Network Working Group
Request for Comments: 3871
Category: Informational

G. Jones, Ed.
The MITRE Corporation

September 2004

Operational Security Requirements for Large Internet Service Provider (ISP) IP Network Infrastructure

Status of this Memo

Copyright Notice

Abstract

This document defines a list of operational security requirements for the infrastructure of large Internet Service Provider (ISP) IP networks (routers and switches). A framework is defined for specifying "profiles", which are collections of requirements applicable to certain network topology contexts (all, core-only, edge-only...). The goal is to provide network operators a clear, concise way of communicating their security requirements to vendors.

Table of Contents

1. Introduction

1.1. Goals

This document defines a list of operational security requirements for the infrastructure of large IP networks (routers and switches). The goal is to provide network operators a clear, concise way of communicating their security requirements to equipment vendors.

1.2. Motivation

Network operators need tools to ensure that they are able to manage their networks securely and to insure that they maintain the ability to provide service to their customers. Some of the threats are outlined in section 3.2 of [RFC2196]. This document enumerates features which are required to implement many of the policies and procedures suggested by [RFC2196] in the context of the infrastructure of large IP-based networks. Also see [RFC3013].

1.3. Scope

The scope of these requirements is intended to cover the managed infrastructure of large ISP IP networks (e.g., routers and switches). Certain groups (or "profiles", see below) apply only in specific situations (e.g., edge-only).

The following are explicitly out of scope:

- general purpose hosts that do not transit traffic including infrastructure hosts such as name/time/log/AAA servers, etc.,
- unmanaged devices,
- customer managed devices (e.g., firewalls, Intrusion Detection System, dedicated VPN devices, etc.),
- SOHO (Small Office, Home Office) devices (e.g., personal firewalls, Wireless Access Points, Cable Modems, etc.),
- confidentiality of customer data,
- integrity of customer data,
- physical security.

This means that while the requirements in the minimum profile (and others) may apply, additional requirements have not be added to account for their unique needs.

While the examples given are written with IPv4 in mind, most of the requirements are general enough to apply to IPv6.

1.4. Definition of a Secure Network

For the purposes of this document, a secure network is one in which:

- The network keeps passing legitimate customer traffic (availability).
- Traffic goes where it is supposed to go, and only where it is supposed to go (availability, confidentiality).
- The network elements remain manageable (availability).
- Only authorized users can manage network elements (authorization).
- There is a record of all security related events (accountability).
- The network operator has the necessary tools to detect and respond to illegitimate traffic.

1.5. Intended Audience

There are two intended audiences: the network operator who selects, purchases, and operates IP network equipment, and the vendors who create them.

1.6. Format

The individual requirements are listed in the three sections below.

- Section 2 lists functional requirements.
- Section 3 lists documentation requirements.
- Section 4 lists assurance requirements.

Within these areas, requirements are grouped in major functional areas (e.g., logging, authentication, filtering, etc.)

Each requirement has the following subsections:

- Requirement (what)
- Justification (why)
- Examples (how)

- Warnings (if applicable)

The requirement describes a policy to be supported by the device. The justification tells why and in what context the requirement is important. The examples section is intended to give examples of implementations that may meet the requirement. Examples cite technology and standards current at the time of this writing. See [RFC3631]. It is expected that the choice of implementations to meet the requirements will change over time. The warnings list operational concerns, deviation from standards, caveats, etc.

Security requirements will vary across different device types and different organizations, depending on policy and other factors. A desired feature in one environment may be a requirement in another. Classifications must be made according to local need.

In order to assist in classification, Appendix A defines several requirement "profiles" for different types of devices. Profiles are concise lists of requirements that apply to certain classes of devices. The profiles in this document should be reviewed to determine if they are appropriate to the local environment.

1.7. Intended Use

It is anticipated that the requirements in this document will be used for the following purposes:

- as a checklist when evaluating networked products,
- to create profiles of different subsets of the requirements which describe the needs of different devices, organizations, and operating environments,
- to assist operators in clearly communicating their security requirements,
- as high level guidance for the creation of detailed test plans.

1.8. Definitions

RFC 2119 Keywords

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The use of the RFC 2119 keywords is an attempt, by the editor, to assign the correct requirement levels ("MUST", "SHOULD", "MAY"...). It must be noted that different organizations, operational environments, policies and legal environments will generate different requirement levels. Operators and vendors should carefully consider the individual requirements listed here in their own context. One size does not fit all.

Bogon.

A "Bogon" (plural: "bogons") is a packet with an IP source address in an address block not yet allocated by IANA or the Regional Internet Registries (ARIN, RIPE, APNIC...) as well as all addresses reserved for private or special use by RFCs. See [RFC3330] and [RFC1918].

CLI.

Several requirements refer to a Command Line Interface (CLI). While this refers at present to a classic text oriented command interface, it is not intended to preclude other mechanisms which may meet all the requirements that reference "CLI".

Console.

Several requirements refer to a "Console". The model for this is the classic RS232 serial port which has, for the past 30 or more years, provided a simple, stable, reliable, well-understood and nearly ubiquitous management interface to network devices. Again, these requirements are intended primarily to codify the benefits provided by that venerable interface, not to preclude other mechanisms that meet all the same requirements.

Filter.

In this document, a "filter" is defined as a group of one or more rules where each rule specifies one or more match criteria as specified in Section 2.8.

In-Band management.

"In-Band management" is defined as any management done over the same channels and interfaces used for user/customer data. Examples would include using SSH for management via customer or Internet facing network interfaces.

High Resolution Time.

"High resolution time" is defined in this document as "time having a resolution greater than one second" (e.g., milliseconds).

IP.

Unless otherwise indicated, "IP" refers to IPv4.

Management.

This document uses a broad definition of the term "management". In this document, "management" refers to any authorized interaction with the device intended to change its operational state or configuration. Data/Forwarding plane functions (e.g., the transit of customer traffic) are not considered management. Control plane functions such as routing, signaling and link management protocols and management plane functions such as remote access, configuration and authentication are considered to be management.

Martian.

Per [RFC1208] "Martian: Humorous term applied to packets that turn up unexpectedly on the wrong network because of bogus routing entries. Also used as a name for a packet which has an altogether bogus (non-registered or ill-formed) Internet address." For the purposes of this document Martians are defined as "packets having a source address that, by application of the current forwarding tables, would not have its return traffic routed back to the sender." "Spoofed packets" are a common source of martians.

Note that in some cases, the traffic may be asymmetric, and a simple forwarding table check might produce false positives. See [RFC3704]

Out-of-Band (OoB) management.

"Out-of-Band management" is defined as any management done over channels and interfaces that are separate from those used for user/customer data. Examples would include a serial console interface or a network interface connected to a dedicated management network that is not used to carry customer traffic.

Open Review.

"Open review" refers to processes designed to generate public discussion and review of proposed technical solutions such as data communications protocols and cryptographic algorithms with the goals of improving and building confidence in the final solutions.

For the purposes of this document "open review" is defined by [RFC2026]. All standards track documents are considered to have been through an open review process.

It should be noted that organizations may have local requirements that define what they view as acceptable "open review". For example, they may be required to adhere to certain national or international standards. Such modifications of the definition of the term "open review", while important, are considered local issues that should be discussed between the organization and the vendor.

It should also be noted that section 7 of [RFC2026] permits standards track documents to incorporate other "external standards and specifications".

Service.

A number of requirements refer to "services". For the purposes of this document a "service" is defined as "any process or protocol running in the control or management planes to which non-transit packets may be delivered". Examples might include an SSH server, a BGP process or an NTP server. It would also include the transport, network and link layer protocols since, for example, a TCP packet addressed to a port on which no service is listening will be "delivered" to the IP stack, and possibly result in an ICMP message being sent back.

Secure Channel.

A "secure channel" is a mechanism that ensures end-to-end integrity and confidentiality of communications. Examples include TLS [RFC2246] and IPsec [RFC2401]. Connecting a terminal to a console port using physically secure, shielded cable would provide confidentiality but possibly not integrity.

Single-Homed Network.

A "single-homed network" is defined as one for which

* There is only one upstream connection

* Routing is symmetric.

See [RFC3704] for a discussion of related issues and mechanisms for multihomed networks.

Spoofed Packet.

A "spoofed packet" is defined as a packet that has a source address that does not correspond to any address assigned to the system which sent the packet. Spoofed packets are often "bogons" or "martians".

2. Functional Requirements

The requirements in this section are intended to list testable, functional requirements that are needed to operate devices securely.

2.1. Device Management Requirements

2.1.1. Support Secure Channels For Management

Requirement.

The device MUST provide mechanisms to ensure end-to-end integrity and confidentiality for all network traffic and protocols used to support management functions. This MUST include at least protocols used for configuration, monitoring, configuration backup and restore, logging, time synchronization, authentication, and routing.

Justification.

Integrity protection is required to ensure that unauthorized users cannot manage the device or alter log data or the results of management commands. Confidentiality is required so that unauthorized users cannot view sensitive information, such as keys, passwords, or the identity of users.

Examples.

See [RFC3631] for a current list of mechanisms that can be used to support secure management.

Later sections list requirements for supporting in-band management (Section 2.2) and out-of-band management (Section 2.3) as well as trade-offs that must be weighed in considering which is appropriate to a given situation.

Warnings.

None.

2.2. In-Band Management Requirements

This section lists security requirements that support secure in-band management. In-band management has the advantage of lower cost (no extra interfaces or lines), but has significant security disadvantages:

- o Saturation of customer lines or interfaces can make the device unmanageable unless out-of-band management resources have been reserved.
- o Since public interfaces/channels are used, it is possible for attackers to directly address and reach the device and to attempt management functions.
- o In-band management traffic on public interfaces may be intercepted, however this would typically require a significant compromise in the routing system.
- o Public interfaces used for in-band management may become unavailable due to bugs (e.g., buffer overflows being exploited) while out-of-band interfaces (such as a serial console device) remain available.

There are many situations where in-band management makes sense, is used, and/or is the only option. The following requirements are meant to provide means of securing in-band management traffic.

2.2.1. Use Cryptographic Algorithms Subject To Open Review

Requirement.

If cryptography is used to provide secure management functions, then there MUST be an option to use algorithms that are subject to "open review" as defined in Section 1.8 to provide these functions. These SHOULD be used by default. The device MAY optionally support algorithms that are not open to review.

Justification.

Cryptographic algorithms that have not been subjected to widespread, extended public/peer review are more likely to have undiscovered weaknesses or flaws than open standards and publicly reviewed algorithms. Network operators may have need or desire to

use non-open cryptographic algorithms. They should be allowed to evaluate the trade-offs and make an informed choice between open and non-open cryptography. See [Schneier] for further discussion.

