OPSEC
Internet-Draft
Expires: March 19, 2007

P. Cain
The Cooper-Cain Group, Inc.
G. Jones
The MITRE Corporation
September 15, 2006

Logging Capabilities for IP Network Infrastructure
draft-cain-logging-caps-01

Status of this Memo

Copyright Notice

Abstract

This document lists logging capabilities originally defined in RFC3871 and needed to support the current practices, including those described in the Operational Security Current Practices document[CURPRAC] Logging is defined as the delivery to another entity or system of messages about the device, the data passing through the device, or the device's interaction with another device.

Capabilities are defined without reference to specific technologies. This is done to leave room for deployment of new technologies that implement the capability. Each capability cites the practices it supports. Current implementations that support the capability are cited. Special considerations are discussed as appropriate listing operational and resource constraints, limitations of current implementations, trade offs, etc.

Table of Contents

1. Introduction

The Framework for Operational Security Capabilities [FRMWK] outlines the proposed effort of the IETF OPSEC working group. This includes producing a series of drafts to codify knowledge gained through operational experience about feature sets that are needed to securely deploy and operate managed network elements providing transit services at the data link and IP layers. Current plans include separate capabilities documents for Packet Filtering; Event Logging; In-Band and Out-of-Band Management; Configuration and Management Interfaces; AAA; and Documentation and Assurance. [CURPRAC] defines the goals, motivation, scope, definitions, intended audience, threat model, potential attacks and give justifications for each of the practices.

1.1. Threat Model

The logging capabilities are derived from real world observations where unexpected activities in a network infrastructure caused concern to the network operator. Examples of such activities are:

An adversary or unauthorized user login into an infrastructure device. The risk is that the configuration or other operating parameter could be modified.

A device becomes overwhelmed, throttles, or crashes. Without logging or some other mechanism to notifify the operator of the condition, the operator will not know that action is required to return the device to optimal operating condition.

Network problems cannot be properly diagnosed without information. Information does not exist unless generated.

Threats to the network devices may be classified into broad categories such as:

* Unexpected device status or configuration change * Failure to send log messages * Interception of log messages * Failure to store log messages

The main technical issues revolve around what events generate a log entry, what mechanisms are used to secure the logged information while it is in transit and while it is stored, and how long are logs retained. Note that guidance in both RFC3871 and the FRAMEWORK documents restrict capabilities to log event generators, so other elements in a logging infrastructure, such as event collection or

archival systems, are not discussed in this document. A good overview of building and operating a log infrastructure can be found in NIST Publication 800-62. [SP800-92]

One unintended threat to the loggin infrastructure is a self-inflicted denial-of-service attack due to an overwheming amount of log messages on the local machine -- such that the local system is using all it's available effort to capture log messages -- or through the network between the log generator and the log collector -- such that the remore system is innaccesible to management operations. Although not specifically a capability, care should be taken when configuring the logging infrastructure to account for this threat.

Although most people equate logging with using the syslog protocol, other protocols such as SNMP [RFC2271] are quite capable of generating a log entry for transmission to a remote log entry collector.

1.2. Capabilities vs. Requirements ?

Capabilities may or may not be requirements. That is a local determination that must be made by each operator with reference to the policies that they must support. This document, together with [CURPRAC] will assist network operators in identifying their security capability requirements and communicating them clearly to vendors.

1.3. Format

Each capability has the following subsections:

- o Capability (what)
- o Discussion
- o Supported Practices (why)
- o Current Implementations (how)
- o Considerations (caveats, resource issues, protocol issues, etc.)

The Capability section describes a feature to be supported by the device. The Supported Practice section cites practices described in [CURPRAC] that are supported by this capability. The Current Implementation section is intended to give examples of implementations of the capability, citing technology and standards current at the time of writing. It is expected that the choice of features to implement the capabilities will change over time. The Considerations section lists operational and resource constraints,

limitations of current implementations, trade offs, etc.

1.4. Definitions

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 author, to assign an expectation level ("MUST", "SHOULD", "MAY") to the each capability. It must be noted that different organizations, operational environments, policies and legal environments will generate different requirement levels.

NOTE: This document defines capabilities. This document does not define requirements, and there is no requirement that any particular capability be implemented or deployed. The use of the terms MUST, SHOULD, and so on are in the context of each capability in the sense that if you conform to any particular capability then you MUST or SHOULD do what is specified for that capability, but there is no requirement that you actually do conform to any particular capability.

