

Description

Title of the invention: METHOD AND SYSTEM FOR MONITORING AND MANAGING DATA TRAFFIC

BACKGROUND

[0001] Electronic and IT devices are used daily by millions of users around the world for the purposes of communicating and sharing information. Communication between these devices is generally facilitated by a communications network, such as the Internet. Communication between devices is generally based on the well-known seven-layer Open Systems Interconnection (OSI) model, which defines the functions of the different protocol layers without specifying the layer protocols themselves. The seven layers of the OSI model, sometimes referred to here as layer 7 to layer 1, are the application, presentation, session, transport, network, data link and physical layers respectively.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Figure 1 illustrates a communication environment implementing a data packet analysis system, according to an example of the present invention;

[0003] Figure 2 illustrates the system diagram, according to an example of the present invention;

[0004] Figure 3 shows another system diagram, according to an example of the present invention;

[0005] Figure 4 illustrates implementation of the system according to the invention,

[0006] Figure 5 illustrates a representation of the system according to the invention on the hardware side;

[0007] Figure 6 illustrates a method of analysing data traffic according to the invention,

[0008] Figure 7 provides a detailed illustration of the method for monitoring and managing data traffic according to the invention,

[0009] Figure 8 illustrates the steps involved in managing digital fingerprints according to the invention;

[0010] Figure 9 illustrates a case of use of the method according to the invention,

[0011] Figure 10 schematically illustrates the emptying and storing step of the process according to the invention, and

[0012] Figure 11 shows a combined view of implementation of the method trigger in the system according to the invention.

DETAILED DESCRIPTION

[0013] Technological developments in communications networks have led to more and more IT devices being connected to each other, resulting in an increase in network traffic. To accommodate the ever-increasing capabilities of IT peripherals, network capabilities are also constantly being expanded. The improved speeds provided by networks have now exceeded network supervision and traffic management capabilities.

[0014] Known network supervision and traffic management equipment, such as network probes, are specialised hardware devices designed to monitor a data stream received via a communications network. With changes in network capabilities leading to an increase in data traffic and an increase in capabilities of IT peripherals, known network probes are not only failing to provide adequate services for monitoring data and managing traffic, but are also failing to provide other much-needed enhanced capabilities, such as data traffic visualisation, monitoring and maintenance of a minimum quality of service (QoS), and security against cyber threats.

[0015] Therefore, whenever network capabilities change, an increase in the influx of data traffic and/or any new measures for monitoring and supervising data traffic stream are identified, known network probes need to be redesigned to modify their processing capabilities in order to handle the increased influx of data traffic and enable the saving of network data of interest, also known as packet capture.

[0016] Known technologies include Tcpdump, the most widely used technology today for capturing and saving network traffic.

[0017] However, Tcpdump technology fails to capture all the packets received when the data rate is too high, resulting in substantial loss of information. The recorded streams are corrupted by what might be termed “half-streams” or “semi-streams”. These half-streams are streams that have begun to be received by the system before saving begins or that will end after saving has finished. Corruption can also occur if saving ends when the network stream is not yet finished.

[0018] Tcpdump technology also has the disadvantage of not being able to be used at bit rates in excess of 500Mb/sec., and does not allow filtering of data packet

saves based on more than five tuples (source IP address/port number, destination IP address/port number and protocol used).

[0019] FPGA-type technologies with very high-performance filters are also known, allowing filtering of data packet saves based on 5-tuples + or 7-tuples.

[0020] An FPGA has the disadvantage of not being able to filter data packets accurately when the number and diversity of protocols to be classified is too complex (multiplexed protocols, complex protocol sequences, tunnelled protocols). FPGA-based solutions will then tend not to classify protocols that do not announce themselves, increasing the inaccuracy of saving while causing a loss of visibility.

[0021] An FPGA cannot be used for relevant identification and classification of protocols beyond layer 4 of the OSI model. The application of an FPGA is therefore unsuitable for high layers and protocols that do not announce themselves, making it difficult to create certain complex filters using metadata associated with protocols at layers higher than 4, or protocols that do not announce themselves.

[0022] In addition, the lack of memory on FPGA cards will not allow the first packets of a session protocol chain to be recorded when saving is triggered late, resulting in corrupted streams.

[0023] Finally, the inability of FPGA technologies to perform deep packet analysis above layer 5 (application) also implies a truncated filtering capability not allowing application of saving filters to all low-level protocols (only the few that an FPGA is capable of implementing), and to protocols above layer 5.

[0024] In addition, in order to filter accurately the nature of the network packets to be recorded, the traffic must be analysed in real time using an analysis probe. However, probes can also be faced with performance issues that can potentially have a negative impact on the quality of the saves made.

[0025] The packets received can be discarded by many hardware or software components, as soon as an overconsumption of resources occurs. This may occur from the network card, which receives the packets and cannot broadcast them to the kernel, or later, somewhere in the kernel or in the detection software itself.

[0026] A network card can saturate its queue if it is unable to perform DMA (Direct Memory Access) writes as quickly as packets arrive from the network. Apart from possible hardware slowness on the communication buses themselves, the main cause of slowness is the filtering of memory accesses by the IOMMU (I/O Memory Management Unit), which acts as a DMA write manager, capable of limiting the

memory regions to which a server's peripherals are able to write, in the same way as a firewall limits access to a network. Its function is crucial for server security, but completely counter-productive if it results in the inability to fulfil its role as an analysis probe.

[0027] The software architecture of the analysis engines can also influence performance if the packets received are poorly distributed between the different processes responsible for their processing. Packet loss occurs when a processor or CPU is saturated/flooded by the processing that an analysis process is required to undertake. This situation arises very easily when the distribution of packets received is not random, but instead tends to concentrate all the packets relating to the same stream on the same analysis process (e.g. TCP sessions). This stream-based distribution is the most common and preferred, as it limits access to shared resources and increases the locality of memory accesses.

[0028] The disadvantage of this method is that it does not correctly distribute traffic including tunnels (IPsec, GRE, L2TP, TLS...).

[0029] This is because, unless the program in charge of distribution performs deep packet inspection (DPI) of the traffic, all the packets in a tunnel will be analysed by the same analysis process, and if the tunnel is highly active, the analysis process will easily be overwhelmed and packets will start to be lost.

[0030] The invention remedies these drawbacks and improves the situation.