Examples.

The following are some algorithms that satisfy the requirement at the time of writing: AES [FIPS.197], and 3DES [ANSI.X9-52.1998] for applications requiring symmetric encryption; RSA [RFC3447] and Diffie-Hellman [PKCS.3.1993], [RFC2631] for applications requiring key exchange; HMAC [RFC2401] with SHA-1 [RFC3174] for applications requiring message verification.

Warnings.

This list is not exhaustive. Other strong, well-reviewed algorithms may meet the requirement. The dynamic nature of the field means that what is good enough today may not be in the future.

Open review is necessary but not sufficient. The strength of the algorithm and key length must also be considered. For example, 56-bit DES meets the open review requirement, but is today considered too weak and is therefore not recommended.

2.2.2. Use Strong Cryptography

Requirement.

If cryptography is used to meet the secure management channel requirements, then the key lengths and algorithms SHOULD be "strong".

Justification.

Short keys and weak algorithms threaten the confidentiality and integrity of communications.

Examples.

The following algorithms satisfy the requirement at the time of writing: AES [FIPS.197], and 3DES [ANSI.X9-52.1998] for applications requiring symmetric encryption; RSA [RFC3447] and Diffie-Hellman [PKCS.3.1993], [RFC2631] for applications requiring key exchange; HMAC [RFC2401] with SHA-1 [RFC3174] for applications requiring message verification.

Note that for *new protocols* [RFC3631] says the following: "Simple keyed hashes based on MD5 [RFC1321], such as that used in the BGP session security mechanism [RFC2385], are especially to be avoided in new protocols, given the hints of weakness in MD5." While use of such hashes in deployed products and protocols is preferable to a complete lack of integrity and authentication checks, this document concurs with the recommendation that new products and protocols strongly consider alternatives.

Warnings.

This list is not exhaustive. Other strong, well-reviewed algorithms may meet the requirement. The dynamic nature of the field means that what is good enough today may not be in the future.

Strength is relative. Long keys and strong algorithms are intended to increase the work factor required to compromise the security of the data protected. Over time, as processing power increases, the security provided by a given algorithm and key length will degrade. The definition of "Strong" must be constantly reevaluated.

There may be legal issues governing the use of cryptography and the strength of cryptography used.

This document explicitly does not attempt to make any authoritative statement about what key lengths constitute "strong" cryptography. See [RFC3562] and [RFC3766] for help in determining appropriate key lengths. Also see [Schneier] chapter 7 for a discussion of key lengths.

2.2.3. Use Protocols Subject To Open Review For Management

Requirement.

If cryptography is used to provide secure management channels, then its use MUST be supported in protocols that are subject to "open review" as defined in Section 1.8. These SHOULD be used by default. The device MAY optionally support the use of cryptography in protocols that are not open to review.

Justification.

Protocols that have not been subjected to widespread, extended public/peer review are more likely to have undiscovered weaknesses or flaws than open standards and publicly reviewed protocols Network operators may have need or desire to use non-open protocols They should be allowed to evaluate the trade-offs and make an informed choice between open and non-open protocols.

Examples.

See TLS [RFC2246] and IPsec [RFC2401].

Warnings.

Note that open review is necessary but may not be sufficient. It is perfectly possible for an openly reviewed protocol to misuse (or not use) cryptography.

2.2.4. Allow Selection of Cryptographic Parameters

Requirement.

The device SHOULD allow the operator to select cryptographic parameters. This SHOULD include key lengths and algorithms.

Justification.

Cryptography using certain algorithms and key lengths may be considered "strong" at one point in time, but "weak" at another. The constant increase in compute power continually reduces the time needed to break cryptography of a certain strength. Weaknesses may be discovered in algorithms. The ability to select a different algorithm is a useful tool for maintaining security in the face of such discoveries.

Examples.

56-bit DES was once considered secure. In 1998 it was cracked by custom built machine in under 3 days. The ability to select algorithms and key lengths would give the operator options (different algorithms, longer keys) in the face of such developments.

Warnings.

None.

2.2.5. Management Functions Should Have Increased Priority

Requirement.

Management functions SHOULD be processed at higher priority than non-management traffic. This SHOULD include ingress, egress, internal transmission, and processing. This SHOULD include at least protocols used for configuration, monitoring, configuration backup, logging, time synchronization, authentication, and routing.

Justification.

Certain attacks (and normal operation) can cause resource saturation such as link congestion, memory exhaustion or CPU overload. In these cases it is important that management functions be prioritized to ensure that operators have the tools needed to recover from the attack.

Examples.

Imagine a service provider with 1,000,000 DSL subscribers, most of whom have no firewall protection. Imagine that a large portion of these subscribers machines were infected with a new worm that enabled them to be used in coordinated fashion as part of large denial of service attack that involved flooding. It is entirely possible that without prioritization such an attack would cause link congestion resulting in routing adjacencies being lost. A DoS attack against hosts has just become a DoS attack against the network.

Warnings.

Prioritization is not a panacea. Routing update packets may not make it across a saturated link. This requirement simply says that the device should prioritize management functions within its scope of control (e.g., ingress, egress, internal transit, processing). To the extent that this is done across an entire network, the overall effect will be to ensure that the network remains manageable.

2.3. Out-of-Band (OoB) Management Requirements

See Section 2.2 for a discussion of the advantages and disadvantages of In-band vs. Out-of-Band management.

These requirements assume two different possible Out-of-Band topologies:

- serial line (or equivalent) console connections using a CLI,
- network interfaces connected to a separate network dedicated to management.

The following assumptions are made about out-of-band management:

- The out-of-band management network is secure.
- Communications beyond the management interface (e.g., console port, management network interface) is secure.
- There is no need for encryption of communication on out-of-band management interfaces, (e.g., on a serial connection between a terminal server and a device's console port).
- Security measures are in place to prevent unauthorized physical access.

Even if these assumptions hold it would be wise, as an application of defense-in-depth, to apply the in-band requirements (e.g., encryption) to out-of-band interfaces.

2.3.1. Support a 'Console' Interface

Requirement.

The device MUST support complete configuration and management via a 'console' interface that functions independently from the forwarding and IP control planes.

Justification.

There are times when it is operationally necessary to be able to immediately and easily access a device for management or configuration, even when the network is unavailable, routing and network interfaces are incorrectly configured, the IP stack and/or operating system may not be working (or may be vulnerable to recently discovered exploits that make their use impossible/ inadvisable), or when high bandwidth paths to the device are unavailable. In such situations, a console interface can provide a way to manage and configure the device.

Examples.

An RS232 (EIA232) interface that provides the capability to load new versions of the system software and to perform configuration via a command line interface. RS232 interfaces are ubiquitous and well understood.

A simple embedded device that provides management and configuration access via an Ethernet or USB interface.

As of this writing, RS232 is still strongly recommended as it provides the following benefits:

* Simplicity. RS232 is far simpler than the alternatives. It is simply a hardware specification. By contrast an Ethernet based solution might require an ethernet interface, an operating system, an IP stack and an HTTP server all to be functioning and properly configured.

* Proven. RS232 has more than 30 years of use.

* Well-Understood. Operators have a great deal of experience with RS232.

* Availability. It works even in the presence of network failure.

* Ubiquity. It is very widely deployed in mid to high end network infrastructure.

* Low-Cost. The cost of adding a RS232 port to a device is small.

* CLI-Friendly. An RS232 interface and a CLI are sufficient in most cases to manage a device. No additional software is required.

* Integrated. Operators have many solutions (terminal servers, etc.) currently deployed to support management via RS232.

While other interfaces may be supplied, the properties listed above should be considered. Interfaces not having these properties may present challenges in terms of ease of use, integration or adoption. Problems in any of these areas could have negative security impacts, particularly in situations where the console must be used to quickly respond to incidents.

Warnings.

It is common practice is to connect RS232 ports to terminal servers that permit networked access for convenience. This increases the potential security exposure of mechanisms available only via RS232 ports. For example, a password recovery mechanism that is available only via RS232 might give a remote hacker to completely reconfigure a router. While operational procedures are beyond the scope of this document, it is important to note here that strong attention should be given to policies, procedures, access mechanisms and physical security governing access to console ports.

2.3.2. 'Console' Communication Profile Must Support Reset

Requirement.

There MUST be a method defined and published for returning the console communication parameters to their default settings. This method must not require the current settings to be known.

Justification.

Having to guess at communications settings can waste time. In a crisis situation, the operator may need to get on the console of a device quickly.

Examples.

One method might be to send a break on a serial line.

Warnings.

None.

2.3.3. 'Console' Requires Minimal Functionality of Attached Devices

Requirement.

The use of the 'console' interface MUST NOT require proprietary devices, protocol extensions or specific client software.

Justification.

The purpose of having the console interface is to have a management interface that can be made to work quickly at all times. Requiring complex or nonstandard behavior on the part of attached devices reduces the likelihood that the console will work without hassles.

Examples.

If the console is supplied via an RS232 interface, then it should function with an attached device that only implements a "dumb" terminal. Support of "advanced" terminal features/types should be optional.

Warnings.

None.

2.3.4. 'Console' Supports Fall-back Authentication

Requirement.

The 'console' SHOULD support an authentication mechanism which does not require functional IP or depend on external services. This authentication mechanism MAY be disabled until a failure of other preferred mechanisms is detected.

Justification.

It does little good to have a console interface on a device if you cannot get into the device with it when the network is not working.

Examples.

Some devices which use TACACS or RADIUS for authentication will fall back to a local account if the TACACS or RADIUS server does not reply to an authentication request.

Warnings.

This requirement represents a trade-off between being able to manage the device (functionality) and security. There are many ways to implement this which would provide reduced security for the device, (e.g., a back door for unauthorized access). Local policy should be consulted to determine if "fail open" or "fail

closed" is the correct stance. The implications of "fail closed" (e.g., not being able to manage a device) should be fully considered.

If the fall-back mechanism is disabled, it is important that the failure of IP based authentication mechanism be reliably detected and the fall-back mechanism automatically enabled...otherwise the operator may be left with no means to authenticate.

2.3.5. Support Separate Management Plane IP Interfaces

Requirement.

The device MAY provide designated network interface(s) that are used for management plane traffic.

Justification.

A separate management plane interface allows management traffic to be segregated from other traffic (data/forwarding plane, control plane). This reduces the risk that unauthorized individuals will be able to observe management traffic and/or compromise the device.

This requirement applies in situations where a separate OoB management network exists.

Examples.

Ethernet port dedicated to management and isolated from customer traffic satisfies this requirement.

Warnings.

The use of this type of interface depends on proper functioning of both the operating system and the IP stack, as well as good, known configuration at least on the portions of the device dedicated to management.

