Alcatel-Lucent Security Management Server (SMS)

Release 9.4

Technical Overview

260-100-022R9.4
Issue 1
June 2009

Contents

About this information product

Purpose

This document is a technical product description and overview of the Alcatel-Lucent VPN Firewall system. It contains descriptions of all system components and features up through and including Release 9.4.

Reason for reissue

Reissued for Release 9.4

VPN firewall solution components

The Alcatel-Lucent VPN Firewall system consists of these components:

- Alcatel-Lucent *VPN Firewall Brick*™ Security Appliance
- Alcatel-Lucent Security Management Server (SMS)
- Alcatel-Lucent IPSec Client

The Brick device is a hardware appliance-based product. The SMS is installed on a general-purpose host. The Alcatel-Lucent IPSec Client is a software component installed on *Windows*® hosts only. The chapters that follow provide a great deal of detail regarding each component, and associated features.

How to comment

To comment on this information product, go to the Online Comment Form (http://www.lucent-info.com/comments/enus/) or e-mail your comments to the Comments Hotline (comments@alcatel-lucent.com).

1 Alcatel-Lucent *VPN Firewall Brick*™ Security Appliance

Basic Physical and Logical Architecture

Overview

The Brick device is a high-speed packet-processing appliance, primarily oriented towards providing security functions. The Brick is offered in several models, providing different physical interface combinations as well as different capacity and throughput ratings. The Brick device product line provides LAN-level Ethernet interfaces, in both 10/100 copper as well as Gigabit fiber and/or copper ports.

Internally, the device has only a solid-state NVRAM disk to store local policy and configuration. The cooling fan in the larger Brick devices is the only continuously moving part. This allows the Brick device to have an extremely long hardware mean time between failures (MTBF) — see data sheets for specific Brick models.

The Brick device does not run as an application on top of a commercial operating system ; rather, it runs as the kernel of a small, highly application-specific operating system developed by Alcatel-Lucent. The Brick device operating system is an evolution of Alcatel-Lucent's Inferno operating system , designed for small embedded security applications. The Brick operating system has no user logins or file system permissions to be overridden, as well as no insecure communication processes (such as Telnet or HTTP) to be broken via a stack smash or buffer overflow.

Brick devices are available in a variety of hardware models; the models differ solely in throughput, capacity, and physical interface types. Currently, 10/100 copper Ethernet and Gigabit fiber-optic (multimode) interfaces, Gigabit copper interfaces, and Gigabit Only interfaces, which are configurable for copper/optical connections, via Small-form Factor Pluggable (SFP) modules (available for Model 1200 and Model 700 Bricks).

The same software image runs on all Brick devices, so all features discussed in the following sections are available on all Brick platforms.

Unlike many router-based firewalls, the Brick device is designed to be a learning bridge, in much the same way Layer-2 Ethernet switches operate. The Brick device can provide Layer-3 forwarding (routing) where desired as well.

□

Packet Forwarding - Bridging and Routing

Overview

Internally, the Brick device is set up much like a classic Layer-2 Ethernet switch. Each packet inbound to a physical port is assigned to a VLAN, and that packet can bridge to any physical port with membership in that VLAN (or VLAN bridge group). Physical ports are associated with a single default VLAN, which is used to associate inbound untagged Ethernet frames, and a list of VLAN memberships.

The Brick device contains a list of static routes used when Layer-2 forwarding is unable to forward a packet. This occurs in one of the following cases:

- A Brick device address has been used as a next-hop gateway by a router or a host
- A packet has come out of a VPN tunnel
- A packet header has been changed via address translation
- A packet is initiated by the Brick device (such as Brick device management traffic)
- A routing entry has become invalid due to, for example, a link failure

In any of these cases, the Brick device has no Layer-2 (MAC) information to use for packet forwarding, so it must resort to making a Layer-3 (IP) decision, via a static route lookup.

As a Layer-2 bridge, the Brick device maintains a cache of connected MAC addresses and the physical ports with which they are associated. The Brick can optionally be configured to require that MAC addresses be bound to the physical port where they were first learned, requiring a manual reset to unlatch and rebind. In addition, the Brick actively verifies the existence of IP/MAC bindings before timing them out of the cache, to discover and proactively respond to changes in L2 architecture. The Brick device also supports the ability to administratively fix IP/MAC/VLAN/interface bindings in highly sensitive environments.

The Brick device can support Jumbo Frames, to achieve higher speed throughput on high-demand networks.

The Brick device will also properly support Broadcast and Multicast packets (although multicast is not supported through IPSec tunnels, since the IPSec standard does not allow it currently). The Brick will also support Microsoft Cluster servers (although this may sometimes require special configuration).

The Brick device can be provisioned to bridge—but not firewall—non-IP Ethernet frames by configuring a list of Ethertypes or DSAP IDs to allow.

□

IEEE 802.1q VLAN Tag Support

Overview

The Brick device supports the use of IEEE 802.1q VLAN tagged Ethernet Frames. Each physical port can be configured to send and/or receive tagged frames, untagged frames, or a combination of both. A fairly unique feature is the ability to preserve tags, if any are present. Since each port can be configured to disallow inbound tagged frames, the Brick device is immune to VLAN tag attacks that have plagued switch vendors in the past.

The Brick device also has the ability to support VLAN Domains, to support the case where VLAN IDs received inbound on one trunk are not logically identical to, but possibly conflicting with, those received inbound from another trunk. VLAN Domains are useful when connecting to multiple VLAN trunks that may use conflicting VLAN IDs.

Note that the Brick device can simultaneously support up to 4094 VLANs on each connected VLAN trunk.

VLAN bridge groups increase this functionality to allow the Brick to bridge among a set of configured VLANs. All VLANs in the bridge group can be accessed by Layer-2 forwarding, eliminating the need to use the Brick as a gateway for packets that simply transition from one VLAN to another. Each VLAN can be then associated with a security level, and packets can transition from one trust level to another by passing through a firewall policy, then being switched to a different VLAN.

□

Brick Devices

Overview

A Brick device can be partitioned into true virtual firewalls. Each virtual firewall has its own routing information, its own set of IP addresses, and its own policy rules, which specify the types of traffic allowed and how that traffic is processed. Policy rules are also referred as Brick zone rulesets or simply zones. Each zone is applied to one or more physical ports of a Brick, qualified by a set of IP addresses, as well as a set of VLAN tags. A given Brick zone ruleset will only apply to traffic to or from those IP addresses on those VLANs. Wildcards may be used in such assignment. Additionally, Brick zone rulesets may be applied to multiple Brick devices.

The use of Brick zone rulesets is not additionally licensed and are only limited by the physical resources of the Brick device to which they are applied.

While Brick devices may be assigned to VLANs, the two features are in no ways interdependent. VLAN tagging may be used with or without virtual firewalls, and Brick devices may be used with or without VLAN tagging.

Each Brick zone can be assigned a single Virtual Brick Address (VBA). A Brick device can be assigned multiple VBAs. A VBA may be used for multiple purposes, including Network Address Translation (NAT) as well as acting as a Tunnel End Point for VPN.

Brick devices can also be used to represent different customers in a multi-tenant environment. Sessions are unique within each given Brick device, and, when used in conjunction with Brick partitions, can be used to ensure session independence in a shared environment.

□

Brick Policy Rules and Zone Rulesets

Brick rules: a primary firewall security mechanism

One of the fundamental security elements of a firewall is a rule, which allows or blocks traffic, based on some set of criteria.

Brick devices use a 6-Tuple Matching system as the first set of rule criteria. The 6-Tuple includes any combination of the following information from the IP header of the packet as it traverses the Brick:

1. IP Source address
2. IP destination address
3. IP protocol
4. TCP or UDP source port (or ICMP/IGMP type)
5. TCP or UDP destination port (or ICMP/IGMP code)
6. VLAN ID

Brick zone ruleset

A set of these rules comprise what is referred to as a *policy* or *Brick zone ruleset*, which is assigned to one or more ports on one or more Brick devices. When you assign a Brick zone ruleset to a port, you specify the IP addresses of the hosts connected to that port and protected by the set of rules in the zone ruleset that has been assigned to the port.

A set of pre-configured Brick zone rulesets are installed with the Alcatel-Lucent VPN Firewall solution. These pre-configured zone rulesets consist primarily of pre-defined system rules that allow management and configuration traffic to pass between the SMS and Bricks. These pre-defined zone rulesets can then be modified with user-defined rules which address the specific data security constraints and activities of the “zone” or portion of the customer’s operational network connected to that Brick port, or copied to a new zone ruleset, modified as needed, and assigned to the port.

Traffic can be further segmented by creating Brick partitions and assigning Virtual LANs (VLANs) to these partitions, for separating traffic from different customers that share physical address points. Each Brick partition can be set up as a distinct virtual firewall, with its own set of host IP addresses, rulesets, and session cache entries.