[0031] The present invention relates to the processing of a data stream comprising batches of packets each defined by a chain of communication protocols associated with at least one session.

[0032] According to a general definition of the invention, the method comprises the following steps:

- receiving a plurality of batches of data packets within a predefined period of time via a communication channel and storing said batches of packets in a temporary memory;
- performing, for each batch of data packets, a protocol analysis enabling the communication protocols in the protocol chain to be identified, and validating at least one session associated with the protocol chain when the identification of the protocol chain is complete;
- calculating at least one digital session fingerprint associated with an identified protocol chain, the session of which is validated during the validation step, and

- storing said calculated digital session fingerprint(s) in at least one hash table, and saving a list of the metadata associated with the identified protocols of the protocol chain of the validated session in a knowledge database;
- generating at least one trigger based on the protocol analysis results, wherein activation of the trigger is indicative of a match between at least one rule for identifying target metadata and the metadata associated with the identified protocols listed in the knowledge database;
- analysing the metadata associated with the identified protocols of the data packets of the validated session, stored in the temporary memory means, against the saved list of metadata associated with the identified protocols in the knowledge database, and checking whether the metadata associated with the identified protocols comply with the trigger's rule for identifying target metadata;
- if the data comply with the rule for identifying target metadata, activating the trigger and assigning a save status to at least one digital session fingerprint of the validated session; and
- emptying the temporary memory of each batch of data packets, the digital session fingerprint of which does not have a save status, and saving on a storage memory for subsequent processing all the data packets for which at least one associated digital session fingerprint has a save status.

[0033] The Applicant has observed that the process according to the invention makes it possible to respond to a very specific issue of triggering saving. Memory management using the method according to the invention enables packets belonging to the same session but the identification of which is delayed to be identified and saved, particularly for protocols requiring special identification packets.

[0034] Advantageously, the method according to the invention allows optimised management of the working memory and furthermore allows precise selection of the object of saving, as well as efficient targeting of sessions intricated in chains of tunnelled or multiplexed protocols and thus targeting and saving only the session of interest without spurious data that can be attributed to other transfers of the same data packet or packets.

[0035] Finally, the method according to the invention allows targeted sessions to be saved without loss of information or visibility, by departing from a 5-tuple type approach and adopting an n-tuple type approach, thus advantageously allowing sessions to be saved to be filtered based on any set of extracted metadata.

[0036] According to one embodiment, the trigger activation step comprises a sub-step of updating the hash table saved following the step of saving at least one digital session fingerprint, by assigning a save status to at least one digital session fingerprint.

[0037] In practice, the step of emptying the temporary memory comprises a sub-step of consulting the hash table to check whether at least one of the digital session fingerprints associated with the data packets subject to emptying has a save status.

[0038] In addition, the step of analysing the metadata associated with the identified protocols of the data packets of the validated session comprises a further sub-step of analysing the metadata associated with the attached content MP of the protocol chain of the validated session, implemented if at least one rule for identifying target metadata of the trigger comprises target metadata in relation to the attached content.

[0039] In practice, the trigger is capable of processing and applying a dynamic list of rules for identifying target metadata.

[0040] According to one embodiment of the invention, the target metadata of the trigger's identification rules belong to the group formed by native metadata and metadata calculated from selected mathematical formulae.

[0041] By way of a non-limiting example, the target metadata are representative of selected network parameters belonging to the group formed by destination IP, source IP, destination port, source port, protocol, IP address, port, QoS quality of service parameters, network tag, session volume, packet size, number of retries, version, encryption algorithm type and version, encryption type, CERT (Computer Emergency Response Team) certificate, SNI (Server Name Indication) value, packet size, returned IP, error flag, domain name, client version, server version, encryption algorithm version, compression algorithm, timestamp, IP version, hostname, lease-time, URL, user agent, number of bytes of content attached, content type, status code, cookie header, client name, request service, error code value, request type, protocol value, response timestamp, privilege level, keyboard type and language, product identification, screen size, or any similar specific metadata extracted from the protocols in one or more data packets of the validated session, similar specific metadata extracted from the content attached to one or more data packets of the validated session.

[0042] In practice, the step of analysing the protocols and the step of analysing the metadata of the data packets of the validated session are performed on the data packets of layer 2 to layer 7 of the OSI model.

[0043] When all the data packets are processed, the updated data packets can be resubmitted to the protocol analysis engine. Repeating the protocol analysis can determine whether the processing of all the data packets has actually dealt with the occurrence of the event. If the event is resolved, the updated data packets can be released and made available to the intended recipient. However, if it is determined that another event has occurred during the analysis of the updated data packets, another trigger can be generated and the updated data packets can be subsequently processed to deal with the occurrence of the event.

[0044] As a result, storing the set of data packets for which an event has occurred, when the trigger is generated, facilitates efficient monitoring and visualisation of the data traffic in a network.

[0045] Furthermore, processing the set of data packets, such as replacing them with a set of corrected data packets or modifying the set of data packets, makes it possible to deal with the cause of the occurrence of the event. In this way, anomalies or malfunctions in data traffic can be identified and dealt with in good time. In addition, data packets can be analysed to preventively detect security risks and address related concerns.

[0046] The present invention also relates to the processing of a data stream comprising batches of packets each defined by a chain of communication protocols attached to at least one session.

[0047] According to another general definition of the invention, the system according to the invention comprises:

- network interface means configured to receive a data stream from a communication channel;
- a processor comprising at least one processing core for processing a predetermined number of data packets per minute ppm;
- a temporary memory, coupled to the processor, capable of storing a plurality of batches of data packets from the network interface means;
- a protocol analysis engine executable on at least one processing core, wherein the protocol analysis engine is capable of:
 - receiving a plurality of batches of data packets via a communication channel;

performing, for each batch of data packets, a protocol analysis enabling the communication protocols in the protocol chain to be identified, and validating at least one associated session;

calculating at least one digital session fingerprint associated with an identified protocol chain, the session of which is validated, and

storing the calculated digital session fingerprint in at least one hash table, and saving a list of the metadata associated with the identified protocols of the protocol chain of the validated session in a knowledge database;

-a monitoring engine executable on at least one processing core, wherein the monitoring engine is capable of:

generating at least one trigger based on the protocol analysis results, wherein activation of the trigger is indicative of a match between at least one rule for identifying target metadata and the list of metadata associated with the identified protocols of the protocol chain of the validated session;

analysing the metadata of the data packets of the validated session, stored in the temporary memory means, against the saved list of metadata associated with the identified protocols in the knowledge database, and checking whether the metadata associated with the identified protocols comply with the trigger's rule for identifying target metadata;

if the data comply with the rule for identifying target metadata, activating the trigger and assigning a save status to at least one digital session fingerprint of the validated session; and

emptying the temporary memory means for each batch of data packets, the digital session fingerprint(s) of which do not have a save status; and

-a storage memory coupled to the processor capable of saving, for subsequent processing, all the data packets, the associated digital session fingerprint of which has a save status.

