(19) NL Octrooicentrum

(11) **2007180**

(12)C **OCTROOI**

(21) Aanvraagnummer: **2007180**

(22) Aanvraag ingediend: **26.07.2011**

(51) Int.Cl.:
H04L 29/06 (2006.01)

(43) Aanvraag gepubliceerd:
-

(47) Octrooi verleend:
29.01.2013

(45) Octrooischrift uitgegeven:
06.02.2013

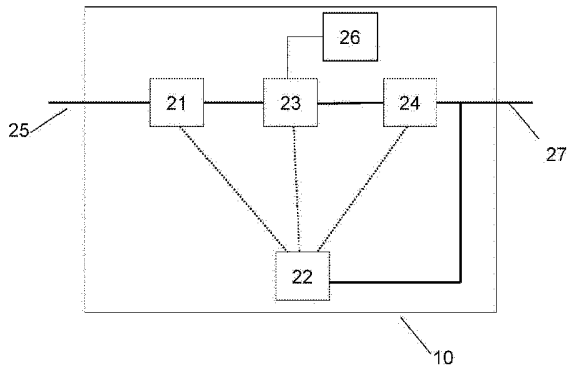
(73) Octrooihouder(s):
Security Matters B.V. te Enschede.

(72) Uitvinder(s):
Emmanuele Zambon te Enschede.

(74) Gemachtigde:
Ir. H.V. Mertens c.s. te Rijswijk.

(54) **Method and system for classifying a protocol message in a data communication network.**

(57) An intrusion detection method for detecting an intrusion in data traffic on a data communication network, the method comprising parsing the data traffic to extract at least one protocol field of a protocol message of the data traffic, associating the extracted protocol field with a model for that protocol field, the model being selected from a set of models, assessing if a contents of the extracted protocol field is in a safe region as defined by the model, and generating an intrusion detection signal in case it is established that the contents of the extracted protocol field is outside the safe region. The set of models may comprise a corresponding model for each protocol field of a set of protocol fields.



NL C 2007180

P30657NL00/HSE

Title: Method and system for classifying a protocol message in a data communication network.

Field of the invention

The invention relates to the field of data communication networks, in particular to the field of classifying messages in data communication networks, for example to detect malicious intrusions in such data communication networks.

Background art

In many data communication networks, detection systems are deployed to detect malicious intrusions. Such intrusions comprise data from attackers or infected computers that may affect the working of servers, computers or other equipment.

There are two main types of such intrusion detection systems: signature-based and anomaly-based intrusion detection systems.

A signature-based intrusion detection system (SBS) relies on pattern-matching techniques. The system contains a database of signatures, i.e. sequences of data, that are known from attacks of the past. These signatures are matched against the tested data. When a match is found, an alert is raised. The database of signatures needs to be updated by experts after a new attack has been identified.

Differently, an anomaly-based intrusion detection system (ABS) first builds a statistical model describing the normal network traffic during a so-called “learning phase”. Then, during a so-called “testing phase” the system then analyses data and classifies any traffic or action that significantly deviates from the model, as an attack. The advantage of an anomaly-based system is that it can detect zero-day attacks, i.e. attacks that not yet have been identified as such by experts.

However, in some data communication networks malicious data is very similar to legitimate data. This may be the case in a so called SCADA (Supervisory Control and Data Acquisition) network. In a SCADA network, protocol messages are exchanged between computers, servers and other equipment on an application layer of the data communication network. These protocol messages may comprise instructions to control machines. A protocol message with a malicious instruction (“set rotational speed at 100 rpm”) may be very similar to a legitimate instruction (“set rotational speed at 10 rpm”).

When the malicious data is very similar to legitimate data, the malicious data may be classified as normal or legitimate data by the anomaly-based intrusion detection system,

which could endanger the working of computers, servers and other equipment in the network.

Summary of the invention

An object of the invention may be to provide an improved intrusion detection system and/or method.

In accordance with an aspect of the invention, there is provided an intrusion detection method for detecting an intrusion in data traffic on a data communication network, the method comprising:

- parsing the data traffic to extract at least one protocol field of a protocol message of the data traffic;
- associating the extracted protocol field with a respective model for that protocol field, the model being selected from a set of models;
- assessing if a contents of the extracted protocol field is in a safe region as defined by the model; and
- generating an intrusion detection signal in case it is established that the contents of the extracted protocol field is outside the safe region.

Parsing the data traffic allows to distinguish individual fields of a protocol (referred to as "protocol fields") in accordance with which data communication over the data network takes place. An association is then made (if successful) between the field ("the protocol field") and a model. Thereto, a set of models is provided. A suitable model for the extracted protocol field is selected, as will be explained in more detail below. The protocol field is then assessed using the model in order to establish if the contents of the protocol field is in a normal, safe, acceptable range or not. In the latter case, a suitable action may be performed. By parsing the protocol message, the data traffic individual protocol fields may be distinguished, and a suitable model for assessment of that particular protocol field, may be selected. Thereby, an adequate assessment can be made, as different protocol fields may be assessed applying different models, for example each protocol field applying a respective model that is tailored to that specific protocol field, for example applying a model that is tailored to the protocol field type and/or contents. The intrusion detection method in accordance with the invention may be a computer implemented intrusion detection method. The parser (i.e. the parsing) may make use of a predefined protocol specification. Also, for example in case the protocol is unknown, the protocol may be learnt by monitoring the data traffic on the network and deriving a protocol specification therefrom.

In the context of this document, the term protocol may be understood as a set of rules that defines a content of some or all of the messages transmitted via the data network. A

network protocol may comprise a definition of protocol messages, also known as Protocol Data Units (PDUs). A protocol message (PDU) may in turn comprise one or more fields. There may be many type of fields. A field may comprise either another PDU, or an “atomic” data entity (for example a number, a string or a binary opaque object). As will be described in more detail below, the network protocol may be organized as a tree, in which nodes are PDUs and leaves of the tree are atomic data entities (fields). For each field (or each relevant field) a separate model may be provided. As an example, assume a protocol message comprises personal data of a person (comprising for example name, address and personal settings): a protocol message that transmits the personal data, could then comprise the fields “name”, “address”, and “personal settings”. The field “name” could for example in turn comprise the fields “surname”, “given name”, “login name”, etc. The field “address” could for example comprise the fields “home address” and “business address”. The fields “home address may for example comprise “home address street”, “home address number”, “home address zip code”, “home address city”, while the “business address” may for example comprise the fields “business address street”, “business address number”, “business address zip code”, “business address city”, etc.. A separate model may be built for each field. For example a separate, respective model could be provided for each one of the fields. In an embodiment, a same model may be applied for a subset of fields, for example the fields “business address city” and “home address city” may apply a same model.

The term data traffic may be understood so as to comprise any data that is communicated via the network, such as a data steam, data packets, etc. The term data network may be understood so as to comprise any data communication establishment that allows a transmission of (e.g. digital) data. The network may comprise or be connected to a public network such as the Internet, and/or may comprise a private network or virtually private network to which only authorized users or authorized equipment is allowed access. Transmission may take place via a wired connection, a glass fiber connection, a wireless connection and/or any other connection. The term model may be understood so as to comprise a rule or set of rules that apply to a protocol field, in order to assess that protocol field. The model may describe normal, legitimate or non-intrusive protocol messages. It may be understood that the more protocol messages in the learning phase are used, the better the model may describe the normal, legitimate or non-intrusive protocol messages.

