Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 27, 2019

L. Xia
D. Zhang
Huawei
Y. Wu
Aliababa Group
R. Kumar
A. Lohiya
Juniper Networks
H. Birkholz
Fraunhofer SIT

October 24, 2018

An Information Model for the Monitoring of Network Security Functions (NSF)
draft-zhang-i2nsf-info-model-monitoring-07

Abstract

The Network Security Functions (NSF) NSF-facing interface exists between the Service Provider's management system (or Security Controller) and the NSF to enforce security policy provisioning and network security status monitoring. This document focuses on the monitoring part and defines the corresponding information model for it.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 27, 2019.

Table of Contents

1. Introduction 3
2. Terminology 4
2.1. Key Words 4
2.2. Definition of Terms 4
3. Use cases for NSF Monitoring Data 4
4. Classification of NSF Monitoring Data 5
4.1. Retention and Emission 5
4.2. Notifications and Events 6
4.3. Unsolicited Poll and Solicited Push 7
4.4. I2NSF Monitoring Terminology for Retained Information . . 8
5. Conveyance of NSF Monitoring Information 8
5.1. Information Types and Acquisition Methods 9
6. Basic Information Model for All Monitoring Data 10
7. Extended Information Model for Monitoring Data 10
7.1. System Alarm 11
7.1.1. Memory Alarm 11
7.1.2. CPU Alarm 11
7.1.3. Disk Alarm 11
7.1.4. Hardware Alarm 12
7.1.5. Interface Alarm 12
7.2. System Events 12
7.2.1. Access Violation 13
7.2.2. Configuration Change 13
7.3. System Log 13
7.3.1. Access Logs 14
7.3.2. Resource Utilization Logs 14
7.3.3. User Activity Logs 15
7.4. System Counters 15
7.4.1. Interface counters 15
7.5. NSF Events 16
7.5.1. DDoS Event 16
7.5.2. Session Table Event 17
7.5.3. Virus Event 17
7.5.4. Intrusion Event 18
7.5.5. Botnet Event 19
7.5.6. Web Attack Event 20

7.6. NSF Logs 21
7.6.1. DDoS Logs 21
7.6.2. Virus Logs 22
7.6.3. Intrusion Logs 22
7.6.4. Botnet Logs 22
7.6.5. DPI Logs 23
7.6.6. Vulnerabillity Scanning Logs 24
7.6.7. Web Attack Logs 24
7.7. NSF Counters 25
7.7.1. Firewall counters 25
7.7.2. Policy Hit Counters 26
8. IANA Considerations 27
9. Security Considerations 27
10. References 27
10.1. Normative References 27
10.2. Informative References 28
Acknowledgements 29
Authors' Addresses 29

1. Introduction

According to [I-D.ietf-i2nsf-terminology], the interface provided by an NSF (e.g., FW, IPS, Anti-DDOS, or Anti-Virus function) to administrative entities (e.g., NMS, security controller) to enable remote management (i.e. configuring and monitoring) is referred to as an "I2NSF NSF-Facing Interface". Monitoring procedures intent to acquire vital types of data at rest with respect to NSF, e.g. alarms, records, or counters, via data in motion, e.g. queries, notifications, or events. The monitoring of NSF plays an important role in the overall security framework, if done in a timely and comprehensive way. The monitoring information generated by an NSF can very well be an early indication of anomalous behavior or malicious activity, such as denial of service attacks.

This draft defines a comprehensive NSF monitoring information model that provides visibility into NSF. This document will not go into the design details of NSF-Facing Interfaces. Instead, it is focused on specifying the information and illustrates the methods that enable NSF to provide the information required in order to be monitored in a scalable and efficient way via the NSF-Facing Interface. The information model for monitoring presented in this document is a complement to the information model for the security policy provisioning part of the NSF-Facing Interface specified in [I-D.xibassnez-i2nsf-capability].

2. Terminology

2.1. Key Words

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].

2.2. Definition of Terms

This document uses the terms defined in [I-D.ietf-i2nsf-terminology].

3. Use cases for NSF Monitoring Data

As mentioned earlier, monitoring plays a critical role in the overall security framework. The monitoring of the NSF provides very valuable information to the security controller in maintaining the provisioned security posture. Besides this, there are various other reasons to monitor the NSF as listed below:

- The security administrator can configure a policy that is triggered on a specific event occurring in the NSF or the network. If a security controller detects the specified event, it configures additional security functions as defined by policies.

- The events triggered by NSF as a result of security policy violation can be used by SIEM to detect any suspicious activity in a larger correlation context.

- The events and activity logs from NSF can be used to build advanced analytics, such as behavior and predictive models to improve security posture in large deplyoments.