[0048] According to one embodiment of the invention, the monitoring engine is configured to update the hash table saved following the step of saving at least one digital session fingerprint, by assigning a save status to at least one digital session fingerprint in case of activation of the trigger.

[0049] In addition, the monitoring engine according to the invention is configured to consult the hash table and check whether the digital session fingerprint(s)

associated with the data packets, the session of which is validated, have a save status.

[0050] According to one particular embodiment of the invention, the monitoring engine is configured to analyse metadata associated with the attached content of the protocol chain of the validated session when analysing the metadata associated with the identified protocols of the data packets of the validated session, if at least one rule for identifying target metadata of the trigger comprises target metadata in relation to the attached content.

[0051] In practice, the temporary memory means are gradually emptied when the use of the associated RAM is between 95% and 98%.

[0052] By way of a non-limiting example, the temporary memory means are emptied chronologically by deleting the oldest data packets at a chosen emptying rate.

[0053] The above techniques are described in greater detail with reference to Figures 1 to 11. It should be noted that the description and numbers merely illustrate the principles of this object and the examples described in this document and should not be interpreted as limited to this object. It is therefore understood that a variety of arrangements may be devised which, although not explicitly described or shown in this document and in the following statements outlining principles, aspects and implementations of this object, together with specific examples thereof, are intended to encompass equivalents thereof.

[0054] Figure 1 illustrates a communication environment 100 implementing a data packet analysis system 102.

[0055] With reference to Figures 1 to 5, the system 102 according to the invention firstly comprises network interface means NIC configured to receive a data stream from a communication channel 106.

[0056] In practice, the network interface means NIC are coupled to the processor 202 and to the temporary memory 108. The network interface card NIC may be an integrated component of the system 102 or a separate component externally coupled to the system 102.

[0057] In an exemplary embodiment of this object, the system 102 may be implemented by a network service provider offering network connectivity to one or more subscribers, such as government organisations, multinational corporations, companies, businesses and other institutions. The network service provider can act

as a connection channel between the communication channel 106 and its subscribers' IT devices. It should be noted that besides the system 102, the network service provider may deploy various other equipment items, devices, network nodes interconnected by one or more wired or wireless network links to provide network connectivity to subscribers. Network nodes may typically include switches, routers, access points and data links capable of facilitating communication between various subscriber hosts (e.g. server computers, client computers, mobile devices, etc.) that can generate and consume data traffic.

[0058] In another example, the system 102 may be deployed by individual institutions/organisations, providing network connectivity and security to one or more of its users.

[0059] In addition, the system 102 can be deployed as a standalone hardware peripheral or can be deployed in known communications equipment, depending on deployment and use.

[0060] As described above, the system 102 can receive data packets 104 from a communication channel 106.

[0061] In practice, the communication channel 106 can be a wireless or wired network, or a combination thereof. The communications network can be a set of individual networks, interconnected with each other and functioning as a single large network.

[0062] Examples of such individual networks include, but are not limited to, the Global System for Mobile Communications (GSM) network, the Universal Mobile Telecommunications System (UMTS) network, the Long Term Evolution (LTE) network, the Personal Communications Services (PCS) network, the Time Division Multiple Access (TDMA) network, the Code Division Multiple Access (CDMA) network, the Next Generation Network (NGN), the Public Switched Telephone Network (PSTN) and the Integrated Services Digital Network (ISDN). According to the terminology, the communications network comprises various network entities, such as gateways, switches and routers; however, these details have been omitted for the sake of brevity of the description.

[0063] In an exemplary implementation, the network interface card NIC can separate the data stream within a predefined period into one or more processing queues and allocate each processing queue to at least one processing core 204-1, 204-2, 204-3, ..., 204-n.

[0064] For example, the network interface card NIC can separate the data stream in multiple processing queues according to the segregation criterion, which may be predefined or based on user preference.

[0065] It should be noted that each of the multiple processing cores 204-1, 204-2, 204-3, ..., 204-n may allow different instances of the multiple engines to be executed thereon. For example, separate instances of the protocol analysis engine 206, the monitoring engine 306 and the correction engine 308 may be executed in parallel on each of the multiple processing cores 204-1, 204-2, 204-3, ..., 204-n.

[0066] However, for ease of understanding, implementation of the system according to the invention has been explained with respect to a single processing core 204-1 of the processor 202, hereinafter referred to as the processing core 204.

[0067] The system 102 according to the invention furthermore comprises at least one processor 202 having at least one processing core 204 for processing a predetermined number of data packets per minute ppm.

[0068] In practice, the processor 202 may include several processing cores 204-1, 204-2, 204-3, ..., 204-n.

[0069] The functions of the functional block labelled as “processor(s)” can be provided through the use of dedicated hardware as well as hardware capable of executing instructions. When provided by a processor, functions may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. Furthermore, use of the term “processor” should not be construed as referring exclusively to the hardware capable of executing instructions and may implicitly include, without limitation, the Digital Signal Processor (DSP) hardware, the network processor, the Application-Specific Integrated Circuit (ASIC), the Read Only Memory (ROM) for storing the instructions, the temporary memory 108 (RAM), and non-volatile storage. Other hardware, standard and/or customised, may also be included.

[0070] The system 102 furthermore comprises a temporary memory 108, coupled to the processor 202, capable of storing a plurality of batches of data packets 104 from the network interface means NIC.

[0071] According to one embodiment of the invention, the system 102 can monitor and manage the data packets 104. The system 102 can first store the data packets 104 from a predefined period in the temporary memory 108.