The term intrusion may be understood so as to comprise any data which may be undesired, possibly harmful to a computer system that receives the data, possibly harmful to an application running on a computer system connected to the data network, or possibly

harmful to an operation of a device, installation, apparatus, etc connected to the data network.

In an embodiment, the set of models comprises a respective model for each protocol field of a set of protocol fields. Thereby, more accurate results may be obtained as for each protocol field, a model specifically tailored to that protocol field, may be applied.

In an embodiment, the set of models comprises two models for one protocol field, a specific one of the two models for the one protocol field being chosen based on the value of another field, so as to possibly further increase a precision of the models.

Similarly, time sequence analysis on the protocol field may be performed in an embodiment wherein the set of models comprises at least two models for one protocol field, a first one of the two models being associated with a first time interval in which the data traffic is observed, and a second one of the models being associated with a second time interval in which the data traffic is observed, the second time interval e.g. non overlapping with the first time interval.

In an embodiment, the model for the field being determined in a learning phase, the learning phase comprising:

- parsing the data traffic to extract at least one protocol field of the protocol applied in the data traffic;
- associating the extracted protocol field with the model for that protocol field, the model being selected from the set of models and
- updating the model for the extracted protocol field using a contents of the extracted protocol field.

Thus, the data traffic may be observed in a learning phase, and the contents of the extracted protocol fields may be applied to update the corresponding models with which the protocol fields are associated. If no association can be made between the extracted protocol field and one of the models, a new model may be created for the extracted protocol field and added to the set of models.

Hence, two phases may be discriminated: a learning phase in which a model of protocol messages is built. These protocol messages in the learning phase may be constructed on the basis of the communication protocol or may be retrieved from data traffic in the data communication network.

Since protocol messages may be described by their structure and the values of the protocol fields, the model may relate to the protocol fields in the learning phase and the values thereof. Different protocol fields in the learning phase may have a different data type, i.e. their value may be an number (such as an integer, a floating point number, etc), a

string., a Boolean or a binary value. This may be defined by the communication protocol. The model may be built in accordance with the data type of the at least one protocol field.

The determined protocol field and/or the determined value of said protocol field are compared with the model and classified on the basis of the comparison. The protocol message may be classified as an anomaly (and thus as a possible danger) on the basis of the comparison.

In the learning phase, the protocol messages that are applied to learn the model, may be obtained from data traffic on the network. Alternatively, or in addition thereto, simulation data may be applied. In the learning phase, possibly intrusive protocol messages may be distinguished by statistical methods, i.e. infrequently used protocol messages or protocol messages having an uncommon contents, may be removed before using the protocol messages for learning the model(s). Additionally, or instead thereof, an operator may identify certain protocol messages as intrusive, and such protocol messages may either be removed before the learning, or the models being corrected accordingly.

Alternatives for learning (i.e. training) the model(s), other than in the above described learning phase may be applied. For example, a model may be derived from inspecting the protocol and the application, creating a set of for example to be expected protocol messages, their fields and/or the values of the fields, there from, and building the model, or a set of models there from. Also, a combination of such building model(s) from inspection, with a learning of the model(s) may be applied: for example first learning the model(s) in a learning phase, and then adapting the learned model(s) based on knowledge of a known behaviour and consequential occurrence and/or contents of protocol messages, their fields and/or the values of the fields.

In an embodiment, the intrusion detection signal is further generated when the parsing cannot establish the field as complying to the protocol, so that an action can be performed also in case a field which is incompliant with the protocol (for example a malformed protocol message) is detected.

In an embodiment, the intrusion detection signal is further generated when the extracted field cannot be associated with any of the models of the set of models, so that an action can be performed also in case the extracted field possibly complies with the protocol, but for which no suitable model is provided. Often, only a subset of the possible protocol fields are used, for example in control applications, allowing for example to raise an alert when a protocol field which complies with the protocol but which is normally not applied, has been retrieved.

The method may be applied on a variety of protocol layers. For example, the protocol may be at least one of an application layer protocol, a session layer protocol, a transport

layer protocol or even lower levels of a network protocol stack. An application layer of a data communication network may be defined by the Open Systems Interconnection model (OSI model), which was determined by the International Organization for Standardization. In the application layer, software running on computers or servers may communicate to each other by sending protocol messages. The protocol messages may be SCADA protocol messages.

The communication between software may follow a certain communication protocol, in which the structure and possible values of (parts of) the protocol messages are defined. The structure of a protocol message may be further described by the protocol fields in the protocol messages. The software may not be able to process protocol messages that are not in accordance with the communication protocol.

In an embodiment, in response to generating the intrusion detection signal, the method further comprises at least one of:

- removing the protocol field or a data packet containing the protocol field; and
- raising and outputting an intrusion alert message. Any other intrusion detection action may be applied, such as for example isolating the protocol field or a data packet containing the protocol field, etc

In an embodiment, the model for the protocol field comprises at least one of

- a set of acceptable protocol field values, and
- a definition of a range of acceptable protocol field values. In case the protocol field comprises a numerical value, a simple model may be provided thereby that may allow to test the protocol field at a low data processing load.

In an embodiment, the model for the protocol field comprises

- a definition of acceptable letters, digits, symbols, and scripts. In case the protocol field comprises a character or string, a simple model may be provided thereby that may allow to test the protocol field at a low data processing load.

In an embodiment, the model for the protocol field comprises a set of predefined intrusion signatures, so that knowledge about known attacks may be taken into account. A combination of a model as described above (comprising e.g. a set of acceptable protocol field values, a definition of a range of acceptable protocol field values, a definition of acceptable letters, digits, symbols, and scripts) with the set of predefined intrusion signatures may be highly effective, as for each specific field a model of its normal contents in combination with one or more specific intrusion signatures for that field, may be applied.

In an embodiment, the protocol comprises primitive protocol fields and composite protocol fields, the composite protocol fields in turn comprising at least one primitive protocol field, wherein a respective model is provided in the set of models for each primitive

protocol field. Thus, efficient intrusion detection may be provided as protocol fields that are composite (i.e. protocol fields that themselves comprise protocol fields, such as “address” comprising “street name”, “number’, “zip code” and city”), may be split up in their elementary (primitive) protocol fields, allowing to apply a suitable model to each of the primitive protocol fields.

Since the model for the at least one protocol field in the learning phase and/or for the value of the at least one protocol field in the learning phase may be built in accordance with the data type of the at least one protocol field in the learning phase, the model may be more accurate in describing normal, legitimate or non-intrusive protocol messages than a model that does not take into account the data type of the protocol fields.

It may be the case that a model optimized for describing a protocol field with a number data type may be less accurate in (or not applicable for) describing a protocol field with a string or binary data type. Likewise, a model optimized for describing a protocol field with a string data type may be less accurate in describing a protocol field with a number or binary data type. Therefore, the accuracy of the model may be improved by taken the data type of the protocol field into account when building the model.