2.3.6. No Forwarding Between Management Plane And Other Interfaces

Requirement.

If the device implements separate network interface(s) for the management plane per Section 2.3.5 then the device MUST NOT forward traffic between the management plane and non-management plane interfaces.

Justification.

This prevents the flow, intentional or unintentional, of management traffic to/from places that it should not be originating/terminating (e.g., anything beyond the customer-facing interfaces).

Examples.

Implementing separate forwarding tables for management plane and non-management plane interfaces that do not propagate routes to each other satisfies this requirement.

Warnings.

None.

2.4. Configuration and Management Interface Requirements

This section lists requirements that support secure device configuration and management methods. In most cases, this currently involves some sort of command line interface (CLI) and configuration files. It may be possible to meet these requirements with other mechanisms, for instance SNMP or a script-able HTML interface that provides full access to management and configuration functions. In the future, there may be others (e.g., XML based configuration).

2.4.1. 'CLI' Provides Access to All Configuration and Management Functions

Requirement.

The Command Line Interface (CLI) or equivalent MUST allow complete access to all configuration and management functions. The CLI MUST be supported on the console (see Section 2.3.1) and SHOULD be supported on all other interfaces used for management.

Justification.

The CLI (or equivalent) is needed to provide the ability to do reliable, fast, direct, local management and monitoring of a device. It is particularly useful in situations where it is not possible to manage and monitor the device in-band via "normal" means (e.g., SSH or SNMP [RFC3410], [RFC3411]) that depend on functional networking. Such situations often occur during security incidents such as bandwidth-based denial of service attacks.

Examples.

Examples of configuration include setting interface addresses, defining and applying filters, configuring logging and authentication, etc. Examples of management functions include displaying dynamic state information such as CPU load, memory utilization, packet processing statistics, etc.

Warnings.

None.

2.4.2. 'CLI' Supports Scripting of Configuration

Requirement.

The CLI or equivalent MUST support external scripting of configuration functions. This CLI SHOULD support the same command set and syntax as that in Section 2.4.1.

Justification.

During the handling of security incidents, it is often necessary to quickly make configuration changes on large numbers of devices. Doing so manually is error prone and slow. Vendor supplied management solutions do not always foresee or address the type or scale of solutions that are required. The ability to script provides a solution to these problems.

Examples.

Example uses of scripting include: tracking an attack across a large network, updating authentication parameters, updating logging parameters, updating filters, configuration fetching/ auditing, etc. Some languages that are currently used for scripting include expect, Perl and TCL.

Warnings.

Some properties of the command language that enhance the ability to script are: simplicity, regularity and consistency. Some implementations that would make scripting difficult or impossible include: "text menu" style interfaces (e.g., "curses" on UNIX) or a hard-coded GUI interfaces (e.g., a native Windows or Macintosh GUI application) that communicate using a proprietary or undocumented protocol not based on a CLI.

2.4.3. 'CLI' Supports Management Over 'Slow' Links

Requirement.

The device MUST support a command line interface (CLI) or equivalent mechanism that works over low bandwidth connections.

Justification.

There are situations where high bandwidth for management is not available, for example when in-band connections are overloaded during an attack or when low-bandwidth, out-of-band connections such as modems must be used. It is often under these conditions that it is most crucial to be able to perform management and configuration functions.

Examples.

The network is down. The network engineer just disabled routing by mistake on the sole gateway router in a remote unmanned data center. The only access to the device is over a modem connected to a console port. The data center customers are starting to call the support line. The GUI management interface is redrawing the screen multiple times...slowly... at 9600bps.

One mechanism that supports operation over slow links is the ability to apply filters to the output of CLI commands which have potentially large output. This may be implemented with something similar to the UNIX pipe facility and "grep" command.

For example,

```
cat largefile.txt | grep interesting-string
```

Another is the ability to "page" through large command output, e.g., the UNIX "more" command:

For example,

```
cat largefile.txt | more
```

Warnings.

One consequence of this requirement may be that requiring a GUI interface for management is unacceptable unless it can be shown to work acceptably over slow links.

2.4.4. 'CLI' Supports Idle Session Timeout

Requirement.

The command line interface (CLI) or equivalent mechanism MUST support a configurable idle timeout value.

Justification.

Network administrators go to lunch. They leave themselves logged in with administrative privileges. They forget to use screen-savers with password protection. They do this while at conferences and in other public places. This behavior presents opportunity for unauthorized access. Idle timeouts reduce the window of exposure.

Examples.

The CLI may provide a configuration command that allows an idle timeout to be set. If the operator does not enter commands for that amount of time, the login session will be automatically terminated.

Warnings.

None.

2.4.5. Support Software Installation

Requirement.

The device MUST provide a means to install new software versions. It MUST be possible to install new software while the device is disconnected from all public IP networks. This MUST NOT rely on previous installation and/or configuration. While new software MAY be loaded from writable media (disk, flash, etc.), the capability to load new software MUST depend only on non-writable media (ROM, etc.). The installation procedures SHOULD support mechanisms to ensure reliability and integrity of data transfers.

Justification.

* Vulnerabilities are often discovered in the base software (operating systems, etc.) shipped by vendors. Often mitigation of the risk presented by these vulnerabilities can only be accomplished by updates to the vendor supplied software (e.g., bug

fixes, new versions of code, etc.). Without a mechanism to load new vendor supplied code, it may not be possible to mitigate the risk posed by these vulnerabilities.

* It is also conceivable that malicious behavior on the part of hackers or unintentional behaviors on the part of operators could cause software on devices to be corrupted or erased. In these situations, it is necessary to have a means to (re)load software onto the device to restore correct functioning.

* It is important to be able to load new software while disconnected from all public IP networks because the device may be vulnerable to old attacks before the update is complete.

* One has to assume that hackers, operators, etc. may erase or corrupt all writable media (disks, flash, etc.). In such situations, it is necessary to be able to recover starting with only non-writable media (e.g., CD-ROM, a true ROM-based monitor).

* System images may be corrupted in transit (from vendor to customer, or during the loading process) or in storage (bit rot, defective media, etc.). Failure to reliably load a new image, for example after a hacker deletes or corrupts the installed image, could result in extended loss of availability.

Examples.

The device could support booting into a simple ROM-based monitor that supported a set of commands sufficient to load new operating system code and configuration data from other devices. The operating system and configuration might be loaded from:

RS232. The device could support uploading new code via an RS232 console port.

CD-ROM. The device could support installing new code from a locally attached CD-ROM drive.

NETWORK. The device could support installing new code via a network interface, assuming that (a) it is disconnected from all public networks and (b) the device can boot an OS and IP stack from some read-only media with sufficient capabilities to load new code from the network.

FLASH. The device could support booting from flash memory cards.

Simple mechanisms currently in use to protect the integrity of system images and data transfer include image checksums and simple serial file transfer protocols such as XMODEM and Kermit.

Warnings.

None.

2.4.6. Support Remote Configuration Backup

Requirement.

The device MUST provide a means to store the system configuration to a remote server. The stored configuration MUST have sufficient information to restore the device to its operational state at the time the configuration is saved. Stored versions of the configuration MAY be compressed using an algorithm which is subject to open review, as long as the fact is clearly identified and the compression can be disabled. Sensitive information such as passwords that could be used to compromise the security of the device MAY be excluded from the saved configuration.

Justification.

Archived configurations are essential to enable auditing and recovery.

Examples.

Possible implementations include SCP, SFTP or FTP over a secure channel. See Section 2.1.1 for requirements related to secure communication channels for management protocols and data.

Warnings.

The security of the remote server is assumed, with appropriate measures being outside the scope of this document.

2.4.7. Support Remote Configuration Restore

Requirement.

The device MUST provide a means to restore a configuration that was saved as described in Section 2.4.6. The system MUST be restored to its operational state at the time the configuration was saved.

Justification.

Restoration of archived configurations allows quick restoration of service following an outage (security related as well as from other causes).

Examples.

Configurations may be restored using SCP, SFTP or FTP over a secure channel. See Section 2.1.1 for requirements related to secure communication channels for management protocols and data.

Warnings.

The security of the remote server is assumed, with appropriate measures being outside the scope of this document.

Note that if passwords or other sensitive information are excluded from the saved copy of the configuration, as allowed by Section 2.4.6, then the restore may not be complete. The operator may have to set new passwords or supply other information that was not saved.

2.4.8. Support Text Configuration Files

Requirement.

The device MUST support display, backup and restore of system configuration in a simple well defined textual format. The configuration MUST also be viewable as text on the device itself. It MUST NOT be necessary to use a proprietary program to view the configuration.

Justification.

Simple, well-defined textual configurations facilitate human understanding of the operational state of the device, enable off-line audits, and facilitate automation. Requiring the use of a proprietary program to access the configuration inhibits these goals.

Examples.

A 7-bit ASCII configuration file that shows the current settings of the various configuration options would satisfy the requirement, as would a Unicode configuration or any other "textual" representation. A structured binary format intended only for consumption by programs would not be acceptable.

Warnings.

Offline copies of configurations should be well protected as they often contain sensitive information such as SNMP community strings, passwords, network blocks, customer information, etc.

"Well defined" and "textual" are open to interpretation. Clearly an ASCII configuration file with a regular, documented command oriented-syntax would meet the definition. These are currently in wide use. Future options, such as XML based configuration may meet the requirement. Determining this will require evaluation against the justifications listed above.

2.5. IP Stack Requirements

2.5.1. Ability to Identify All Listening Services

Requirement.

The vendor MUST:

* Provide a means to display all services that are listening for network traffic directed at the device from any external source.

* Display the addresses to which each service is bound.

* Display the addresses assigned to each interface.

* Display any and all port(s) on which the service is listing.

* Include both open standard and vendor proprietary services.

Justification.

This information is necessary to enable a thorough assessment of the security risks associated with the operation of the device (e.g., "does this protocol allow complete management of the device without also requiring authentication, authorization, or accounting?"). The information also assists in determining what steps should be taken to mitigate risk (e.g., "should I turn this service off ?")

Examples.

If the device is listening for SNMP traffic from any source directed to the IP addresses of any of its local interfaces, then this requirement could be met by the provision of a command which displays that fact.

Warnings.

None.

2.5.2. Ability to Disable Any and All Services

Requirement.

The device MUST provide a means to turn off any "services" (see Section 1.8).

Justification.

The ability to disable services for which there is no operational need will allow administrators to reduce the overall risk posed to the device.

Examples.

Processes that listen on TCP and UDP ports would be prime examples of services that it must be possible to disable.

Warnings.

None.

2.5.3. Ability to Control Service Bindings for Listening Services

Requirement.

The device MUST provide a means for the user to specify the bindings used for all listening services. It MUST support binding to any address or net-block associated with any interface local to the device. This must include addresses bound to physical or non-physical (e.g., loopback) interfaces.