[0072] By way of a non-limiting example, the temporary memory 108 can be implemented on a local memory, an external memory or a combination thereof, using registers, primary memories, cache memories or secondary memories, and can be implemented on any computer-readable medium, including, for example, volatile memory (e.g. RAM) and/or non-volatile memory (e.g. EPROM, flash memory, etc.).

[0073] The temporary memory 108 can provide data packets 110 for protocol and data analysis. It would be appreciated if a data traffic stream received from the communication channel 106 within the predetermined time period, such as the data packets 104, could be separated into one or more different processing queues before being saved on the temporary memory 108. The segregation of the data packets 104 into one or more processing queues can be based on a segregation criterion, which can be predefined or based on user preferences.

[0074] The system 102 may furthermore include the engines 302, wherein the engines 302 may include a protocol analysis engine 206, a monitoring engine 306.

[0075] For example, the engines 302 can be implemented as a combination of hardware and firmware. In the examples described here, such combinations of hardware and firmware can be deployed in a number of different ways. For example, the engine firmware may consist of executable processor instructions stored on a non-transitory machine-readable storage medium and the engine hardware may include a processing resource (e.g. implemented as a single processor or a combination of several processors) to execute these instructions.

[0076] In the present examples, the machine-readable storage medium may store instructions which, when executed by the processing resource, implement the engine functionalities. In such examples, the system 102 may include the machine-readable storage medium storing the instructions and the processing resource for executing the instructions. In other examples of this object, the machine-readable storage medium may be situated at a different location, accessible however to the system 102 and the processor 202.

[0077] The system 102 also includes data 304, which serve, among other aspects, as a repository for storing data that can be retrieved, processed, received or generated by the protocol analysis engine 206, the monitoring engine 306 and the correction engine 308. The data 304 may include the protocol analysis data 310, the event data 312, the trigger data 314 and other data 316. In one example, the data 304 can be stored in the memory 108.

[0078] The protocol (or communication session) analysis engine 206 and data analysis engine (hereinafter referred to as the analysis engine 206) of the system according to the invention is capable of being executed on at least one processing core 204, and is configured to allow the reception S0 of a plurality of batches of data packets 104 within a predefined time period via a communication channel 106.

[0079] In practice, the data packets 104 provided for the protocol chain analysis engine 206 may correspond to a processing queue of the temporary memory 108 and the data packets in other processing queues (not illustrated) may be separately accessible from the temporary memory 108 for the protocol chain analysis engine 112.

[0080] The protocol analysis engine 206 is furthermore capable of implementing a protocol and data analysis of the data packets 104, and is configured to perform S10, for each batch of data packets 104, a protocol analysis DAPD making it possible to identify the communication protocols in the protocol chain, and to validate at least one associated session.

[0081] The analysis DAPD of the data packets 104 can allow determination of the communication protocol(s) of the data packets, in addition to analysis and extraction of metadata from the analysed data packets. The protocol analysis engine 206 can perform an analysis DAPD of the data packets 104 in a number of ways.

[0082] In one example, the protocol analysis engine 206 can perform an explicit classification analysis to check whether an identified protocol does not announce the following protocol(s).

[0083] The protocol analysis engine (206) can also perform detection by session classification, making it possible to check whether a conclusion can be drawn about the nature of the protocol being analysed from the sequence of protocols already identified.

[0084] Finally, the protocol analysis engine 206 can also perform Deep Packet Inspection (DPI) detection on the data packets. In this example, the protocol analysis engine 206 can perform DPI on layers 2 to 7 of the OSI (Open Systems Interconnection) model.

[0085] Subsequently, the protocol analysis engine 206 can perform data packet analysis and extraction for the data packets based on the identified communication protocols.

[0086] In addition, the protocol analysis engine 206 is configured to calculate S21 at least one digital session fingerprint HS associated with an identified protocol chain PID, the session of which is validated, according to a chosen hash function.

[0087] In practice, the protocol analysis engine 206 is configured to store S22 the calculated digital session fingerprint HS in at least one hash table TH, and generate and save a list LM of the metadata MPID associated with the identified protocols PID of the protocol chain of the validated session in a knowledge database BDDS.

[0088] The system 102 according to the invention furthermore comprises the monitoring engine 306 executable on at least one processing core 204, which engine is capable of generating S30 at least one trigger 114 based on the results of protocol analysis DAPD, wherein the activation of the trigger 114 is indicative of the match between at least one rule for identifying target metadata MC, and the list LM of the metadata MPID associated with the identified protocols PID of the protocol chain of the validated session.

[0089] In practice, the trigger 114 thus generated is capable of processing and applying a dynamic list of rules for identifying target metadata MC.

[0090] The target metadata MC of the trigger's 114 identification rules belong to the group formed by native metadata and metadata calculated from selected mathematical formulae.

[0091] By way of a non-limiting example, the target metadata (MC) are representative of selected network parameters belonging to the group formed by destination IP, source IP, destination port, source port, protocol, IP address, port, QoS quality of service parameters, network tag, session volume, packet size, number of retries, version, encryption algorithm type and version, encryption type, CERT (Computer Emergency Response Team) certificate, SNI (Server Name Indication) value, packet size, returned IP, error flag, domain name, client version, server version, encryption algorithm version, compression algorithm, timestamp, IP version, hostname, lease-time, URL, user agent, number of bytes of content attached, content type, status code, cookie header, client name, request service, error code value, request type, protocol value, response timestamp, privilege level, keyboard type and language, product identification, screen size, or any similar specific metadata extracted from the protocols in one or more data packets of the validated session, similar specific metadata extracted from the content attached to one or more data packets of the validated session.

[0092] According to one embodiment of the invention, the occurrence of an event, such as a deterioration in QoS, a change in the configuration of the IT peripherals, a change in the network capabilities, a security risk or the like can influence the nature of the rules applied for identifying the target metadata MC.

[0093] In one example, the event data 312 can store a set of parameters that can be used as a reference for identifying the occurrence of an event during the analysis S10 of the protocol chains of the set of data packets and thus participate in updating the rules applied for identifying the target metadata MC.

[0094] In addition to identifying the occurrence of events, the set of parameters can also indicate a user-defined criterion for filtering data packets. For example, a network administrator may be aware that data packets received from a particular IP address may pose a security risk. In such a situation, the set of parameters stored in the event data 312 can be updated to identify and filter the set of data packets received from the particular IP address. Similarly, the set of parameters can include a list of predetermined protocols, sessions and other factors that can increase network vulnerability.

