolithic OS by using a modular architecture. One of the benefits of this approach is to allow network administrators the ability to add, change or remove ExtremeXOS loadable application modules, for example the SSH2 feature, without impacting network operation. This capability is critical for converged networks that need the levels of availability historically associated with the public telephone network, where the standard networking practice of rebooting switches to enable a new feature is no longer acceptable.

We first downloaded the SSH2 ExtremeXOS application module onto the BlackDiamond® 8810 switch. While Example 1 is done on the BlackDiamond 8810, the same process can be executed on all ExtremeXOS-based switches. The successful install was verified with the command 'show version image,' The Extreme Networks security framework is designed with flexibility in mind, and control can be handled at the access or core layers depending on the IT organization's requirements and preferences.

```
BD-8810.6 # download image 10.103.0.98
bd10K-11.0.0.23-ssh.xmod
Do you want to install image after downloading?
(y - yes, n - no, <cr> - cancel) Yes
```

Example 1

Dynamic Modularity (See Figure 2)

1. Non-SSH2 ExtremeXOS software image installed. Management (Mgmt) Station B is unable to connect via SSH2.
2. ExtremeXOS pulls the SSH2 module (requests a download) from Mgmt Station A.
3. ExtremeXOS software on the BlackDiamond 8810 dynamically updates its software with SSH2 module.
4. During steps 2 and 3, network connectivity (including EAPS ring) is not broken. Traffic between IXIA ports is maintained with no packet loss, and Voice-over-IP (VoIP) calls are maintained. Wireless communications are maintained.
5. Mgmt Station B is now able to connect to the BlackDiamond 8810 via SSH2.

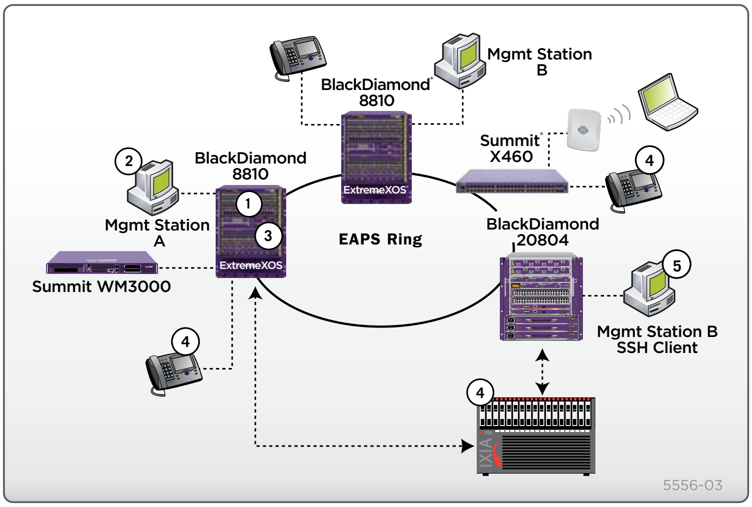


Figure 2. Dynamic Mobility

```
Downloading to MSM-A..
Downloading to MSM-B
Installing to primary partition!
Installing to MSM-B........
Installing to MSM-A.......
Image installed successfully
```

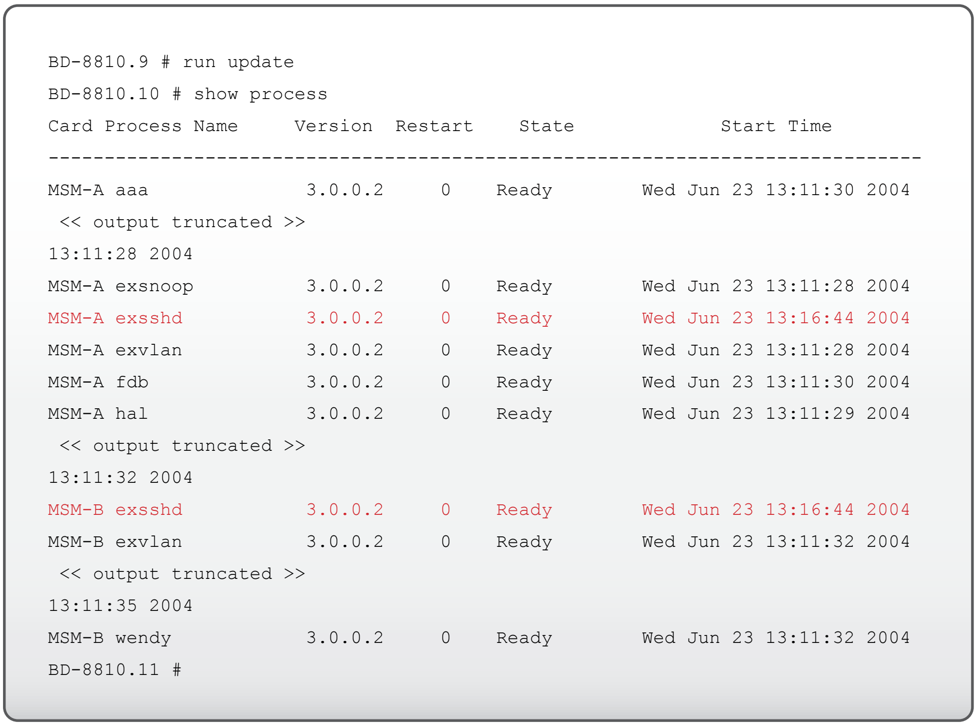
Example 2

```
BD-8810.7 # sh ver image
Card  Partition     Installation Date        Version     Name
----------------------------------------------------------------
MSM-A primary   Tue Jun 22 13:46:23 UTC 2004 11.0.0.23 bd8K-
11.0.0.23.xos
MSM-A primary   Wed Jun 23 13:13:39 UTC 2004 11.0.0.23 bd8K-
11.0.0.23-ssh.xmod
MSM-A secondary Wed May 26 13:59:01 UTC 2004 11.0.0.18 bd8K-
11.0.0.18.xos
MSM-B primary   Tue Jun 22 13:43:16 UTC 2004 11.0.0.23 bd8K-
11.0.0.23.xos
MSM-B primary   Wed Jun 23 13:13:34 UTC 2004 11.0.0.23 bd8K-
11.0.0.23-ssh.xmod
MSM-A secondary Wed May 26 13:59:01 UTC 2004 11.0.0.18 bd8K-
11.0.0.18.xos

BD-8810.8 #
```