Justification.

It is a common practice among operators to configure "loopback" pseudo-interfaces to use as the source and destination of management traffic. These are preferred to physical interfaces

because they provide a stable, routable address. Services bound to the addresses of physical interface addresses might become unreachable if the associated hardware goes down, is removed, etc.

This requirement makes it possible to restrict access to management services using routing. Management services may be bound only to the addresses of loopback interfaces. The loopback interfaces may be addressed out of net-blocks that are only routed between the managed devices and the authorized management networks/hosts. This has the effect of making it impossible for anyone to connect to (or attempt to DoS) management services from anywhere but the authorized management networks/hosts.

It also greatly reduces the need for complex filters. It reduces the number of ports listening, and thus the number of potential avenues of attack. It ensures that only traffic arriving from legitimate addresses and/or on designated interfaces can access services on the device.

Examples.

If the device listens for inbound SSH connections, this requirement means that it should be possible to specify that the device will only listen to connections destined to specific addresses (e.g., the address of the loopback interface) or received on certain interfaces (e.g., an Ethernet interface designated as the "management" interface). It should be possible in this example to configure the device such that the SSH is NOT listening to every address configured on the device. Similar effects may be achieved with the use of global filters, sometimes called "receive" or "loopback" ACLs, that filter traffic destined for the device itself on all interfaces.

Warnings.

None.

2.5.4. Ability to Control Service Source Addresses

Requirement.

The device MUST provide a means that allows the user to specify the source addresses used for all outbound connections or transmissions originating from the device. It SHOULD be possible to specify source addresses independently for each type of outbound connection or transmission. Source addresses MUST be limited to addresses that are assigned to interfaces (including loopbacks) local to the device.

Justification.

This allows remote devices receiving connections or transmissions to use source filtering as one means of authentication. For example, if SNMP traps were configured to use a known loopback address as their source, the SNMP workstation receiving the traps (or a firewall in front of it) could be configured to receive SNMP packets only from that address.

Examples.

The operator may allocate a distinct block of addresses from which all loopbacks are numbered. NTP and syslog can be configured to use those loopback addresses as source, while SNMP and BGP may be configured to use specific physical interface addresses. This would facilitate filtering based on source address as one way of rejecting unauthorized attempts to connect to peers/servers.

Warnings.

Care should be taken to assure that the addresses chosen are routable between the sending and receiving devices, (e.g., setting SSH to use a loopback address of 10.1.1.1 which is not routed between a router and all intended destinations could cause problems).

Note that some protocols, such as SCTP [RFC3309], can use more than one IP address as the endpoint of a single connection.

Also note that [RFC3631] lists address-based authentication as an "insecurity mechanism". Address based authentication should be replaced or augmented by other mechanisms wherever possible.

2.5.5. Support Automatic Anti-spoofing for Single-Homed Networks

Requirement.

The device MUST provide a means to designate particular interfaces as servicing "single-homed networks" (see Section 1.8) and MUST provide an option to automatically drop "spoofed packets" (Section 1.8) received on such interfaces where application of the current forwarding table would not route return traffic back through the same interface. This option MUST work in the presence of dynamic routing and dynamically assigned addresses.

Justification.

See sections 3 of [RFC1918], sections 5.3.7 and 5.3.8 of [RFC1812], and [RFC2827].

Examples.

This requirement could be satisfied in several ways. It could be satisfied by the provision of a single command that automatically generates and applies filters to an interface that implements anti-spoofing. It could be satisfied by the provision of a command that causes the return path for packets received to be checked against the current forwarding tables and dropped if they would not be forwarded back through the interface on which they were received.

See [RFC3704].

Warnings.

This requirement only holds for single-homed networks. Note that a simple forwarding table check is not sufficient in the more complex scenarios of multi-homed or multi-attached networks, i.e., where the traffic may be asymmetric. In these cases, a more extensive check such as Feasible Path RPF could be very useful.

2.5.6. Support Automatic Discarding Of Bogons and Martians

Requirement.

The device MUST provide a means to automatically drop all "bogons" (Section 1.8) and "martians" (Section 1.8). This option MUST work in the presence of dynamic routing and dynamically assigned addresses.

Justification.

These sorts of packets have little (no?) legitimate use and are used primarily to allow individuals and organization to avoid identification (and thus accountability) and appear to be most often used for DoS attacks, email abuse, hacking, etc. In addition, transiting these packets needlessly consumes resources and may lead to capacity and performance problems for customers.

See sections 3 of [RFC1918], sections 5.3.7 and 5.3.8 of [RFC1812], and [RFC2827].

Examples.

This requirement could be satisfied by the provision of a command that causes the return path for packets received to be checked against the current forwarding tables and dropped if no viable return path exists. This assumes that steps are taken to assure that no bogon entries are present in the forwarding tables (for example filtering routing updates per Section 2.7.5 to reject advertisements of unassigned addresses).

See [RFC3704].

Warnings.

This requirement only holds for single-homed networks. Note that a simple forwarding table check is not sufficient in the more complex scenarios of multi-homed or multi-attached networks, i.e., where the traffic may be asymmetric. In these cases, a more extensive check such as Feasible Path RPF could be very useful.

2.5.7. Support Counters For Dropped Packets

Requirement.

The device MUST provide accurate, per-interface counts of spoofed packets dropped in accordance with Section 2.5.5 and Section 2.5.6.

Justification.

Counters can help in identifying the source of spoofed traffic.

Examples.

An edge router may have several single-homed customers attached. When an attack using spoofed packets is detected, a quick check of counters may be able to identify which customer is attempting to send spoofed traffic.

Warnings.

None.

2.6. Rate Limiting Requirements

2.6.1. Support Rate Limiting

Requirement.

The device MUST provide the capability to limit the rate at which it will pass traffic based on protocol, source and destination IP address or CIDR block, source and destination port, and interface. Protocols MUST include at least IP, ICMP, UDP, and TCP and SHOULD include any protocol.

Justification.

This requirement provides a means of reducing or eliminating the impact of certain types of attacks. Also, rate limiting has the advantage that in some cases it can be turned on a priori, thereby offering some ability to mitigate the effect of future attacks prior to any explicit operator reaction to the attacks.

Examples.

Assume that a web hosting company provides space in its data-center to a company that becomes unpopular with a certain element of network users, who then decide to flood the web server with inbound ICMP traffic. It would be useful in such a situation to be able to rate-filter inbound ICMP traffic at the data-center's border routers. On the other side, assume that a new worm is released that infects vulnerable database servers such that they then start spewing traffic on TCP port 1433 aimed at random destination addresses as fast as the system and network interface of the infected server is capable. Further assume that a data center has many vulnerable servers that are infected and simultaneously sending large amounts of traffic with the result that all outbound links are saturated. Implementation of this requirement, would allow the network operator to rate limit inbound and/or outbound TCP 1433 traffic (possibly to a rate of 0 packets/bytes per second) to respond to the attack and maintain service levels for other legitimate customers/traffic.

Warnings.

None.

2.6.2. Support Directional Application Of Rate Limiting Per Interface

Requirement.

The device MUST provide support to rate-limit input and/or output separately on each interface.

Justification.

This level of granular control allows appropriately targeted controls that minimize the impact on third parties.

Examples.

If an ICMP flood is directed a single customer on an edge router, it may be appropriate to rate-limit outbound ICMP only on that customers interface.

Warnings.

None.

2.6.3. Support Rate Limiting Based on State

Requirement.

The device MUST be able to rate limit based on all TCP control flag bits. The device SHOULD support rate limiting of other stateful protocols where the normal processing of the protocol gives the device access to protocol state.

Justification.

This allows appropriate response to certain classes of attack.

Examples.

For example, for TCP sessions, it should be possible to rate limit based on the SYN, SYN-ACK, RST, or other bit state.

Warnings.

None.

2.7. Basic Filtering Capabilities

2.7.1. Ability to Filter Traffic

Requirement.

The device MUST provide a means to filter IP packets on any interface implementing IP.

Justification.

Packet filtering is important because it provides a basic means of implementing policies that specify which traffic is allowed and which is not. It also provides a basic tool for responding to malicious traffic.

Examples.

Access control lists that allow filtering based on protocol and/or source/destination address and or source/destination port would be one example.

Warnings.

None.

2.7.2. Ability to Filter Traffic TO the Device

Requirement.

It MUST be possible to apply the filtering mechanism to traffic that is addressed directly to the device via any of its interfaces - including loopback interfaces.

Justification.

This allows the operator to apply filters that protect the device itself from attacks and unauthorized access.

Examples.

Examples of this might include filters that permit only BGP from peers and SNMP and SSH from an authorized management segment and directed to the device itself, while dropping all other traffic addressed to the device.

Warnings.

None.

2.7.3. Ability to Filter Traffic THROUGH the Device

Requirement.

It MUST be possible to apply the filtering mechanism to traffic that is being routed (switched) through the device.

Justification.

This permits implementation of basic policies on devices that carry transit traffic (routers, switches, etc.).

Examples.

One simple and common way to meet this requirement is to provide the ability to filter traffic inbound to each interface and/or outbound from each interface. Ingress filtering as described in [RFC2827] provides one example of the use of this capability.

Warnings.

None.

2.7.4. Ability to Filter Without Significant Performance Degradation

Requirement.

The device MUST provide a means to filter packets without significant performance degradation. This specifically applies to stateless packet filtering operating on layer 3 (IP) and layer 4 (TCP or UDP) headers, as well as normal packet forwarding information such as incoming and outgoing interfaces.

The device MUST be able to apply stateless packet filters on ALL interfaces (up to the maximum number possible) simultaneously and with multiple filters per interface (e.g., inbound and outbound).

Justification.

This enables the implementation of filtering wherever and whenever needed. To the extent that filtering causes degradation, it may not be possible to apply filters that implement the appropriate policies.

Examples.

Another way of stating the requirement is that filter performance should not be the limiting factor in device throughput. If a device is capable of forwarding 30Mb/sec without filtering, then it should be able to forward the same amount with filtering in place.

Warnings.

The definition of "significant" is subjective. At one end of the spectrum it might mean "the application of filters may cause the box to crash". At the other end would be a throughput loss of less than one percent with tens of thousands of filters applied. The level of performance degradation that is acceptable will have to be determined by the operator.

Repeatable test data showing filter performance impact would be very useful in evaluating conformance with this requirement. Tests should include such information as packet size, packet rate, number of interfaces tested (source/destination), types of interfaces, routing table size, routing protocols in use, frequency of routing updates, etc. See [bmwg-acc-bench].

This requirement does not address stateful filtering, filtering above layer 4 headers or other more advanced types of filtering that may be important in certain operational environments.