Rule packet filtering

The Brick, as a bridging device, “listens” to all data traffic that crosses its ports. When a data packet crosses a Brick port, the Brick examines the packet header information and compares it with the rules contained in the zone ruleset that has been assigned to the port. If the Brick finds a rule that matches the information and traffic type contained in the packet header, it takes the action dictated by the rule for that type of

data packet: it passes, drops, tunnels (that is, encrypts/decrypts the packet) or tunnel proxies the packet, and establishes a session cache for every packet session that was passed, dropped, or proxied. If the Brick decides to pass the first packet of a session, it places an entry in its session cache. For each subsequent packet, the Brick checks its session cache first to see if there has already been a packet which matched and passed by its rules. If so, it passes all subsequent packets of the same type. For some protocols, the Brick can inspect the contents of the packet further by applying an application layer filter before it decides to pass or drop the packet.

If the Brick cannot find a rule in the zone ruleset that matches the packet header, the packet is dropped by a "drop all packets" rule, which is automatically inserted into each ruleset. If the Brick cannot find a zone ruleset that matches the source/destination or traffic protocol defined in the packet header, it drops the packet.

Advanced ruleset features

Besides the filtering criteria applied by rules, the Brick supports a full range of features that can be set for any rule in a ruleset. These include the capability to:

- Activate/deactivate individual rules, and set days of the week and start/end times for when a rule will be dynamically activated/deactivated
- Set up Network Address Translation (NAT) or Port Address Translation (PAT) for packets passing through a rule, and specify whether source or destination NAT/PAT is applied, and the mode of address assignment: Direct, Pooled, Local, or Dynamic
- Set minimum and maximum bandwidth for packets passing through the rule, and whether to apply the limits and guarantees per session or for the entire rule.
- Set TOS bits and DiffServ Code points, and whether to apply separate throughput and delay limits for packets that meet or do not meet bandwidth requirements, as well as set bandwidth alarms to monitor quality of service guarantees and Service Level Agreements (SLAs)

Rules-based routing

A rule can be configured for any type of protocol traffic to route all packets that match the rule to a proxy server, router, or other device, utilizing third party software, to perform content filtering functions such as command blocking, URL filtering, and virus scanning. The exception is for dynamic pinholes that are opened for RTSP, H.323 and SIP protocol traffic; this traffic will not be routed by the rules-based routing feature. The rule can be set up to route all packets of a certain traffic type that are coming into the zone, going out of the zone, or both.

Rules-based routing can be set up to load balance routing of packets between multiple servers that are defined in a host group to perform content filtering. Options can be set to selectively skip hosts that are pinged and unresponsive by a set interval or do not respond with a SYN-ACK message.

The Rules-based routing feature expands the Brick's flexibility to monitor and filter incoming and outgoing traffic with the following capabilities:

- Route based on any UDP or TCP port to any IP address
- Return route. Send to scan then return to Brick for additional routing and scanning
- Interface with any third party equipment
- Load balance over several proxies or WAN links
- Bypass scanning equipment for data that does not require scanning, thereby reducing traffic bottlenecks
- Scan data to be passed onto the network or redirected back to the Brick for further filtering and routing (data scanning and routing is transparent to users)
- Load balance forwarding and content filtering of traffic between multiple content filters/hosts
- Preserve rules-based routes in the event of Brick hardware failure/failover

The Rules-based routing feature will work in tandem with any third party vendor, and allows you to retain your existing proxy services for URL filtering and antivirus applications, and load balance content filtering and URL checking processes across multiple servers.

Brick-specific rules

In some cases, the security constraints for various types of traffic, even within the same zone, may be different for each Brick and the related portion of the network being protected. To allow for this contingency, a zone ruleset can be customized by adding specific rules within a ruleset for a specific Brick or by modifying the parameters of an existing rule in a zone for a specific Brick. In this way, a Brick zone ruleset can be defined with generic rules that apply to all Bricks in a zone and Brick-specific rules that only apply to a particular Brick in a zone.

Rule and ruleset maintenance

A full range of editing and maintenance functions is provided by the Alcatel-Lucent Security Management Server (SMS) application to update rules and rulesets to keep pace with ongoing changes in security requirements and traffic routing. These editing and maintenance functions include the capability to:

- Modify rule parameters
- Activate/deactivate rules on demand or by day of the week/time of day
- Duplicate a rule within a ruleset and modify parameters in the duplicate rule
- Copy a rule or range of rules, and paste them in the same zone ruleset or another ruleset, and modify the copied rules as needed, essentially creating a new rule or set of rules

- Renumber a rule by moving it up or down within a ruleset to change the order/priority in which a rule is applied to an incoming/outgoing packet
- Delete a user-defined rule
- Copy and rename a Brick zone ruleset
- Move a zone ruleset to a different zone ruleset folder in a different group
- Delete a zone ruleset

Traffic matcher tool

The SMS allows you to test a Brick zone ruleset by entering a specification for simulated traffic, and performing a search for which rules in that ruleset would be triggered by the traffic pattern entered. The rules are matched and displayed, based on the general traffic characteristics entered, not on the specific action that would be taken on the traffic or traffic originating from a certain NAT address. The tool allows you to display all rules that match the "virtual traffic" details entered, or all rules in the ruleset with those that match the "virtual traffic" details highlighted in the display.

Rule statistics report

A Rule Statistics report is available which provides hit counts by rule for a Brick zone ruleset, or hit counts by zone ruleset for a selected group or one or more Bricks, during a specified time period.

Related information

For complete details about creation of security rules and Brick zone rulesets, refer to the *SMS Policy Guide*.

□

Stateful Packet Filtering

Overview

Every packet processed by the Brick device is considered part of a ″session″, regardless of IP type or higher-layer protocol. A session is simply a stateful entity tracked in memory on the Brick - a record of a conversation between two or more parties. The conversation may be unidirectional, and it may be between multiple parties, as in the multicast case. Regardless, the concept of a ″session″ still applies.

Each packet is not processed individually, as in non-stateful devices, such as routers. Rather, the first packet in a session is subject to ordered processing by the Brick zone rules, and an entry is made in that Brick device RAM cache. The following packets in that session are processed using a mathematical transform that allows the RAM cache entry to be directly accessed, supplying the associated disposition of that packet (pass, drop, address-translate, and so forth) Of course, this explanation is vastly simplified; many criteria are used to evaluate packets as they flow through the Brick, depending on the type of packet and Brick device configuration.

Each Brick zone ruleset consists primarily of an ordered list of rules. These rules consist of ″matching criteria″, used to evaluate the packet headers to determine if a given rule should be applied to a particular packet, as well as a set of action criteria, used to determine what should be done with the packet.

Examples of matching criteria are as follows:

- Source IP Address
- Destination IP Address
- IP Protocol (ICMP, TCP, UDP, etc.)
- Layer-4 Source Port
- Layer-4 Destination Port
- Source and/or Destination User Group
- VLAN
- ToS/DiffServ tag
- Time of Day (local to the Brick, or global time)
- Dependency Mask (used to establish an ″if-then″ pair of rules)
- Application Protocol attributes (application filters)

Examples of Action Criteria are as follows:

- Pass/Drop/Proxy
- VPN Internal/External/Both
- TCP Validation / Strengthening Parameters
- SYN Flood Protection

- Rule Alarms
- Source Address Translate
- Destination Address Translate
- Destination Port Address Translate
- Quality of Service parameters
- Quality of Service alarms
- ToS/DiffServ tag marking

It is worthwhile to note that the Brick device processes all packets in a stateful manner. Those protocols that do not have explicit connection establishment protocols (as does TCP), are processed using idle timeouts. That is, a session is created upon seeing a new such packet, and torn down when no more packets are seen within a configurable period of time.

All Brick devices in the system will synchronize their local time with the time set on its SMS server, plus or minus an administrator specified offset.

The Brick device has been certified by the ICSA for firewall and IPSec functionality.

□

Application Filters

Overview

The Brick device has the ability to perform inspection at the application layer of packet-based traffic passing through it using its unique Application Filter architecture. This inspection is performed for different purposes, depending on the application protocol, including to secure the protocol commands themselves, to open dynamic TCP or UDP ports as required by the semantics of the protocol, or to filter specific contents.