According to another aspect of the invention, there is provided an intrusion detection system for detecting an intrusion in data traffic on a data communication network, the system comprising:

- a parser for parsing the data traffic to extract at least one protocol field of a protocol message of the data traffic;
- an engine for associating the extracted protocol field with a respective model for that protocol field, the model being selected from a set of models;
- a model handler for assessing if a contents of the extracted protocol field is in a safe region as defined by the model; and
- an actuator for generating an intrusion detection signal in case it is established that the contents of the extracted protocol field is outside the safe region.

With the system according to the invention, the same or similar effects may be achieved as with the method according to the invention. Also, the same or similar embodiments may be provided as described with reference to the method according to the invention, achieving the same or similar effects. The parser, engine, model handler and actuator may be implemented by means of suitable software instructions to be executed by a data processing device. They may be implemented in a same software program that is to be executed by a same data processing device, or may be executed at two or more different data processing devices. For example, the parser may be executed locally at a location where the data traffic passes, while the engine, model handler and actuator may be

located remotely, for example at a safe location. Also, data from different sites may be monitored, whereby for example a parser may be provided at each site, output data from each parser being sent to a single engine, model handler and actuator.

Brief description of the figures

Further effects and features of the invention will be described, by way of example only, with reference to the below description and accompanying schematic drawings in which non limiting embodiments are disclosed, wherein:

Figure 1 schematically depicts an example of a data communication network comprising an intrusion detection system according to an embodiment of the invention;

Figure 2 schematically depicts an overview of an intrusion detection system according to an embodiment of the invention;

Figure 3 schematically depicts an overview of a learning phase of a method according to an embodiment of the invention; and

Figure 4 schematically depicts an overview of an intrusion detection phase of a method according to an embodiment of the invention.

Detailed description of the invention

In figure 1 a schematic overview is depicted of an example of a data communication network with an intrusion detection system for classifying a protocol message according to an embodiment of the invention. In this network personal computers (or workstations) 14 and 15 are connected with a server 13. The network may be connected to the internet 16 via a firewall 17.

In the data communication network an intrusion or an attack may originate from the Internet 16 or from a personal computer 14, when it has been infected with malicious software.

The data communication network may be a SCADA (Supervisory Control and Data Acquisition) network. In such a network, a machinery 12 may be controlled by software running on a remote terminal unit (RTU) 11, or on a programmable logic controller (PLC). Software running on the server 13 may send protocol messages to the software running on the RTU 11. The software on the RTU 11 may send protocol messages to the machinery, on which also software may be running.

A user may communicate with server 13 via software running on the personal computer 14 or work station 15 by exchanging protocol messages between the software running on the personal computer 14 or work station 15 and the software running on server 13.

The intrusion detection system 10 may be positioned between the RTU 11 and a remainder of the network, as is shown in figure 1, or between the RTU 11 and the machinery 12 (not shown). The intrusion detection system 10 may retrieve protocol messages from the data communication network, that may be exchanged between the software running on the personal computer 14 or work station 15 and the software running on server 13, between the software running on server 13 and the software running on RTU 11 or between the software running on RTU 11 and software running on a data processing device of the machinery 12.

A communication protocol may be defined as a formal description of digital protocol message formats and the rules for exchanging those messages in or between (software running on) computing systems. The communication protocol may include descriptions for syntax, semantics, and synchronization of communication. Protocol messages on an application layer in a data communication network may contain one or more fields, which can be characterized by their data types. For instance, a field can represent the entire length of a message, with a number value or a string value.

With more information about the protocol messages, a model describing normal, legitimate or non-intrusive protocol message may include more information about the normal or legitimate values of each protocol field of each protocol message that is exchanged in the data communication network. The model may then be used (e.g. real time) to classify protocol messages from live data traffic in data communication network in order to find anomalies, i.e. something that deviates from the normal behavior of the data communication network as it is described by the model.

Figure 2 shows a schematic overview of an embodiment of an intrusion detection system 10 according to an embodiment of the invention. The intrusion detection system 10 comprises a network protocol parser 21, arranged for retrieving at least one protocol field in a protocol message in (for example) an application layer of the data communication network. In the learning phase, the protocol messages may be obtained from the network via input 25. The network protocol parser 21 may be used during an optional learning phase as well as during regular operation of the intrusion detection system. Information about the extracted protocol message may be transferred to engine 23.

The intrusion detection system further comprises engine 23, a set of models 26 and a model handler 24. The engine 23 is arranged to associate the extracted protocol field with a model. Thereto, the engine comprises or has access to a set of models 26. The engine associates the extracted protocol field with a model that is specific for that protocol field. Thereto, the set of models 26 comprises different models, each model for a specific one (or more) of the protocol fields. In a learning phase, the engine may, in case no model is

available yet for the extracted protocol field, create a model for the extracted protocol field and add it to the set of models. Information about the extracted protocol field may be transferred to handler 24.

The handler 24 then makes an assessment whether or not the extracted protocol field conforms to the model, so as to asses if the contents of the extracted protocol field may be considered an intrusion or not. In the learning phase, the model may be updated using the contents of the extracted protocol field. The handler may output the messages via output 27.

The intrusion detection system may further comprise an actuator 22 to generate an intrusion detection signal in case the protocol field has been identified as an intrusion. In response to generating the intrusion detection signal, an intrusion detection action may be performed e.g. comprising raising an alert, filtering the data packet or protocol field (thereby e.g. removing the data packet or protocol field). The intrusion detection signal may also be generated in case the parser could not identify the protocol field (which would imply that the data packet is incompliant with the protocol), and/or in case the model handler during intrusion detection operation could not associate the extracted protocol field with a model from the set (which would imply that the data packet does not comprise the protocol fields that are normally transmitted).

For each protocol field, a specific model is used, preferably using a different model for each different protocol field, so that for a most optimal assessment may be performed for each protocol field, as a model that is specifically dedicated to that protocol field, may be used for assessment of that protocol field.

In an embodiment, the models have been built using at least two model types, wherein a first model type of the at least two model types is optimized for a protocol field with a first data type and wherein a second model type of the at least two model types is optimized for a protocol field with a second data type. It may be the case that the first model type is optimized for a protocol field with one of an number data type, a string data type or a binary data type and the second model type is optimized for a protocol field with another of a number data type, a string data type or a binary data type.

For example, for the value of a protocol field A1 with a number data type, model M-I-A1 may be built that is intended for describing number values. For the value of a protocol field A2 with an number data type, model M-I-A2 may be built that is likewise intended for describing number values. For the value of a protocol field A3 with a string data type, model M-S-A3 may be built that is optimized for describing string values. The models for different protocol fields that have the same data type, for example models M-I-A1 and M-I-A2, may be built using the same model architecture, but having a different contents (e.g. a different

allowable range, different set of allowable values, etc) so as to express the differences between the protocol fields A1 and A2.

It may be understood that a model with a model type for describing number values and a model with a model type describing string values may be better or more accurate in describing the values of a protocol message comprising both number values and string values in its protocol fields, than a single model that would be optimized for describing all values, both number values and string values, of a protocol message.