Example 3

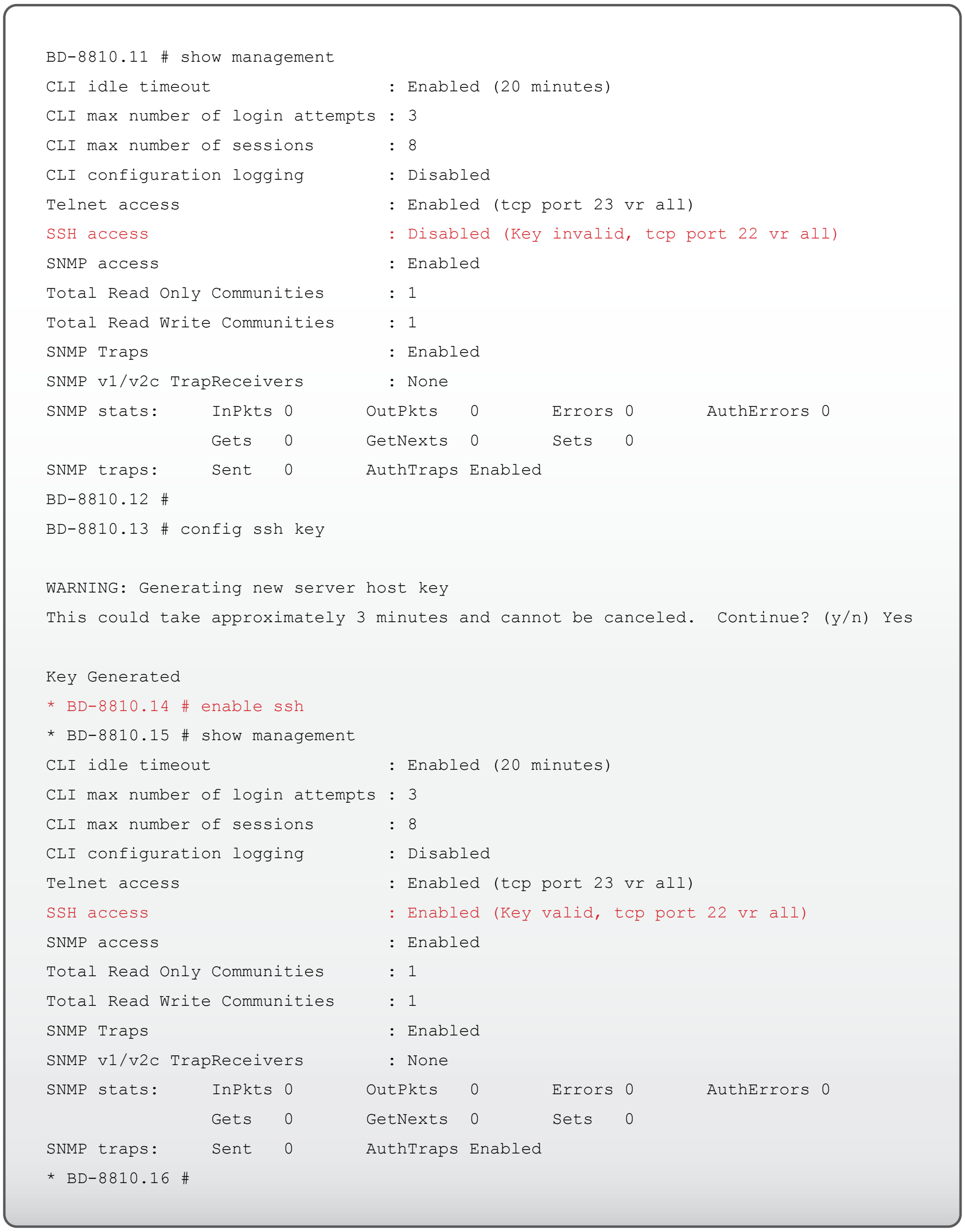
Next, we refreshed the process list and readied the SSH2 module for use by entering the command 'run update'. We verified that the SSH2 process (exsshd) is loaded and in the 'Ready' state on both MSM modules as shown below in Example 4.



```
BD-8810.9 # run update
BD-8810.10 # show process
Card Process Name     Version  Restart    State             Start Time
------------------------------------------------------------------------------
MSM-A aaa              3.0.0.2     0    Ready        Wed Jun 23 13:11:30 2004
 << output truncated >>
13:11:28 2004
MSM-A exsnoop          3.0.0.2     0    Ready        Wed Jun 23 13:11:28 2004
MSM-A exsshd           3.0.0.2     0    Ready        Wed Jun 23 13:16:44 2004
MSM-A exvlan           3.0.0.2     0    Ready        Wed Jun 23 13:11:28 2004
MSM-A fdb              3.0.0.2     0    Ready        Wed Jun 23 13:11:30 2004
MSM-A hal              3.0.0.2     0    Ready        Wed Jun 23 13:11:29 2004
 << output truncated >>
13:11:32 2004
MSM-B exsshd           3.0.0.2     0    Ready        Wed Jun 23 13:16:44 2004
MSM-B exvlan           3.0.0.2     0    Ready        Wed Jun 23 13:11:32 2004
 << output truncated >>
13:11:35 2004
MSM-B wendy            3.0.0.2     0    Ready        Wed Jun 23 13:11:32 2004
BD-8810.11 #
```

Example 4

The SSH2 module is now loaded and ready to be configured. We configured an SSH2 key, which is used to uniquely identify the BlackDiamond 8800 series switches during the client authentication and login process. Once the key is generated, we enabled SSH2 as shown below in Example 5.



```
BD-8810.11 # show management
CLI idle timeout                 : Enabled (20 minutes)
CLI max number of login attempts : 3
CLI max number of sessions       : 8
CLI configuration logging        : Disabled
Telnet access                    : Enabled (tcp port 23 vr all)
SSH access                       : Disabled (Key invalid, tcp port 22 vr all)
SNMP access                      : Enabled
Total Read Only Communities      : 1
Total Read Write Communities     : 1
SNMP Traps                       : Enabled
SNMP v1/v2c TrapReceivers        : None
SNMP stats:     InPkts 0       OutPkts   0       Errors 0       AuthErrors 0
                Gets   0       GetNexts  0       Sets   0
SNMP traps:     Sent   0       AuthTraps Enabled
BD-8810.12 #
BD-8810.13 # config ssh key

WARNING: Generating new server host key
This could take approximately 3 minutes and cannot be canceled.  Continue? (y/n) Yes

Key Generated
* BD-8810.14 # enable ssh
* BD-8810.15 # show management
CLI idle timeout                 : Enabled (20 minutes)
CLI max number of login attempts : 3
CLI max number of sessions       : 8
CLI configuration logging        : Disabled
Telnet access                    : Enabled (tcp port 23 vr all)
SSH access                       : Enabled (Key valid, tcp port 22 vr all)
SNMP access                      : Enabled
Total Read Only Communities      : 1
Total Read Write Communities     : 1
SNMP Traps                       : Enabled
SNMP v1/v2c TrapReceivers        : None
SNMP stats:     InPkts 0       OutPkts   0       Errors 0       AuthErrors 0
                Gets   0       GetNexts  0       Sets   0
SNMP traps:     Sent   0       AuthTraps Enabled
* BD-8810.16 #
```

Example 5

The SSH2 server on the Extreme Networks switches is now ready for use and to be accessed by any SSH2 client.

Upgrade of new software features is a common activity throughout the lifetime of a switching platform. Network administrators no longer have to schedule after-hour upgrades and schedule network downtime.

New software features can be dynamically loaded to the ExtremeXOS OS any time of the day without impacting IT staffing hours or daily network operation.

