Lucent Technologies
Bell Labs Innovations

Lucent Security Management Server

Release 9.1

Technical Overview

260-100-022R9.1
Issue 1
August 2006

Contents

About this information product

Purpose

This document is a technical product description and overview of the Lucent VPN Firewall system. It contains descriptions of all system components and features up through and including Release 9.1.

Reason for reissue

Reissued for Release 9.1

VPN firewall solution components

The Lucent VPN Firewall system consists of these components:

- VPN Firewall *Brick*®
- Lucent Security Management Server (LSMS)
- Lucent IPSec Client

The *Brick*® is a hardware appliance-based product. The LSMS is installed on a general-purpose host. The 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 email your comments to the Comments Hotline (comments@lucent.com).

1 VPN Firewall *Brick*®

Basic Physical and Logical Architecture

Overview

The VPN Firewall *Brick*® is a high-speed packet-processing appliance, primarily oriented towards providing security functions. The Brick is offered is 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 Bell Labs. The Brick device operating system is an evolution of Lucent's Inferno operating system , designed for small embedded security applications. The Brick OS 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) and Gigabit copper interfaces are available.

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 will device configurably 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.

□

Virtual Firewalls

Overview

Every Brick device supports the use of Virtual Firewalls. Each Virtual Firewall is a set of policy rules which specific what types of traffic are allowed, and how that traffic is processed. Virtual Firewalls are also known as Brick Zone Rulesets, often shortened to 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 Virtual Firewall will only apply to traffic to or from those IP addresses on those VLANs. Wildcards may be used in such assignment. Additionally, Virtual Firewalls may be applied to multiple Brick devices, via the Lucent Security Management Server (see associated section in this document).

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

While Virtual Firewalls may be assigned to VLANs, the two features are in no ways interdependent. VLAN tagging may be used with or without Virtual Firewalls, and Virtual Firewalls may be used with or without VLAN tagging.

Each Virtual Firewall may also have a single Virtual Brick Address (VBA) associated with it; this VBA may be used for multiple purposes, including Network Address Translation as well as acting as a Tunnel End Point for VPN (see below).

Virtual Firewalls can also be used to represent different customers in a multi-tenant environment. Sessions are unique within each given Virtual Firewall, and, when used in conjunction with Brick partitions (see below) can be used to ensure session independence in a shared environment.

□

Stateful Packet Filtering

Overview

Every packets 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 Virtual Firewall 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, etc.) Of course, this explanation is vastly simplified; many criteria are used to evaluate packets as they flow through the packet, depending on the type of packet and Brick device configuration.

Each Virtual Firewall 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 LSMS server, plus or minus an administrator specified offset.

The Lucent VPN Firewall 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]
- SMTP [Protocol Anomalies Checks, Commands filtering, Hide banner information, Block Spoofed Outgoing Mails, filter MIME types/attachments] (New in R8.0)
- FTP [Protocol anomaly check, Commands Filtering, prevent connection stealing, restrict Dynamic Ports, restrict users, prevent dictionary attacks, etc.] (New in R8.0).
- 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]
- 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 a preconfigured known DHCP server, 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 LSMS 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 LVF Brick will automatically open 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 LVF Brick will perform NAT on all addresses, whether in the IP header, or H.323 payload. This includes H.323/Q.931 messages, as well as H.245 messages, along with 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 5

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

For Strong User Authentication

The following encryption 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

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

Both AH and ESP types of IPSec are supported. Only tunnel mode is supported. VPN is supported on all models of the Lucent VPN Firewall; however, certain models support an optional hardware VPN encryption acceleration card (EAC).

The Lucent VPN Firewall 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). This mechanism is supported between Lucent Brick devices (LAN-LAN tunnels) and the Lucent IPSec Client (Client-LAN tunnels) only.

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

LSMS interoperates with IPSec Clients from multiple vendors, e.g. 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 three modes: Direct, Pool, 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). 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 three modes: Direct, Pool, 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. 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), provided care is taken to ensure ARP/MAC issues are addressed.

□

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 in a multi-tenant application. 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 four explicit protections 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 Lucent Technologies.

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.

□

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 Firwalls. 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.

□

Dynamic Address Support (including DHCP)

Overview

The Brick device has to 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 a DHCP server. 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 LSMS on first contact. The LSMS 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 LSMS will reregister that Brick device and use the new address.

DHCP Relay

The Brick device will recognize inbound DHCP messages and forward them to a known, preconfigured DHCP server.

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 LSMS, 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 now 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.

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 (LSMS, see 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 five distinct types of log messages:

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

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 VPN tunnel status and keying information, as well as simple ″reconfiguration 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. These are sent every 30 seconds, nonconfigurably.

User Authentication Logs

The User Authentication Log contains messages that record successful or unsuccessful user authentication requests to the LSMS 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, and messages. 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 LSMS 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 LSMS may be reached (this is generally only enabled in the most security-sensitive applications). The Brick device will choose one of two LSMS 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 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 Lucent Security Management System is the only way to affect a change in the Brick device policy. The Brick device console can be used for debugging or troubleshooting.