Application Filter protocols [and their associated functions] currently supported by the Brick device are as follows:

Internet

- HTTP (HyperText Transfer Protocol) [URL logging, URI pattern match blocking, root directory traversal blocking, HTTP request protocol anomaly detection]
- RTSP (Real Time Streaming Protocol) [command checking for filename-based attacks, protects against buffer overflow attacks]
- SMTP [Protocol Anomalies Checks, Commands filtering, Hide banner information, Block Spoofed Outgoing Mails, filter MIME types/attachments]
- FTP [Protocol anomaly check, Commands Filtering, prevent connection stealing, restrict Dynamic Ports, restrict users, prevent dictionary attacks, and so forth]
- DNS (Domain Name Service) [protocol anomaly detection and protocol specific field blocking, dynamic channel opening]

VoIP

- SIP (Session Initiation Protocol) [protocol anomaly detection and protocol specific field blocking, dynamic channel opening]
- NOE [inspects and validates voice and data traffic between IP touch phones and other network elements in a VoIP network]
- H.323 [full v2 support, dynamic channel opening, address translation, FastStart, H.245 tunneling]
- H.323 RAS [address translation]

Mobility

- GTP (GPRS Tunneling Protocol)[Stateful, protocol anomaly detection and protocol specific field blocking, dynamic channel opening

Other

- DHCP Relay (allows DHCP messages to be translated and sent to pre-configured known DHCP server(s), on an arbitrary IP network)
- TFTP (Trivial File Transfer Protocol) [dynamic channel opening, address translation]
- Oracle SQL*Net [dynamic channel opening]

- Microsoft NetBIOS [address translation]
- RPC (Remote Procedure Call) [logging, filter by procedure and program, dynamic channel opening]

These Application Filters are customizable for the particular environment in which they are being used. Different versions of the same Application Filters can be applied simultaneously to different traffic as it flows through the Brick. Application Filters are fully integrated into the Brick Virtual Firewall model, as well as the SMS Object Model.

Some Application Filters, such as the H.323 App Filter, also support user-configurable limits on the numbers and timeouts for dynamic channel support.

Important! *H.323 VOICE-OVER-IP APPLICATION FILTER NOTES*

The H.323 Voice-over-IP Application Filter, along with the H.323 RAS application filter are of particular note. These features are designed to fully support the H.323 version 2 (2/98) standard, including complex features such as Fast-Start and H.245 tunneling.

These Application Filters include a full ASN.1 PER decoder to fully and transparently inspect the H.323 and related set of data streams. When dynamic channels are required, by either endpoint, the Brick device automatically opens a self-sealing ″pinhole″ specific to that call. Only a single session will be allowed through that pinhole; once that session is begun, the pinhole vanishes. If the pinhole is never used, it will timeout.

Additionally, if Network Address Translation is desired, the Brick device performs NAT on all addresses, whether in the IP header or H.323 payload. This includes H.323/Q.931 messages, H.245 messages, and RAS messages.

□

Virtual Private Networking (VPN)

Overview

VPN is a core security component offered by the Brick device . While firewall rules can prevent obviously invalid or malicious traffic from entering a protected perimeter, a VPN can prevent all unauthenticated traffic from entering it. This feature can provide state-of-the-art cryptographic protection against attacks by requiring strong end user authentication in conjunction with confidentiality and integrity verification of messages.

The Brick device offers both LAN-LAN VPN as well as Client-to-LAN VPN, using the IPSec protocol. Cryptographic parameters supported are as follows:

For Session Establishment

The following encryption methods are available:

- Diffie-Hellman Group 1
- Diffie-Hellman Group 2
- Diffie-Hellman Group 14
- Diffie-Hellman Group 15
- Diffie-Hellman Group 16
- Diffie-Hellman Group 5

For Key Negotiation

The following key negotiation methods are supported:

- Internet Key Exchange Version 1 (IKEv1)
- Internet Key Exchange Version 2 (IKEv2)

For Confidentiality

The following encryption methods are available:

- DES
- 3DES
- AES (CBC-128, CBC-192, CBC-256)

For Integrity

The following encryption methods are available:

- SHA-1
- MD5
- AES-XCBC-MAC (for client tunnel endpoints only)

For Strong User Authentication

The following authentication methods are available:

- X.509 compatible certificates
- PKCS #12 Certificate Import
- SecurID (Client-to-LAN only)
- RADIUS (Client-to-LAN only)
- Locally managed shared secrets
- Pre-shared keys

There is an integrated utility in SMS to convert Entrust certificates to standard X.509 format.

Both AH and ESP types of IPSec are supported. Both Tunnel Mode and Transport Mode are supported. VPN is supported on all models of the Alcatel-Lucent Brick device. However, certain models support an optional hardware VPN encryption acceleration card (EAC).

The Alcatel-Lucent Brick device has been certified by the ICSA for VPN and is a member of the ICSA 1.0B VPN reference platform set.

Advanced features include the ability to tunnel IPSec traffic (IP type 50 or 51) inside UDP (IP type 17). With the implementation of Internet Key Exchange Version 2 (IKEv2), standard encryption algorithms from many different vendors are supported.

The Brick device can be configured to ″splice″ tunnels, that is, forward packets between two different tunnels terminating on the same Brick to provide a dynamic end-to-end secure connection.

Supported IPSec Clients

The Alcatel-Lucent Brick device interoperates with IPSec Clients from multiple vendors, including Alcatel-Lucent, Safenet, and Certicom (Movian).

□

Network Address Translation (NAT)

Overview

As with many other Brick device features, Network Address Translation and Port Address Translation are performed on the Policy Rule level, within a given Virtual Firewall. Every policy rule may have an Address Translation entry. Each Address Translation entry consists of any of the following three types of translation:

- Source Address Translation
- Destination Address Translation
- Destination Port Translation

Source Address Translation

Source Address Translation will translate the source IP address (and possibly layer-4 source port) of all forward packets within the session, and retranslate the destination IP address and possibly ports on all reverse packets within the session. Source address translation is available in four modes: Direct, Pool, Dynamic, and Local. Direct source address translation, provides a one-to-one map between an inbound set of address and a translated set of addresses. Pool source address translation allows a large number of inbound addresses to be multiplexed to a smaller number of translated addresses, using other protocol fields (such as source TCP or UDP port) to establish a unique socket. This capability is also commonly called Port Address Translation (PAT) or Network Port Address Translation (NPAT). Dynamic source address translation is a variation of NAT in which the original (usually private) IP address of a client that is connecting to a service provider network is dynamically mapped to another (usually public) IP address by its supporting Brick from a pool of IP addresses, usually on a per-zone basis. Local source address translation is only used in conjunction with Client VPN and is used to give an inbound client VPN connection a ″local″ address on the protected network.

Destination Address Translation

Destination Address Translation will translate the destination IP address (and possibly layer-4 destination port) of all forward packets within the session and retranslate the source IP addresses and possibly ports on all reverse packets within the session. Destination address translation is available in four modes: Direct, Pool, Dynamic, and Local. Direct destination address translation, provides a one-to-one map between an inbound set of address and a translated set of addresses, usually to provide public images for servers with private addresses. Pool destination address translation is used to provide session-based server-load-balancing. Dynamic destination address translation is a variation of NAT in which the public IP address of an inbound packet is dynamically mapped to a private client IP address by its supporting Brick from a pool

of IP addresses, usually on a per-zone basis. Local destination address translation is only used in conjunction with Client VPN and is used to give an inbound client VPN connection a ″local″ address on the protected network, for reverse-initiated connections (such as an X-Windows Server).

Destination Port Translation

Destination Port Translation is used to change the destination TCP or UDP port of an inbound session. This feature is usually used for special purposes, like mapping all inbound connections to a fixed port, regardless of the actual port requested.

NAT may be performed even if the Brick is bridging (Layer-2).

□

User Authentication

Overview

Strong user authentication is often a critical component of a security architecture. If a resource must be accessed, perhaps it is reasonable to maintain an audit trail of who accessed it and when, so any malfeasance may be traced back to an individual.

Users are collected into objects called User Groups. As discussed in the Stateful Packet Filtering section above, User Groups may be used as matching criteria in a firewall rule. This allows the administrator to configure sets of rules that apply only to users in a given User Group, once the users have authenticated.

Every user in a User Group may have their own individual mechanism for authentication. Default authentication mechanisms are also provided for those who do not wish to recopy user lists stored in external databases; the authentication requests are simply passed through to the default server, if so configured.

The Brick device supports three types of general-purpose authentication verification mechanisms:

- SecurID/ACE Server (RSA)
- RADIUS protocol authentication and accounting server access
- Local Password Database

Additionally, VPN Certificate authentication is available for use only with the VPN Client as well.

Windows domain authentication can be supported via certain RADIUS server implementations.

One additional feature is the ability to receive parameters from a RADIUS server. Certain parameters, in addition to success or failure of authentication, may be returned from a RADIUS server. The Brick allows those parameters to be used within the scope of the Brick's security mechanisms. Parameters which may be configured via RADIUS are:

- Authentication Timeout
- User Group
- Local IP address (Client VPN only)
- DNS primary and secondary servers (Client VPN only)
- WINS primary and secondary servers (Client VPN only)

Authentication may be used with either of two authentication processes: firewall authentication and VPN Client user authentication.