The SSH2 module can also be uninstalled without impacting network operation. In Example 6, we first terminated the 'exsshd' process as shown using a BlackDiamond 8800 series switch.

```
* BlackDiamond 8810 series# terminate
  process "exsshd" graceful
```

Example 6

Next, we executed the 'uninstall image' command to remove the SSH module from the desired partition as shown in Example 7.

```
BlackDiamond 8810 series # uninstall image orion-11.1.1.3-ssh.xmod primary
Uninstallation of the EXOS module
Uninstalling from primary partition!
./bin/exsshd exos
./config/clidef/exsshd.xml exos
BlackDiamond 8810 series # sho management
CLI idle timeout                 : Enabled (20 minutes)
CLI max number of login attempts : 3
CLI max number of sessions       : 8
CLI paging                       : Enabled (this session only)
CLI space-completion             : Disabled (this session only)
CLI configuration logging        : Disabled
Telnet access                    : Enabled (tcp port 23 vr all)
SSH Access                       : ssh module not loaded.
SNMP access                      : Enabled
Total Read Only Communities      : 1
Total Read Write Communities     : 1
RMON                             : Disabled
SNMP Traps                       : Enabled
SNMP v1/v2c TrapReceivers        : None
BlackDiamond 8810 series # show version images
Card Partition Installation Date Version Name
--------------------------------------------------------------------
MSM-A primary Fri Nov 19 16:20:33 UTC 2004 11.1.1.3 orion-11.1.1.3.xos
MSM-B secondary Fri Nov 5 15:35:41 UTC 2004 11.1.0.25 orion-11.1.1.2.xos
MSM-B primary Fri Nov 19 16:16:16 UTC 2004 11.1.1.3 orion-11.1.1.3.xos
MSM-A secondary Fri Nov 5 15:40:59 UTC 2004 11.1.0.25 orion-11.1.1.2.xos
```

Example 7

The SSH2 module is now removed. TCP port 22 is no longer listening or available.

We terminated a process and unloaded an application module, while maintaining full switch operation on both the

BlackDiamond 8800 series switches. Our wireless connections continued to operate seamlessly and our VoIP calls neither dropped nor showed any quality loss.

The modular function of ExtremeXOS also allows for self-healing process recovery. Each software process is monitored for its health status and a preconfigured recovery action is taken when a process becomes unresponsive. A recovery action can involve a restart of a process without impacting other processes, or a failover to the backup MSM, where a switch reboot is not required. A system trap message is then sent to the operator for notification of the exception. The operator can then take the appropriate actions to address the underlying problem that caused the process restart.

A simple demonstration proves the point. We begin by enabling Telnet on an Extreme Networks switch. Users access the switch via Telnet to verify its operation. The Telnet service is then terminated to simulate a process crash. ExtremeXOS detects the problem and automatically restarts the Telnet service without manual user intervention. Users are able to continue managing the switch via Telnet without taking the switch offline or rebooting (see Figure 3).

Self-healing Process

1. Telnet is enabled on the ExtremeXOS software image.
2. Users access the switch via Telnet to verify it is enabled.
3. The Telnet service is then terminated to simulate a process crash or DoS attack against the switch's Telnet service.
4. ExtremeXOS detects the problem and automatically restarts the Telnet service without manual user intervention or requiring a switch reboot.
5. During these steps, all other network traffic forwarding is unaffected.
6. If this was done on a monolithic OS, the entire OS package would crash or lock up. A switch reboot would then take place. All network communication would be temporarily lost.

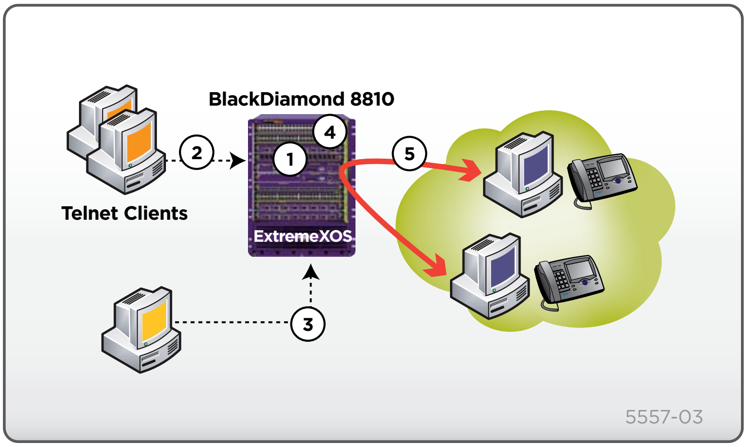
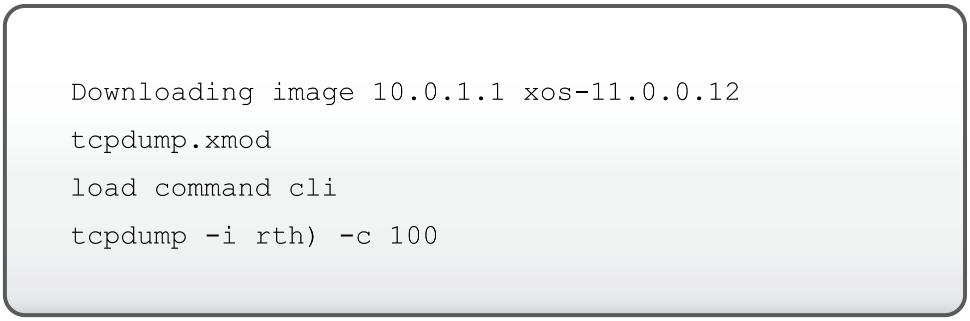


Figure 3. Self-healing Process

Feature: POSIX Compliance

Benefit = Extensible Functionality

POSIX is an open API based on the UNIX operating system architecture. POSIX is an IEEE standard and is accepted by the ISO and ANSI standards bodies. POSIX support ensures code portability between systems and is becoming an increasing requirement for inter-network computing and government contracts, especially with regard to Federal Information Processing Standard 151-2 (FIPS-151-2). The Portable Applications Standards Committee is responsible for defining POSIX and is a part of the IEEE computer society. Extreme Networks offers this functionality in a networking infrastructure to allow seamless porting of POSIX-compliant third party and open development community applications, loadable modules and services. For example, Extreme Networks can port a well-known POSIX-compliant tool such as tcpdump within ExtremeXOS. As we have demonstrated in the previous section, we can also dynamically load the tcpdump ExtremeXOS application module by executing the following commands as shown in Example 8.



```
Downloading image 10.0.1.1 xos-11.0.0.12
tcpdump.xmod
load command cli
tcpdump -i rth) -c 100
```

Example 8

Feature: Authentication, Encryption, Integrity, Protection

Benefit = Secure Networking and Operational Efficiency

While a modular and an open-standards OS provide the required flexibility, ExtremeXOS architecture is resilient against users with malicious intent.

ExtremeXOS delivers on four elements to enable a safe network operation: Authentication, Encryption, Integrity, and Protection.