2. Functional Capabilities of Log Generating Systems

The capabilities in this section are intended to list testable, functional capabilities that are needed to operate devices securely and meet the obligations of Section 1.1Threat Model.

2.1. Logging Facility Uses Protocols Subject To Open Review

Capability

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

Discussion

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. .

Supported Practices.

* syslog

* syslog with reliable delivery

* SNMP with applicable security controls

Current Implementations

This capability can be satisfied by the use of one or more of syslog [RFC3164], syslog with reliable delivery [RFC3195], TACACS+ [RFC1492], RADIUS [RFC2865] or SNMP [RFC2271].

The current best solution seems to be the following:

* Implement syslog [RFC3164].

* Consider implementing syslog with reliable delivery [RFC3195].

Considerations

None.

2.2. Logs Sent To Remote Servers

Capability

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

Discussion

None.

Supported Practices

* Use multiple collection devices to enhance reliability.

* Use different collection devices to segregate different event sensitivity levels.

Current Implementations

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

Considerations

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.

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.

2.3. Ability to Select Reliable Delivery

Capability

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

Discussion

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 instead of a true record of events.

Supported Practices

* Use syslog-ng.

* Tunnel the logging stream over a TCP-based connection.

* Use an out-of-band network to connect critical logging devices to the collection device.

Current Implementations

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

Considerations

Reliable delivery should be used if the path from log event generator to the collection device transits administrative domains or uses unreliable channels, as it is important that the entire stream of leg events is captured.

2.4. Ability to Remotely Log Securely

Capability

The log data stream SHOULD be able to be delivered to the collection device in a confidential manner.

Discussion

While syslog *could* provide this capability, it has many security issues and by itself does not address issues from the threat model. See the security considerations section of [RFC3164] for a list of issues. Syslog with reliable delivery 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.

Supported Practices

* Log data tunnelled within IPSec or SSH.

* Use syslog-ng.

Current Implementations

There is no common implementation of this capability.

Considerations

As is the reliable delivery capability, delivering log data across untrusted streams or including sensitive data in a event data may require additional countermeasures to protect the data. This concern should not be addressed lightly.

ISPs are fully aware that there is no security with syslog but IPSec is considered too operationally expensive and cumbersome to deploy. Syslog-ng and stunnel could be used for better authentication and integrity protected solutions. Mechanisms to prevent unauthorized personnel from tampering with logs is constrained to auditing who has access to the logging servers and files.

2.5. Ability to Log Locally

Capability

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

Discussion

Logging of failed authentication attempts to local non-volatile storage is critical as it provides a record of events if the device gets isolated from its authentication interfaces or an attack overwhelms the console interface. Local logging is also important for viewing information when connected to the device and it provides some backup of log data in case remote logging fails.

Local logging also provides a way to quickly view logs relevant to one device without having to sort through a possibly large set of logs from other devices at the collection device.

Supported Practices

* To conserve space, only failed device logins and network connectivity issues are logged locally.

Current Implementations

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).

Considerations

Storage on the device may be limited.; high volumes of log messages may quickly fill the 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.

2.6. Ability to Log Different Severities to Different Destinations

Capability

The device SHOULD allow specified severity levels of log message to be delivered to different collection destinations.

Discussion

A network of multiple devices may generate a significant amount of log data. The ability to send critical log messages, for example a root login, to a specific destination device will enhance the ability of the network operator to notice the critical event.

Supported Practices

* Email critical event notices to a 24-hour watched mailbox.

* Send critical event notices to a separate log collector that scrolls received messages upon a large display in the NOC.

Current Implementations

There are no common implementations of this capability.

Considerations

The use of multiple collectors will incur maintenance and reliability issues. In some cases, multiple filters watching a single collection point may be more efficient than using multiple collectors.

2.7. Ability to Log to Multiple Destinations

Capability

The device SHOULD allow log message to be delivered to multiple collection destinations.

Discussion

All ISPs have multiple syslog servers - some ISPs choose to use separate syslog servers for varying infrastructure devices (i.e., one syslog server for backbone routers, one syslog server for customer edge routers, etc.) This provides a backup mechanism to see what is going on in the network in the event that a device may

'forget' to do syslog if the CPU is busy.

Supported Practices

* Use multiple log servers to enhance reliability.

Current Implementations

Most ISPs use multiple, sometimes gegraphic-driven, log collectors.

Considerations

None.

2.8. Ability to Maintain Accurate System Time

Capability

The device MUST maintain accurate, "high resolution" system time.