Firewall Authentication

Firewall authentication is provided via a HTTP or HTTPS/Web Browser access, using a two-step authentication procedure. First, the end user accesses a preconfigured IP address (the Virtual Brick Address of the associated Virtual Firewall) from his web browser. The Brick then provides a generic username/password web page. Upon successful authentication, the user is informed of his required reauthentication interval, and allowed to pass traffic, subject to configured firewall policy. Note that this process allows any protocol to be authenticated, even if the protocol itself doesn't support an authentication mechanism.

Additionally, it is possible to configure, via policy, a set of rules which forces any unauthenticated outbound HTTP or HTTPS traffic to be redirected to the authentication server, so that users need not have a priori knowledge of the authentication address.

VPN Client User Authentication

VPN Client User Authentication is provided via one of the supported IPSec Client softwares, installed on the user's workstation. This software provides IPSec-tunneled traffic from the user's workstation to the Brick device. Part of the tunnel establishment procedure involves authenticating the user; once authenticated, the tunnel is established, but the user may only access resources specifically granted his user group. Note that each Virtual Firewall can have its own Tunnel End Point, for true Virtual Firewall independence.

□

Quality of Service / Bandwidth Management

Overview

Bandwidth Management features provide the ability to both guarantee service as well as limit overloads, thereby helping to ensure the end-user experience is not compromised, even during an attempted attack. Additionally, these features are designed to help the Service Provider manage individual Customer bandwidth. This feature works in conjunction with the specific Denial of Service Protection features described in the next section.

Quality of Service features are provided via a Class-Based Queuing (CBQ) model. Resources are allocated in a tree-like structure, by dividing them downwards into Classes, from the root (the physical interface) all the way to the leaves (individual sessions through a given Virtual Firewall). Packets that required more resource than allocated can borrow resources if permitted, or are queued otherwise. Queued packets will remain queued until either the queue fills up, in which case they will be cleared in an as-needed basis, or until sufficient resources are freed, in which case they will be transmitted.

The CBQ class hierarchy is predefined on the Brick; it has four distinct levels:

- Physical Port
- Virtual Firewall
- Policy Rule
- Session

Again, all QoS enforcement is provided on stateful traffic that traverses the Brick device, so the session is affected, not just individual packets. Note that session-level Quality-of-Service control provides a direct control and effect with respect to the user experience.

Quality of service parameters may be specified at any level in the tree. In particular, every Virtual Firewall, and every Firewall Policy Rule may have QoS parameters configured. Offered parameters differ depending on the level, but the choices range from the following:

Guarantees and limits on:

- New Sessions Per Second
- Packets Per Second
- Bits Per Second

Guarantees provide ways to ensure that a given session can borrow and burst up to whatever capacity is desired, while still ensuring enough bandwidth for all users at times of peak demand. Limits provide hard controls that the Brick device will enforce - by dropping packets, if it becomes necessary.

The Brick device can also provide IP Type-of-Service field tagging, using either ToS templates or DiffServ codepoints. The IP ToS field can be set to a given value, configurable on both the Virtual Firewall as well as the Firewall Policy Rule level. It can also be set to differing values depending on whether or not the Brick device had to borrow bandwidth to meet the demand of a given session.

Additionally, the Brick device can generate both explicit and implicit bandwidth alarms. Explicit alarms can be configured on each Firewall Policy Rule to fire whenever a rule exceeds a specified bits/second, packets/second, new sessions/second rate. Implicit alarms will fire whenever a configured QoS boundary is crossed (or attempted to be crossed).

There are no additional license requirements for the QoS feature; it is included in the standard product, and supported on all Brick hardware models.

□

Denial of Service Protection

Overview

Denial of Service can be directed at two distinct points in the network: (1) at the protected hosts, such as web servers etc., and (2), at the network elements themselves, with the likely targets being firewalls and routers.

The Brick device offers five unique Denial of Service protection mechanisms. While each protects against a specific class of attack, the protections are general-purpose and can be tailored to both existing attacks as well as newly-emerging attacks not yet seen. The five explicit methods of protection are:

- Intelligent Cache Management (ICM)
- SYN Flood Protection
- TCP State Verification and Strengthening
- Robust Fragment Reassembly
- Application Protocol Anomaly Checks

Additionally, the Quality-of-Service features described above can be used to provide limits on connections, packets, and bits per second, an effective tool for use against flooding DoS in general.

Intelligent cache management (ICM)

ICM is used to ensure that the Brick device cache cannot be exhausted in a brute-force session-flood attack. Once enabled and triggered, the ICM feature proactively scans the Brick cache memory to target and purge cache entries that have been configured as lower priority, to ensure that highest-priority sessions have room in the cache. Without an ICM-like feature, any stateful device such as a firewall is subject to a trivial resource-consumption attack, easily launchable via a single 56k modem, resulting in a potential denial-of-service on the entire protected network. ICM is enabled and configured for the entire Brick device, since it is designed to protect the Brick itself from attack. This feature is patented by Acatel-Lucent.

SYN flood protection

SYN Flood protection is a specific protection from TCP SYN attacks on servers. Sending a flood of invalid SYN packets to a server may causes it to cease accepting new inbound TCP sessions, an effective Denial of Service.

The Brick device allows SYN Flood protection to be configured and customized on every Firewall Policy Rule. Configurable parameters are a half-open limit, to specify the number of half-open connections to each destination server required to activate the feature, as well as a half-open timer, to specify the number of seconds each session is allowed to be half-open. Once the limit threshold is exceeded, each session that

remains half-open beyond the timer will have a TCP reset (RST) packet sent by the Brick device to the affected server, to ensure that associated resources may be cleaned up and reallocated.

Since this second-generation SYN Flood protection incorporates both an activation counter as well as a session timer, it may be tweaked much more finely than can implementations that include one or the other.

TCP state verification and strengthening

Each Firewall Policy Rule can also have Strict TCP Validation enabled. Strict TCP Validation follows the series of TCP messages as the connection is established and ensures that only a valid TCP handshake can start a TCP session. All sequence number and acknowledgement numbers are verified to be in-window, for all packets in the TCP stream. TCP sessions must be closed with either a valid pair of acknowledged FIN exchanges, or a valid, in-window RST packet. If the session isn't in a valid TCP Established (fully-open) state, no data packets will be allowed to flow between the two endpoints. The Brick device will also protect against bad combinations of TCP flags, as appropriate to the current TCP state of each connection.

Additionally, the Initial Sequence Number for any TCP-based connection through the Brick may be optionally strengthened by rewriting the existing sequence number with a new, Brick-generated pseudorandom number. This can help protected servers or network elements be protected against ISN-prediction attacks.

Robust fragment reassembly

The Brick device will always reassemble IP fragments that pass through it. Overlapping fragments or duplicate fragments will be discarded. Packets that do not fully reassemble will be discarded without forwarding. The Brick will re-fragment packets as necessary according to the MTU on the destination network.

The Brick device itself is protected against resource starvation attacks designed to overload fragment reassembly queues. Continuous, sophisticated packet fragment attacks directed at the firewall will simply be discarded by the Brick device, while other traffic will continue unaffected.

Application protocol anomaly checks

Application protocol anomaly checks can be performed, which parses all significant protocol fields to determine field lengths for incoming messages, as a protection from buffer overrun attacks.

□

Brick Device Partitions

Overview

Brick device partitions provide a way to truly share a Brick device among multiple customers, placing no requirements on the customers and their IP space. Brick device partitions are used in conjunction with virtual firewalls to provide true isolation between different logical Brick devices in the same physical device.

Each Partition has its own set of VLANs, along with its own set of routing tables and virtual firewalls. Therefore, each Brick device partition may be used independently, even if multiple protected networks use overlapping IP addresses (e.g. RFC 1918 reserved addresses such as 10.0.0.0/8).

Although packets may not pass Partition boundaries ordinarily, there is a mechanism designed to permit carefully controlled inter-Partition interactions. This design can avoid hair-pinning packets out to an attached router then back into the Brick device, if desired.

□

Brick Device Failover/Redundancy & State Sharing

Overview

Brick device failover and state sharing is accomplished by installing two Brick devices, each connected to the same sets of networks on both sides. The Brick devices are bootstrapped identically, even down to the IP addresses and VLANs. The two Bricks then are booted and discover each other using Layer-2 multicast healthcheck messages, sent out all physical ports. One Brick device then elects to be the Active device, and one becomes the Standby device, using an empirical, deterministic algorithm.

The Active Brick device processes all packets through each interface. The Standby Brick does not process any packets, but does maintain communications with the Active Brick device. When health check information ceases to be heard by the Standby Brick, or when health check information indicates that the Active is less healthy than the Standby (determined by the number of physical interfaces up and available), the Standby Brick transitions to the Active state. If sane, the formerly Active Brick transitions to Standby state, brings link integrity down on all physical interfaces, and reboots.