- The security controller can use events from the NSF for achieving high availability. It can take corrective actions such as restarting a failed NSF, horizontally scaling the NSF, etc.

- The events and activity logs from the NSF can aid in the root cause analysis of an operational issue - therefore improve debugging.

- The activity logs from the NSF can be used to build historical data for operational and business reasons.

4. Classification of NSF Monitoring Data

In order to maintain a strong security posture, it is not only necessary to configure NSF security policies but also to continuously monitor NSF by consuming acquirable and therefore observable information. This enables security admins to assess the state of the network topology in a timely fashion. It is not possible to block all the internal and external threats based on static security posture. A more practical approach is supported by enabling dynamic security measures - for which continuous visibility is required. This draft defines a set of information elements (and their scope) that can be acquired from NSF and can be used as monitoring information. In essence, these types of monitoring information can be leveraged to support constant visibility on multiple levels of granularity and can be consumed by corresponding functions.

Three basic domains about the monitoring of information originating from a system entity [RFC4949] or a NSF are highlighted in this document.

- Retention and Emission
- Notifications and Events
- Unsolicited Poll and Solicited Push

The Alarm Management Framework in [RFC3877] defines an Event as "something that happens which may be of interest. A fault, a change in status, crossing a threshold, or an external input to the system, for example." In the I2NSF domain, I2NSF events [I-D.ietf-i2nsf-terminology] are created and the scope of the Alarm Management Framework Events is still applicable due to its broad definition. The model presented in this document elaborates on the work-flow of creating I2NSF events in the context of NSF monitoring and on how initial I2NSF events are created.

As with I2NSF components, every generic system entity can include a set of capabilities [I-D.ietf-i2nsf-terminology] that creates information about the context, composition, configuration, state or behavior of that system entity. This information is intended to be provided to other consumers of informations--and in the scope of this document, to monitor that information in an automated fashion.

4.1. Retention and Emission

Typically, a system entity populates standardized interface, such as SNMP, NETCONF, RESTCONF or CoMI to provide and emit created information directly via NSF-Facing Interfaces

[I-D.ietf-i2nsf-terminology]. Alternatively, the created information is retained inside the system entity (or hierarchy of system entities in a composite device) via records or counters that are not exposed directly via NSF-Facing Interfaces.

Information emitted via standardized interfaces can be consumed by an I2NSF Agent [I-D.ietf-i2nsf-terminology] that includes the capability to consume information not only via I2NSF Interfaces but also via interfaces complementary to the standardized interfaces a generic system entity provides.

Information retained on a system entity requires a corresponding I2NSF Agent to access aggregated records of information, typically in the form of logfiles or databases. There are ways to aggregate records originating from different system entities over a network, for examples via Syslog [RFC5424] or Syslog over TCP [RFC6587]. But even if records are conveyed, the result is the same kind of retention in form of a bigger aggregate of records on another system entity.

An I2NSF Agent is required to process fresh [RFC4949] records created by I2NSF Functions in order to provide them to other I2NSF Components via corresponding I2NSF Interfaces timely. This process is effectively based on homogenizing functions that can access and convert specific kinds of records into information that can be provided and emitted via I2NSF interfaces.

Retained or emitted, the information required to support monitoring processes has to be processed by an I2NSF Agent at some point in the work-flow. Typical locations of these I2NSF Agents are:

- a system entity that creates the information
- a system entity that retains an aggregation of records
- an I2NSF Component that includes the capabilities of using standardized interfaces provided by other system entities that are not I2NSF Components
- an I2NSF Component that creates the information

4.2. Notifications and Events

A specific task of I2NSF Agents is to process I2NSF Policy Rules [I-D.ietf-i2nsf-terminology]. Rules are composed of three clauses: Events, Conditions, and Actions. In consequence, an I2NSF Event is required to trigger an I2NSF Policy Rule. "An I2NSF Event is defined as any important occurrence in time of a change in the system being

managed, and/or in the environment of the system being managed." [I-D.ietf-i2nsf-terminology], which aligns well with the generic definition of Event from [RFC3877].

The model illustrated in this document introduces a complementary type of information that can be conveyed--notification.

Notification: An occurrence of a change of context, composition, configuration, state or behavior of a system entity that can be directly or indirectly observed by an I2NSF Agent and can be used as input for an event-clause in I2NSF Policy Rules.

A notification is similar to an I2NSF Event with the exception that it is created by a system entity that is not an I2NSF Component and that its importances is yet to be assessed. Semantically, a notification is not an I2NSF Event in the context of I2NSF, although they can potentially use the exact same information or data model. In respect to [RFC3877], a Notification is a specific subset of events, because they convey information about "something that happens which may be of interest". In consequence, Notifications may contain information with very low expressiveness or relevance. Hence, additional post-processing functions, such as aggregation, correlation or simple anomaly detection, might have to be employed to satisfy a level of expressiveness that is required for an event-clause of an I2NSF Policy Rule.