[0095] The monitoring engine 306 according to the invention is furthermore configured to analyse S40 the metadata of the data packets of the validated session, stored in the temporary memory means 108, against the saved list LM of metadata MPID associated with the identified protocols PID in the knowledge database BDDS, and to check whether the metadata MPID associated with the identified protocols PID comply with the trigger's 114 rule for identifying target metadata MC.

[0096] The read list LM is based on n-tuple metadata extraction.

[0097] Advantageously, this n-tuple approach makes it possible to extract associated metadata MPID specific to the identified protocol PID, thereby overcoming the constraints involved in the 5-tuple approach and the restricted visibility associated with it.

[0098] The following metadata are extracted from the following identified protocols PID, by way of non-limiting examples:

PROTOCOLS	EXAMPLES OF EXTRACTED METADATA
VLAN and VxLAN	Network tag
MPLS	Id
TCP	Session volume, packet size, number of retries

TLS	version, type and version of the encryption algorithm, CERT of the certificate, SNI value
DNS	Packet size, returned IP, error flag, domain name, rcode
SSH	Client version, server version, encryption algorithm version, compression algorithm version
DHCP	Timestamp, IP version, server IP addresses, endpoint IP address, originating port/protocol - TCP or UDP - server or endpoint hostname, lease-time,...
HTTP	Url, user agent, number of payload bytes, content type, status code, cookie header, etc.
Kerberos	client name, service requested, error code value, request type, protocol value, response timestamp, privilege level, encryption type, ...
LDAP	session duration, number of logon errors, end-of-session flag, query result code, error code, etc.
RDP	cookie username, keyboard type and language, client version, product ID, screen size, ...

Table of examples of extractable metadata for a list of selected protocols.

[0099] In other words, the saved list LM of metadata MPID associated with the identified protocols PID serves as a reading grid for said monitoring engine 306 in order to advantageously save resources.

[00100] In the event of compliance with the rule(s) for identifying target metadata MC, the trigger 114 is activated S50 and the monitoring engine 306 then assigns a save status SV to at least one digital session fingerprint HS of the validated session stored on the hash table TH.

[00101] The assignment of a save status SV by the monitoring engine 306 comprises the updating S51 of the hash table TH saved following the step of storing S22 by the analysis engine 206 of at least one digital session fingerprint HS, by assigning a save status SV to at least one of the digital session fingerprints HS of the validated session and storing this status on the hash table TH.

[00102] According to one particular embodiment of the invention, the monitoring engine 306 is configured to analyse metadata associated with the attached content MP of the protocol chain of the validated session during the analysis S40 of the metadata MPID associated with the identified protocols PID of the data packets of the validated session.

[00103] Such an analysis is implemented if at least one rule for identifying target metadata MC of the trigger 114 includes target metadata MC related to an attached content.

[00104] The monitoring engine 306 according to the invention is furthermore configured to implement emptying S60 of the temporary memory means 108 for each batch of data packets, the digital session fingerprint(s) HS of which do not have a save status SV.

[00105] According to one embodiment of the invention, the temporary memory means 108 are gradually emptied when the use of the associated RAM is between 95% and 98%.

[00106] According to one particular embodiment of the invention, the temporary memory means 108 are emptied chronologically by deleting the oldest data packets at a chosen emptying rate.

[00107] In practice, the monitoring engine 306 is configured to consult S61 the hash table TH and check whether the digital session fingerprint(s) HS associated with the data packets TH, the session of which is validated, have a save status SV.

[00108] The monitoring engine 306 according to the invention is furthermore configured to implement saving S61 on a storage memory 116, of all the data packets for which at least one associated digital session fingerprint HS has a save status SV.

[00109] According to a first particular embodiment of the invention, based on the generation S30 of the trigger 114, the monitoring engine 306 can generate a notification indicating the severity of the event. The monitoring engine 306 can also include in the notification a set of actions that can be carried out in order to mitigate the event.

[00110] According to a second particular embodiment of the invention, based on the generation S30 of the trigger 114 for the set of data packets, the system according to the invention may comprise a correction engine 308.

[00111] Said correction engine 308 is capable of performing an additional processing operation on the set of data packets. For instance, in one example, the correction engine 308 can replace the set of data packets with a set of corrected data packets. In another example, the correction engine 308 can modify all the data packets.

[00112] Once the correction engine 308 has processed all the data packets, the protocol analysis engine 206 can resubmit the data packets for an analysis DAPD. Repeating the analysis can determine whether the processing of the set of data packets has actually dealt with or mitigated the occurrence of the event. The way in which the analysis DAPD can be performed on the data packets is described above and has not been reproduced for the sake of brevity. If the protocol analysis engine 206 determines that the event has been resolved, the protocol analysis engine 206 may release the updated data packets for consumption by a destination IT peripheral.

[00113] However, if the protocol analysis engine 206 determines that another event has occurred during the DAPD of the updated data packets, the protocol analysis engine 206 may generate another trigger. In such a situation, the protocol analysis engine 206 can reprocess another set of data packets that may have caused the other event. In one example, reprocessing may include replacing the set of data packets with a set of corrected data packets or modifying the set of data packets.

[00114] In this way, anomalies or malfunctions in data traffic can be identified and dealt with in good time. In addition, data packets can be analysed to preventively detect security risks and address related concerns.

[00115] The system 102 according to the invention furthermore comprises a storage memory 116 coupled to the processor 202 capable of saving 508 for subsequent processing, all the data packets, the associated digital session fingerprint HS of which has a save status SV, the trigger 114 thus forming a saving filter.

[00116] For instance, the storage memory 116 may consist of local, or remote, storage means, such as the Cloud.

[00117] Furthermore, the system 102 can be coupled to a data packet knowledge database BDDS, wherein the data packet knowledge database BDDS corresponds to the knowledge base BDC for protocol detection and the dynamic session database BDDS for session detection.

[00118] In an exemplary implementation of the system according to the invention, the network interface card NIC can receive a data stream of data packets 104 via a communications network. The network interface card NIC can subsequently separate the data stream of data packets 104 into multiple processing queues, wherein each of the multiple processing queues is to be processed by a separate

processing core of the processor 202. The network interface card NIC can subsequently transfer the multiple processing queues to the temporary memory 108 and can simultaneously inform the processor 202 of the address at which each of the multiple processing queues is stored on the temporary memory 108.