Failover may be initiated manually from the Brick device management server, as well as from the Brick device console.

Failover detection and full activation occurs in about 400 milliseconds, preserving all state on the previously Active Brick device.

The Active Brick device also exchanges state information with the Standby Brick device over a specific link. That link can be chosen heuristically by the Brick device or a preferred link may be user-configured. In either case, if the chosen state-sharing path becomes unavailable, the Brick device will again heuristically search for the next best available link.

State information is sent from the active to the standby in real-time. All information is sent with an authenticating hash, unless otherwise configured. Critical information, such as VPN keys, are sent encrypted as well. New critical state information is shared in real-time at the maximum new session rate supported by a given Brick device; less-critical information is sent in batch mode a few times per second. Critical messages are also acknowledged by the standby.

Additionally, all policy modifications are transferred from the active to the standby securely, and success is only reported back to the management system if both systems accept and verify the change. Finally, OS updates are also transferred from the active to the standby.

Active/Standby Brick pairs share IP addresses and MAC addresses. When failover occurs, the now-Active Brick will perform a gratuitous ARP for each of the IP addresses on the shared MAC addresses, so connected switching elements will update

their MAC/interface binding. Additionally, the Brick will perform gratuitous ARPs for all entries in its MAC cache, to help ensure that session entries are properly switched as well.

One of the Brick devices in a failover pair can be designated as the *Primary* Brick device. The Primary Brick device will be the active Brick device at all times, unless it has experienced some failure or has lost its LAN connectivity. If the Primary Brick device is currently in standby state, and the currently active Brick device detects that the Primary Brick device has been up and running and has LAN connectivity, the active Brick device will initiate failover to the designated Primary Brick device after a provisionable Failback Delay time period has elapsed.

The exception to the above is if the Primary Brick initiated failover to the secondary Brick as a result of an IP tracking failure. In this case, the currently active Brick (the secondary Brick) will *not* yield control back to the Primary Brick after the Primary Brick reboots, unless some other failure occurs, IP tracking recovers and then fails again, or the failover is initiated manually.

□

Dynamic Address Support (including DHCP)

Overview

The Brick device has the ability to exist in a dynamic address environment. The Brick device can register its public address with its management server when used behind a many-to-one-NAT device. Additionally, the Brick can support having its own addresses assigned via DHCP or PPPoE as well as allowing DHCP requests to be forwarded to DHCP servers. The Brick device supports two simultaneous PPPoE address assignments for use in a redundant environment. These features, possibly used in conjunction with the UDP encapsulation supporting VPN tunnels, provide an effective CPE (customer premises equipment) solution for the small to medium size premise-based market.

Mobile Brick Device

The Brick device can be installed behind a many-to-one NAT (also known as PAT or NPAT). The Brick device management address is a private address, but this is translated to a public address upon making an outbound connection. The Brick device will register this public address with the SMS on first contact. The SMS will then use this public address for contacting the Brick device when necessary, rather than the Brick device actual (private) address. If that public address changes, the SMS will reregister that Brick device and use the new address.

DHCP Relay

The Brick device will recognize inbound DHCP messages and forward them to known, pre-configured DHCP servers.

DHCP Client

The Brick device will act as a DHCP client to acquire a DHCP address. The Brick device will renew that address as appropriately specified in the DHCP lease. The Brick device will register this address with the SMS, if appropriate, to use for management communications. The address acquired via DHCP may be mapped to any of the following purposes:

- Interface / VLAN IP Address
- Virtual Brick Address (VBA), in particular for Network Address Translation
- Tunnel End Point Address (TEP)

PPPoE Support

In a DSL network environment, dynamic IP addresses are assigned by DSL modems using PPPoE. The Brick device can operate in such a setting by being assigned an IP address by way of a PPPoE session. Up to two such sessions can be used to support several network configurations and they can also be treated as a redundant pair.

□

SNMP Agent on the Brick

Overview

The SNMP agent on the Brick, when enabled, allows an NMS to monitor and retrieve management data directly from a Brick without having to go through the SMS proxy.

When enabled, the SNMP agent on the Brick reports configuration information, operational status, and statistical information for that Brick.

The SNMP agent software on the Brick does not permit write or `SET` protocol operations to set or modify a Brick configuration for security reasons.

MIB

The MIB is an Alcatel-Lucent private enterprise MIB which largely mirrors MIB-II.

A private branch of the Brick module provides SNMP access to configuration, statistical, interface statistics, and tunnel endpoint information about the individual Brick, as well as LAN-LAN tunnel information and those Brick objects that map directly to a standard MIB or MIB-II object. The SNMP agent on the SMS does not report Object Identifiers (OIDs) from this branch.

Port (Link) Aggregation

Overview

This feature allows two or more physical Brick device ports to be combined into one logical, aggregated port. The zone(s) assigned to the aggregated port can now support a higher bandwidth.

□

Logging

Overview

All logging is performed in real-time from the Brick device to its management server (SMS, as described below). Log messages are sent via TCP for reliable delivery, encrypted over a mutually-authenticated channel. This logging mechanism has been empirically tested to range from 0.1% to about 1% of the inband data rate (in bits-per-second) depending on the application-layer protocol mix.

The Brick device generates the following types of log messages:

- Session logs
- Administrative Event logs
- Proactive Monitoring Statistic logs
- User Authentication logs
- VPN logs

The log data that is collected can be viewed in real-time or as historical data.

Session Logs

The following details pertain to session logs:

- One log message is sent on session establishment, one is sent on session completion. Session completions are explicit for TCP-based sessions, and based on timeouts for all other IP sessions. Session logs are sent in a batch if possible, but not held more than a fraction of a second to avoid troubleshooting latency. All session logs contain at least the following information (this is only a brief example; many fields are available in each log record):
- Date/Time stamp
- Physical Brick name
- Virtual Firewall Name
- Firewall Rule Number
- VLAN ID
- Source and Destination IP address information
- Source and Destination Layer-4 port information
- Source and Destination NAT addresses and ports (if applicable)

Session Completion Logs additionally contain:

- Session Duration
- Packet Counts (forward & reverse)
- Byte Counts (forward & reverse)
- Bad TCP packet counts (if applicable, forward and reverse)

- Session Termination reason (if applicable)
- Dynamic rule creation/usage (if enabled)
- Detailed command logging (if enabled)
- Application filter disposition (if enabled)

Administrative Event Logs

Administrative Events are generated by the Brick device for a variety of reasons, ranging from security audit attack information to simple ″re-configuration successful″ messages. Message content for Administrative Events depends strongly on the type of event; however, all Brick-based events contain a date/time stamp and a physical Brick device name designator.

Proactive Monitoring Statistic Logs

Proactive Monitoring statistic logs are sent periodically by each Brick device to its active management server. These logs contain MIB-II-like statistic information, such as packets in and out each interface, bytes in and out each interface, as well as overall Brick statistic information, such as CPU busy percentage, along with firewall and VPN policy statistic information. The default reporting interval of promon statistics log data by the Brick device is 30 seconds; this interval can be configured.

User Authentication Logs

The User Authentication Log contains messages that record successful or unsuccessful user authentication requests to the SMS or other external servers, such as RADIUS or Secure ID servers.

VPN Logs

The VPN Log contains records that pertain to all VPN tunnel transactions including all errors, events, messages, status, and keying information. The information allows easier debugging of VPN tunnel problems.

Resiliency of Log Transmission

Log messages are sent via encrypted TCP for reliable, secure delivery of messages. If no SMS hosts are reachable via TCP, the Brick will queue messages up to available RAM, and then discard additional ones. Note that the Brick will throttle duplicate log record transmission to avoid floods based on audit records.

Also, if desired, the Brick device will cease to forward any packets until such time as an SMS may be reached (this is generally only enabled in the most security-sensitive applications). The Brick device will choose one of two SMS servers or Compute

Servers to use at any given time for log transmission. If multiple servers are available, the Brick device can be configured to either latch at the current server, or prefer a given server.

Debugging/Command-Line Interface

The Brick device has a console used for debugging purpose only, which is password protected. The console can report overall status on Brick configuration as well as existing cache information and other status information. It can also perform packet and audit tracing capabilities. The console is also capable of performing ping and Traceroute functions. A reboot can be initiated, and, in the case of a failover pair of Bricks the console can report failover status as well as initiate a failover.

The Brick device console is available from a variety of different sources:

- a locally attached VGA monitor and PS/2 keyboard
- a locally attached RS-232 serial terminal
- the Alcatel-Lucent Security Management Server Remote Navigator (Graphical User Interface)

The first two access mechanisms are local only, and the third, while remote in-band, is strongly secured.

It is important to point out that no policy modifications may be made from the Brick console whatsoever. The required Alcatel-Lucent Security Management System (SMS) is the only way to affect a change in the Brick device policy. The Brick device console can be used for debugging or troubleshooting.