2.7.5. Support Route Filtering

Requirement.

The device MUST provide a means to filter routing updates for all protocols used to exchange external routing information.

Justification.

See [RFC3013] and section 3.2 of [RFC2196].

Examples.

Operators may wish to ignore advertisements for routes to addresses allocated for private internets. See eBGP.

Warnings.

None.

2.7.6. Ability to Specify Filter Actions

Requirement.

The device MUST provide a mechanism to allow the specification of the action to be taken when a filter rule matches. Actions MUST include "permit" (allow the traffic), "reject" (drop with appropriate notification to sender), and "drop" (drop with no notification to sender). Also see Section 2.7.7 and Section 2.9

Justification.

This capability is essential to the use of filters to enforce policy.

Examples.

Assume that you have a small DMZ network connected to the Internet. You want to allow management using SSH coming from your corporate office. In this case, you might "permit" all traffic to port 22 in the DMZ from your corporate network, "rejecting" all others. Port 22 traffic from the corporate network is allowed through. Port 22 traffic from all other addresses results in an ICMP message to the sender. For those who are slightly more paranoid, you might choose to "drop" instead of "reject" traffic from unauthorized addresses, with the result being that *nothing* is sent back to the source.

Warnings.

While silently dropping traffic without sending notification may be the correct action in security terms, consideration should be given to operational implications. See [RFC3360] for consideration of potential problems caused by sending inappropriate TCP Resets.

2.7.7. Ability to Log Filter Actions

Requirement.

It MUST be possible to log all filter actions. The logging capability MUST be able to capture at least the following data:

* permit/deny/drop status,

* source and destination IP address,

* source and destination ports (if applicable to the protocol),

* which network element received the packet (interface, MAC address or other layer 2 information that identifies the previous hop source of the packet).

  Logging of filter actions is subject to the requirements of Section 2.11.

Justification.

Logging is essential for auditing, incident response, and operations.

Examples.

A desktop network may not provide any services that should be accessible from "outside." In such cases, all inbound connection attempts should be logged as possible intrusion attempts.

Warnings.

None.

2.8. Packet Filtering Criteria

2.8.1. Ability to Filter on Protocols

Requirement.

The device MUST provide a means to filter traffic based on the value of the protocol field in the IP header.

Justification.

Being able to filter on protocol is necessary to allow implementation of policy, secure operations and for support of incident response.

Examples.

Some denial of service attacks are based on the ability to flood the victim with ICMP traffic. One quick way (admittedly with some negative side effects) to mitigate the effects of such attacks is to drop all ICMP traffic headed toward the victim.

Warnings.

None.

2.8.2. Ability to Filter on Addresses

Requirement.

The function MUST be able to control the flow of traffic based on source and/or destination IP address or blocks of addresses such as Classless Inter-Domain Routing (CIDR) blocks.

Justification.

The capability to filter on addresses and address blocks is a fundamental tool for establishing boundaries between different networks.

Examples.

One example of the use of address based filtering is to implement ingress filtering per [RFC2827].

Warnings.

None.

2.8.3. Ability to Filter on Protocol Header Fields

Requirement.

The filtering mechanism MUST support filtering based on the value(s) of any portion of the protocol headers for IP, ICMP, UDP and TCP. It SHOULD support filtering of all other protocols supported at layer 3 and 4. It MAY support filtering based on the headers of higher level protocols. It SHOULD be possible to specify fields by name (e.g., "protocol = ICMP") rather than bit-offset/length/numeric value (e.g., 72:8 = 1).

Justification.

Being able to filter on portions of the header is necessary to allow implementation of policy, secure operations, and support incident response.

Examples.

This requirement implies that it is possible to filter based on TCP or UDP port numbers, TCP flags such as SYN, ACK and RST bits, and ICMP type and code fields. One common example is to reject "inbound" TCP connection attempts (TCP, SYN bit set+ACK bit clear or SYN bit set+ACK,FIN and RST bits clear). Another common

example is the ability to control what services are allowed in/out of a network. It may be desirable to only allow inbound connections on port 80 (HTTP) and 443 (HTTPS) to a network hosting web servers.

Warnings.

None.

2.8.4. Ability to Filter Inbound and Outbound

Requirement.

It MUST be possible to filter both incoming and outgoing traffic on any interface.

Justification.

This requirement allows flexibility in applying filters at the place that makes the most sense. It allows invalid or malicious traffic to be dropped as close to the source as possible.

Examples.

It might be desirable on a border router, for example, to apply an egress filter outbound on the interface that connects a site to its external ISP to drop outbound traffic that does not have a valid internal source address. Inbound, it might be desirable to apply a filter that blocks all traffic from a site that is known to forward or originate lots of junk mail.

Warnings.

None.

2.9. Packet Filtering Counter Requirements

2.9.1. Ability to Accurately Count Filter Hits

Requirement.

The device MUST supply a facility for accurately counting all filter hits.

Justification.

Accurate counting of filter rule matches is important because it shows the frequency of attempts to violate policy. This enables resources to be focused on areas of greatest need.

Examples.

Assume, for example, that a ISP network implements anti-spoofing egress filters (see [RFC2827]) on interfaces of its edge routers that support single-homed stub networks. Counters could enable the ISP to detect cases where large numbers of spoofed packets are being sent. This may indicate that the customer is performing potentially malicious actions (possibly in violation of the ISPs Acceptable Use Policy), or that system(s) on the customers network have been "owned" by hackers and are being (mis)used to launch attacks.

Warnings.

None.

2.9.2. Ability to Display Filter Counters

Requirement.

The device MUST provide a mechanism to display filter counters.

Justification.

Information that is collected is not useful unless it can be displayed in a useful manner.

Examples.

Assume there is a router with four interfaces. One is an up-link to an ISP providing routes to the Internet. The other three connect to separate internal networks. Assume that a host on one of the internal networks has been compromised by a hacker and is sending traffic with bogus source addresses. In such a situation, it might be desirable to apply ingress filters to each of the internal interfaces. Once the filters are in place, the counters can be examined to determine the source (inbound interface) of the bogus packets.

Warnings.

None.

2.9.3. Ability to Display Filter Counters per Rule

Requirement.

The device MUST provide a mechanism to display filter counters per rule.

Justification.

This makes it possible to see which rules are matching and how frequently.

Examples.

Assume that a filter has been defined that has two rules, one permitting all SSH traffic (tcp/22) and the second dropping all remaining traffic. If three packets are directed toward/through the point at which the filter is applied, one to port 22, the others to different ports, then the counter display should show 1 packet matching the permit tcp/22 rule and 2 packets matching the deny all others rule.

Warnings.

None.

2.9.4. Ability to Display Filter Counters per Filter Application

Requirement.

If it is possible for a filter to be applied more than once at the same time, then the device MUST provide a mechanism to display filter counters per filter application.

Justification.

It may make sense to apply the same filter definition simultaneously more than one time (to different interfaces, etc.). If so, it would be much more useful to know which instance of a filter is matching than to know that some instance was matching somewhere.

Examples.

One way to implement this requirement would be to have the counter display mechanism show the interface (or other entity) to which the filter has been applied, along with the name (or other designator) for the filter. For example if a filter named

"desktop_outbound" applied two different interfaces, say, "ethernet0" and "ethernet1", the display should indicate something like "matches of filter 'desktop_outbound' on ethernet0 ..." and "matches of filter 'desktop_outbound' on ethernet1 ..."

Warnings.

None.

2.9.5. Ability to Reset Filter Counters

Requirement.

It MUST be possible to reset counters to zero on a per filter basis.

For the purposes of this requirement it would be acceptable for the system to maintain two counters: an "absolute counter", C[now], and a "reset" counter, C[reset]. The absolute counter would maintain counts that increase monotonically until they wrap or overflow the counter. The reset counter would receive a copy of the current value of the absolute counter when the reset function was issued for that counter. Functions that display or retrieve the counter could then display the delta (C[now] - C[reset]).

Justification.

This allows operators to get a current picture of the traffic matching particular rules/filters.

Examples.

Assume that filter counters are being used to detect internal hosts that are infected with a new worm. Once it is believed that all infected hosts have been cleaned up and the worm removed, the next step would be to verify that. One way of doing so would be to reset the filter counters to zero and see if traffic indicative of the worm has ceased.

Warnings.

None.

2.9.6. Filter Counters Must Be Accurate

Requirement.

Filter counters MUST be accurate. They MUST reflect the actual number of matching packets since the last counter reset. Filter counters MUST be capable of holding up to 2^32 - 1 values without overflowing and SHOULD be capable of holding up to 2^64 - 1 values.

Justification.

Inaccurate data can not be relied on as the basis for action. Underreported data can conceal the magnitude of a problem.

Examples.

If N packets matching a filter are sent to/through a device, then the counter should show N matches.

Warnings.

None.

2.10. Other Packet Filtering Requirements

2.10.1. Ability to Specify Filter Log Granularity

Requirement.

It MUST be possible to enable/disable logging on a per rule basis.

Justification.

The ability to tune the granularity of logging allows the operator to log only the information that is desired. Without this capability, it is possible that extra data (or none at all) would be logged, making it more difficult to find relevant information.

Examples.

If a filter is defined that has several rules, and one of the rules denies telnet (tcp/23) connections, then it should be possible to specify that only matches on the rule that denies telnet should generate a log message.

Warnings.

None.

2.11. Event Logging Requirements

2.11.1. Logging Facility Uses Protocols Subject To Open Review

Requirement.

The device MUST provide a logging facility that is based on protocols subject to open review. See Section 1.8. Custom or proprietary logging protocols MAY be implemented provided the same information is made available.

Justification.

The use of logging based on protocols subject to open review permits the operator to perform archival and analysis of logs without relying on vendor-supplied software and servers.

Examples.

This requirement may be satisfied by the use of one or more of syslog [RFC3164], syslog with reliable delivery [RFC3195], TACACS+ [RFC1492] or RADIUS [RFC2865].

Warnings.

While [RFC3164] meets this requirement, it has many security issues and by itself does not meet the requirements of Section 2.1.1. See the security considerations section of [RFC3164] for a list of issues. [RFC3195] provides solutions to most/all of these issues....however at the time of this writing there are few implementations. Other possible solutions might be to tunnel syslog over a secure transport...but this often raises difficult key management and scalability issues.

The current best solution seems to be the following:

* Implement [RFC3164].

* Consider implementing [RFC3195].

2.11.2. Logs Sent To Remote Servers

Requirement.

The device MUST support transmission of records of security related events to one or more remote devices. There MUST be configuration settings on the device that allow selection of servers.

Justification.

This is important because it supports individual accountability. It is important to store them on a separate server to preserve them in case of failure or compromise of the managed device.