Authenticating Trusted Users

Before sending or receiving encrypted communication to and from the switch, ExtremeXOS confirms that the person on the other end of the line is the person he or she claims to be (authentication). ExtremeXOS provides three methods to authenticate users accessing the switch:

- RADIUS Client: Remote Authentication Dial In User Service (RADIUS, RFC 2138) is a mechanism for authenticating and centrally administrating access to network nodes. The ExtremeXOS RADIUS client implementation enables authentication for Telnet or console access to the switch.
- TACACS+: Terminal Access Controller Access Control System Plus (TACACS+) is a mechanism for providing authentication, authorization, and accounting on a centralized server, similar in function to the RADIUS client. The ExtremeXOS version of TACACS+ is used to authenticate prospective users who are attempting to administer the switch. TACACS+ is used to communicate between the switch and an authentication database.
- Local database of accounts and passwords: ExtremeXOS supports two levels of management accounts: User and Administrator. A user-level account can view but not change all manageable parameters, with the exception of the user account database and SNMP community strings. An administrator-level account can view and change all manageable parameters.
- Identity Manager: Identity Manager provides role-based access control at the network level. It gathers information about users, devices, and location and then allows IT administrators to create granular access policies for employees or third-party users. The result is tighter network access, easier security/compliance audits, and greater business flexibility.

ExtremeXOS has the ability to authenticate users to the network with a feature called Network Login. Network Login enables the operator to control network access to users that are properly authenticated. Network Login is controlled on a per-port, per-VLAN basis. When Network Login is enabled on a port in a VLAN, that port does not forward any traffic until authentication takes place.

Network Login encompasses two types of active authentication, Web-based and 802.1x. Unlike 802.1x, the unique Web-based Network Login does not require any specific client software and can work with any HTTP-compliant Web browser. A clientless authentication simplifies the way IT manages the desktop.

When the Web-based Network Login is enabled on a switch port, that port is placed into a non-forwarding state until authentication takes place. To authenticate network access, a user is only required to open a Web browser and provide the appropriate credentials. These credentials are either approved, in which case the port is placed in forwarding mode, or not approved, in which case the port remains blocked. Three failed login attempts will disable the port for a configured length of time.

As seen in Figure 4, we demonstrate what a typical user would experience with the three-step client-less Web-based network authentication.

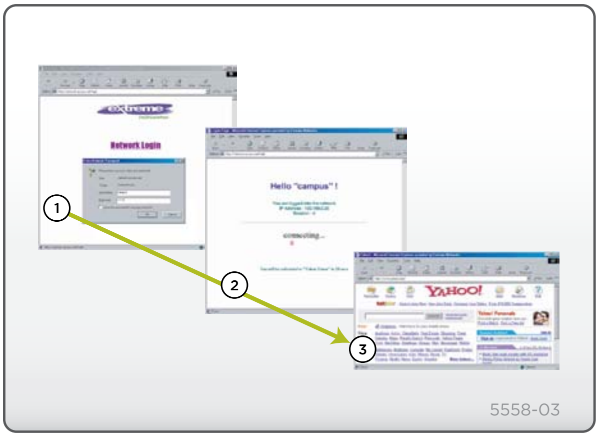


Figure 4: Network Login

Network Login

1. Login attempt. User attempts to access the network. The user is redirected to the Network Login page.
2. Successful login. User is successfully authenticated upon validation of user name and password.
3. Redirect after authentication. User is redirected to the requested URL page.

Step 1: The user attempts to access the network. An Extreme Networks switch intercepts any URL headed for a server on the internal network or out to the Internet (e.g. www.yahoo.com). If the user has not yet been authenticated, the switch performs "URL hijacking" by returning a logon screen to the user.

Step 2: The user's credentials are validated against the authentication, authorization, and accounting server. The user is successfully authenticated upon validation of user name and password.

Step 3: After successful authentication, the user is redirected to the requested URL page.

Network Login also ensures an additional level of security with multiple-supplicant authentication for shared port configurations. A single user authentication will not be able to open that port for all other users when the switched port is shared.

With the multiple-supplicant authentication, every client attached to the same switch port must be authenticated for network access. This safeguard prevents unauthorized access where a user might attach a hub and multiple devices to a switch port.

Location-based Tracking of Network Users with Kerberos Snooping

With NetLogin, ExtremeXOS interacts with the client and various authentication backend data stores to obtain the authentication status of a client. However, Enterprise IT organizations may not want to actively enforce network authentication, but have a need to know "who" or "what" is attached to their internal network. To meet this demand, ExtremeXOS has the ability to passively monitor Active Directory domain user authentication through Kerberos Snooping. Kerberos Snooping increases network-wide visibility of connected users without interrupting the user workflow.

Kerberos authentication is used by Microsoft Active Directory and by various Unix systems (including Linux and MAC OSX). ExtremeXOS has the ability to "snoop" the Kerberos communication between the user and the Active Directory Domain server. Once a valid user logs into the domain, ExtremeXOS binds the MAC and IP address to the user identity and ties all relevant network information to the user. Through this process, ExtremeXOS also has the ability to monitor