The intrusion detection system 10 may be arranged for building a model during a learning phase. The working of the intrusion detection system 10 and method according to embodiments of the invention will further be described with reference to figures 3 and 4. Figure 3 schematically illustrates the learning phase and figure 4 schematically illustrates the intrusion detection phase.

In figure 3, steps of the learning phase have been schematically depicted: Step a1: parsing the data traffic to extract at least one protocol field of a protocol applied in the data traffic. Step a2: associating the extracted protocol field with the model for that protocol field, the model being selected from the set of models,
Step a3: in case no association can be made with the existing models of the set of models, creating a new model for the extracted protocol field and adding the new model to the set of models. Step a4: updating the model for the extracted protocol field using the contents of the extracted protocol field.

In general, a protocol message may comprise primitive protocol fields and composite protocol fields. A composite protocol field comprises two or more sub protocol fields, which may each be a primitive protocol field or a composite protocol field. A primitive protocol field can not be divided or split into more protocol fields. In this way a protocol message can be said to comprise a tree structure of protocol fields. For example, in a protocol message the composite protocol field “msg_body” comprise of a primitive protocol field “msg_len” and composite protocol field “msg_data”. The composite protocol field “msg_data” may comprise primitive protocol fields “msg_typeA” and “msg_typeB”. The term protocol field in this document may refer to any primitive protocol field at any level of such a tree structure.

Different model types may be used. For example, a model type of the protocol field may for example be one of: a number model type, a string model type or a binary model type. In case it is found that the extracted protocol field comprises a number value, a number model type may be applied for that protocol field. In case it is found that the extracted protocol field comprises a string value, a string model type may be applied for that protocol field. It may be the case that, when in the learning phase the network protocol parser is unable to

establish that the data type of the protocol field is a number data type or a string data type, a binary data type model is applied as a more universal model type.

As explained above, the set of models may comprise a respective model for each protocol field. A model for a protocol field with a number data type may be differently built (i.e. may be of a different kind or having a different model architecture) than a model for a protocol field with a string data type. Since the models may be optimized for each data type, the model may be more accurate in describing normal, legitimate or non-intrusive protocol messages than models that do not take into account the data type of the protocol fields.

Examples of different kind of model types for different kinds of data types are explained below. For the number data type two model types may be applied, a first one for protocol fields representing lengths and a second one for protocol fields representing enumerations.

If the protocol field represents an enumeration, the model may comprise a set S with all values of the protocol field that have been retrieved in the learning phase. After starting with an empty set, during the learning phase each value that is identified for the protocol field may be added to the set. In the intrusion detection phase, a protocol message may be classified as anomalous, when the value of the corresponding determined protocol field is for example not part of set S.

If the protocol field represents a length, the model may be built on a approximation of the distribution of the values of the protocol field during the learning phase. During the learning phase, the mean μ and the variance σ^2 of the approximation of the distribution may be calculated on the basis of the sample mean and the sample variance of all the values that have been determined as a content of that protocol field. With the mean μ and the variance σ^2 of the approximation of the distribution, a probability may be calculated for all values. During the intrusion detection phase, when the probability of a determined value of the protocol field is smaller than a given threshold, the protocol message with this value may be classified anomalous.

A model for a Boolean type protocol field may for example monitor a Boolean value averaged over a number of samples and compare the averaged value to a predetermined threshold. An example of such a model is described below:

During the learning phase a probability Pt is computed that a value of the field is true, and a probability Pf (1-Pt) is computed that the value of the field is false.

2 - During the intrusion detection a sequence of n samples for the field value is considered and then compute a binomial probability of observing such a sequence of values, given Pt and Pf. We then compare the probability with a certain threshold t and raise an alert if p_sample < t. For example, suppose that during the learning phase we observe an equal amount of true and false values. Then Pt ~ 1/2 and Pf ~ 1/2. We set a probability threshold

for sequences of 5 values to 0.1. Now, consider that during the intrusion detection phase we observe the sequence [false, false, false, false, false]. The binomial probability of p_sample = P(true=0) = 0.03125 < 0.1. In this case we raise an alert. An example of a model type for strings that can handle ASCII and Unicode strings, is described below. First, a model type for ASCII strings is described.

The model type for ASCII string comprises two Boolean values and a list. The first Boolean value (*letters*) is set to true if we have seen letters, the second Boolean value (*digits*) is set to true if we have seen digits, and the set (*symbols*) keeps track of all the symbols we have seen. Given a string field s, a function $f(s)$ is defined that tells whether the string contains letters, numbers and which symbols. For example for the string "userName?#!" we have:

$$f(\text{"userName?\#!"}) = \begin{Bmatrix} \boldsymbol{letters}: & \boldsymbol{true} \\ \boldsymbol{digits}: & \boldsymbol{false} \\ \boldsymbol{symbols}: & \{!, \#, ?\} \end{Bmatrix}$$

During the learning phase, given a string s the model M is updated as follows:

$$M = \begin{Bmatrix} letters: & M.letters \vee f(s).letters \\ digits: & M.digits \vee f(s).digits \\ symbols: & M.symbols \cup f(s).symbols \end{Bmatrix}$$

The string characters are evaluated one after the other. For each character the engine verifies the type, and in case the character is either a letter or a digit, the engine updates the model accordingly by setting the corresponding flag to "true". In case the current character is a symbol, it is added to the current symbol set. In case the symbol is already present, it is not added twice.

During the intrusion detection phase, given a string s, an alert may be raised if:

$$\begin{gathered} (f(s).letters \wedge \neg M.letters) \vee \\ (f(s).digits \wedge \neg M.digits) \vee \\ (f(s).symbols \not\subseteq M.symbols) \end{gathered}$$

The string characters are again evaluated one after the other. The verification process is straightforward. If the current character is either a letter (or a digit), the engine verifies that letter characters (or digits) have been observed before for the given field. When this verification fails, an alert is raised. In case the character is a symbol, the engine verifies that the given symbol has been observed before. When this verification fails, an alert is raised.

At a beginning, the model M is defined as follows:

$$M = \begin{Bmatrix} letters: & false \\ digits: & false \\ symbols: & \emptyset \end{Bmatrix}$$

Another example of a model type for stings, as may be used for Unicode strings, is described below, For Unicode strings, the modeling and detection technique may be similar to the modeling for ASCII strings. The Unicode characters that are not ASCII are treated as ASCII letters, i.e. if a string contains a Unicode character, the boolean value “letters” is set to true. In addition the set of the Unicode scripts (e.g. latin, cyrillic, arabic) as seen during the learning phase, is memorized. With this additional information it is detected, for example, if strange Unicode characters (that probably belongs to a different script than the one seen in the learning phase) are present in a string.

In some more detail, given a Unicode string field s, we define a function f’(s) that tells whether the string contains letters, numbers, which symbols and which Unicode scripts. For example, for the string ”mu3sòafà?#!” we have:

$$f'(\text{"mu3sòafà?\#!"}) = \begin{Bmatrix} \boldsymbol{letters}: & \boldsymbol{true} \\ \boldsymbol{digits}: & \boldsymbol{false} \\ \boldsymbol{symbols}: & \{!, \#, ?\} \\ \boldsymbol{scripts}: & \{\boldsymbol{latin}\} \end{Bmatrix}$$

For Unicode strings the model M is initialised and updated by performing the same or similar operations as for ASCII strings and by handling the additional field “scripts”, similarly to the field “symbols”.