[00119] In an exemplary implementation, one of the instances of the protocol analysis engine 206 running on one of the processing cores, for example the processing core 204-1, can perform a DAPD on the data packets included in the corresponding processing queue. In one example, the protocol analysis engine 206 can analyse a first set of data packets in the processing queue to determine the characteristics of the first set of data packets. The protocol analysis engine 206 can subsequently perform the DAPD for a first data packet following the first set of data packets to predict a communication protocol for the data packet. The protocol analysis engine 206 can perform the DAPD on the first data packet based on a dynamic decision-making tree correlated to the data packet knowledge database BDDS. The protocol analysis engine 206 can subsequently analyse and extract the first data packet according to the data label associated with it.

[00120] In one example, the protocol analysis engine 206 can perform DAPD and data analysis and extraction in parallel for the data packets included in the processing queue. For example, once the protocol analysis engine 206 has predicted the communication protocol for the first data packet, the first data packet can be transferred for data analysis and extraction. While the protocol analysis engine 206 is analysing and extracting data on the first data packet, the protocol analysis engine 206 can simultaneously begin performing DAPD for a second data packet following the first data packet in the data stream.

[00121] The simultaneous execution of DAPD and data analysis and extraction for data packets on a single processor core facilitates scalability to handle any alterations to the influx of data received via the communications network. In one example, the techniques described above can facilitate the processing of data streams with data traffic in excess of 100 Gbps.

[00122] With reference to Figures 6 to 11, the invention also relates to a method of processing of a data stream comprising batches of packets 104 each defined by a chain of communication protocols associated with at least one session.

[00123] The method according to the invention comprises a first step of receiving S0 a plurality of batches of data packets 104 within a predetermined time via a

communication channel 106, and storing said batches of packets 104 in a temporary memory 108.

[00124] For instance, the plurality of batches of data packets 104 may be received by a network interface card NIC of the system 102.

[00125] The method also includes a step S10 of protocol analysis DAPD making it possible to identify the communication protocols in the protocol chain, and for extracting metadata from the plurality of data packets analysed.

[00126] The step S10 of protocol analysis DAPD consists of a succession of conditional classification methods, including explicit classification detection, configured to check whether a given protocol announces the next protocol in the protocol chain.

[00127] The protocol analysis DAPD furthermore comprises a so-called session detection configured to identify the next protocol(s) using the previously identified protocol chains.

[00128] Finally, the protocol analysis DAPD comprises deep packet inspection detection, which involves identifying the communication protocol of the next packet according to a dynamic decision-making tree correlated to a knowledge database BDC comprising protocol analysis parameters and a database of specific markers, also called labels, specific to each known protocol.

[00129] If the next protocol is not identified, the protocol analysis DAPD is configured to issue a list of potential candidate protocols to be taken into account according to at least two possible detection branches, each detection being attached to a sub-session determined to analyse the next protocol or protocols by repeating the analyses according to the classification by explicit detection, by session detection, and by deep packet inspection until at least one protocol, the identity of which is certain is identified.

[00130] In practice, the step S10 of protocol analysis DAPD comprises a sub-step of validating S11 at least one session associated with the protocol chain when the entire protocol chain has been identified and classified.

[00131] A session means a communication session representative of a sequence of protocols.

[00132] Validation S11 of a session means complete identification of the associated protocol chain.

[00133] The method according to the invention furthermore includes a step of calculating S21 at least one digital session fingerprint HS associated with an identified protocol chain PID, the session of which is validated during the validation step S11.

[00134] In practice, calculation S21 of the digital session fingerprint HS is performed after each protocol analysed in the protocol chain according to a chosen hash function.

[00135] Calculation S21 of the digital fingerprint HS is based on the extraction of metadata from each protocol identified during the analysis step S10, and the hashing of a dynamically defined list specific to each type of protocol, adopting an n-tuple approach. Each protocol will therefore have a defined number of parameters used to calculate digital session fingerprint HS, so that once the session has been validated, a completely unique session identifier can be obtained.

[00136] The step of calculation S21 is followed by a step S22 of storing said calculated digital session fingerprint(s) HS in at least one hash table TH, as well as saving a list LM of the metadata MPID associated with the identified protocols PID of the protocol chain of the validated session in a knowledge database BDDS.

[00137] Advantageously, the list LM of metadata MPID associated with the identified protocols PID in the protocol chain of the validated session makes it possible to identify the metadata on which any subsequent analysis can be based and thus save resources by concentrating only on information that is available with certainty.

[00138] The method according to the invention also comprises a step of generating S30 at least one trigger 114 based on the results of protocol analysis DAPD.

[00139] Said trigger 114 comprises a series of commands executed by activation according to selected conditions, wherein activation is indicative of a match between at least one rule for identifying target metadata MC and the metadata MPID associated with the identified protocols PID, the list LM of which is saved in the knowledge database BDDS.

[00140] According to one embodiment of the invention, the trigger 114 is capable of processing and applying a dynamic list of rules for identifying target metadata MC.

[00141] Target metadata MC are any metadata, native or calculated from selected mathematical formulae, capable of enabling precise selection of part of a data stream 104.

[00142] By way of a non-limiting example, the target metadata (MC) are representative of selected network parameters belonging to the group formed by destination IP, source IP, destination port, source port, protocol, IP address, port, QoS quality of service parameters, network tag, session volume, packet size, number of retries, version, encryption algorithm type and version, encryption type, CERT (Computer Emergency Response Team) certificate, SNI (Server Name Indication) value, packet size, returned IP, error flag, domain name, client version, server version, encryption algorithm version, compression algorithm, timestamp, IP version, hostname, lease-time, URL, user agent, number of bytes of content attached, content type, status code, cookie header, client name, request service, error code value, request type, protocol value, response timestamp, privilege level, keyboard type and language, product identification, screen size, or any similar specific metadata extracted from the protocols in one or more data packets of the validated session, similar specific metadata extracted from the content attached to one or more data packets of the validated session.

[00143] According to one particular embodiment of the invention, the target metadata MC may comprise metadata associated with the attached content MP of the protocol chain of the validated session.

[00144] According to one embodiment of the invention, the trigger 114 can be generated following the occurrence of an event during the step S10 of protocol analysis DAPD.