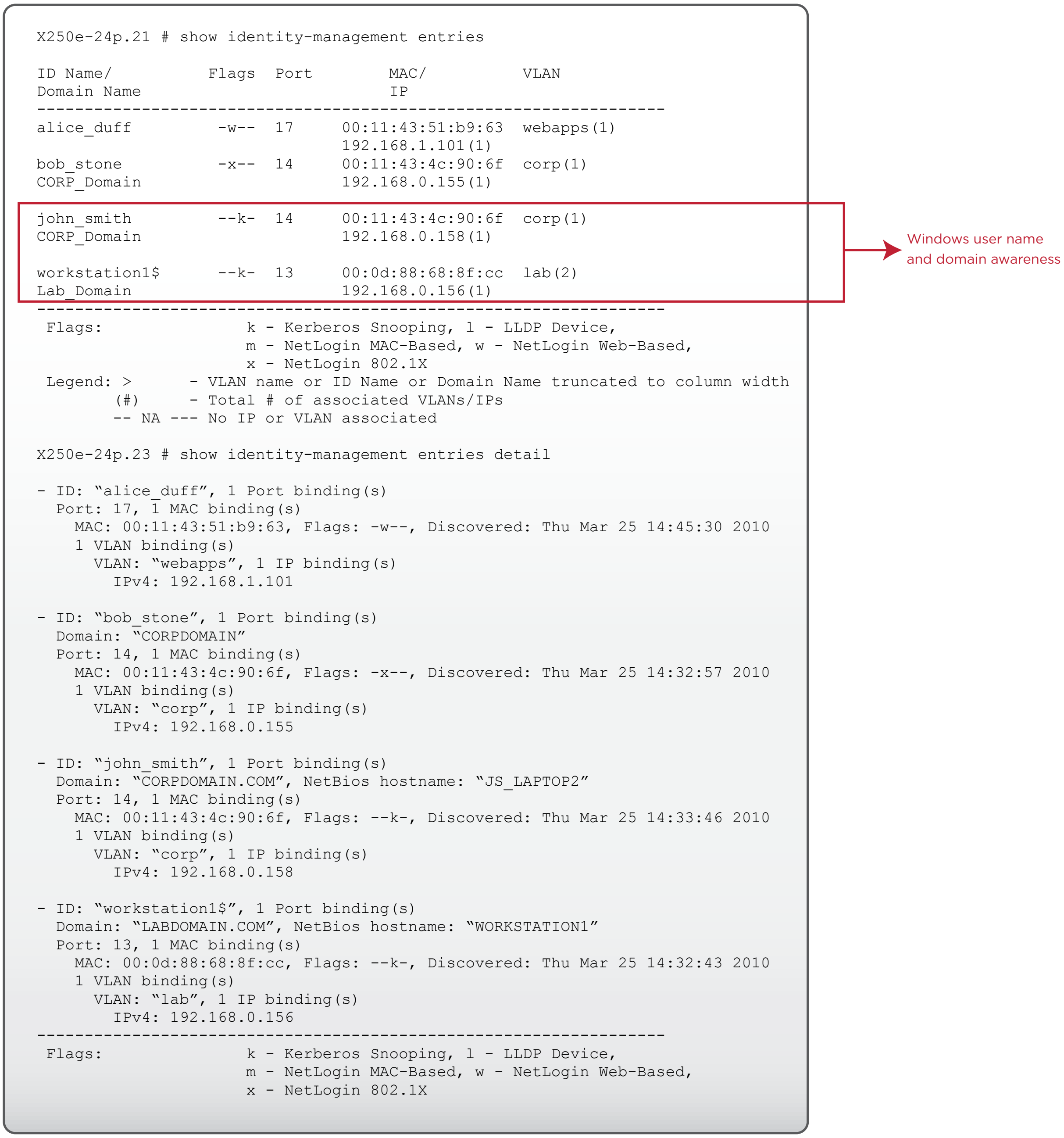
Kerberos Snooping extends the definition of connected endpoints to include:

- User name (account name)

- Host name (computer name)
- Realm/Domain Name
- Time at which the user or device was discovered
- Login time and logoff time (if applicable)
- Typical IP and MAC address information

With this information, detailed reports can aid in preparing information required for compliance and internal audits.

In the following example, ExtremeXOS displays a summarized view of the Kerberos users (as well as NetLogin users). Each ExtremeXOS-enabled switch has the ability to send the identity information to EPICenter® for centralized reporting and network-wide visibility.



```
X250e-24p.21 # show identity-management entries

ID Name/           Flags  Port        MAC/          VLAN
Domain Name                           IP
-------------------------------------------------------------------
alice_duff          -w--  17      00:11:43:51:b9:63  webapps(1)
                                  192.168.1.101(1)
bob_stone           -x--  14      00:11:43:4c:90:6f  corp(1)
CORP_Domain                       192.168.0.155(1)

john_smith          --k-  14      00:11:43:4c:90:6f  corp(1)
CORP_Domain                       192.168.0.158(1)

workstation1$       --k-  13      00:0d:88:68:8f:cc  lab(2)
Lab_Domain                        192.168.0.156(1)
-------------------------------------------------------------------
 Flags:               k - Kerberos Snooping, l - LLDP Device,
                      m - NetLogin MAC-Based, w - NetLogin Web-Based,
                      x - NetLogin 802.1X
 Legend: >       - VLAN name or ID Name or Domain Name truncated to column width
        (#)      - Total # of associated VLANs/IPs
        -- NA --- No IP or VLAN associated

X250e-24p.23 # show identity-management entries detail

- ID: “alice_duff”, 1 Port binding(s)
  Port: 17, 1 MAC binding(s)
    MAC: 00:11:43:51:b9:63, Flags: -w--, Discovered: Thu Mar 25 14:45:30 2010
    1 VLAN binding(s)
      VLAN: “webapps”, 1 IP binding(s)
        IPv4: 192.168.1.101

- ID: “bob_stone”, 1 Port binding(s)
  Domain: “CORPDOMAIN”
  Port: 14, 1 MAC binding(s)
    MAC: 00:11:43:4c:90:6f, Flags: -x--, Discovered: Thu Mar 25 14:32:57 2010
    1 VLAN binding(s)
      VLAN: “corp”, 1 IP binding(s)
        IPv4: 192.168.0.155

- ID: “john_smith”, 1 Port binding(s)
  Domain: “CORPDOMAIN.COM”, NetBios hostname: “JS_LAPTOP2”
  Port: 14, 1 MAC binding(s)
    MAC: 00:11:43:4c:90:6f, Flags: --k-, Discovered: Thu Mar 25 14:33:46 2010
    1 VLAN binding(s)
      VLAN: “corp”, 1 IP binding(s)
        IPv4: 192.168.0.158

- ID: “workstation1$”, 1 Port binding(s)
  Domain: “LABDOMAIN.COM”, NetBios hostname: “WORKSTATION1”
  Port: 13, 1 MAC binding(s)
    MAC: 00:0d:88:68:8f:cc, Flags: --k-, Discovered: Thu Mar 25 14:32:43 2010
    1 VLAN binding(s)
      VLAN: “lab”, 1 IP binding(s)
        IPv4: 192.168.0.156
-------------------------------------------------------------------
 Flags:               k - Kerberos Snooping, l - LLDP Device,
                      m - NetLogin MAC-Based, w - NetLogin Web-Based,
                      x - NetLogin 802.1X
```

Windows user name
and domain awareness

Example 9

Encrypted Communication

Once a trusted user is authenticated, management communication to an Extreme Networks device is encrypted to protect information from prying eyes.

Secure Shell (SSH2) allows the encryption of a Telnet session data between an SSH2 client and an Extreme Networks switch. ExtremeXOS also enables the switch to function as an SSH2 client, which allows it to send encrypted data to an SSH2 server on a remote system.

Transferring of software image files, configuration files, and/or ACL config files are done using the Secure Copy Protocol 2 (SCP2), which enables 168-bit (3DES) encryption. SCP2 is used to securely copy files over the network to avoid potential packet sniffers from extracting usable information in the data packets. The program uses the SSH2 protocol for data transfer, and provides the same security as SSH2. Network operators can be assured that inquisitive users on the network will not be able to compromise the switch configuration files.

ExtremeXOS also provides SNMPv3 communication to increase security and privacy of SNMP access to managed devices. The prior standard versions of SNMP (v1 and v2) provided no privacy and no security. SNMPv3 is designed to be secure against:

- Modification of information, where an in-transit message is altered.
- Masquerades, where an unauthorized entity assumes the identity of an authorized entity.
- Message stream modification, where packets are delayed and/or replayed.
- Disclosure, where packet exchanges are sniffed (examined) and information is learned about the contents.

ExtremeXOS, combined with the Extreme Networks EPICenter management platform, enables the network operator to utilize SNMPv3 and SSH2 for secure communication.