Discussion

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

The ability to correlate network events from different devices is directly related to the accuracy of the log timestamps. If a timeline cannot be constructed, the event logs and forensic data is useless.

Supported Practices

* The time is derived from NTP which is generally configured as a flat hierarchy at stratum1 and stratum2 servers to have less configuration and less maintenance issues.

* Each router is configured with one stratum1 peer both locally and remotely.

Current Implementations

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

Considerations

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..

2.9. Display Timezone And UTC Offset

Capability

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

Discussion

None.

Supported Practices

* The log timestamps include a timezone indicator like "-05:00".

Current Implementations

Considerations

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

Notice that a physical location may have different offsets from UTC during a year as summertime, daylight savings time, or other local customs are applied.

2.10. Default Timezone Should Be UTC

Capability

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.

Discussion

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

Supported Practices

* The timezone offset can be entered as part of configuration of a device.

Implementation

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.

Considerations

None.

2.11. Log Entries Must Be Timestamped

Capability

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.

Discussion

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

Supported Practices

* Each entry into the log contains a time value.

Current Implementations

This capability may be satisfied by writing timestamps into syslog messages.

Considerations

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..

2.12. Log on Exception or Identifed Event

Capability

Log entries should be generated on exceptions (e.g., failures) or event matching (e.g., generate a log entry if an event happens) via a configurable value.

Discussion

Traditionally log events are generated on exceptions -- failures or errors. Many times this is not sufficient as a network operator cannot tell if an attacker failed to log into a device once, or failed once and then succeeded on the second try. Devices should be configurable to allow for log messages on failures, successes, or everything.

Supported Practices

* Log all login events to a device but only have the collection device alert on failures.

* Log on successful device configuration changes since one must be aware of all modifications on some types of devices.

Current Implementations

Some ISPs put in passive devices to see routing updates and withdrawals and not rely solely on the device for log files.

Considerations

None.

2.13. Logs Contain Untranslated IP Addresses

Capability

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.

Discussion

Although some times less obtuse than DNS names, IP address assignments tend to be more stable than DNS entries. If an operator is trying to correlate a historical event, the DNS name may have been changed from that used at the event. TO ease this confusion, the IP address of the source of the action that caused the log event should be retained in the log entry.

Supported Practices

* Include the source IP address in all log messages.

* Although a corresponding DNS name is useful, DNS lookups can be slow and consume resources.

Current Implementations

Most devices include the source IP in event logs

Considerations

A failed network login should generate a record with the source address of the login attempt, but the Source addresses may be spoofed. Network-based attacks often use spoofed source addresses so they should not be completely trusted unless verified by other means. Having accurate timestamps in the logs increases the chances that the use of an address can be correlated to an individual.

2.14. Logs Contain Records Of Security Events

Capability

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 including filtering (or ACL) exceptions, routing protocol state changes, all device access (regardless of authentication success or failure), all commands issued to a device, and all routing events (boot-up/flaps).

Discussion

The main function of any of these log messages is to see what the device is doing as well as to try and ascertain what certain malicious attackers are trying to do.

Typically the data logged will contain the source and destination IP addresses and layer 4 port numbers as well as a timestamp.

Supported Practices

* Examples of events recorded include: user logins, bad login attempts, logouts, user privilege level changes, individual configuration commands issued by users and system startup/ shutdown events.

Current Implementations

Most devices crudely support this capability.

Considerations

This list is far from complete. Note that there may be privacy or legal considerations when logging/monitoring user activity or personal information.

This is an important capability because it supports individual accountability and auditing as well as forensics. See section 4.5.4.4 of .

2.15. Logs Do Not Contain Passwords

Capability

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

Discussion

A user may make small mistakes in entering a password such as using incorrect capitalization ("my password" vs. "My Password"). Event logs are traditional disperse widely so unexpected events will be noticed. Unauthorized access to event logs that contain these mistakes may compromise more than just the network devices as most users do not have independent passwords for every system.

Supported Practices

* Login failure log messages include the failed username, timestamp, and source IP address, but not the password used.

Current Implementations

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 also give hints about actual passwords. See section 4.5.4.4 of [RFC2196]

Considerations

There may be situations where it is appropriate/required to log passwords, such as when performing real-time attack analysis. Caution is advised in these rare circumstances.

2.16. Devices Should Log Every Message

Capability

Devices should be configurable to either log every event or to drop events due to congestion.

Discussion

Many devices implement logging as an afterthought with the device dropping log messages or failing to log critical events when the device is "busy". This behaviour makes forensic analysis difficult, if not impossible. Devices should be configurable to not drop log events at those operator-defined times when this behaviour s expected.