It is important to note that the consumer of a notification (the observer) assesses the importance of a notification and not the producer. The producer can include metadata in a notification that supports the observer in assessing the importance (even metadata about severity), but the deciding entity is an I2NSF Agent.

4.3. Unsolicited Poll and Solicited Push

The freshness of the monitored information depends on the acquisition method. Ideally, an I2NSF Agent is accessing every relevant information about the I2NSF Component and is emitting I2NSF Events to a monitoring NSF timely. Publication of events via a pubsub/broker model, peer-2-peer meshes, or static defined channels are only a few examples on how a solicited push of I2NSF Events can be facilitated. The actual mechanic implemented by an I2NSF Component is out of the scope of this document.

Often, corresponding management interfaces have to be queried in intervals or on-demand if required by an I2NSF Policy rule. In some cases, a collection of information has to be conducted via login mechanics provided by a system entity. Accessing records of

information via this kind of unsolicited polls can introduce a significant latency in regard to the freshness of the monitored information. The actual definition of intervals implemented by an I2NSF Component is also out of scope of this document.

4.4. I2NSF Monitoring Terminology for Retained Information

Records: Unlike information emitted via notifications and events, records do not require immediate attention from an analyst but may be useful for visibility and retroactive cyber forensic. Depending on the record format, there are different qualities in regard to structure and detail. Records are typically stored in logfiles or databases on a system entity or NSF. Records in the form of logfiles usually include less structures but potentially more detailed information in regard to changes of an system entity's characteristics. In contrast, databases often use more strict schemas or data models, therefore enforcing a better structure, but inhibit storing information that do not match those models ('closed world assumption'). Records can be continuously processed by I2NSF Agents that act as I2NSF Producer and emit events via functions specifically tailored to a certain type of record. Typically, records are information generated by NSF or system entity about their operational and informational data, or various changes in system characteristics, such as user activities, network/traffic status, network activity, etc. They are important for debugging, auditing and security forensic.

Counters: A specific representation of continuous value changes of information elements that potentially occur in high frequency. A prominent example are network interface counters, e.g. PDU amount or byte amount, drop counters, error counters etc. Counters are useful in debugging and visibility into operational behavior of the NSF. An I2NSF Agent that observes the progression of counters can act as an I2NSF Producer and emit events in respect to I2NSF Policy Rules.

5. Conveyance of NSF Monitoring Information

As per the use cases of NSF monitoring data, information needs to be conveyed to various I2NSF Consumers based on requirements imposed by I2NSF Capabilities and work-flows. There are multiple aspects to be considered in regard to emission of monitoring information to requesting parties as listed below:

- Pull-Push Model: A set of data can be pushed by a NSF to the requesting party or pulled by the requesting party from a NSF. Specific types of information might need both the models at the same time if there are multiple I2NSF Consumers with varying

requirements. In general, any I2NSF Event including a high severity assessment is considered to be of great importance and should be processed as soon as possible (push-model). Records, in contrast, are typically not as critical (pull-model). The I2NSF Architecture does not mandate a specific scheme for each type of information and is therefore out of scope of this document.

- Pub-Sub Model: In order for an I2NSF Provider to push monitoring information to multiple appropriate I2NSF Consumers, a subscription can be maintained by both I2NSF Components. Discovery of available monitoring information can be supported by an I2NSF Controller that takes on the role of a broker and therefore includes I2NSF Capabilities that support registration.

- Export Frequency: Monitoring information can be emitted immediately upon generation by a NSF to requesting I2NSF Consumers or can be pushed periodically. The frequency of exporting the data depends upon its size and timely usefulness. It is out of the scope of I2NSF and left to each NSF implementation.

- Authentication: There may be a need for authentication between I2NSF Producer of monitoring information and corresponding I2NSF Consumer to ensure that critical information remains confidential. Authentication in the scope of I2NSF can also require a corresponding content authorization. This may be necessary, for example, if a NSF emits monitoring information to I2NSF Consumer outside its administrative domain. The I2NSF Architecture does not mandate when and how specific authentication has to be implemented.

- Data-Transfer Model: Monitoring information can be pushed by NSF using a connection-less model that does require a persistent connection or streamed over a persistent connection. An appropriate model depends on the I2NSF Consumer requirements and the semantics of the information to be conveyed.