□

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 Lucent VPN Firewall team directly for up-to-date performance statistics.

□

Certifications

Overview

The LVF Brick device has been approved by the following certifications:

- FIPS 140-1 level 2 (FIPS 140-2 level 2 in progress)
- ICSA Firewall 3.0
- ICSA VPN 1.0B (1.0D in progress)
- EAL 2 (EAL 4 in progress)
- OSMINE
- NEBS-3

□

2 Lucent Security Management Server

Basic Design

Overview

The Lucent Security Management Server (LSMS) is a software-based system used to manage a network of Lucent VPN Firewall *Bricks*®. Currently, over 5000 Bricks may be managed from a given Lucent Security Management Server host. The LSMS software is designed to run on *Solaris*® and *Windows*™ 2000 Professional, 2000 Server, XP Professional, or Server 2003.

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

□

Tiered Model

Overview

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

Once the LSMS Remote Navigator is installed on a user's workstation, that software can access different LSMS servers - even if they are running different versions of the LSMS software.

Firewall administrators use the LSMS 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 LSMS Navigator provides full and complete access to all aspects of Brick device management.

□

LSMS Policy Objects

Overview

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

Brick devices, Firewall and VPN Policies, Users, and other policy objects are contained within LSMS Groups. 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, etc, may be dynamically nested inside one another, to allow flexible and sensible object hierarchy.

The LSMS comes pre-installed with dozens 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 LSMS object model exists called the System Group. Policy objects in other Groups may be applied to Brick devices in the System Group, thereby created 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 policy (but not modified) by other LSMS 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 LSMS have one of two roles: LSMS Administrators and Group Administrators. LSMS Administrators have full access to all aspects of the LSMS, 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 LSMS Groups. In general, all objects reside within a given LSMS 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 LSMS 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 Lucent Security Management Server 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-LSMS 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 LSMS.

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

Generally, the Brick device is bootstrapped by configuring its basic parameters via the LSMS 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 LSMS) 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 LSMS 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 LSMS. 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 LSMS as well as the administrator who applied the Virtual Firewall policy to that Brick device. This mechanism alone makes unauthorized reprovisioning of the Brick device-while not impossible-cryptographically infeasible.

Since the LSMS 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 LSMS 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 LSMS use this unified administrator database for AAA access.

□

Log Collection System

Overview

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

- Firewall Session Logs
- Administrative Event Logs
- User Authentication Logs
- Proactive Monitoring Statistic Logs
- Packet Trace Logs
- VPN Log

The LSMS 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 LSMS disk. Logs may also be scheduled for automatic transfer from the LSMS via FTP to an arbitrary FTP server, if so desired.

Logs are stored in a well-defined manner on the LSMS 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 LSMS 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 LSMS Administrators.

□

Compute Servers

Overview

The fact that all Brick devices send log information to the centralized LSMS 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 Lucent's security solution, a set of new servers, known as Compute Servers were introduced in LSMS R8.0. These new servers act as log collection points and will increase the total number of bricks as well as total log traffic that can be logged by these bricks. Sometime for network efficiency reasons, it may be desirable to deploy local / regional log collection and the compute servers are an ideal solution providing substantial saving of WAN bandwidth used by log transmission. Compute servers are also managed by the LSMS.

The Primary and secondary LSMSs both contain their own databases which are kept synchronized. The Compute Servers focus on enhancing the capabilities of LSMS by providing a large number of Log collection points and do not contain any database.

They access the database on the associated LSMS. The LSMS and all its related Compute Servers are referred to as a unit called the LSMS Cluster. Each LSMS Cluster contains a LSMS (primary, secondary, or a standalone) and few Compute Servers. The compute servers within a cluster can be geographically distributed and will communicate securely. One LSMS server can support up to five Compute Servers, each Compute Server can support about 1,000 Bricks. Currently compute servers are supported only on Windows platform.

The Compute Servers get all their data from the database on the LSMS. Each Brick (or other managed device) is configured with a list of Log Collection Points (Note that both Compute Servers as well as LSMSs are valid Log Collection Points) with a preference order and will send logs to the Log Collector it is currently homed to. Thus a LSMS Cluster is able to manage a larger number of bricks as well accept a larger amount of Logs from the bricks.

Most of the LSMS tools are also available on the Compute Servers. 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 Compute Server.

□

Configuration/Change Management

Overview

Each object modification performed by an administrator is logged by the LSMS 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 LSMS 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 LSMS 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 LSMS via the LSMS Command Line mechanism (see below).

□

Reporting System

Overview

The LSMS 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 LSMS 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, as well as administrator events and configuration changes.

The LSMS 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.