SMS reports

SMS reports provide a useful tool for identifying the nature of firewall and tunnel problems. You can use the SMS reports Memorize function to build a library of reports to assist in diagnosis of specific kinds of transmission failures.

SMS Administrators can run reports for any group. Group Administrators can only run reports for groups for which they have at least *View* privileges.

Reports can be run from any machine that is running the SMS Navigator or Remote Navigator. While reports cannot display real time information, as logs can, they do allow access to the same information as contained in the Historical logs from any location.

Web browser software must be installed on your workstation in order to view SMS reports. The following browsers are supported by the SMS:

- *Microsoft*® Internet Explorer 5.5 or later
- Netscape Navigator 4.7 or later
- Firefox 2.0 or later

If you are running the SMS application on a *Solaris*® workstation, you will be prompted for the path to the browser software (for example, `/usr/sfw/bin/mozilla`) in order to bring up the browser and view the selected report.

The SMS provides the following types of reports:

- *Administrative Events Report*
  The Administrative Events report can be used to monitor network events and, in particular, troubleshoot problems related to routing, LAN-LAN tunnels, and Client-LAN tunnels.
- *Session Log Report*
  The Sessions Logged report can be used to monitor traffic on the network, identify potential problem areas, and troubleshoot security operations.
- *Closed Session Details Report*
  The Closed Session Details report can be used to monitor traffic on the network, identify potential problem areas, and troubleshoot security operations.
- *Alarms Logged report*
  The Alarms Logged report can be used to troubleshoot network problems by tracking and analyzing alarms
- *User Authentication Report*
  The User Authentication Report can show all login attempts, successful and unsuccessful. It can show user authentications performed by the local SMS database, as well as authentications performed by any RADIUS or SecurID servers referenced by the SMS.
- *VPN Events Report*
  The VPN Events Report is used to generate a report for all VPN-related events, such as setting up and taking down of VPN tunnels.
- *Audit Trail Report*
  The Audit Trail Report can be used to show any changes made by an administrator (additions, deletions, modifications) to an object managed by the SMS, such as Bricks, Brick zone rulesets, host groups, service groups, application filters, LAN-LAN tunnels, user accounts, and Alarm triggers. An archive file of the change(s) to an object is created with each modification, showing the object's current parameter settings.
- *Rule Statistics Report*
  The Rule Statistics Report is used to show hit counts on rules of a given Brick zone ruleset or hit counts on a given zone ruleset for a group or one or more Bricks during a specified time period.

□

Capacity/Throughput

Overview

Capacity and throughput varies strongly with each Brick device model. Overall throughput is a function of the Brick device hardware architecture as well as the speed and model CPU in the Brick device . Overall capacity is largely a function of the amount of physical RAM installed in the device.

Important! Please contact the Alcatel-Lucent VPN Firewall team directly for up-to-date performance statistics.

□

Certifications

Overview

Product safety and emissions certifications depend on the Brick model. Refer to the User's Guide of the respective Brick device for a list of certifications.

□

2 Alcatel-Lucent Security Management Server (SMS)

Basic Design

Overview

The Alcatel-Lucent Security Management Server (SMS) is a software-based system that is used to manage a network of Brick devices. Currently, up to 1,000 Brick devices may be managed from a given SMS host. The SMS software is designed to run on *Solaris*®, *Windows*® XP Professional or Server 2003, *Microsoft*® *Vista*®, and Linux server platforms.

The SMS is the central repository for configuration management, audit collection, alarm generation, and secure communications for the VPN Firewall system. All changes to the Brick device configuration must be performed using the SMS.

□

Tiered Model

Overview

The SMS is installed in a central location, with logical access to all Brik devices via an IP network. The SMS is accessed by administrators using a built-in utility called Navigator. SMS can also be accessed remotely using the SMS Remote Navigator, an included component, which may be downloaded from the SMS via HTTP/S and installed locally on multiple management workstations. SMS Remote Navigator provides the full functionality of a SMS Navigator.

Once the SMS Remote Navigator is installed on a user's workstation, that software can access different SMS servers.

Firewall administrators use the SMS Navigator to control all aspects of every Brick device in the system, ranging from IP addresses and VLAN information all the way to firewall policy and VPN tunnels. No other mechanism is necessary; the SMS Navigator provides full and complete access to all aspects of Brick device management.

Levels of administrative control

To promote the efficient management of Bricks, policies, tunnels, and users on a local basis, the SMS allows an administrator to organize the entire constellation of managed objects into separate groups, and then to further subdivide objects within a named group into folders and subfolders. Some objects can be made globally visible to all other groups.

Groups can be administered by both SMS Administrators and Group Administrators. SMS Administrators have full privileges over all groups, which means they can access all folders in all groups and make any additions, modifications, or deletions they deem necessary.

Group Administrators, on the other hand, can only access the specific groups to which they are assigned. In addition, Group Administrators can be given three levels of privilege over the folders in their groups: None, View and Full.

This means that multiple Group Administrators can be created for a group, each with different privileges, to administer different aspects of the group's operations. For example, one Group Administrator could have Full privileges over devices, but only View privileges over policy, while a second Group Administrator could have View privileges over devices and Full privileges over policy.

Concurrency control

The SMS application is a carrier grade centralized management system capable of managing a large number of objects (Bricks, zone rulesets, Host groups, tunnels, service groups, and so forth). The group-based model allows the creation of multiple

management domains, where each group contains a set of resources. Multiple administrators may have access and the desire to modify the same object simultaneously. This raises the potential for problems caused by simultaneous changes to an object by multiple administrators, or editing of an outdated instance within an object by an administrator.

The SMS has a Concurrency Control feature that prevents changes from being made to a managed object, such as a Brick or zone ruleset, by more than one administrator at a time. With Concurrency Control enabled, an object opened for **Edit** by an administrator is “locked out” to other administrators until the managing administrator completes and saves the changes.

Multiple administrators can open the same object in **View** mode, but only one administrator can open the object in **Edit** mode.

□

SMS Policy Objects

Overview

SMS resources are divided into SMS Groups, each containing sets of resources. In a Service Provider model, SMS Groups may be used to designate Customers of that Service Provider. Enterprises can use a single Group or decide to use multiple SMS Groups to delineate geographical regions, operating divisions, etc.

Brick devices, Firewall and VPN Policies, Users, and other policy objects are contained within SMS group. Some objects within a group (such as a host group) can be provisioned to be displayed and used globally across any group. Arbitrary combinations of IP addresses and ranges can be contained in Host Group objects. Collections of IP protocols, source and destination Layer-4 ports are contained in Service Groups. Users are collected into User Groups. Certain policy objects, such as Host Groups, Service Groups, for instance, may be dynamically nested inside one another, to allow flexible and sensible object hierarchy.

The SMS comes pre-installed with more than 60 of the most commonly-used Service Groups, as well as several Brick Zone Ruleset templates to speed up initial deployment.

A ″special″ group in the SMS object model exists called the *system* Group. Policy objects in other Groups may be applied to Brick devices in the *system* Group, thereby creating a set of ″shared″ devices. The Brick device may in fact be managed by a single entity, and the policies installed on it may be managed by a set of other entities.

Additionally, policy objects may be made ″global″ in the sense that they may be used in a Brick zone ruleset (but not modified) by other SMS Groups. These Global objects are dynamically modifiable within the original Group from which they were made Global. Both Global and Nested objects help the administrator avoid ongoing data duplication within objects.

□

Permissions Model

Overview

Administrators of the SMS have one of two roles: SMS Administrators and Group Administrators. SMS Administrators have full access to all aspects of the SMS, managed devices, security policies, VPN tunnels, and Users. Group Administrators may have configurable Read/Write (full), Read Only (view), or No access over the set of configured SMS Groups. In general, all objects reside within a given SMS Group.

For example, the Virtual Firewall policies resident within a given group are constructed using the policy objects within that Group, and are applied to Brick devices contained within that Group.

Permissions verifications are pervasive across SMS system tools; in general, tools require authentication and will only allow access to those resources to which the administrator has permissions.

Secure Communications

Overview

The original purpose of the Alcatel-Lucent Security Management Server (SMS) was to provide a secure mechanism to remotely provision Brick devices. It was designed with 20/20 hindsight of other firewalls at the time that had been compromised due to attacks on the remote provisioning mechanisms.

The foundation for the Brick-to-SMS security is a cryptographic system of digital signatures and authentication keychains such as to provide a high degree of assurance that the Brick devices may not be re-provisioned by any individual or system other than the dedicated SMS.

Each Brick device is allocated a certificate by the SMS. That certificate is used to verify mutual authenticity, as well as the foundation for confidential, authenticated, and integrity-verified channel. Diffie-Hellman, DSS, 3DES and SHA-1 are all used to provide a secure channel between the SMS and the Brick device.