- Data Model and Interaction Model for Data in Motion: There are a lot of transport mechanisms such as IP, UDP, TCP. There are also open source implementations for specific set of data such as systems counter, e.g. IPFIX [RFC7011] or NetFlow [RFC3954]. The I2NSF does not mandate any specific method for a given data set, it is up to each implementation.

5.1. Information Types and Acquisition Methods

In this document most information types defined, benefit from high visibility with respect to value changes, e.g. alarms or records. In contrast, values that change monotonically in a continuous way do not

benefit from this high visibility. On the contrary, emitting each change would result in a useless amount of value updates. Hence, values, such as counter, are best acquired in periodic intervals.

The mechanism provided by YANG Push [I-D.ietf-netconf-yang-push] and YANG Subscribed Notifications [I-D.ietf-netconf-subscribed-notifications] address exactly these set of requirements. YANG also enables semantically well-structured information, as well as subscriptions to datatores or event streams - on-change or periodically.

In consequence, this information model is intended to support data models used in solicited or unsolicited event streams that potentially are facilitated by subscription mechanism. A subset of information elements defined in the information model address this domain of application.

6. Basic Information Model for All Monitoring Data

As explained in the above section, there is a wealth of data available from the NSF that can be monitored. Firstly, there must be some general information with each monitoring message sent from an NSF that helps consumer in identifying meta data with that message, which are listed as below:

- message_version: Indicate the version of the data format and is a two-digit decimal numeral starting from 01
- message_type: Event, Alert, Alarm, Log, Counter, etc
- time_stamp: Indicate the time when the message is generated
- vendor_name: The name of the NSF vendor
- NSF_name: The name (or IP) of the NSF generating the message
- Module_name: The module name outputting the message
- Severity: Indicates the level of the logs. There are total eight levels, from 0 to 7. The smaller the numeral is, the higher the severity is.

7. Extended Information Model for Monitoring Data

This section covers the additional information associated with the system messages. The extended information model is only for the structured data such as alarm. Any unstructured data is specified with basic information model only.

7.1. System Alarm

Characteristics:

- acquisition_method: subscription
- emission_type: on-change
- dampening_type: no-dampening

7.1.1. Memory Alarm

The following information should be included in a Memory Alarm:

- event_name: 'MEM_USAGE_ALARM'
- module_name: Indicate the NSF module responsible for generating this alarm
- usage: specifies the amount of memory used
- threshold: The threshold triggering the alarm
- severity: The severity of the alarm such as critical, high, medium, low
- message: 'The memory usage exceeded the threshold'

7.1.2. CPU Alarm

The following information should be included in a CPU Alarm:

- event_name: 'CPU_USAGE_ALARM'
- usage: Specifies the amount of CPU used
- threshold: The threshold triggering the event
- severity: The severity of the alarm such as critical, high, medium, low
- message: 'The CPU usage exceeded the threshold'

7.1.3. Disk Alarm

The following information should be included in a Disk Alarm:

- event_name: 'DISK_USAGE_ALARM'

- usage: Specifies the amount of disk space used
- threshold: The threshold triggering the event
- severity: The severity of the alarm such as critical, high, medium, low
- message: 'The disk usage exceeded the threshold'

7.1.4. Hardware Alarm

The following information should be included in a Hardware Alarm:

- event_name: 'HW_FAILURE_ALARM'
- component_name: Indicate the HW component responsible for generating this alarm
- threshold: The threshold triggering the alarm
- severity: The severity of the alarm such as critical, high, medium, low
- message: 'The HW component has failed or degraded'

7.1.5. Interface Alarm

The following information should be included in a Interface Alarm:

- event_name: 'IFNET_STATE_ALARM'
- interface_Name: The name of interface
- interface_state: 'UP', 'DOWN', 'CONGESTED'
- threshold: The threshold triggering the event
- severity: The severity of the alarm such as critical, high, medium, low
- message: 'Current interface state'

7.2. System Events

Characteristics:

- acquisition_method: subscription

- emission_type: on-change
- dampening_type: on_repetition

7.2.1. Access Violation

The following information should be included in this event:

- event_name: 'ACCESS_DENIED'
- user: Name of a user
- group: Group to which a user belongs
- login_ip_address: Login IP address of a user
- authentication_mode: User authentication mode. e.g., Local Authentication, Third-Party Server Authentication, Authentication Exemption, SSO Authentication
- message: 'access denied'

7.2.2. Configuration Change

The following information should be included in this event:

- event_name: 'CONFIG_CHANGE'
- user: Name of a user
- group: Group to which a user belongs
- login_ip_address: Login IP address of a user
- authentication_mode: User authentication mode. e.g., Local Authentication, Third-Party Server Authentication, Authentication Exemption, SSO Authentication
- message: 'Configuration modified'

7.3. System Log

Characteristics:

- acquisition_method: subscription
- emission_type: on-change

- dampening_type: on_repetition

7.3.1. Access Logs

Access logs record administrators' login, logout, and operations on the device. By analyzing them, security vulnerabilities can be identified. The following information should be included in operation report:

- Administrator: Administrator that operates on the device
- login_ip_address: IP address used by an administrator to log in
- login_mode: Specifies the administrator logs in mode e.g. root, user
- operation_type: The operation type that the administrator execute, e.g., login, logout, configuration, etc
- result: Command execution result
- content: Operation performed by an administrator after login.