Integrity Checking

Encryption is vital to prevent the theft of information. However, the network operator must have assurance that the downloaded software is not tampered with during its journey (integrity). How does the network operator know that the ExtremeXOS application module (e.g. SSH2 or tcpdump) does not contain malicious code?

ExtremeXOS incorporates an advanced integrity protection scheme that prevents the network operator from dynamically loading a malicious application module by accident. In ExtremeXOS, the loadable ExtremeXOS application module must carry out a signature verification process. If the signature verification process is successful, only then will ExtremeXOS allow the dynamic loading of an ExtremeXOS application module.

A failed signature verification process would imply that the loadable ExtremeXOS application module was altered and it did not originate from Extreme Networks. In the following example, we created an invalid ExtremeXOS application module and tried to dynamically upload the module to the ExtremeXOS OS. As a result, ExtremeXOS rejected the non-conforming module with the error as shown in Example 10.

```
* BD-8810.30 # download image 10.255.97.34
bd8K-11.0.0.23-ssh.xmod Do you want to
install image after downloading?
(y - yes, n - no, <cr> - cancel) Yes
Downloading to MSM-A Error: Failed to
download image - Error: File is not an
upgrade package.
* BD-8810.31 #
```

Example 10

Since the altered module failed the signature verification process, ExtremeXOS rejected the application module without any impact to network operation. Network operators can be assured that the modular operation of ExtremeXOS is hardened against malicious alterations to the software code.

Protection against Denial of Service

Upon failed attempts to hack into a switch, a hacker's last effort is to launch a DoS attack at the switch. A DoS attack is an explicit attempt by an attacker to degrade or disable the switch system resources (e.g. CPU, bandwidth, memory). A successful attack will consume resources to cause the switch operation and/or performance to degrade or fail.

> Various well-known public attacks were tested against the ExtremeXOS operating system. The outcome had zero impact to the ExtremeXOS operating system along with the hardware resources on which it runs. A switch crash did not occur and the CPU resource remained unaffected.
>
> The following is a sample of some well-known public attacks that were directed at the BlackDiamond 10808 and the BlackDiamond 8800 series switch, both running ExtremeXOS:
>
> - ICMP Flood
> - ICMP Echo Sweep
> - UDP/TCP Flood
> - Teardrop
> - Land
> - Boink
> - Bonk
> - OpenTear
> - NewTear
> - Nestea
> - Ping of death
> - Syndrop
> - Misfrag

CPU DoS Protection

The detection module has the intelligence to classify packets that are directed to the switch CPU. The detection module can ignore packets from trusted sources or classify packets for counting. If packet counting occurs, the packets are classified into the following categories, with each having an individual threshold value:

1. IP Packets
2. ICMP Packets
3. UDP Packets
4. TCP Packets

Packet counts are compared against two thresholds: a lower threshold (threshold-1) that indicates an attack may be in progress, and an upper threshold (threshold-2) that triggers a response.

If a counter exceeds threshold-1 (lower threshold), an event is triggered. At this point, the packet header of each packet is examined by the DoS protect analysis module for predefined patterns. If the examined packet counter exceeds threshold-2 (upper threshold), a hardware ACL is automatically created to block further packets of the same type from disrupting the switch CPU or system resources. The automatic creation of ACLs will match the patterns detected. For example:

- Flood ACL
- Random source flood ACL
- Sweep ACL

The ACL is put in place for a configurable period of time to prevent further DoS attacks directed at the switch CPU.

After that time has elapsed, the ACL is removed. The events are then logged and sent to the network operator for further prevention.

Prevention is important when protecting the network. ExtremeXOS allows users to prevent many attacks by separating network traffic or isolating user communities with the use of virtual switch domains.

Feature: Flexible Policy Language

Benefit = Simplicity without compromise

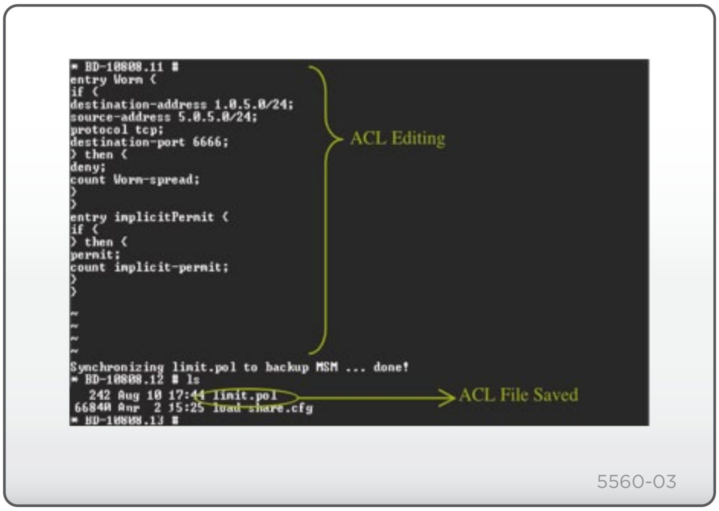
ExtremeXOS includes a file-based method for configuring ACLs. A file-based ACL is far less error-prone than the old command line way of entering ACLs and is simpler to manage.

Traditionally, the creation of ACLs (e.g. a 100 line access list) requires the network operator to type each entry through the CLI or through the inband management interface. The traditional workaround is to create a console script to automate the typing of each entry. However, once the entries are entered into the switch, the network operator cannot edit or reorder the entries within the policy. This monolithic approach is cumbersome and if a mistake is made or if the entries need reshuffling, the operator would have to delete the policy and restart the entry process—a time-consuming procedure.

ExtremeXOS allows the operator to create an ACL policy text file directly on the switch or as a file on a separate machine. The context of the ACL policy has a structured interface similar to the C programming language. A sample context of the ACL policy can be seen in Example 11. The policy text file is then saved or transferred to the switch file management system. The policy file is loaded into a policy database which can be applied to VLANs, ports on the switch, or to the user identity for role-based access policies.

```
entry TCP-SYN {
if {
destination-address 1.0.1.1/32;
source-address 5.0.1.1/32;
protocol tcp;
tcp-flags syn;
} then {
permit;
count tcp-syn;
}
}
entry Worm {
if {
destination-address 1.0.5.0/24;
source-address 5.0.5.0/24;
protocol tcp;
destination-port 6666;
} then {
deny;
count Worm-spread;
}
}
```

Example 11: Sample ACL Context



```
* BD-10808.11 #
entry Worm {
if {
destination-address 1.0.5.0/24;
source-address 5.0.5.0/24;
protocol tcp;
destination-port 6666;
} then {
deny;
count Worm-spread;
}
}
entry implicitPermit {
if {
} then {
permit;
count implicit-permit;
}
}

~
~
~
~
Synchronizing limit.pol to backup MSM ... done!
* BD-10808.12 # ls
  242 Aug 10 17:44 limit.pol
 66840 Apr  2 15:25 load-share.cfg
* BD-10808.13 #
```

Figure 6: File-based ACL Editing

The network operator can choose to create or edit the policy file directly on any Extreme Networks ExtremeXOS based switches using the ExtremeXOS text-based editor. In the example below, the network operator can create or edit a policy file named 'limit.pol' directly on the switch console by launching a text-based editor with the following command:

```
* edit policy limit.pol
```

Within the text editor, the operator can delete, reorder, or add a new ACL entry to better manage network traffic (see Figure 6).