Supported Practices

* Use multiple logging devices and collectors to capture enough extra messages to be able to recreate a full log.

Current Implementations

Use multiple logging devices.

Considerations Improper configuration or implementation of this capability may open a device, network, or logging infrastructure to a self-inflicted denial-of-service attack.

None.

2.17. Syslog-specific Capabilties

The predominant logging mechanism within network infrastructures is BSD-syslog and its' variants. With such widespread uses, this section identifies capabilities specific to syslog.

2.17.1. Configurable Facility Values

Capability

The device SHOULD allow for the selection of the syslog facility number via configuration.

Discussion

A network operator may have many similar devices in their network. The ability to segregate different severity events by the strategic use of the syslog facility number is extremely useful.

Supported Practices

* Authentication log entries are marked at a different facility code to allow for easier segregation at the event collector.

Current Implementations

Some devices support this capability via a configuration variable.

Considerations

None.

2.17.2. Configurable Destination UDP Port

Capability

Devices should allow for the configuration of the destination syslog UDP port number.

Discussion

In large logging environments, spreading the load amongst multiple receiving daemons is a useful optimization. This capability also allows to differentiate different device functions very easily, for example all backbone router log to port 512 and all access router log to port 513.

Supported Practices

* Send all backbone routers log to port 512 and all access router log to port 513.

Current Implementations

Some devices support this capability via a configuration variable.

Considerations

None.

2.18. SNMP-specific capabilities

Another commonly used logging mechansim is using the trap and notification messages of the Simple Network Management Protocol.

2.18.1. Read-only Operations Supported

Capability

Devices should support the disablement of SNMP write operations to the device.

Discussion

Since SNMP is used as a management protocol in adition to its logging functionality, the ability to disable operations that would change the device operations should be supported for those devices whach aren't using the management functions.

Supported Practices

* Disable SNMP write operations.

Current Implementations

Some devices support this capability via a configuration variable.

Considerations

None.

2.18.2. Restrict Returning Data to specific Hosts

Capability

Devices should allow for restricting the IPAddresses that can query the SNMP interface for event data.

Discussion

Since event data can educate an adversary, devices should be able to only send event data ("responses") to certain, configured IP Addresses, not any system that interrogates them.

Supported Practices

* Configure devices to only accept SNMP requests from authorized addresses.

Current Implementations

Some devices support this capability via a configuration variable.

Considerations

None.

2.18.3. Only Return Specific Data to Requestor

Capability

Devices should support the delivery of specific managed object data (e.g., values linked to a specific OID) instead of returning all event data in the device (e.g., an entire OID subtree).

Discussion

Since event data can educate an adversary, devices should be able to only send specific event data instead of returning all the data in every query.

Supported Practices

* Queries request specific OID values instead of dumping the entire MIB. This practice reduces event data volume in addition to attaining security.

Current Implementations

Most devices support this capability.

Considerations

None.

3. Additional Operational Practices

This section describes practices not covered in [CURPRAC]. They are included here to provide justification for capabilities that reference them.

This section will be populated from comments received on this internet-draft.

4. Security Considerations

Security capabilities of network devices is the subject matter of this entire memo. The capabilities listed cite practices in [CURPRAC] that they are intended to support. [CURPRAC] also defines the general threat model, practices, and lists justifications for each practice.

5. IANA Considerations

There are no IANA actions required by this document.

6. Normative References

[CURPRAC] Kaeo, M., "Operational Security Current Practices", May 2006.

[FRMWK] Jones, G., Ed., "Framework for Operational Security Capabilities for IP Network Infrastructure, draft-ietf-opsec-framework-03 (work in progress)", October 2005, <draft-ietf-opsec-framework>.

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

[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", RFC 2196, September 1997.

[RFC2271] "An Architecture for Describing SNMP Management Frameworks", RFC 2271, January 1998.

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

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

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

[SP800-92] Souppaya, M. and K. Kent, "Guide to Security Log Management", FIPS 800-92, April 2006.

Authors' Addresses

Patrick Cain
The Cooper-Cain Group, Inc.
P.O. Box 400992
Cambridge, MA 02140
U.S.A.

Phone: +1 617-848-1950
Email: pcain@coopercain.com

George Jones
The MITRE Corporation
7515 Colshire Drive, M/S WEST
McLean, Virginia 22102-7508
USA

Phone: +1 703 488 974
Fax:
Email: gmjones@mitre.org
URI:

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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