7.3.2. Resource Utilization Logs

Running reports record the device system's running status, which is useful for device monitoring. The following information should be included in running report:

- system_status: The current system's running status
- CPU_usage: Specifies the CPU usage
- memory_usage: Specifies the memory usage
- disk_usage: Specifies the disk usage
- disk_left: Specifies the available disk space left
- session_number: Specifies total concurrent sessions
- process_number: Specifies total number of system processes
- in_traffic_rate: The total inbound traffic rate in pps
- out_traffic_rate: The total outbound traffic rate in pps
- in_traffic_speed: The total inbound traffic speed in bps

- out_traffic_speed: The total outbound traffic speed in bps

7.3.3. User Activity Logs

User activity logs provide visibility into users' online records (such as login time, online/lockout duration, and login IP addresses) and the actions users perform. User activity reports are helpful to identify exceptions during user login and network access activities.

- user: Name of a user
- group: Group to which a user belongs
- login_ip_address: Login IP address of a user
- authentication_mode: User authentication mode. e.g., Local Authentication, Third-Party Server Authentication, Authentication Exemption, SSO Authentication
- access_mode: User access mode. e.g., PPP, SVN, LOCAL
- online_duration: Online duration
- lockout_duration: Lockout duration
- type: User activities. e.g., Successful User Login, Failed Login attempts, User Logout, Successful User Password Change, Failed User Password Change, User Lockout, User Unlocking, Unknown
- cause: Cause of a failed user activity

7.4. System Counters

Characteristics:

- acquisition_method: subscription or query
- emission_type: periodical
- dampening_type: none

7.4.1. Interface counters

Interface counters provide visibility into traffic into and out of NSF, bandwidth usage.

- interface_name: Network interface name configured in NSF

- in_total_traffic_pkts: Total inbound packets
- out_total_traffic_pkts: Total outbound packets
- in_total_traffic_bytes: Total inbound bytes
- out_total_traffic_bytes: Total outbound bytes
- in_drop_traffic_pkts: Total inbound drop packets
- out_drop_traffic_pkts: Total outbound drop packets
- in_drop_traffic_bytes: Total inbound drop bytes
- out_drop_traffic_bytes: Total outbound drop bytes
- in_traffic_ave_rate: Inbound traffic average rate in pps
- in_traffic_peak_rate: Inbound traffic peak rate in pps
- in_traffic_ave_speed: Inbound traffic average speed in bps
- in_traffic_peak_speed: Inbound traffic peak speed in bps
- out_traffic_ave_rate: Outbound traffic average rate in pps
- out_traffic_peak_rate: Outbound traffic peak rate in pps
- out_traffic_ave_speed: Outbound traffic average speed in bps
- out_traffic_peak_speed: Outbound traffic peak speed in bps.

7.5. NSF Events

Characteristics:

- acquisition_method: subscription
- emission_type: on-change
- dampening_type: none

7.5.1. DDoS Event

The following information should be included in a DDoS Event:

- event_name: 'SEC_EVENT_DDoS'

- sub_attack_type: Any one of Syn flood, ACK flood, SYN-ACK flood, FIN/RST flood, TCP Connection flood, UDP flood, Icmp flood, HTTPS flood, HTTP flood, DNS query flood, DNS reply flood, SIP flood, and etc.

- dst_ip: The IP address of a victum under attack

- dst_port: The port numbers that the attrack traffic aims at.

- start_time: The time stamp indicating when the attack started

- end_time: The time stamp indicating when the attack ended. If the attack is still undergoing when sending out the alarm, this field can be empty.

- attack_rate: The PPS of attack traffic

- attack_speed: the bps of attack traffic

- rule_id: The ID of the rule being triggered

- rule_name: The name of the rule being triggered

- profile: Security profile that traffic matches.