[00145] Furthermore, an event may include: a deterioration in the quality of service, a change in the configuration of the IT peripherals, a change in the network capabilities or a security risk, for which a trigger will be generated specifically according to the parameters of the event encountered.

[00146] The method according to the invention furthermore comprises a step of analysis S40 of the metadata MPID associated with the identified protocols PID of the data packets of the validated session, stored in the temporary memory means 108 as a function of the saved list LM of metadata MPID associated with the identified protocols PID in the knowledge database BDDS.

[00147] In other words, the saved list LM of metadata MPID associated with the identified protocols PID serves as a reading grid during analysis S40 of the metadata MPID associated with the identified protocols PID.

[00148] In practice, the step of analysis S40 of the metadata MPID associated with the identified protocols PID furthermore includes checking the presence or absence of a match between the metadata MPID associated with the identified protocols PID and at least one rule for identifying target metadata MC of the trigger 114.

[00149] If the data comply with the rule for identifying target metadata MC, the method according to the invention comprises a step of activation S50 of the trigger 114, allowing assignment of a save status SV to at least one saved digital session fingerprint HS of the validated session.

[00150] In practice, the step of activation S50 of the trigger 114 comprises a sub-step of updating S51 the hash table TH saved following the step of saving S22 at least one digital session fingerprint HS, by assigning a save status SV to at least one digital session fingerprint HS, the target metadata MC of which match the analysed metadata.

[00151] According to one embodiment of the invention, the step of analysis S40 the metadata associated with the identified protocols PID of the data packets of the validated session comprises a further sub-step of analysing the metadata associated with the attached content MP of the protocol chain of the validated session, implemented if at least one rule for identifying target metadata MC of the trigger 114 comprises target metadata MC in relation to the attached content MP.

[00152] The method according to the invention furthermore comprises a step of emptying the temporary memory 108.

[00153] The step of emptying S60 is conditional, and configured to empty S60 the temporary memory 108 of each stored batch of packets of data, the digital session fingerprint HS of which does not have a save status SV.

[00154] At the same time, the process furthermore comprises a sub-step of storing S61 all the data packets for which at least one associated digital session fingerprint HS has a save status SV.

[00155] Such a selective storing step S61 also enables the specific saving of the session of interest, the target metadata MC or associated rules of the trigger 114 of

which have been recognised, and indeed with no constraint as to the diversity and the layer on which the protocols of the validated session are located.

[00156] In practice, the storing step S61 is carried out to a storage memory 116 for subsequent processing of the filtered data packets.

[00157] By way of a non-limiting example, the data packets saved in the storage memory 116 are converted to PCAP format.

[00158] According to one embodiment of the invention, the steps of emptying S60 the temporary memory 108 and storing S61 comprise a prior sub-step of consulting S61 the hash table TH to check whether at least one of the digital session fingerprints HS associated with the data packets subject to emptying S60 has a save status SV.

[00159] In practice, the step S10 of protocol analysis DAPD and the step of analysing S40 the metadata of the data packets of the validated session MPID are carried out on the data packets of layer 2 to layer 7 of the OSI model.

[00160] According to a particular embodiment of the invention, the set of data packets saved following the storing step S61 can be replaced by a corrected set of data packets in the plurality of data packets. In one example, the set of data packets can be replaced by the set of data packets corrected by a correction engine 308 of the system 102.

[00161] For instance, the corrected set of data packets is analysed once again using the method according to the invention, in order to determine whether the replacement of the set of data packets by the corrected set of data packets has effectively resolved the occurrence of the event.

[00162] It can be understood that steps S10 to S60 can be carried out in the system 102. The blocks of methods S0, S10, S11, S21, S22, S30, S40, S50, S51, S60, S61 and S62 can be executed based on instructions stored in a non-transient computer-readable medium, as will be readily understood. The non-transient computer-readable medium may include, for example, digital memories, magnetic storage media, such as magnetic disks and tapes, hard disks or optically readable digital data storage media.

[00163] Advantageously, such a process makes it possible to use all the metadata from all the protocols to generate triggers 114, the identification rules of which form saving filters, making said filters versatile and the architecture for applying the filters adaptable to any needs, and also making it possible to save traffic very accurately.

[00164] Although examples of the present invention have been described in language specific to the methods and/or structural features, it should be understood that the present invention is not limited to the specific methods or features described. Instead, the specific methods and features are disclosed and explained by way of examples of the present invention.

Claims

[Claim 1] A method of processing of a data stream comprising batches of packets (104) each defined by a chain of communication protocols associated with at least one session, characterised in that it comprises the following steps:

- receiving (S0) a plurality of batches of data packets (104) via a communication channel (106) and storing said batches of packets (104) in a temporary memory (108);
- performing (S10), for each batch of data packets (104), a protocol analysis (DAPD) enabling the communication protocols in the protocol chain to be identified, and validating (S11) at least one session associated with the protocol chain when the identification of the protocol chain is complete;
- Calculating (S21) at least one digital session fingerprint (HS) associated with an identified protocol chain (PID), the session of which is validated during the validation step (S11), and
- storing (S22) said calculated digital session fingerprint(s) (HS) in at least one hash table (TH), and saving a list (LM) of the metadata (MPID) associated with the identified protocols (PID) of the protocol chain of the validated session in a knowledge database (BDDS);
- generating (S30) at least one trigger (114) based on the protocol analysis results (DAPD), wherein activation of the trigger (114) is indicative of a match between at least one rule for identifying target metadata (MC) and the metadata (MPID) associated with the identified protocols (PID), the list (LM) of which is saved in the knowledge database (BDDS);
- analysing (S40) the metadata (MPID) associated with the identified protocols (PID) of the data packets of the validated session, stored in the temporary memory means (108), against the saved list (LM) of metadata (MPID) associated with the identified protocols (PID) in the knowledge database (BDDS), and checking whether the metadata (MPID) associated with the identified protocols (PID) comply with the trigger's (114) rule for identifying target metadata (MC);

- if the data comply with the rule for identifying target metadata (MC), activating (S50) the trigger (114) and assigning a save status (SV) to at least one digital session fingerprint (HS) of the validated session; and
- emptying (S60) the temporary memory (108) of each batch of data packets, the digital session fingerprint of which does not have a save status (SV), and saving (S61) on a storage memory (116) for subsequent processing all the data packets for which at least one associated digital session fingerprint (HS) has a save status (SV).