Some further example of a model type for binary protocol fields is provided below:

For the binary data type a model may be applied from known anomaly-based intrusion detection systems based on an analysis of the payload.

An example of binary model is based on 1-gram analysis. An n-gram in a sequence of n consecutive bytes.

Given a binary field b of length l bytes, we first compute a vector f containing the relative frequency of each byte. In other words, given a byte value v, the element of f corresponding to v is given by:

$$\bar{\boldsymbol{f}}[\boldsymbol{v}] = \frac{\sum_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{l}} \boldsymbol{1}, \boldsymbol{if}\ \boldsymbol{b}[\boldsymbol{i}] = \boldsymbol{v}}{\boldsymbol{l}}$$

During the learning phase, a vector of relative frequencies is applied to compute a mean and standard deviation for each byte value. Therefore, given a sequence of n binary fields b1..bn, and their associated vectors of relative byte frequency (f1..fn), two vectors μ and σ are computed that contain respectively the mean and standard deviation of each byte value (from 0 to 255). These two vectors in this example form the binary model.

During the testing phase, given a binary field value s, an associated vector of relative frequencies fs is computed first. Then, an appropriate function F (e.g. a normalised Euclidean distance) is applied to determine a distance between fs and the model as built during learning phase. If the resulting distance exceeds a predetermined threshold, an alert may be raised.

A more accurate version of the model described above may be obtained by splitting the set of learning values b1..bn into subsets. To split the learning set into subsets a clustering algorithm may be applied, such as a Self Organizing Map (SOM), on the input values (b1..bn). A separate model (i.e. the array pair μ, σ) may then be built for each subset.

During the intrusion detection phase, a cluster algorithm is run on the binary field value (s). The test as described above may then be applied on the model associated to the resulting cluster.

A third example of a binary model is a so-called network emulator. A network emulator is an algorithm that is able to determine if dangerous executable instructions are contained inside a set of bytes. Given a sequence of bytes, the algorithm first translates existing byte values into the relative assembly instructions (disassembly). Afterwards, it tries to find sequences of instructions that can be recognised as dangerous or suspicious (for example long sequences of NOP instructions, which are typically found inside malicious shell codes of known attacks). In case such sequences are found, an alert is raised. Note that this type of binary model does not require a training phase.

Furthemore, for the string data type a model may be applied as is described in "Bolzoni, D. and Etalle, S. (2008), Boosting Web Intrusion Detection Systems by Inferring Positive Signatures. In: Confederated International Conferences On the Move to Meaningful Internet Systems (OTM)". For the binary data type a sub-model may be applied from known anomaly-based intrusion detection systems based on the analysis of the payload. An example may be found in “Anomalous payload-based network intrusion detection” (RAID, pages 203-222, 2004) by Ke Wang and Salvatore J. Stolfo. In this work the authors present a system, named PAYL, which leverages n-gram analysis to detect anomalies. An n-gram in a sequence of n consecutives bytes. The relative frequency and standard deviation of 1-grams (sequences of 1 byte) are analyzed and stored into detection models built during the

learning phase. Then, in the intrusion detection phase, an appropriate model is selected (using the payload length value) and used to compare the incoming traffic.

Another example may be found in “POSEIDON: a 2-tier Anomaly-based Network” (IWIA, pages 144–156. IEEE Computer Society, 2006) by Damiano Bolzoni, Emmanuele Zambon, Sandro Etalle, and Pieter Hartel. In this paper the authors build on the top of PAYL an improved system by discarding the payload length to select (and build) the detection models, but use instead a neural network that pre-process the payload data and whose output is used to select the appropriate detection mode.

A still further example may be found in Michalis Polychronakis, Kostas G. Anagnostakis, and Evangelos P. Markatos. Comprehensive Shellcode Detection using Runtime Heuristics. In Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC). December 2010, Austin, TX, USA. In this paper the authors present a “network emulator”. This software component implements heuristics and simulates via software a physical CPU. The network emulator can test whether the input data contains executable (and harmful) code.In an embodiment, the parsing process may comprises the steps of:

i) collecting data packets from the data communication network;

ii) defragmenting IP packets;

iii) reassembling TCP segments;

iv) retrieving application data; and

v) retrieving protocol messages.

Figure 4 schematically depicts the steps of the intrusion detection process: step b1: parsing the data traffic to extract at least one protocol field of a protocol message of the data traffic, step b2: associating the extracted protocol field with a model for that protocol field, the model being selected from a set of models, step b3: assessing if a contents of the extracted protocol field is in a safe region as defined by the model, and step b4:generating an intrusion detection signal (e.g. followed by filtering the extracted protocol field or protocol message comprising the protocol field, generating an alarm to a user, or any other intrusion detection action) in case it is established that the contents of the extracted protocol field is outside the safe region.

In an embodiment, the intrusion detection signal may further be generated when the parsing cannot establish the field as complying to the protocol or when the extracted field cannot be associated with any of the models of the set of models.

It is to be understood that the disclosed embodiments are merely exemplary of the invention, which can be embodied in various forms. Therefore, specific structural and functional details disclosed herein are not to be interpreted as limiting, but merely as a basis for the claims and as a representative basis for teaching one skilled in the art to variously

employ the present invention in virtually any appropriately detailed structure. Furthermore, the terms and phrases used herein are not intended to be limiting, but rather, to provide an understandable description of the invention. Elements of the above mentioned embodiments may be combined to form other embodiments.

The terms "a" or "an", as used herein, are defined as one or more than one. The term another, as used herein, is defined as at least a second or more. The terms including and/or having, as used herein, are defined as comprising (i.e., not excluding other elements or steps). Any reference signs in the claims should not be construed as limiting the scope of the claims or the invention. The mere fact that certain measures are recited in mutually different dependent claims does not indicate that a combination of these measures cannot be used to advantage. The scope of the invention is only limited by the following claims.

C O N C L U S I E S

1. Intrusiedetectie werkwijze voor het detecteren van een intrusie in dataverkeer op een datacommunicatienetwerk, waarbij de werkwijze omvat:
   - het parsen van het dataverkeer voor het extraheren van ten minste een protocolveld van een protocolbericht van het dataverkeer;
   - het associëren van het geëxtraheerde protocolveld met een respectief model voor dat protocolveld, waarbij het model is geselecteerd uit een set met modellen;
   - het beoordelen of een inhoud van het geëxtraheerde protocolveld in een veilig gebied is zoals gedefinieerd door het model; en
   - het genereren van een intrusiedetectiesignaal in geval is vastgesteld dat de inhoud van het geëxtraheerde protocolveld buiten het veilige gebied is.

2. Intrusiedetectie werkwijze volgens conclusie 1, waarbij de set met modellen een respectief model voor elk protocolveld van de set protocolvelden omvat.