7.5.2. Session Table Event

The following information should be included in a Session Table Event:

- event_name: 'SESSION_USAGE_HIGH'

- current: The number of concurrent sessions

- max: The maximum number of sessions that the session table can support

- threshold: The threshold triggering the event

- message: 'The number of session table exceeded the threshold'

7.5.3. Virus Event

The following information should be included in a Virus Event:

- event_Name: 'SEC_EVENT_VIRUS'

- virus_type: Type of the virus, e.g., trojan, worm, macro Virus type
- virus_name
- dst_ip: The destination IP address of the packet where the virus is found
- src_ip: The source IP address of the packet where the virus is found
- src_port: The source port of the packet where the virus is found
- dst_port: The destination port of the packet where the virus is found
- src_zone: The source security zone of the packet where the virus is found
- dst_zone: The destination security zone of the packet where the virus is found
- file_type: The type of the file where the virus is hided within
- file_name: The name of the file where the virus is hided within
- virus_info: The brief introduction of virus
- raw_info: The information describing the packet triggering the event.
- rule_id: The ID of the rule being triggered
- rule_name: The name of the rule being triggered
- profile: Security profile that traffic matches.

7.5.4. Intrusion Event

The following information should be included in a Intrustion Event:

- event_name: The name of event: 'SEC_EVENT_Intrusion'
- sub_attack_type: Attack type, e.g., brutal force, buffer overflow
- src_ip: The source IP address of the packet
- dst_ip: The destination IP address of the packet

- src_port:The source port number of the packet
- dst_port: The destination port number of the packet
- src_zone: The source security zone of the packet
- dst_zone: The destination security zone of the packet
- protocol: The employed transport layer protocol, e.g.,TCP, UDP
- app: The employed application layer protocol, e.g.,HTTP, FTP
- rule_id: The ID of the rule being triggered
- rule_name: The name of the rule being triggered
- profile: Security profile that traffic matches
- intrusion_info: Simple description of intrusion
- raw_info: The information describing the packet triggering the event.

7.5.5. Botnet Event

The following information should be included in a Botnet Event:

- event_name: the name of event: 'SEC_EVENT_Botnet'
- botnet_name: The name of the detected botnet
- src_ip: The source IP address of the packet
- dst_ip: The destination IP address of the packet
- src_port: The source port number of the packet
- dst_port: The destination port number of the packet
- src_zone: The source security zone of the packet
- dst_zone: The destination security zone of the packet
- protocol: The employed transport layer protocol, e.g.,TCP, UDP
- app: The employed application layer protocol, e.g.,HTTP, FTP
- role: The role of the communicating parties within the botnet:

1. the packet from zombie host to the attacker
2. The packet from the attacker to the zombie host
3. The packet from the IRC/WEB server to the zombie host
4. The packet from the zombie host to the IRC/WEB server
5. The packet from the attacker to the IRC/WEB server
6. The packet from the IRC/WEB server to the attacker
7. The packet from the zombie host to the victim

- botnet_info: Simple description of Botnet
- rule_id: The ID of the rule being triggered
- rule_name: The name of the rule being triggered
- profile: Security profile that traffic matches
- raw_info: The information describing the packet triggering the event.

7.5.6. Web Attack Event

The following information should be included in a Web Attack Alarm:

- event_name: the name of event: 'SEC_EVENT_WebAttack'
- sub_attack_type: Concret web attack type, e.g., sql injection, command injection, XSS, CSRF
- src_ip: The source IP address of the packet
- dst_ip: The destination IP address of the packet
- src_port: The source port number of the packet
- dst_port: The destination port number of the packet
- src_zone: The source security zone of the packet
- dst_zone: The destination security zone of the packet
- req_method: The method of requirement. For instance, 'PUT' or 'GET' in HTTP

- req_url: Requested URL
- url_category: Matched URL category
- filtering_type: URL filtering type, e.g., Blacklist, Whitelist, User-Defined, Predefined, Malicious Category, Unknown
- rule_id: The ID of the rule being triggered
- rule_name: The name of the rule being triggered
- profile: Security profile that traffic matches.

7.6. NSF Logs

Characteristics:

- acquisition_method: subscription
- emission_type: on-change
- dampening_type: on_repetition

7.6.1. DDoS Logs

Besides the fields in an DDoS Alarm, the following information should be included in a DDoS Logs:

- attack_type: DDoS
- attack_ave_rate: The average pps of the attack traffic within the recorded time
- attack_ave_speed: The average bps of the attack traffic within the recorded time
- attack_pkt_num: The number attack packets within the recorded time
- attack_src_ip: The source IP addresses of attack traffics. If there are a large amount of IP addresses, then pick a certain number of resources according to different rules.
- action: Actions against DDoS attacks, e.g., Allow, Alert, Block, Discard, Declare, Block-ip, Block-service.