[Claim 2] The method according to claim 1, wherein the step of activation (S50) of the trigger (114) comprises a sub-step of updating (S51) the hash table (TH) saved following the step of saving (S22) at least one digital session fingerprint (HS), by assigning a save status (SV) to at least one digital session fingerprint (HS).

[Claim 3] The method according to claim 1 or 2, wherein the step of emptying (S60) the temporary memory (108) comprises a sub-step of consulting (S61) the hash table (TH) to check whether at least one of the digital session fingerprints (HS) associated with the data packets subject to emptying (S60) has a save status (SV).

[Claim 4] The method according to any of claims 1 to 3, characterised in that in the step of analysis (S40) of the metadata associated with the identified protocols (PID) of the data packets of the validated session comprises a further sub-step of analysing the metadata associated with the attached content (MP) of the protocol chain of the validated session, implemented if at least one rule for identifying target metadata (MC) of the trigger (114) comprises target metadata (MC) in relation to the attached content (MP).

[Claim 5] The method according to any of claims 1 to 4, characterised in that in the trigger (114) is capable of processing and applying a dynamic list of rules for identifying target metadata (MC).

[Claim 6] The method according to any of claims 1 to 5, characterised in that the target metadata (MC) of the trigger's (114) identification rules belong to the group

formed by native metadata and metadata calculated from selected mathematical formulae.

[Claim 7] The method according to any of claims 1 to 6, wherein the target metadata (MC) are representative of selected network parameters belonging to the group formed by destination IP, source IP, destination port, source port, protocol, IP address, port, QoS quality of service parameters, network tag, session volume, packet size, number of retries, version, encryption algorithm type and version, encryption type, CERT (Computer Emergency Response Team) certificate, SNI (Server Name Indication) value, packet size, returned IP, error flag, domain name, client version, server version, encryption algorithm version, compression algorithm, timestamp, IP version, hostname, lease-time, URL, user agent, number of bytes of content attached, content type, status code, cookie header, client name, request service, error code value, request type, protocol value, response timestamp, privilege level, keyboard type and language, product identification, screen size, or any similar specific metadata extracted from the protocols in one or more data packets of the validated session, similar specific metadata extracted from the content attached to one or more data packets of the validated session.

[Claim 8] The method according to any of claims 1 to 7, wherein the step (S10) of protocol analysis (DAPD) and the step of analysing (S40) the metadata of the data packets of the validated session (MPID) are carried out on the data packets of layer 2 to layer 7 of the OSI model.

[Claim 9] A system (102) for processing of a data stream comprising batches of packets each defined by a chain of communication protocols associated with at least one session, characterised in that it comprises

- network interface means (NIC) configured to receive a data stream from a communication channel (106);
- a processor (202) comprising at least one processing core (204) for processing a predetermined number of data packets per (minute ppm);
- a temporary memory (108), coupled to the processor (202), capable of storing a plurality of batches of data packets (104) from the network interface means (NIC);

- a protocol analysis engine (206) executable on at least one processing core (204), wherein the protocol analysis engine (206) is capable of:
 - receiving (S0) a plurality of batches of data packets (104) within a predetermined time via a communication channel (106);
 - performing (S10), for each batch of data packets, a protocol analysis (DAPD) enabling the communication protocols in the protocol chain to be identified, and validating at least one associated session;
 - calculating (S21) at least one digital session fingerprint (HS) associated with an identified protocol chain (PID), the session of which is validated, and
 - storing (S22) said calculated digital session fingerprint (HS) in at least one hash table (TH), and saving a list (LM) of the metadata (MPID) associated with the identified protocols (PID) of the protocol chain of the validated session in a knowledge database (BDDS);
- a monitoring engine (306) executable on at least one processing core (204), wherein the monitoring engine (306) is capable of:
 - generating (S30) at least one trigger (114) based on the protocol analysis results (DAPD), wherein activation of the trigger (114) is indicative of a match between at least one rule for identifying target metadata (MC) and the list (LM) of metadata (MPID) associated with the identified protocols (PID) of the protocol chain of the validated session;
 - analysing (S40) the metadata of the data packets of the validated session, stored in the temporary memory means (108), against the saved list (LM) of metadata (MPID) associated with the identified protocols (PID) in the knowledge database (BDDS), and checking whether the metadata (MPID) associated with the identified protocols (PID) comply with the trigger's (114) rule for identifying target metadata (MC);
 - if the data comply with the rule for identifying target metadata (MC), activating (S50) the trigger (114) and assigning a save status (SV) to at least one digital session fingerprint (HS) of the validated session; and

emptying (S50) the temporary memory means (108) for each batch of data packets, the digital session fingerprint(s) (HS) of which do not have a save status (SV); and

-a storage memory (116) coupled to the processor (202) capable of saving, (508) for subsequent processing, all the data packets, the associated digital session fingerprint (HS) of which has a save status (SV).

[Claim 10] System (102) according to claim 9, characterised in that the monitoring engine (306) is configured to update (S51) the hash table (TH) saved following the step of saving (S22) at least one digital session fingerprint (HS), by assigning a save status (SV) to at least one digital session fingerprint (HS) in case of activation (S50) of the trigger (114).

[Claim 11] System (102) according to claim 9 or 10, characterised in that in the monitoring engine (306) is configured to consult (S61) the hash table (TH) and check whether the digital session fingerprint(s) (HS) associated with the data packets, the session of which is validated, have a save status (SV).

[Claim 12] System (102) according to any of claims 9 to 11, characterised in that in the monitoring engine (306) is configured to analyse metadata associated with the attached content (MP) of the protocol chain of the validated session during the analysis (S40) of the metadata (MPID) associated with the identified protocols (PID) of the data packets of the validated session, if at least one rule for identifying target metadata (MC) of the trigger (114) comprises target metadata (MC) in relation to the attached content.

[Claim 13] System (102) according to any of claims 9 to 12, characterised in that in the temporary memory means (108) are gradually emptied when the use of the associated RAM is between 95% and 98%.

[Claim 14] System (102) according to claims 9 to 13, characterised in that the temporary memory means (108) are emptied chronologically by deleting the oldest data packets at a chosen emptying rate. |