A text-based editor can be invoked within ExtremeXOS to edit and rearrange ACL entries.

File-based ACL Editing

To further flex the benefits of file-based ACLs, the operator can dynamically load, unload, or refresh a policy file without impact to network performance and switch resources.

To prove this, we simulated the deployment and processing of ACL policies on a BlackDiamond 8800 running ExtremeXOS. We measured the impact of this configuration on VoIP application performance. An ACL policy of 30,000 rules was created with each rule containing several match criteria and unique counter/log actions.

When transmitting VoIP test traffic that simulated 3,000 VoIP sessions, we applied the ACL policy to simulate the deployment and live 'in-service' provisioning of large scale ACL policies. ExtremeXOS processed 30,000 ACLs without affecting our PSQM VoIP quality as shown in Figure 7.

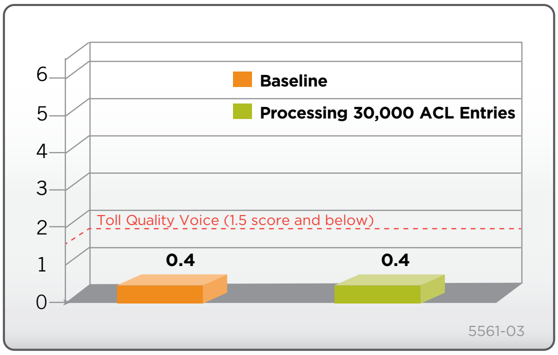


Figure 7: VoIP Quality (PSQM) When Processing 30,000 ACL entries (Lower is Better Quality)

When transmitting VoIP test traffic that simulated 3,000 VoIP sessions, we applied the ACL policy to simulate the deployment and live 'in-service' provisioning of large scale ACL policies. ExtremeXOS processed 30,000 ACLs without affecting our PSQM VoIP quality as shown in Figure 7.

A file-based ACL method also provides a precautionary measure for viewing the access list configuration. Traditionally, a user can view every access list entry (applications, ports, hosts, or subnets that are restricted and/or allowed on the network) in the global switch configuration when a 'show config' command is executed. In contrast to this monolithic approach, ExtremeXOS does not store nor display the access list entries in the global switch configuration. With the benefit of using a file-based ACL, the entire access list configuration is stored as a separate file under a user-defined name.

The ExtremeXOS file-based ACL simplifies the way an operator controls network devices without compromising network operation and performance.

Built to Extend

The ExtremeXOS native XML infrastructure enables extensibility and cooperation with best-of-breed solutions and/or customer-defined applications. The Tcl infrastructure enables user-customizable look and feel of the CLI to drive management simplicity.

ExtremeXOS XML API will provide a means for external programs and scripts to access the configuration and operational data states of Extreme Networks switches and rearrange ACL entries.

XML API

Feature: Extensible Markup Language (XML)

Benefit: Best-of-Breed Ecosystem

In recent years, Extensible Markup Language (XML) has emerged as the preferred technology for data representation for both Web-based and traditional software applications. This development has accelerated the standardization of XML and the widespread support of XML libraries, utilities, and applications. XML is increasingly the preferred foundation for integrating various enterprise applications with one another. XML extends Hypertext Markup Language (HTML) by providing a new language toolkit. The new toolkit allows programmers to develop their own markup languages, while automatically providing the benefit of being compatible with existing deployed XML code.

ExtremeXOS has been designed from the ground up with underlying XML technology. XML is used by all management communication schemes in ExtremeXOS, which includes CLI, SNMP, and Web-based services.

Once unlocked, ExtremeXOS XML API provides a consistent infrastructure for configuration and operational application both internally within the switch for the ExtremeXOS application and externally outside the switch for a management or security applications (see Figure 8)

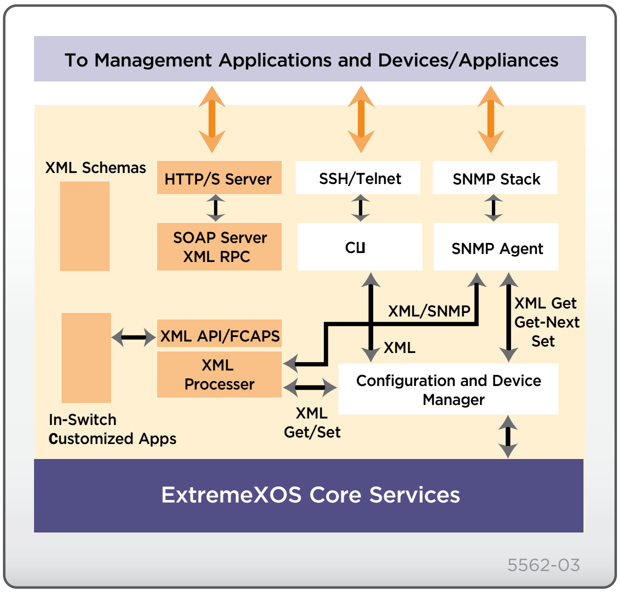


Figure 8: ExtremeXOS Core Services

Traditional management of Ethernet switches is done via two main mechanisms for configuring and managing a network switch, the Command Line Interface (CLI) and the Simple Network Management Protocol (SNMP).

CLI is a good interface for switch configuration. However, CLI is not an ideal interface for management software to exchange configuration with the embedded device. CLI usually represents a simple one-line ASCII command. Output, however, is often screen-oriented with enhancements being merged into screens. Hence, parsing output of the CLI can be complex and cumbersome and typically is not compatible across major releases.

On the other hand, SNMP provides a better interface for management software to perform device configuration. It requires a well-defined set of MIB tables to exchange configuration settings. However, SNMP lacks the flexibility that CLI offers. SNMP uses an unreliable UDP transport and requires the user to walk through tables in order to access a particular element within the MIB structure. It also follows a master and agent approach, with limited communication capabilities originated in the agent. Many MIB implementations still only provide read access, due to the historic lack of authentication and privacy of SNMPv1/v2.

A mechanism is needed between these two methods; one that is more usable for a machine interface than CLI, but more complete, powerful and flexible than SNMP. An on-switch XML based programming interface meets this demand.

ExtremeXOS XML API reduces the cost and complexity of creating and maintaining the management infrastructure within ExtremeXOS. It supports more expressive and complex, yet simple, management information. ExtremeXOS XML API allows Extreme Networks switches to be integrated with newly deployed Web-based services standardized by the World Wide Web consortium. These services use Web services description language to describe their capabilities and simple object access protocol or remote procedure call to transport the XML-based messages between the various Web services components and the ExtremeXOS switches.

ExtremeXOS XML API also provides a means for external programs and scripts to access (read/write) the configuration and operational data and states of Extreme Networks switches, such as forwarding databases, VLAN, and Access Control List configuration. Instead of MIB queries, external applications will take advantage of direct access to individual ExtremeXOS APIs.