7.6.2. Virus Logs

Besides the fields in an Virus Alarm, the following information should be included in a Virus Logs:

- attack_type: Virus
- protocol: The transport layer protocol
- app: The name of the application layer protocol
- times: The time of detecting the virus
- action: The actions dealing with the virus, e.g., alert, block
- os: The OS that the virus will affect, e.g., all, android, ios, unix, windows

7.6.3. Intrusion Logs

Besides the fields in an Intrusion Alarm, the following information should be included in a Intrusion Logs:

- attack_type: Intrusion
- times: The times of intrusions happened in the recorded time
- os: The OS that the intrusion will affect, e.g., all, android, ios, unix, windows
- action: The actions dealing with the intrusions, e.g., e.g., Allow, Alert, Block, Discard, Declare, Block-ip, Block-service
- attack_rate: NUM the pps of attack traffic
- attack_speed: NUM the bps of attack traffic

7.6.4. Botnet Logs

Besides the fields in an Botnet Alarm, the following information should be included in a Botnet Logs:

- attack_type: Botnet
- botnet_pkt_num:The number of the packets sent to or from the detected botnet

- action: The actions dealing with the detected packets, e.g., Allow, Alert, Block, Discard, Declare, Block-ip, Block-service, etc

- os: The OS that the attack aiming at, e.g., all, android, ios, unix, windows, etc.

7.6.5. DPI Logs

DPI Logs provide statistics on uploaded and downloaded files and data, sent and received emails, and alert and block records on websites. It's helpful to learn risky user behaviors and why access to some URLs is blocked or allowed with an alert record.

- type: DPI action types. e.g., File Blocking, Data Filtering, Application Behavior Control

- file_name: The file name

- file_type: The file type

- src_zone: Source security zone of traffic

- dst_zone: Destination security zone of traffic

- src_region: Source region of the traffic

- dst_region: Destination region of the traffic

- src_ip: Source IP address of traffic

- src_user: User who generates traffic

- dst_ip: Destination IP address of traffic

- src_port: Source port of traffic

- dst_port: Destination port of traffic

- protocol: Protocol type of traffic

- app: Application type of traffic

- policy_id: Security policy id that traffic matches

- policy_name: Security policy name that traffic matches

- action: Action defined in the file blocking rule, data filtering rule, or application behavior control rule that traffic matches.

7.6.6. Vulnerabillity Scanning Logs

Vulnerability scanning logs record the victim host and its related vulnerability information that should to be fixed. the following information should be included in the report:

- victim_ip: IP address of the victim host which has vulnerabilities
- vulnerability_id: The vulnerability id
- vulnerability_level: The vulnerability level. e.g., high, middle, low
- OS: The operating system of the victim host
- service: The service which has vulnerabillity in the victim host
- protocol: The protocol type. e.g., TCP, UDP
- port: The port number
- vulnerability_info: The information about the vulnerability
- fix_suggestion: The fix suggestion to the vulnerability.

7.6.7. Web Attack Logs

Besides the fields in an Web Attack Alarm, the following information should be included in a Web Attack Report:

- attack_type: Web Attack
- rsp_code: Response code
- req_clientapp: The client application
- req_cookies: Cookies
- req_host: The domain name of the requested host
- raw_info: The information describing the packet triggering the event.

7.7. NSF Counters

Characteristics:

- acquisition_method: subscription or query
- emission_type: periodical
- dampening_type: none

7.7.1. Firewall counters

Firewall counters provide visibility into traffic signatures, bandwidth usage, and how the configured security and bandwidth policies have been applied.

- src_zone: Source security zone of traffic
- dst_zone: Destination security zone of traffic
- src_region: Source region of the traffic
- dst_region: Destination region of the traffic
- src_ip: Source IP address of traffic
- src_user: User who generates traffic
- dst_ip: Destination IP address of traffic
- src_port: Source port of traffic
- dst_port: Destination port of traffic
- protocol: Protocol type of traffic
- app: Application type of traffic
- policy_id: Security policy id that traffic matches
- policy_name: Security policy name that traffic matches
- in_interface: Inbound interface of traffic
- out_interface: Outbound interface of traffic
- total_traffic: Total traffic volume

- in_traffic_ave_rate: Inbound traffic average rate in pps
- in_traffic_peak_rate: Inbound traffic peak rate in pps
- in_traffic_ave_speed: Inbound traffic average speed in bps
- in_traffic_peak_speed: Inbound traffic peak speed in bps
- out_traffic_ave_rate: Outbound traffic average rate in pps
- out_traffic_peak_rate: Outbound traffic peak rate in pps
- out_traffic_ave_speed: Outbound traffic average speed in bps
- out_traffic_peak_speed: Outbound traffic peak speed in bps.

7.7.2. Policy Hit Counters

Policy Hit Counters record the security policy that traffic matches and its hit count. It can check if policy configurations are correct.