3. Intrusiedetectie werkwijze volgens conclusie 1 of 2, waarbij het model voor het veld wordt bepaald in een leerfase, waarbij de leerfase omvat:
   - het parsen van het dataverkeer voor het extraheren van ten minste een protocolveld van het protocol dat wordt gebruikt in het dataverkeer;
   - het associëren van het geëxtraheerde protocolveld met het model voor dat protocolveld, waarbij het model is geselecteerd uit de set met modellen en
   - het updaten van het model voor het geëxtraheerde protocolveld gebruik makend van een inhoud van het geëxtraheerde protocolveld.

4. Intrusiedetectie werkwijze volgens conclusie 3, waarbij wanneer er geen associatie kan worden gemaakt tussen het geëxtraheerde protocolveld en een van de modellen,
   - het creëren van een nieuw model voor het geëxtraheerde protocolveld en het toevoegen van het nieuwe model aan de set met modellen.

5. Intrusiedetectie werkwijze volgens een van de voorgaande conclusies, waarbij het intrusiedetectiesignaal voorts wordt gegenereerd wanneer het parsen niet kan vaststellen dat het veld voldoet aan het protocol.

6. Intrusiedetectie werkwijze volgens een van de voorgaande conclusies, waarbij het intrusiedetectiesignaal voorts wordt gegenereerd wanneer het geëxtraheerde veld met geen van de modellen van de set met modellen kan worden geassocieerd.

7. Intrusiedetectie werkwijze volgens een van de voorgaande conclusies, waarbij het protocol ten minste een van een applicatielaag protocol, een sessielaag protocol, een transportlaag protocol of een lagere laag protocolstapel protocol omvat.

8. Intrusiedetectie werkwijze volgens een van de voorgaande conclusies, waarbij de werkwijze voorts omvat, in antwoord op het genereren van het intrusiedetectiesignaal, ten minste een van:
   - het verwijderen van het protocolveld of een datapakket dat het protocolveld omvat;
   - het genereren en uitvoeren van een intrusiealertbericht.

9. Intrusiedetectie werkwijze volgens een van de voorgaande conclusies, waarbij het model voor het protocolveld ten minste een omvat van
   - een set met acceptabele protocolveldwaarden, en
   - een definitie van een gebied met acceptabele protocolveldwaarden.

10. Intrusiedetectie werkwijze volgens een van de voorgaande conclusies, waarbij het model voor het protocolveld omvat
   - een definitie van acceptabele letters, cijfers, symbolen en scripts.

11. Intrusiedetectie werkwijze volgens een van de voorgaande conclusies, waarbij het model voor het protocolveld een set met tevoren gedefinieerde Intrusiesignaturen omvat.

12. Intrusiedetectie werkwijze volgens een van de voorgaande conclusies, waarbij de set met modellen twee modellen voor een protocolveld omvat, waarbij een specifieke van de twee modellen wordt geassocieerd met het ene protocolveld gebaseerd op de waarde van een ander protocolveld.

13. Intrusiedetectiesysteem voor het detecteren van een intrusie in dataverkeer op een datacommunicatie netwerk, waarbij het systeem omvat:

- een parser voor het parsen van het dataverkeer voor het extraheren van ten minste een protocolveld uit een protocolbericht van het dataverkeer;
- een engine voor het associëren van het geëxtraheerde protocolveld met een respectief model voor dat protocolveld, waarbij het model is geselecteerd uit een set met modellen;
- een modelhanteerinrichting voor het beoordelen of een inhoud van het geëxtraheerde protocolveld in een veilig gebied is zoals gedefinieerd door het model; en
- een actuator voor het genereren van een instructiedetectiesignaal in geval is vastgesteld dat de inhoud van het geëxtraheerde protocolveld buiten het veilige gebied is.

14. Intrusiedetectiesysteem volgens conclusie 13, waarbij de set met modellen een respectief model omvat voor elke protocolveld van een set met protocolvelden.

15. Intrusiedetectiesysteem volgens conclusie 13 of 14, voorts ingericht om te worden bedreven in een leerfase, de leerfase voor het leren van ten minste een van de modellen, waarbij de modelhanteerinrichting is ingericht voor het updaten in de leerfase van het model voor het geëxtraheerde protocolveld gebruik maken van het inhoud van het geëxtraheerde protocolveld.

16. Intrusiedetectiesysteem volgens conclusie 15, waarbij de engine voorts is ingericht voor het in de leerfase, wanneer geen associatie kan worden gemaakt tussen het geëxtraheerde protocolvel en een van de modellen, creëren van een nieuw model voor het geëxtraheerde protocolveld en toevoegen van het model aan de set met modellen.

17. Intrusiedetectiesysteem volgens een van conclusie 13 – 16, waarbij de actuator voorts is ingericht voor het genereren van het intrusiedetectiesignaal in antwoord op een indicatie van de parser dat de parser niet kan vaststellen dat het veld voldoet aan het protocol.

18. Intrusiedetectiesysteem volgens een van conclusies 13 – 17, waarbij de actuator voorts is ingericht voor het genereren van het intrusiedetectiesignaal in antwoord op een indicatie van de engine dat het geëxtraheerd protocolveld met geen van de modellen van de set met modellen kan worden geassocieerd.

19. Intrusiedetectiesysteem volgens een van conclusies 13 – 18, waarbij het protocol ten minste een is van een applicatielaag protocol, een sessielaag protocol, een transportlaag protocol of een lagere laag protocolstapel protocol.

20. Intrusiedetectiesysteem volgens een van conclusie 13 – 19, waarbij de actuator is ingericht voor, in antwoord op het genereren van het intrusiedetectiesignaal:
- het verwijderen van het protocolveld of een datapakket omvattende het protocolveld; en
- het genereren en uitvoeren van een intrusiealertbericht.

21. Intrusiedetectiesysteem volgens een van conclusies 13 – 20, waarbij het model voor het protocolveld ten minste een omvat van
- een set met acceptabele protocolveldwaarden, en
- een definitie van een bereik met acceptabele protocolveldwaarden.

22. Intrusiedetectiesysteem volgens een van conclusies 13 – 21, waarbij het model voor het protocolveld omvat:
- een definitie van acceptabele letters, cijfers, symbolen en scripts.

23. Intrusiedetectiesysteem volgens een van conclusies 13 – 22, waarbij het model voor het protocolveld een set met tevoren gedefinieerde intrusiesignaturen omvat.

24. Intrusiedetectiesysteem volgens een van conclusies 13 – 23, waarbij de set met modellen twee modellen omvat voor een protocolveld, waarbij de engine is ingericht voor het associëren van een specifieke van de twee modellen met het ene protocolveld gebaseerd op de waarde van een ander protocolveld.