Example scenarios that show various capabilities of the ExtremeXOS XML element are:

- XML-based configuration displayed in an Internet Explorer browser
- Managing an Extreme Networks infrastructure with Microsoft C# programming environment
- Native XML interfaces exposed over SOAP and RPC transport protocols
- Personalization of the CLI (see Tool Command Line paragraph)
- Extensibility of the CLI
- Installation of a POSIX application

Support for Tcl infrastructure will allow ExtremeXOS to customize the CLI.

Tcl Scripting

Feature: Tool Command Language (Tcl)

Benefit = Simple to Manage

ExtremeXOS Tcl infrastructure allows the operator to change the look and feel of the CLI management.

With a native Tcl infrastructure, ExtremeXOS allows the personalization of CLI commands. For example, the output of a single CLI command can be altered to include outputs of multiple CLI commands. Instead of having to input five, ten, or twenty CLI commands on a daily basis, the Tcl infrastructure allows for combining these commonly used syntaxes into a single CLI command. ExtremeXOS CLI personalization will simplify the operator's daily command tasks.

As a security measure, ExtremeXOS CLI processor interfaces to the authentication, authorization, and accounting engine for CLI command authorization and validation. The authorization is performed before the front-end Tcl code is executed. In addition, per-user and per-session command statistics are recorded and made available as a security measure.

Feature: Universal Port

Benefit = Operational Efficiency through Automation

In the past, when users, devices, or applications were added, moved, or changed, IT personnel had to be available to physically and manually configure the network. Universal Port provides the ability to automatically configure the network interface ports on Extreme Networks switches and automatically provide configuration information to the attached device. This is done through event triggering of ExtremeXOS scripts and dynamic runtime variables.

The ExtremeXOS Universal Port is a flexible framework that allows the switch to take actions based on events. Leveraging the ExtremeXOS CLI scripting capability, Universal Port activates dynamic or static profiles that are created and managed via the ExtremeXOS CLI or through EPICenter Universal Port Manager. These profiles provide configuration attributes against switches for items such as security policy, QoS settings and VLAN assignment at the port level. A profile is a variable command set that can take action based on different types of events. For example, a profile can automatically provision an IP phone and the attached switch port with appropriate power and QoS settings.

Universal Port supports the following trigger events:

- Device discovery
- User or device authentication
- Time of day

To learn how Universal Port can help decrease administration time and cost, while increasing operational efficiency and supporting green initiatives, please visit Extreme Networks EtherNation portal at: http://www.ethernation.net/Default.aspx?tabid=87.

Feature: XNV (ExtremeXOS Network Virtualization)

XNV is a set of licensable software modules for both the ExtremeXOS® based switching product portfolio, as well as for Extreme Networks EPICenter®, a network provisioning and management application. XNV brings insight, control and automation for highly virtualized data centers to the network.

XNV enables the following capabilities:

- XNV provides centralized network-based virtual machine (VM) inventory, VM location history and VM provisioning. XNV achieves this through EPICenter, which interfaces through standard application programming interfaces (APIs) to virtual machine management platforms such as VMware vCenter, Citrix and others.
- XNV allows centralized network-based configuration and distributed network-based enforcement of network-level capabilities down to the individual virtual machine level. XNV does this through a virtual port profile (VPP) which can be associated with individual virtual machines in a centralized manner through EPICenter. VPPs allow configuration of access control lists (ACLs), Quality of Service (QoS), rate limiting, and other capabilities to individual virtual machines. VPPs are enforced through the ExtremeXOS enabled network switches running XNV.
- XNV provides automated VM lifecycle tracking of virtual machines in the network as VMs migrate from server to server, as well as the ability to automatically move the VM's VPP to the appropriate network switch and enforce the VPP-based parameters and policies in real time.

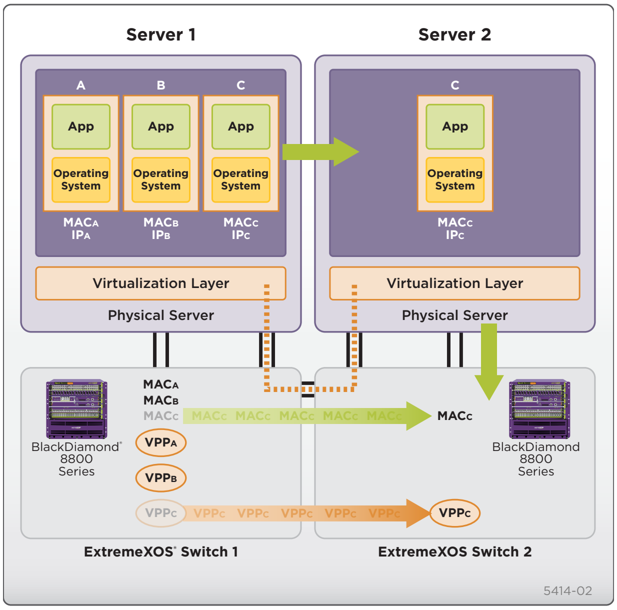


Figure 9

XNV does not require any changes to the server operating environment and works across multiple hypervisor technologies.

For a demonstration of XNV capabilities, visit http://info.extremenetworks.com/XNVPart1-VirtualizationManagementVideo.html and http://info.extremenetworks.com/XNVPart2-VirtualizationManagementVideo.html

Feature: ID Manager

Identity Manager allows network managers to track users who access their network. User identity is captured based on NetLogin authentication, LLDP discovery and Kerberos snooping. ExtremeXOS uses the information to then report on the MAC, VLAN, computer hostname, and port location of the user. Further, Identity Manager can create both roles and policies, and then bind them together to create role-based profiles based on organizational structure or other logical groupings, and apply them across multiple users to allow appropriate access to network resources.

In addition, support for Wide Key ACLs further improves security by going beyond source/destination and MAC address as identification criteria to examine the IP address and VLAN of the user as well.

Feature: M-LAG

Multi-Switch Link Aggregation Groups (M-LAG) can address bandwidth limitations and improve network resiliency, in part by routing network traffic around bottlenecks, reducing the risks of a single point of failure and allowing load balancing across multiple switches.

Conclusion

ExtremeXOS brings the latest operating system technology to networking and enables several cutting edge features such as modular architecture; open-standards POSIX compliance; secure operation with authentication, encryption; integrity, and protection; flexible policy language with file-based ACLs; along with XML and Tcl languages.

Working in tandem, these features allow network managers to build an infrastructure for converged applications that enable plug-and-play connectivity, resilient connectivity, and secured connectivity.

ExtremeXOS is the next generation operating system.

Make Your Network Mobile

www.extremenetworks.com

Corporate
and North America
Extreme Networks, Inc.
3585 Monroe Street
Santa Clara, CA 95051 USA
Phone +1 408 579 2800

Europe, Middle East, Africa
and South America
Phone +31 30 800 5100

Asia Pacific
Phone +65 6836 5437

Japan
Phone +81 3 5842 4011