- src_zone: Source security zone of traffic
- dst_zone: Destination security zone of traffic
- src_region: Source region of the traffic
- dst_region: Destination region of the traffic
- src_ip: Source IP address of traffic
- src_user: User who generates traffic
- dst_ip: Destination IP address of traffic
- src_port: Source port of traffic
- dst_port: Destination port of traffic
- protocol: Protocol type of traffic
- app: Application type of traffic
- policy_id: Security policy id that traffic matches
- policy_name: Security policy name that traffic matches

- hit_times: The hit times that the security policy matches the specified traffic.

8. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

9. Security Considerations

The monitoring information of NSF should be protected by the secure communication channel, to ensure its confidentiality and integrity. In another side, the NSF and security controller can all be faked, which lead to undesireable results, i.e., leakage of NSF's important operational information, faked NSF sending false information to mislead security controller. The mutual authentication is essential to protected against this kind of attack. The current mainstream security technologies (i.e., TLS, DTLS, IPSEC, X.509 PKI) can be employed approriately to provide the above security functions.

In addition, to defend against the DDoS attack caused by a lot of NSFs sending massive monitoring information to the security controller, the rate limiting or similar mechanisms should be considered in NSF and security controller, whether in advance or just in the process of DDoS attack.

10. References

10.1. Normative References

[I-D.ietf-core-yang-cbor]
Veillette, M., Pelov, A., Somaraju, A., Turner, R., and A. Minaburo, "CBOR Encoding of Data Modeled with YANG", draft-ietf-core-yang-cbor-07 (work in progress), September 2018.

[I-D.ietf-netconf-subscribed-notifications]
Voit, E., Clemm, A., Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Customized Subscriptions to a Publisher's Event Streams", draft-ietf-netconf-subscribed-notifications-17 (work in progress), September 2018.

[I-D.ietf-netconf-yang-push]
Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "Subscription to YANG Datastores", draft-ietf-netconf-yang-push-20 (work in progress), October 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3877] Chisholm, S. and D. Romascanu, "Alarm Management Information Base (MIB)", RFC 3877, DOI 10.17487/RFC3877, September 2004, <https://www.rfc-editor.org/info/rfc3877>.

[RFC4765] Debar, H., Curry, D., and B. Feinstein, "The Intrusion Detection Message Exchange Format (IDMEF)", RFC 4765, DOI 10.17487/RFC4765, March 2007, <https://www.rfc-editor.org/info/rfc4765>.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <https://www.rfc-editor.org/info/rfc4949>.

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, DOI 10.17487/RFC5424, March 2009, <https://www.rfc-editor.org/info/rfc5424>.

[RFC6587] Gerhards, R. and C. Lonvick, "Transmission of Syslog Messages over TCP", RFC 6587, DOI 10.17487/RFC6587, April 2012, <https://www.rfc-editor.org/info/rfc6587>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/info/rfc7011>.

10.2. Informative References

[I-D.ietf-i2nsf-framework]
Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", draft-ietf-i2nsf-framework-10 (work in progress), November 2017.

[I-D.ietf-i2nsf-terminology]
Hares, S., Strassner, J., Lopez, D., Xia, L., and H. Birkholz, "Interface to Network Security Functions (I2NSF) Terminology", draft-ietf-i2nsf-terminology-06 (work in progress), July 2018.

[I-D.xia-i2nsf-capability-interface-im]
Xia, L., Strassner, J., Li, K., Zhang, D., Lopez, E., Bouthors, N., and L. Fang, "Information Model of Interface to Network Security Functions Capability Interface", draft-xia-i2nsf-capability-interface-im-06 (work in progress), July 2016.

[I-D.xibassnez-i2nsf-capability]
Xia, L., Strassner, J., Basile, C., and D. Lopez, "Information Model of NSFs Capabilities", draft-xibassnez-i2nsf-capability-02 (work in progress), July 2017.

[RFC3954] Claise, B., Ed., "Cisco Systems NetFlow Services Export Version 9", RFC 3954, DOI 10.17487/RFC3954, October 2004, <https://www.rfc-editor.org/info/rfc3954>.

Acknowledgements

Authors' Addresses

Liang Xia
Huawei

Email: frank.xialiang@huawei.com

Dacheng Zhang
Huawei

Email: dacheng.zhang@huawei.com

Yi Wu
Aliababa Group

Email: anren.wy@alibaba-inc.com

Rakesh Kumar
Juniper Networks

Email: rkkumar@juniper.net

Anil Lohiya
Juniper Networks

Email: alohiya@juniper.net

Henk Birkholz
Fraunhofer SIT

Email: henk.birkholz@sit.fraunhofer.de