Examples.

This requirement may be satisfied by the use of one or more of: syslog [RFC3164], syslog with reliable delivery [RFC3195], TACACS+ [RFC1492] or RADIUS [RFC2865].

Warnings.

Note that there may be privacy or legal considerations when logging/monitoring user activity.

High volumes of logging may generate excessive network traffic and/or compete for scarce memory and CPU resources on the device.

2.11.3. Ability to Select Reliable Delivery

Requirement.

It SHOULD be possible to select reliable delivery of log messages.

Justification.

Reliable delivery is important to the extent that log data is depended upon to make operational decisions and forensic analysis. Without reliable delivery, log data becomes a collection of hints.

Examples.

One example of reliable syslog delivery is defined in [RFC3195]. Syslog-ng provides another example, although the protocol has not been standardized.

Warnings.

None.

2.11.4. Ability to Log Locally

Requirement.

It SHOULD be possible to log locally on the device itself. Local logging SHOULD be written to non-volatile storage.

Justification.

Local logging of failed authentication attempts to non-volatile storage is critical. It provides a means of detecting attacks where the device is isolated from its authentication interfaces and attacked at the console.

Local logging is important for viewing information when connected to the device. It provides some backup of log data in case remote logging fails. It provides a way to view logs relevant to one device without having to sort through a possibly large set of logs from other devices.

Examples.

One example of local logging would be a memory buffer that receives copies of messages sent to the remote log server. Another example might be a local syslog server (assuming the device is capable of running syslog and has some local storage).

Warnings.

Storage on the device may be limited. High volumes of logging may quickly fill available storage, in which case there are two options: new logs overwrite old logs (possibly via the use of a circular memory buffer or log file rotation), or logging stops.

2.11.5. Ability to Maintain Accurate System Time

Requirement.

The device MUST maintain accurate, "high resolution" (see definition in Section 1.8) system time.

Justification.

Accurate time is important to the generation of reliable log data. Accurate time is also important to the correct operation of some authentication mechanisms.

Examples.

This requirement may be satisfied by supporting Network Time Protocol (NTP), Simple Network Time Protocol (SNTP), or via direct connection to an accurate time source.

Warnings.

System clock chips are inaccurate to varying degrees. System time should not be relied upon unless it is regularly checked and synchronized with a known, accurate external time source (such as an NTP stratum-1 server). Also note that if network time synchronization is used, an attacker may be able to manipulate the clock unless cryptographic authentication is used.

2.11.6. Display Timezone And UTC Offset

Requirement.

All displays and logs of system time MUST include a timezone or offset from UTC.

Justification.

Knowing the timezone or UTC offset makes correlation of data and coordination with data in other timezones possible.

Examples.

Bob is in Newfoundland, Canada which is UTC -3:30. Alice is somewhere in Indiana, USA. Some parts of Indiana switch to daylight savings time while others do not. A user on Bob’s network attacks a user on Alice’s network. Both are using logs with local timezones and no indication of UTC offset. Correlating these logs will be difficult and error prone. Including timezone, or better, UTC offset, eliminates these difficulties.

Warnings.

None.

2.11.7. Default Timezone Should Be UTC

Requirement.

The default timezone for display and logging SHOULD be UTC. The device MAY support a mechanism to allow the operator to specify the display and logging of times in a timezone other than UTC.

Justification.

Knowing the timezone or UTC offset makes correlation of data and coordination with data in other timezones possible.

Examples.

Bob in Newfoundland (UTC -3:30) and Alice in Indiana (UTC -5 or UTC -6 depending on the time of year and exact county in Indiana) are working an incident together using their logs. Both left the default settings, which was UTC, so there was no translation of time necessary to correlate the logs.

Warnings.

None.

2.11.8. Logs Must Be Timestamped

Requirement.

By default, the device MUST timestamp all log messages. The timestamp MUST be accurate to within a second or less. The timestamp MUST include a timezone. There MAY be a mechanism to disable the generation of timestamps.

Justification.

Accurate timestamps are necessary for correlating events, particularly across multiple devices or with other organizations. This applies when it is necessary to analyze logs.

Examples.

This requirement MAY be satisfied by writing timestamps into syslog messages.

Warnings.

It is difficult to correlate logs from different time zones. Security events on the Internet often involve machines and logs from a variety of physical locations. For that reason, UTC is preferred, all other things being equal.

2.11.9. Logs Contain Untranslated IP Addresses

Requirement.

Log messages MUST NOT list translated addresses (DNS names) associated with the address without listing the untranslated IP address where the IP address is available to the device generating the log message.

Justification.

Including IP address of access list violations authentication attempts, address lease assignments and similar events in logs enables a level of individual and organizational accountability and is necessary to enable analysis of network events, incidents, policy violations, etc.

DNS entries tend to change more quickly than IP block assignments. This makes the address more reliable for data forensics.

DNS lookups can be slow and consume resources.

Examples.

A failed network login should generate a record with the source address of the login attempt.

Warnings.

* Source addresses may be spoofed. Network-based attacks often use spoofed source addresses. Source addresses should not be completely trusted unless verified by other means.

* Addresses may be reassigned to different individual, for example, in a desktop environment using DHCP. In such cases the individual accountability afforded by this requirement is weak. Having accurate time in the logs increases the chances that the use of an address can be correlated to an individual.

* Network topologies may change. Even in the absence of dynamic address assignment, network topologies and address block assignments do change. Logs of an attack one month ago may not give an accurate indication of which host, network or organization owned the system(s) in question at the time.

2.11.10. Logs Contain Records Of Security Events

Requirement.

The device MUST be able to send a record of at least the following events:

* authentication successes,

* authentication failures,

* session Termination,

* authorization changes,

* configuration changes,

* device status changes.

The device SHOULD be able to send a record of all other security related events.

Justification.

This is important because it supports individual accountability. See section 4.5.4.4 of [RFC2196].

Examples.

Examples of events for which there must be a record include: user logins, bad login attempts, logouts, user privilege level changes, individual configuration commands issued by users and system startup/shutdown events.

Warnings.

This list is far from complete.

Note that there may be privacy or legal considerations when logging/monitoring user activity.

2.11.11. Logs Do Not Contain Passwords

Requirement.

Passwords SHOULD be excluded from all audit records, including records of successful or failed authentication attempts.

Justification.

Access control and authorization requirements differ for accounting records (logs) and authorization databases (passwords). Logging passwords may grant unauthorized access to individuals with access to the logs. Logging failed passwords may give hints about actual passwords. See section 4.5.4.4 of [RFC2196].

Examples.

A user may make small mistakes in entering a password such as using incorrect capitalization ("my password" vs. "My Password").

Warnings.

There may be situations where it is appropriate/required to log passwords.

2.12. Authentication, Authorization, and Accounting (AAA) Requirements

2.12.1. Authenticate All User Access

Requirement.

The device MUST provide a facility to perform authentication of all user access to the system.

Justification.

This functionality is required so that access to the system can be restricted to authorized personnel.

Examples.

This requirement MAY be satisfied by implementing a centralized authentication system. See Section 2.12.5. It MAY also be satisfied using local authentication. See Section 2.12.6.

Warnings.

None.

2.12.2. Support Authentication of Individual Users

Requirement.

Mechanisms used to authenticate interactive access for configuration and management MUST support the authentication of distinct, individual users. This requirement MAY be relaxed to support system installation Section 2.4.5 or recovery of authorized access Section 2.12.15.

Justification.

The use of individual accounts, in conjunction with logging, promotes accountability. The use of group or default accounts undermines individual accountability.

Examples.

A user may need to log in to the device to access CLI functions for management. Individual user authentication could be provided by a centralized authentication server or a username/password database stored on the device. It would be a violation of this rule for the device to only support a single "account" (with or without a username) and a single password shared by all users to gain administrative access.

Warnings.

This simply requires that the mechanism to support individual users be present. Policy (e.g., forbidding shared group accounts) and enforcement are also needed but beyond the scope of this document.

2.12.3. Support Simultaneous Connections

Requirement.

The device MUST support multiple simultaneous connections by distinct users, possibly at different authorization levels.

Justification.

This allows multiple people to perform authorized management functions simultaneously. This also means that attempted connections by unauthorized users do not automatically lock out authorized users.

Examples.

None.

Warnings.

None.

2.12.4. Ability to Disable All Local Accounts

Requirement.

The device MUST provide a means of disabling all local accounts including:

* local users,

* default accounts (vendor, maintenance, guest, etc.),

* privileged and unprivileged accounts.

A local account defined as one where all information necessary for user authentication is stored on the device.

Justification.

Default accounts, well-known accounts, and old accounts provide easy targets for someone attempting to gain access to a device. It must be possible to disable them to reduce the potential vulnerability.

Examples.

The implementation depends on the types of authentication supported by the device.

Warnings.

None.

2.12.5. Support Centralized User Authentication Methods

Requirement.

The device MUST support a method of centralized authentication of all user access via standard authentication protocols.

Justification.

Support for centralized authentication is particularly important in large environments where the network devices are widely distributed and where many people have access to them. This reduces the effort needed to effectively restrict and track access to the system by authorized personnel.

Examples.

This requirement can be satisfied through the use of DIAMETER [RFC3588], TACACS+ [RFC1492], RADIUS [RFC2865], or Kerberos [RFC1510].

The secure management requirements (Section 2.1.1) apply to AAA.

See [RFC3579] for a discussion security issues related to RADIUS.

Warnings.

None.

2.12.6. Support Local User Authentication Method

Requirement.

The device SHOULD support a local authentication method. If implemented, the method MUST NOT require interaction with anything external to the device (such as remote AAA servers), and MUST work in conjunction with Section 2.3.1 (Support a 'Console' Interface) and Section 2.12.7 (Support Configuration of Order of Authentication Methods).

Justification.

Support for local authentication may be required in smaller environments where there may be only a few devices and a limited number of people with access. The overhead of maintaining centralized authentication servers may not be justified.

Examples.

The use of local, per-device usernames and passwords provides one way to implement this requirement.

Warnings.

Authentication information must be protected wherever it resides. Having, for instance, local usernames and passwords stored on 100 network devices means that there are 100 potential points of failure where the information could be compromised vs. storing authentication data centralized server(s), which would reduce the potential points of failure to the number of servers and allow protection efforts (system hardening, audits, etc.) to be focused on, at most, a few servers.

2.12.7. Support Configuration of Order of Authentication Methods

Requirement.

The device MUST support the ability to configure the order in which supported authentication methods are attempted. Authentication SHOULD "fail closed", i.e., access should be denied if none of the listed authentication methods succeeds.

Justification.

This allows the operator flexibility in implementing appropriate security policies that balance operational and security needs.

Examples.