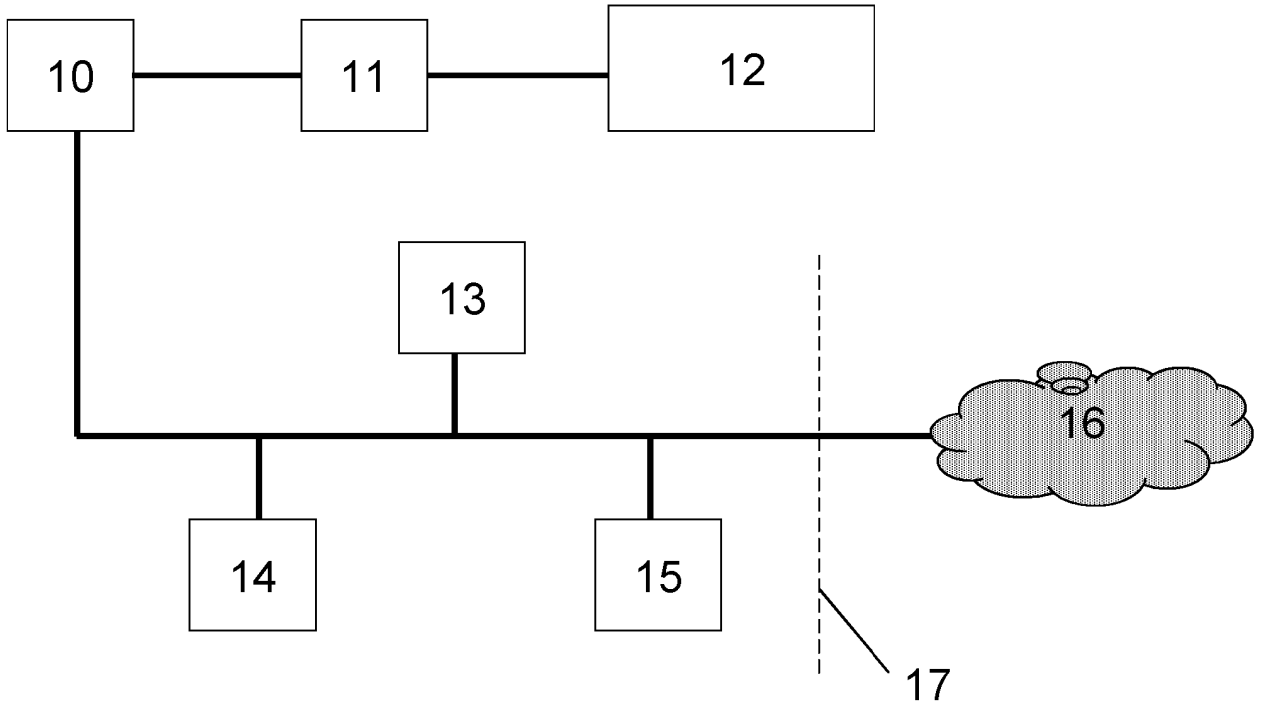


Figure 1

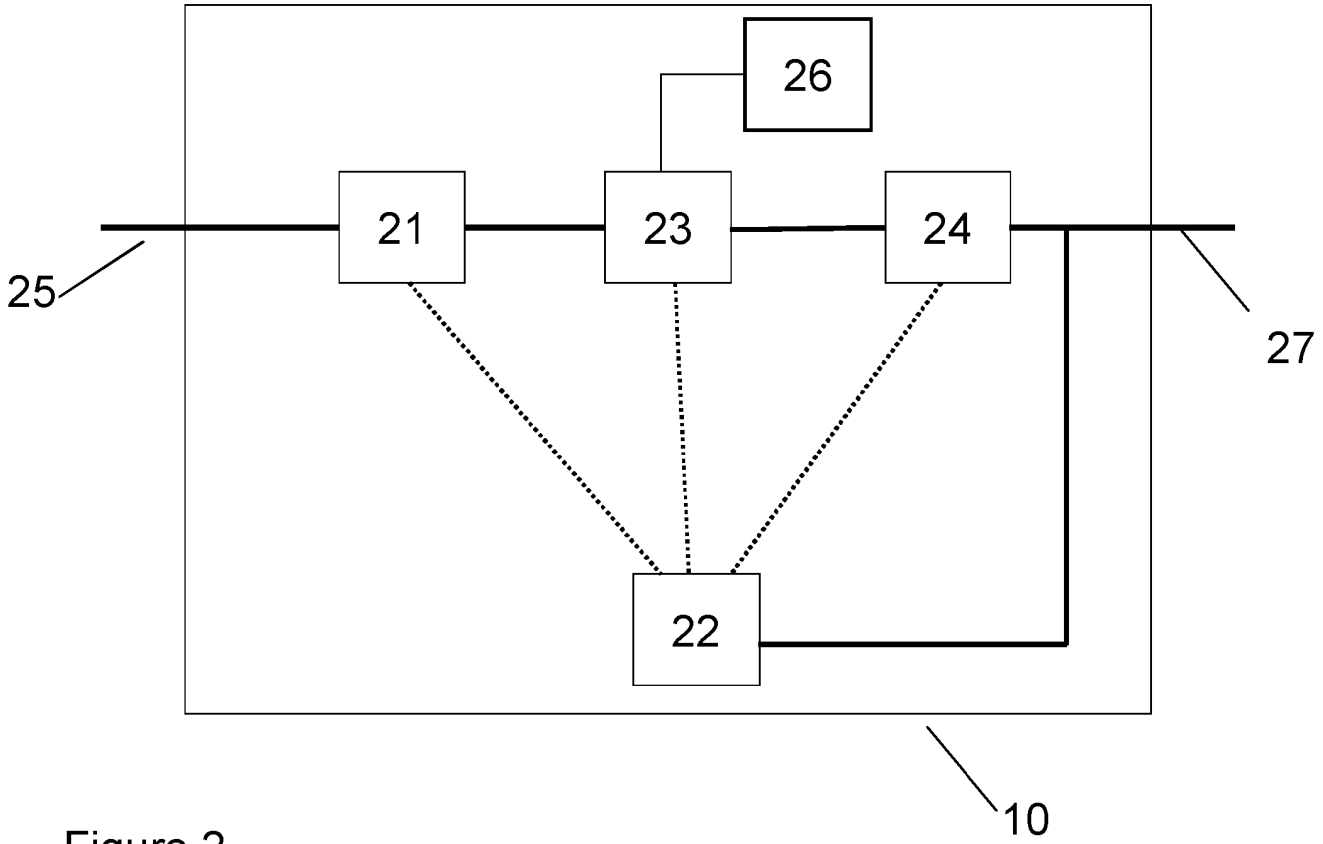


Figure 2

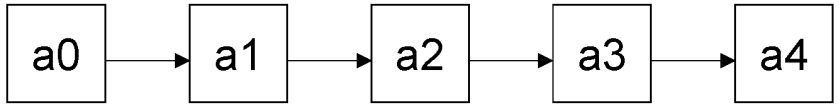


Figure 3

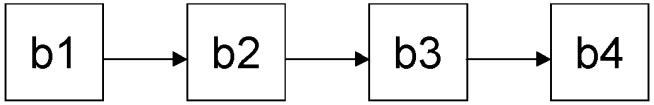


Figure 4

SAMENWERKINGSVERDRAG (PCT)

RAPPORT BETREFFENDE NIEUWHEIDSONDERZOEK VAN INTERNATIONAAL TYPE

IDENTIFICATIE VAN DE NATIONALE AANVRAGE	KENMERK VAN DE AANVRAGER OF VAN DE GEMACHTIGDE
	P30657NL00/HSE
Nederlands aanvraag nr. **2007180**	Indieningsdatum **26-07-2011**
	Ingeroepen voorrangsdatum
Aanvrager (Naam) **Security Matters B.V.**	
Datum van het verzoek voor een onderzoek van internationaal type **05-11-2011**	Door de Instantie voor Internationaal Onderzoek aan het verzoek voor een onderzoek van internationaal type toegekend nr. **SN57160**