Generally, the Brick device is bootstrapped by configuring its basic parameters via the SMS and then generating a ″boot floppy″ disk. Newer models of Brick devices will use a portable USB storage device in place of the floppy drive or USB flash drive. This disk is physically inserted into the floppy drive on the Brick, which then copies its operating system, and basic boot parameters (but NOT its security policy) from the floppy to the local flash disk. The disk is then removed and destroyed, and the Brick is booted. When the Brick device boots, it contacts its management server (the SMS) for a secure transfer of its policy. The physical floppy disk may be created from any workstation, including one that does not have access to the SMS directly. Additionally, a Brick device may be bootstrapped via its serial port with a terminal application .

Each administrator is also allocated a certificate by the SMS. That certificate is used to verify all tasks performed by that administrator. In fact, the policy stored on the Brick’s NVRAM disk is digitally signed by the SMS as well as the administrator who applied the Virtual Firewall policy to that Brick device. This mechanism alone makes unauthorized re-provisioning of the Brick device-while not impossible-cryptographically infeasible.

Since the SMS is a required part of the architecture, it is used to facilitate certain types of maintenance issues as well. The Brick operating system can be pushed to each Brick from the SMS system, without physically interacting with the device, and in a secure fashion. With a failover pair of Bricks, this OS upgrade can even be done with no downtime, maintaining all sessions.

Administrators can use either local password authentication or external database authentication with either SecurID or RADIUS servers. The authentication mechanism (which may include a pointer to an external database) is configurable on a per-administrator basis. All logins to the SMS use this unified administrator database for AAA access.

□

Log Collection System

Overview

The SMS is the central point for log collection in the Lucent VPN Firewall system. Audit logs fall into one of the following categories:

- Audit Trail Log
- Administrative Event Logs
- Proactive Monitoring Statistic Logs
- Firewall Session Logs
- User Authentication Logs
- VPN Log

The SMS creates a new log file for each type once per day or when the existing file reaches a size user-configurable by file type. Finally, each log file type has a user-configurable total maximum size, to avoid filling up the SMS disk. Logs may also be scheduled for automatic transfer from the SMS via FTP to an arbitrary FTP server, if so desired.

Logs are stored in a well-defined manner on the SMS host, using colon-delimited ASCII text fields. Wherever non-ASCII information must be represented, it is converted into ASCII via an encoding scheme.

Logs may be viewed in Real-Time, using the SMS Log Viewer (see below) or historically using either the Log Viewer or the Reporting System (see below).

Log viewing is protected by permissions; an administrator will only be permitted to view logs specific to the devices over which he has at least Read Only access. Log records pertaining to the entire system are only viewable by SMS Administrators.

□

Compute Servers

Overview

The fact that all Brick devices send log information to the centralized SMS could become a bottleneck for an extremely large network with thousands of Brick device or having very high traffic. To further enhance the scalability of the Alcatel-Lucent security solution, a set of additional servers, known as Compute Servers (CSs), expand the logging and data collection capabilities of an SMS. The CS(s) acts as log collection points and increases the total number of supported Bricks as well as total log traffic that can be logged by these Bricks. Sometimes, for network efficiency reasons, it may be desirable to deploy local / regional log collection and the CSs are an ideal solution, providing substantial saving of WAN bandwidth used by log transmission. CSs are also managed by the SMS.

The Primary and secondary SMSs both contain their own databases which are kept synchronized. The CSs focus on enhancing the capabilities of SMS by providing a large number of log collection points and do not contain any database.

The CS accesses the database on its associated SMS. The SMS and all its related CSs are referred to as a unit called the SMS Cluster. Each SMS Cluster contains a SMS (A Primary SMS and up to three other Secondary SMSs) and one or more CSs. The CSs within a cluster can be geographically distributed and will communicate securely. One SMS server can support up to five CSs. Each CS, in turn, can collect logging data from up to 1,000 Brick devices.

The CSs obtain all of their data from the SMS database. Each Brick (or other managed device) is configured with a list of log collection points (Note that both CSs as well as SMSs are valid log collection points) with a preference order and will send logs to the log collection server to which is it is currently homed. Thus, an SMS Cluster is able to manage a larger number of Bricks as well as collect a larger volume of log data from Brick devices.

Most of the SMS tools are also available on the CSs. Administrators will be able to log into the Compute Servers to do most management activities including creating and updating Bricks and Policies for the Brick devices that are associated with that CS.

□

Configuration/Change Management

Overview

Each object modification performed by an administrator is logged by the SMS in the Administrative Event Log. Additionally, the full and complete state of that object post-modification is stored in an additional Change History folder on the SMS host, in an individual file for each time the object is modified. This file is then hashed using an HMAC, and the hash stored in the Administrative Event log.

An SMS utility provides a mechanism to verify that the hash in the log file matches that in the Change History folder.

The Change History files are stored in a format that is easily pushed back into the SMS via the SMS Command Line mechanism (see below).

□

Reporting System

Overview

The SMS has the ability to generate HTML-based reports, and serve them via its own internal secure web server (HTTP or HTTPS). These reports are basically reformatted versions of the SMS logs, with full filtering and sorting capabilities. Reports may be limited to specific physical devices, Virtual Firewalls, time period ranges, or several other criteria. Reports may also be memorized such that they may be run again later, with other matching criteria.

Reports include sessions over time, policy snapshots, administrator events, auditing of modifications made by administrators to objects managed by the SMS, and Brick zone rule and ruleset usage counts.

The SMS provides a number of preconfigured reports, to allow fast initial deployment.

Reports are stored per-administrator, so each system administrator may keep track of his own information, as needed.

□

Alarm System

Overview

The SMS generates alarms based on Brick device log messages, as well as locally generated log messages from the various SMS subsystems. Alarms consist of two parts: triggers and actions. Alarms are configured per-administrator, so each system administrator may configure the alarms in which they are interested, and be notified by methods appropriate to the administrator, as well as the specific alarm.

The SMS provides a number of preconfigured alarms, to allow fast initial deployment.

Triggers

Triggers cause the alarm to be fired. Triggers contain the criteria matching information, thresholding information, applicable devices or subsystems, etc. Each trigger must be mapped to one or more action.

Some of the configurable trigger types are as follows:

- Application Process Hung
- SMS Error
- Brick Error
- Brick Lost/Found
- Brick Interface Up/Down
- Proactive Monitoring Threshold Crossing
- Brick Redundancy Alarms
- SMS Redundancy Alarms
- Brick SLA Round Trip Delay Alarms
- QoS Bandwidth Alarms
- Alarm Code
- VPN Proactive Monitoring
- User Authentication
- Unauthorized SMS login attempts
- LAN-LAN tunnel lost/up
- Brick ICM Alarm

Actions

Actions allow each trigger to cause some response to be taken by the SMS. These actions all center around notifying the administrator. Actions are customizable so as to send to each administrator in the most convenient notification mechanism for that particular administrator.

Notification mechanisms include:

- Console Alarm (via the SMS Remote Navigator)
- Email
- Out-of-band modem-dialed alphanumeric message sent to pager (via the TAP protocol)
- SNMP Trap (V1 or V2c)
- SYSLOG Message (with configurable SYSLOG level)

□

Real-Time Display (Status, Graphs, Logs)

Overview

The SMS Remote Navigator provides multiple mechanisms for reporting real-time information regarding the status of the system.

Brick Status Viewers

Brick status is provided via real-time windows for each Brick, and overall for aggregate collections of Bricks. Single-Brick status provides up-down statistics for each physical port, along with packet, byte, and session statistic collection information. Physical status provides information specific to the physical port, including DOT3 errors, in-out counters, and other information. Quality-of-Service graphs display throughput and performance relative to configured guarantees and limits for an at-a-glance view of the offered traffic as a function of configured bandwidth.

VPN Tunnel Status Viewers

The status of all VPN Tunnels can be viewed at-a-glance. Each tunnel status is reported and summarized. Individual users using Client VPN can even be disabled in real-time, if so desired. Additionally Service-Level Agreements (SLAs) can be monitored for VPN tunnel round-trip delay.

Administrator and SMS Status Viewer

All logged-in administrators may be viewed in real-time, along with their connection statistics. The connection status of each SMS can also be viewed in real-time.

System Status Graphs

Many aspects of the system can also be graphed, using dynamically-updating strip charts. The starting place for this view is an integrated Dashboard, which displays in a single window the entire status of the system, including current status as well as local history. Overall graphs include:

- Total number of Bricks up/down
- Total number of currently logged-in users, by type
- Total number of VPN tunnels up/down, by key negotiation method
- Overall Brick sessions by Pass/Drop/Proxy//NAT/IPSec
- Overall sessions by IP protocol
- Overall packet counts in/out
- Number of SMSs/CSs up/down

Real-time device status can be synchronized across redundant SMSs or multi-site SMSs. Graphs provide both real-time as well as historical data.