If, for example, a device supports RADIUS authentication and local usernames and passwords, it should be possible to specify that RADIUS authentication should be attempted if the servers are available, and that local usernames and passwords should be used for authentication only if the RADIUS servers are not available. Similarly, it should be possible to specify that only RADIUS or only local authentication be used.

Warnings.

None.

2.12.8. Ability To Authenticate Without Plaintext Passwords

Requirement.

The device MUST support mechanisms that do not require the transmission of plaintext passwords in all cases that require the transmission of authentication information across networks.

Justification.

Plaintext passwords can be easily observed using packet sniffers on shared networks. See [RFC1704] and [RFC3631] for a through discussion.

Examples.

Remote login requires the transmission of authentication information across networks. Telnet transmits plaintext passwords. SSH does not. Telnet fails this requirement. SSH passes.

Warnings.

None.

2.12.9. No Default Passwords

Requirement.

The initial configuration of the device MUST NOT contain any default passwords or other authentication tokens.

Justification.

Default passwords provide an easy way for attackers to gain unauthorized access to the device.

Examples.

Passwords such as the name of the vendor, device, "default", etc. are easily guessed. The SNMP community strings "public" and "private" are well known defaults that provide read and write access to devices.

Warnings.

Lists of default passwords for various devices are readily available at numerous websites.

2.12.10. Passwords Must Be Explicitly Configured Prior To Use

Requirement.

The device MUST require the operator to explicitly configure "passwords" prior to use.

Justification.

This requirement is intended to prevent unauthorized management access. Requiring the operator to explicitly configure passwords will tend to have the effect of ensuring a diversity of passwords. It also shifts the responsibility for password selection to the user.

Examples.

Assume that a device comes with console port for management and a default administrative account. This requirement together with No Default Passwords says that the administrative account should come with no password configured. One way of meeting this requirement would be to have the device require the operator to choose a password for the administrative account as part of a dialog the first time the device is configured.

Warnings.

While this device requires operators to set passwords, it does not prevent them from doing things such as using scripts to configure hundreds of devices with the same easily guessed passwords.

2.12.11. Ability to Define Privilege Levels

Requirement.

It MUST be possible to define arbitrary subsets of all management and configuration functions and assign them to groups or "privilege levels", which can be assigned to users per Section 2.12.12. There MUST be at least three possible privilege levels.

Justification.

This requirement supports the implementation of the principal of "least privilege", which states that an individual should only have the privileges necessary to execute the operations he/she is required to perform.

Examples.

Examples of privilege levels might include "user" which only allows the initiation of a PPP or telnet session, "read only", which allows read-only access to device configuration and operational statistics, "root/superuser/administrator" which allows update access to all configurable parameters, and "operator" which allows updates to a limited, user defined set of

parameters. Note that privilege levels may be defined locally on the device or on centralized authentication servers.

Warnings.

None.

2.12.12. Ability to Assign Privilege Levels to Users

Requirement.

The device MUST be able to assign a defined set of authorized functions, or "privilege level", to each user once they have authenticated themselves to the device. Privilege level determines which functions a user is allowed to execute. Also see Section 2.12.11.

Justification.

This requirement supports the implementation of the principal of "least privilege", which states that an individual should only have the privileges necessary to execute the operations he/she is required to perform.

Examples.

The implementation of this requirement will obviously be closely coupled with the authentication mechanism. If RADIUS is used, an attribute could be set in the user’s RADIUS profile that can be used to map the ID to a certain privilege level.

Warnings.

None.

2.12.13. Default Privilege Level Must Be ’None’

Requirement.

The default privilege level SHOULD NOT allow any access to management or configuration functions. It MAY allow access to user-level functions (e.g., starting PPP or telnet). It SHOULD be possible to assign a different privilege level as the default. This requirement MAY be relaxed to support system installation per Section 2.4.5 or recovery of authorized access per Section 2.12.15.

Justification.

This requirement supports the implementation of the principal of "least privilege", which states that an individual should only have the privileges necessary to execute the operations he/she is required to perform.

Examples.

Examples of privilege levels might include "user" which only allows the initiation of a PPP or telnet session, "read-only", which allows read-only access to device configuration and operational statistics, "root/superuser/administrator" which allows update access to all configurable parameters, and "operator" which allows updates to a limited, user defined set of parameters. Note that privilege levels may be defined locally on the device or on centralized authentication servers.

Warnings.

It may be required to provide exceptions to support the requirements to support recovery of privileged access (Section 2.12.15) and to support OS installation and configuration (Section 2.4.5). For example, if the OS and/or configuration has somehow become corrupt an authorized individual with physical access may need to have "root" level access to perform an install.

2.12.14. Change in Privilege Levels Requires Re-Authentication

Requirement.

The device MUST re-authenticate a user prior to granting any change in user authorizations.

Justification.

This requirement ensures that users are able to perform only authorized actions.

Examples.

This requirement might be implemented by assigning base privilege levels to all users and allowing the user to request additional privileges, with the requests validated by the AAA server.

Warnings.

None.

2.12.15. Support Recovery Of Privileged Access

Requirement.

The device MUST support a mechanism to allow authorized individuals to recover full privileged administrative access in the event that access is lost. Use of the mechanism MUST require physical access to the device. There MAY be a mechanism for disabling the recovery feature.

Justification.

There are times when local administrative passwords are forgotten, when the only person who knows them leaves the company, or when hackers set or change the password. In all these cases, legitimate administrative access to the device is lost. There should be a way to recover access. Requiring physical access to invoke the procedure makes it less likely that it will be abused. Some organizations may want an even higher level of security and be willing to risk total loss of authorized access by disabling the recovery feature, even for those with physical access.

Examples.

Some examples of ways to satisfy this requirement are to have the device give the user the chance to set a new administrative password when:

* The user sets a jumper on the system board to a particular position.

* The user sends a special sequence to the RS232 console port during the initial boot sequence.

* The user sets a "boot register" to a particular value.

Warnings.

This mechanism, by design, provides a "back door" to complete administrative control of the device and may not be appropriate for environments where those with physical access to the device can not be trusted.

Also see the warnings in Section 2.3.1 (Support a 'Console' Interface).

2.13. Layer 2 Devices Must Meet Higher Layer Requirements

Requirement.

If a device provides layer 2 services that are dependent on layer 3 or greater services, then the portions that operate at or above layer 3 MUST conform to the requirements listed in this document.

Justification.

All layer 3 devices have similar security needs and should be subject to similar requirements.

Examples.

Signaling protocols required for layer 2 switching may exchange information with other devices using layer 3 communications. In such cases, the device must provide a secure layer 3 facility. Also, if higher layer capabilities (say, SSH or SNMP) are used to manage a layer 2 device, then the rest of the requirements in this document apply to those capabilities.

Warnings.

None.

2.14. Security Features Must Not Cause Operational Problems

Requirement.

The use of security features specified by the requirements in this document SHOULD NOT cause severe operational problems.

Justification.

Security features which cause operational problems are not useful and may leave the operator with no mechanism for enforcing appropriate policy.

Examples.

Some examples of severe operational problems include:

* The device crashes.

* The device becomes unmanageable.

* Data is lost.

* Use of the security feature consumes excessive resources (CPU, memory, bandwidth).

Warnings.

Determination of compliance with this requirement involves a level of judgement. What is "severe"? Certainly crashing is severe, but what about a %5 loss in throughput when logging is enabled? It should also be noted that there may be unavoidable physical limitations such as the total capacity of a link.

2.15. Security Features Should Have Minimal Performance Impact

Requirement.

Security features specified by the requirements in this document SHOULD be implemented with minimal impact on performance. Other sections of this document may specify different performance requirements (e.g., "MUST"s).

Justification.

Security features which significantly impact performance may leave the operator with no mechanism for enforcing appropriate policy.

Examples.

If the application of filters is known to have the potential to significantly reduce throughput for non-filtered traffic, there will be a tendency, or in some cases a policy, not to use filters.

Assume, for example, that a new worm is released that scans random IP addresses looking for services listening on TCP port 1433. An operator might want to investigate to see if any of the hosts on their networks were infected and trying to spread the worm. One way to do this would be to put up non-blocking filters counting and logging the number of outbound connection 1433, and then to block the requests that are determined to be from infected hosts. If any of these capabilities (filtering, counting, logging) have the potential to impose severe performance penalties, then this otherwise rational course of action might not be possible.

Warnings.

Requirements for which performance is a particular concern include: filtering, rate-limiting, counters, logging and anti-spoofing.

3. Documentation Requirements

The requirements in this section are intended to list information that will assist operators in evaluating and securely operating a device.

3.1. Identify Services That May Be Listening

Requirement.

The vendor MUST provide a list of all services that may be active on the device. The list MUST identify the protocols and default ports (if applicable) on which the services listen. It SHOULD provide references to complete documentation describing the service.

Justification.

This information is necessary to enable a thorough assessment of the potential security risks associated with the operation of each service.

Examples.

The list will likely contain network and transport protocols such as IP, ICMP, TCP, UDP, routing protocols such as BGP and OSPF, application protocols such as SSH and SNMP along with references to the RFCs or other documentation describing the versions of the protocols implemented.

Web servers "usually" listen on port 80. In the default configuration of the device, it may have a web server listening on port 8080. In the context of this requirement "identify ... default port" would mean "port 8080".

Warnings.

There may be valid, non-technical reasons for not disclosing the specifications of proprietary protocols. In such cases, all that needs to be disclosed is the existence of the service and the default ports (if applicable).

3.2. Document Service Defaults

Requirement.

The vendor MUST provide a list of the default state of all services.

Justification.

Understanding risk requires understanding exposure. Each service that is enabled presents a certain level of exposure. Having a list of the services that is enabled by default makes it possible to perform meaningful risk analysis.

Examples.

The list may be no more than the output of a command that implements Section 2.5.1.

Warnings.

None.

3.3. Document Service Activation Process

Requirement.

The vendor MUST concisely document which features enable and disable services.

Justification.

Once risk has been assessed, this list provides the operator a quick means of understanding how to disable (or enable) undesired (or desired) services.

Examples.

This may be a list of commands to enable/disable services one by one or a single command which enables/disables "standard" groups of commands.

Warnings.

None.

3.4. Document Command Line Interface

Requirement.

The vendor MUST provide complete documentation of the command line interface with each software release. The documentation SHOULD include highlights of changes from previous versions. The documentation SHOULD list potential output for each command.

Justification.

Understanding of inputs and outputs is necessary to support scripting. See Section 2.4.2.

Examples.

Separate documentation should be provided for each command listing the syntax, parameters, options, etc. as well as expected output (status, tables, etc.).

Warnings.

None.

3.5. 'Console' Default Communication Profile Documented

Requirement.