Additionally, the LSMS comes with a tool to allow fully-automated integration into the WebTrends Firewall Reporting Suite using the WebTrends Enhanced Log Format (WELF). This tool, when used with WebTrends, can provide a web-based customer-specific reporting system, automated with no human intervention.

□

Alarm System

Overview

The LSMS generates alarms based on Brick device log messages, as well as locally generated log messages from the various LSMS 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 LSMS 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:

- LSMS Error
- Brick Error
- Brick Lost/Found
- Brick Interface Up/Down
- Proactive Monitoring Threshold Crossing
- Brick Redundancy Alarms
- LSMS Redundancy Alarms
- ISS RealSecure Alarms
- QoS Bandwidth Alarms
- Alarm Code
- VPN Proactive Monitoring
- User Authentication
- Unauthorized LSMS login attempt
- LAN-LAN tunnel lost/up
- SLA Alarms
- Brick ICM Alarm

Actions

Actions allow each trigger to cause some response to be taken by the LSMS. 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 LSMS Remote Navigator)
- Email
- Out-of-band modem-dialed alphanumeric message sent to pager (via the TAP protocol)
- SNMP Trap (V1 of V2c)
- SYSLOG Message (with configurable SYSLOG level)

□

Real-Time Display (Status, Graphs, Logs)

Overview

The LSMS 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 LSMS Status Viewer

All logged-in administrators may be viewed in real-time, along with their connection statistics. The connection status of each LSMS 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
- Overall sessions by Pass/Drop/Proxy/VPN
- Overall sessions by IP protocol
- Overall packet counts in/out

Real-time device status can be synchronized across redundant LSMSs or multi-site LSMSs. 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 LSMS Remote Navigator. It has the ability to display log records in real-time, as received from all *Bricks*® 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.

LSMS Messenger

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

LSMS Process Status Viewer

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

□

SNMP Agent

Overview

The LSMS provides an SNMP agent to use for accessing limited configuration and statistic information regarding the system and associated Bricks in a Read-Only fashion. Absolutely NO information may be configured via SNMP. The MIB is available in SNMP v2c format.

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

Note that although the LSMS provides information on behalf of managed Bricks, the SNMP agent is NOT a proxy agent in the strict sense. Requests to the LSMS SNMP agent is serviced by information local to the LSMS, and will not result in a query to any Brick. Bricks do NOT respond to SNMP or any variation thereof.

Redundancy and Availability

Overview

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

Since both the Primary and Secondary LSMS servers are simultaneously active, each Brick device can be configured from either LSMS. Each Brick maintains a list of LSMS/compute servers, and can be configured to prefer one LSMS/compute server over the other, if available. Each Brick device sends log files to its currently-active LSMS/compute server. A Brick device can be manually ″rehomed″ to the other server, if available.

The LSMS 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 LSMS Command Line feature is designed to allow administrators the ability to configure many LSMS components and policy objects by using a text file-based interface. This command-line is available local to the LSMS host only; remote access to the host is the responsibility of the administrator. This feature is designed to be easily scriptable from an external application running on the LSMS host.

Configuration Assistant

Overview

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

□

Brick Device Remote Console

Overview

The LSMS 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. This console is both secure from the user's workstation to the LSMS, as well as from the LSMS to the Brick. No policy modifications may be made from this Remote Console (or any Brick console interface).

□

3 Lucent IPSec Client

Overview

Purpose

The 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 Lucent VPN Firewall system; some features require the use of an LSMS or Brick device to function as described herein.

Contents

Platforms and Compatibility

Overview

The IPSec Client can be installed on the following Windows platforms:

- *Windows*™ 95
- *Windows*™ 98
- *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 Lucent VPN Firewall team directly for this information.

Although the IPSec Client strongly integrates with the Lucent VPN Firewall, it also interoperates with the Lucent Access Point.

□

Supported Standards

Overview

The Lucent IPSec Client supports the following security standards

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

Personal Firewall

Overview

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

Overview

The Lucent 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 Lucent proprietary and not designed to interoperate with other non-Lucent products.

Local Presence

Overview

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 LSMS, or one-at-a-time using the RADIUS parameter download feature.

Split Tunnels

Overview

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 LSMS, and can be configured to disallow clear-text traffic entirely if so desired.

□

Entrust Integration

Overview

The IPSec client has the ability to perform strong host authentication using X.509 certificates as well as pre-shared keys. When used with the Entrust PKI, the IPSec client can retrieve an X.509 certificate using the LDAP protocol stored in an Entrust X.500 directory server. The client can then use this certificate for basic authentication.

□

Strong Authentication

Overview

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

Overview

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

Overview

Upon tunnel establishment, the Lucent 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

Overview

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

Overview

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

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

□

Pleasant Push Software Upgrade

Overview

If a new version of the IPSec client is available on the LSMS, 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

Overview

The 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

Overview

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

Client Log

Overview

The IPSec client maintains logs of connection attempts, including detailed IKE and IPSec negotiation, to aid troubleshooting.

Windows Tray Icon

Overview

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.