Additionally, specific elements of individual Brick devices may be graphed.

Also, sets of Brick devices may be collected into a ″monitored″ group, with statistics provided overall for that group, disregarding all other Bricks in the system.

An administrator may login using only ″Status Monitor″ view, providing a view into the system which cannot be used to modify any configuration; this is ideal for a Network Operations Center wall screen or other persistent monitoring system.

Real-time Log Viewer

The Log Viewer application is launchable from the SMS Remote Navigator. It has the ability to display log records in real-time, as received from all Brick devices in the network, from one centralized point. These messages can be filtered and sorted, and the filters can be stored for future use. The Log Viewer also provides a historical record search capabilities, within specified time parameters. This real-time Log Viewer is typically used for troubleshooting and debugging, as well as conducting security audits of attacks in progress.

SMS Messenger

The SMS Messenger allows logged-in administrators to send text messages to each other. Note that this mechanism is controlled completely within the SMS and requires no commercial ″external″ servers or services. Additionally, the feature works with administrators logged in to either SMS in a redundant environment.

SMS Process Status Viewer

For those administrators responsible for overall operations of the SMS host itself, a real-time SMS Process Monitor application is also available remotely. This application can display and graph real-time resource utilization of the SMS host.

□

SNMP Agent on the SMS

Overview

The SMS provides an SNMP agent to use for accessing configuration information for all Bricks, management status and hardware alarm status for all Bricks, and statistical information about the state and health of all Bricks that are homed to that SMS and the SMS itself, by a Network Management System (NMS) issuing GET or GETBULK SNMP requests. Requests to set or modify information via the SNMS agent are not accepted. SNMPv1 and SNMPv2c are supported, only over User Datagram Protocol (UDP). SNMPv3 is currently not supported.

The MIB is an Alcatel-Lucent private enterprise MIB which largely mirrors MIB-II along with selected parts of the bridge and Etherlike MIBs, including DOT3 statistics.

Although the SMS provides information on behalf of managed Bricks, the SNMP agent is *not* a proxy agent in the strict sense. Requests to the SMS SNMP agent is serviced by information local to the SMS, and will not result in a query to any Brick.

Redundancy and Availability

Overview

Basic redundancy is provided by two SMS servers that are installed in an active/active fashion. These two active SMS servers maintain their configuration databases across the network via real-time database replication. All inter-SMS communication is secured. Starting with Release 9.0, for additional capacity and security in large-scale, multi-site network designs, up to three Secondary SMSs can be connected to a Primary SMS.

Since both the Primary and Secondary SMS servers are simultaneously active, each Brick device can be configured from any SMS. Each Brick maintains a list of SMSs/Compute Servers (CSs), and can be configured to prefer one SMS/CS over the other servers (if it is available). Each Brick device sends log files to its currently-active SMS/CS. A Brick device can be manually ″rehomed″ to the other server(s), if available.

The SMS will automatically back up its internal database to a local disk once a day. Additional backups can be scheduled at any time; backup files can be transferred to a remote site for archival storage and disaster recovery. This single backup file contains ALL policy, configuration, and security information for ALL configured devices and policies.

□

Command-Line Interface

Overview

The SMS Command Line (CLI) feature is designed to allow administrators the ability to configure many SMS components and policy objects by using a text file-based interface. This CLI is available from the SMS host, from a host running the SMS GUI remotely, or from the Brick serial port console (remote access to the SMS host must be set up by SMS administrator). This feature is designed to be easily scriptable from an external application running on the SMS host.

Configuration Assistant

Overview

The SMS Configuration Assistant, securely available from the SMS Remote Navigator, allow SMS Administrators the ability to edit system-wide parameters, such as login timeouts and log file parameters.

Brick Device Remote Console

Overview

The SMS Remote Navigator Remote Console allows administrators the ability to bring up a secure remote console to a given Brick device and execute Brick debugging/troubleshooting commands using the Brick CLI. This console is both secure from the user's workstation to the SMS, as well as from the SMS to the Brick. No policy modifications may be made from this Remote Console (or any Brick console interface).

3 Alcatel-Lucent IPSec Client

Overview

Purpose

The Alcatel-Lucent IPSec Client is a software component designed to provide secure Client-to-Gateway IPSec-based connectivity. The IPSec Client provides a host of security and interoperability features designed to allow the roaming user to securely and conveniently connect back into his main network over an untrusted network like the Internet. The IPSec Client has been designed to work with the Alcatel-Lucent VPN Firewall system; some features require the use of an SMS or Brick device to function as described herein.

Contents

Platforms and Compatibility

Overview

The IPSec Client can be installed on the following *Microsoft® Windows®* server platforms:

- *Windows®* ME
- *Windows®* 2000
- *Windows®* XP

Third-party testing has been performed documenting installation of the IPSec Client on a variety of PCs from different vendors. Please contact the Alcatel-Lucent VPN Firewall team directly for this information.

Supported standards

The IPSec Client supports the following security standards

- IPSec
- IKE
- Diffie-Hellman Group 1, Group 2, Group 14, Group 15, Group 16, and Group 5
- DES
- 3DES
- SHA-1
- MD5
- IPComp (LZS compression)
- X.509
- PKCS #12

Personal firewall

The IPSec Client supports a stateful personal firewall. The firewall can be set differently depending on whether or not a tunnel is currently established. In normal operational mode (no tunnel is up), the firewall setting is under the control of the end user. However, when the tunnel is established, the firewall setting is controlled by the administrator. The personal firewall currently has three modes:

- Block All
- Pass All
- Pass only Client-initiated (outbound sessions only)

UDP encapsulation

The IPSec Client has the ability to tunnel IPSec inside of UDP packets, for the explicit purpose of using in a many-to-one NAT/PAT environment. The method of UDP-encapsulation is Alcatel-Lucent proprietary and not designed to interwork with other non Alcatel-Lucent products.

Local presence

The local presence feature allows the client's PC to be assigned an address local to the network to which they are connecting. This allows complex connections, such as X-Windows, to be directed back from other hosts to the client host, properly using established network routing paths. The local addresses are assigned using a local pool managed by the SMS, or one-at-a-time using the RADIUS parameter download feature.

Split tunnels

The IPSec Client has the ability to permit simultaneous traffic in clear-text as well as through the tunnel. The endpoint IP networks behind the tunnel are configured by the system administrator on the SMS, and can be configured to disallow clear-text traffic entirely if so desired.

Strong authentication

In addition to basic IKE authentication, the IPSec Client supports the use of Strong Authentication mechanisms. The client can provide both XAuth or proprietary strong authentication protocols, depending on the endpoint to which it is terminating. The client will support RADIUS, SecurID, and local passwords, including SecurID time sync mode.

Multiple tunnel configurations with redundancy

The IPSec Client can be configured and saved with a number of tunnels, each with a different endpoint and other configurations. Additionally, each tunnel can have its own backup tunnel endpoint, in case the primary tunnel endpoint is not reachable at tunnel establishment time.

DNS/WINS

Upon tunnel establishment, the IPSec Client will automatically configure local primary and secondary DNS (Domain Name Server) and WINS (Windows Information Name Server) addresses. This information is configured by the Administrator for each tunnel.

Windows domain authentication

If desired, the user can be automatically logged-on to a remote Windows Domain. Upon authenticating, the user will have access to any configured domain resources, including file servers and print servers.

RADIUS parameter download

The RADIUS attribute containing the applicable information is user-configurable.

The administrator has the ability to configure certain user-specific parameters in their RADIUS database, some of which can be used when an IPSec Client VPN user establishes a tunnel. Parameters that can be configured include:

- Local Presence address
- Primary/Secondary DNS
- Primary/Secondary WINS
- Login Timeout
- Idle Timeout
- User Group
- Login Banner
- Client Program Information

Pleasant push software upgrade

If a new version of the IPSec Client is available on the SMS, upon tunnel establishment, the user is prompted to upgrade their software, if desired. A single click will accomplish download and installation of new IPSec Client software on the user's PC.

customization and branding

The Alcatel-Lucent IPSec Client contains features to allow an organization to customize the graphical appearance to be customized as desired. This includes images as well as text both in the installation process as well as the runtime software.

Message of the day

The IPSec Client can be configured to display a message of the day (MOTD) set by the SMS Administrator. This message is downloaded and displayed upon tunnel establishment, and must be acknowledged by the user before continuing.

Client log

The IPSec Client maintains logs of connection attempts, including detailed IKE and IPSec negotiation, to aid troubleshooting. These are viewable by the user.

Windows tray icon

The IPSec Client displays an icon in the Windows Menu Bar Tray (usually lower right corner). This icon indicates status of the tunnel (up/down) with a color change to help the user visually confirm their tunnel status at a glance.

□