The console default profile of communications parameters MUST be published in the system documentation.

Justification.

Publication in the system documentation makes the settings accessible. Failure to publish them could leave the operator having to guess.

Examples.

None.

Warnings.

None.

4. Assurance Requirements

The requirements in this section are intended to

- identify behaviors and information that will increase confidence that the device will meet the security functional requirements.
- Provide information that will assist in the performance of security evaluations.

4.1. Identify Origin of IP Stack

Requirement.

The vendor SHOULD disclose the origin or basis of the IP stack used on the system.

Justification.

This information is required to better understand the possible security vulnerabilities that may be inherent in the IP stack.

Examples.

"The IP stack was derived from BSD 4.4", or "The IP stack was implemented from scratch."

Warnings.

Many IP stacks make simplifying assumptions about how an IP packet should be formed. A malformed packet can cause unexpected behavior in the device, such as a system crash or buffer overflow which could result in unauthorized access to the system.

4.2. Identify Origin of Operating System

Requirement.

The vendor SHOULD disclose the origin or basis of the operating system (OS).

Justification.

This information is required to better understand the security vulnerabilities that may be inherent to the OS based on its origin.

Examples.

"The operating system is based on Linux kernel 2.4.18."

Warnings.

None.

5. Security Considerations

General

Security is the subject matter of this entire memo. The justification section of each individual requirement lists the security implications of meeting or not meeting the requirement.

SNMP

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in the MIB.

It is recommended that implementors consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Furthermore, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to MIB objects is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

6. References

6.1. Normative References

[ANSI.X9-52.1998] American National Standards Institute, "Triple Data Encryption Algorithm Modes of Operation", ANSI X9.52, 1998.

[FIPS.197] National Institute of Standards and Technology, "Advanced Encryption Standard", FIPS PUB 197, November 2001, <http://csrc.nist.gov/publications/fips/fips197/fips-197.ps>.

[PKCS.3.1993] RSA Laboratories, "Diffie-Hellman Key-Agreement Standard, Version 1.4", PKCS 3, November 1993.

[RFC1208] Jacobsen, O. and D. Lynch, "Glossary of networking terms", RFC 1208, March 1991.

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[RFC1492] Finseth, C., "An Access Control Protocol, Sometimes Called TACACS", RFC 1492, July 1993.

[RFC1510] Kohl, J. and C. Neuman, "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993.

[RFC1704] Haller, N. and R. Atkinson, "On Internet Authentication", RFC 1704, October 1994.

[RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2196] Fraser, B., "Site Security Handbook", FYI 8, RFC 2196, September 1997.

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method", RFC 2631, June 1999.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC3013] Killalea, T., "Recommended Internet Service Provider Security Services and Procedures", BCP 46, RFC 3013, November 2000.

[RFC3164] Lonvick, C., "The BSD Syslog Protocol", RFC 3164, August 2001.

[RFC3174] Eastlake, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, September 2001.

[RFC3195] New, D. and M. Rose, "Reliable Delivery for syslog", RFC 3195, November 2001.

[RFC3309] Stone, J., Stewart, R. and D. Otis, "Stream Control Transmission Protocol (SCTP) Checksum Change", RFC 3309, September 2002.

[RFC3330] IANA, "Special-Use IPv4 Addresses", RFC 3330, September 2002.

[RFC3360] Floyd, S., "Inappropriate TCP Resets Considered Harmful", BCP 60, RFC 3360, August 2002.

[RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

[RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.

[RFC3562] Leech, M., "Key Management Considerations for the TCP MD5 Signature Option", RFC 3562, July 2003.

[RFC3579] Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)", RFC 3579, September 2003.

[RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

[RFC3631] Bellovin, S., Schiller, J., and C. Kaufman, Eds., "Security Mechanisms for the Internet", RFC 3631, December 2003.

6.2. Informative References

[RFC3766] Orman, H. and P. Hoffman, "Determining Strengths For Public Keys Used For Exchanging Symmetric Keys", BCP 86, RFC 3766, April 2004.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.

[bmwg-acc-bench] Poretsky, S., "Framework for Accelerated Stress Benchmarking", Work in Progress, October 2003.

[Schneier] Schneier, B., "Applied Cryptography, 2nd Ed., Publisher John Wiley & Sons, Inc.", 1996.

Appendix A. Requirement Profiles

This Appendix lists different profiles. A profile is a list of list of requirements that apply to a particular class of devices. The minimum requirements profile applies to all devices.

A.1. Minimum Requirements Profile

The functionality listed here represents a minimum set of requirements to which managed infrastructure of large IP networks should adhere.

The minimal requirements profile addresses functionality which will provide reasonable capabilities to manage the devices in the event of attacks, simplify troubleshooting, keep track of events which affect system integrity, help analyze causes of attacks, as well as provide administrators control over IP addresses and protocols to help mitigate the most common attacks and exploits.

- Support Secure Channels For Management
- Use Protocols Subject To Open Review For Management
- Use Cryptographic Algorithms Subject To Open Review
- Use Strong Cryptography
- Allow Selection of Cryptographic Parameters
- Management Functions Should Have Increased Priority
- Support a 'Console' Interface
- 'Console' Communication Profile Must Support Reset
- 'Console' Default Communication Profile Documented
- 'Console' Requires Minimal Functionality of Attached Devices.
- Support Separate Management Plane IP Interfaces
- No Forwarding Between Management Plane And Other Interfaces
- 'CLI' Provides Access to All Configuration and Management Functions
- 'CLI' Supports Scripting of Configuration

- 'CLI' Supports Management Over 'Slow' Links
- Document Command Line Interface
- Support Software Installation
- Support Remote Configuration Backup
- Support Remote Configuration Restore
- Support Text Configuration Files
- Ability to Identify All Listening Services
- Ability to Disable Any and All Services
- Ability to Control Service Bindings for Listening Services
- Ability to Control Service Source Addresses
- Ability to Filter Traffic
- Ability to Filter Traffic TO the Device
- Support Route Filtering
- Ability to Specify Filter Actions
- Ability to Log Filter Actions
- Ability to Filter Without Significant Performance Degradation
- Ability to Specify Filter Log Granularity
- Ability to Filter on Protocols
- Ability to Filter on Addresses
- Ability to Filter on Protocol Header Fields
- Ability to Filter Inbound and Outbound
- Packet Filtering Counter Requirements
- Ability to Display Filter Counters
- Ability to Display Filter Counters per Rule

- Ability to Display Filter Counters per Filter Application
- Ability to Reset Filter Counters
- Filter Counters Must Be Accurate
- Logging Facility Uses Protocols Subject To Open Review
- Logs Sent To Remote Servers
- Ability to Log Locally
- Ability to Maintain Accurate System Time
- Display Timezone And UTC Offset
- Default Timezone Should Be UTC
- Logs Must Be Timestamped
- Logs Contain Untranslated IP Addresses
- Logs Contain Records Of Security Events
- Authenticate All User Access
- Support Authentication of Individual Users
- Support Simultaneous Connections
- Ability to Disable All Local Accounts
- Support Centralized User Authentication Methods
- Support Local User Authentication Method
- Support Configuration of Order of Authentication Methods
- Ability To Authenticate Without Plaintext Passwords
- Passwords Must Be Explicitly Configured Prior To Use
- No Default Passwords
- Ability to Define Privilege Levels
- Ability to Assign Privilege Levels to Users

- Default Privilege Level Must Be 'None'
- Change in Privilege Levels Requires Re-Authentication
- Support Recovery Of Privileged Access
- Logs Do Not Contain Passwords
- Security Features Must Not Cause Operational Problems
- Security Features Should Have Minimal Performance Impact
- Identify Services That May Be Listening
- Document Service Defaults
- Document Service Activation Process
- Identify Origin of IP Stack
- Identify Origin of Operating System
- Identify Origin of IP Stack
- Identify Origin of Operating System
- Layer 2 Devices Must Meet Higher Layer Requirements

A.2. Layer 3 Network Edge Profile

This section builds on the minimal requirements listed in A.1 and adds more stringent security functionality specific to layer 3 devices which are part of the network edge. The network edge is typically where much of the filtering and traffic control policies are implemented.

An edge device is defined as a device that makes up the network infrastructure and connects directly to customers or peers. This would include routers connected to peering points, switches connecting customer hosts, etc.

- Support Automatic Anti-spoofing for Single-Homed Networks
- Support Automatic Discarding Of Bogons and Martians
- Support Counters For Dropped Packets
- Support Rate Limiting

- Support Directional Application Of Rate Limiting Per Interface
- Support Rate Limiting Based on State
- Ability to Filter Traffic THROUGH the Device

Appendix B. Acknowledgments

This document grew out of an internal security requirements document used by UUNET for testing devices that were being proposed for connection to the backbone.

The editor gratefully acknowledges the contributions of:

- Greg Sayadian, author of a predecessor of this document.
- Eric Brandwine, a major source of ideas/critiques.
- The MITRE Corporation for supporting continued development of this document. NOTE: The editor's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the editor.
- The former UUNET network security team: Jared Allison, Eric Brandwine, Clarissa Cook, Dave Garn, Tae Kim, Kent King, Neil Kirr, Mark Krause, Michael Lamoureux, Maureen Lee, Todd MacDermid, Chris Morrow, Alan Pitts, Greg Sayadian, Bruce Snow, Robert Stone, Anne Williams, Pete White.
- Others who have provided significant feedback at various stages of the life of this document are: Ran Atkinson, Fred Baker, Steve Bellovin, David L. Black, Michael H. Behringer, Matt Bishop, Scott Blake, Randy Bush, Pat Cain, Ross Callon, Steven Christey, Owen Delong, Sean Donelan, Robert Elmore, Barbara Fraser, Barry Greene, Jeffrey Haas, David Harrington, Dan Hollis, Jeffrey Hutzelman, Merike Kaeo, James Ko, John Kristoff, Chris Lonvick, Chris Liljenstolpe, James W. Laferriere, Jared Mauch, Perry E. Metzger, Mike O'Connor, Alan Paller, Rob Pickering, Pekka Savola, Gregg Schudel, Juergen Schoenwaelder, Don Smith, Rodney Thayer, David Walters, Joel N. Weber II, Russ White, Anthony Williams, Neal Ziring.
- Madge B. Harrison and Patricia L. Jones, technical writing review.
- This listing is intended to acknowledge contributions, not to imply that the individual or organizations approve the content of this document.

- Apologies to those who commented on/contributed to the document and were not listed.

Author's Address

George M. Jones, Editor
The MITRE Corporation
7515 Colshire Drive, M/S WEST
McLean, Virginia 22102-7508
U.S.A.

Phone: +1 703 488 9740
EMail: gmj3871@pobox.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.