I. CLASSIFICATIE VAN HET ONDERWERP (bij toepassing van verschillende classificaties, alle classificatiesymbolen opgeven)

Volgens de internationale classificatie (IPC)

H04L29/06

II. ONDERZOCHTE GEBIEDEN VAN DE TECHNIEK

Onderzochte minimumdocumentatie

Classificatiesysteem	Classificatiesymbolen
IPC	**H04L**

Onderzochte andere documentatie dan de minimum documentatie, voor zover dergelijke documenten in de onderzochte gebieden zijn opgenomen

III.	☐	**GEEN ONDERZOEK MOGELIJK VOOR BEPAALDE CONCLUSIES**	(opmerkingen op aanvullingsblad)
IV.	☐	**GEBREK AAN EENHEID VAN UITVINDING**	(opmerkingen op aanvullingsblad)

ONDERZOEKSRAPPORT BETREFFENDE HET RESULTAAT VAN HET ONDERZOEK NAAR DE STAND VAN DE TECHNIEK VAN HET INTERNATIONALE TYPE

Nummer van het verzoek om een onderzoek naar de stand van de techniek
NL 2007180

A. CLASSIFICATIE VAN HET ONDERWERP
INV. H04L29/06
ADD.

Volgens de Internationale Classificatie van octrooien (IPC) of zowel volgens de nationale classificatie als volgens de IPC.

B. ONDERZOCHTE GEBIEDEN VAN DE TECHNIEK

Onderzochte miminum documentatie (classificatie gevolgd door classificatiesymbolen)
H04L

Onderzochte andere documentatie dan de mimimum documentatie, voor dergelijke documenten, voor zover dergelijke documenten in de onderzochte gebieden zijn opgenomen

Tijdens het onderzoek geraadpleegde elektronische gegevensbestanden (naam van de gegevensbestanden en, waar uitvoerbaar, gebruikte trefwoorden)
EPO-Internal, INSPEC, WPI Data

C. VAN BELANG GEACHTE DOCUMENTEN

Categorie °	Geciteerde documenten, eventueel met aanduiding van speciaal van belang zijnde passages	Van belang voor conclusie nr.
X	US 2011/167493 A1 (SONG YINGBO [US] ET AL) 7 juli 2011 (2011-07-07) * samenvatting * * alinea [0010] - alinea [0014] * * alinea [0020] - alinea [0094] * * figuren 1-4 * -----	1-24

☐ Verdere documenten worden vermeld in het vervolg van vak C.

☒ Leden van dezelfde octrooifamilie zijn vermeld in een bijlage

° Speciale categorieën van aangehaalde documenten

"A" niet tot de categorie X of Y behorende literatuur die de stand van de techniek beschrijft

"D" in de octrooiaanvrage vermeld

"E" eerdere octrooi(aanvrage), gepubliceerd op of na de indieningsdatum, waarin dezelfde uitvinding wordt beschreven

"L" om andere redenen vermelde literatuur

"O" niet-schriftelijke stand van de techniek

"P" tussen de voorrangsdatum en de indieningsdatum gepubliceerde literatuur

"T" na de indieningsdatum of de voorrangsdatum gepubliceerde literatuur die niet bezwarend is voor de octrooiaanvrage, maar wordt vermeld ter verheldering van de theorie of het principe dat ten grondslag ligt aan de uitvinding

"X" de conclusie wordt als niet nieuw of niet inventief beschouwd ten opzichte van deze literatuur

"Y" de conclusie wordt als niet inventief beschouwd ten opzichte van de combinatie van deze literatuur met andere geciteerde literatuur van dezelfde categorie, waarbij de combinatie voor de vakman voor de hand liggend wordt geacht

"&" lid van dezelfde octrooifamilie of overeenkomstige octrooipublicatie

Datum waarop het onderzoek naar de stand van de techniek van internationaal type werd voltooid	Verzenddatum van het rapport van het onderzoek naar de stand van de techniek van internationaal type
26 april 2012	

Naam en adres van de instantie	De bevoegde ambtenaar
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Dujardin, Corinne

Formulier PCT/ISA/201 (tweede blad) (Januari 2004)

ONDERZOEKSRAPPORT BETREFFENDE HET RESULTAAT VAN HET ONDERZOEK NAAR DE STAND VAN DE TECHNIEK VAN HET INTERNATIONALE TYPE

Informatie over leden van dezelfde octrooifamilie

Nummer van het verzoek om een onderzoek naar de stand van de techniek
NL 2007180

In het rapport genoemd octrooigeschrift	Datum van publicatie	Overeenkomend(e) geschrift(en)	Datum van publicatie
US 2011167493 A1	07-07-2011	US 2011167493 A1	07-07-2011
		WO 2010011411 A1	28-01-2010

Agentschap NL
Ministerie van Economische Zaken,
Landbouw en Innovatie

WRITTEN OPINION

File No.	Filing date *(day/month/year)*	Priority date *(day/month/year)*	Application No.
SN57160	26.07.2011		NL2007180

International Patent Classification (IPC)
INV. H04L29/06

Applicant
Security Matters B.V.

This opinion contains indications relating to the following items:

- ☒ Box No. I Basis of the opinion
- ☐ Box No. II Priority
- ☐ Box No. III Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
- ☐ Box No. IV Lack of unity of invention
- ☒ Box No. V Reasoned statement with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
- ☐ Box No. VI Certain documents cited
- ☐ Box No. VII Certain defects in the application
- ☐ Box No. VIII Certain observations on the application

	Examiner
	Dujardin, Corinne

Application number

WRITTEN OPINION

NL2007180

Box No. I Basis of this opinion

1. This opinion has been established on the basis of the latest set of claims filed before the start of the search.

2. With regard to any **nucleotide and/or amino acid sequence** disclosed in the application and necessary to the claimed invention, this opinion has been established on the basis of:

   a. type of material:

   ☐ a sequence listing

   ☐ table(s) related to the sequence listing

   b. format of material:

   ☐ on paper

   ☐ in electronic form

   c. time of filing/furnishing:

   ☐ contained in the application as filed.

   ☐ filed together with the application in electronic form.

   ☐ furnished subsequently for the purposes of search.

3. ☐ In addition, in the case that more than one version or copy of a sequence listing and/or table relating thereto has been filed or furnished, the required statements that the information in the subsequent or additional copies is identical to that in the application as filed or does not go beyond the application as filed, as appropriate, were furnished.

4. Additional comments:

Box No. V Reasoned statement with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. Statement

Novelty	Yes:	Claims	5, 6, 10-12, 17, 18, 22-24
	No:	Claims	1-4, 7-9, 13-16, 19-21
Inventive step	Yes:	Claims	
	No:	Claims	1-24
Industrial applicability	Yes:	Claims	1-24
	No:	Claims	

2. Citations and explanations

see separate sheet

Re Item V

Reasoned statement with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1 Reference is made to the following document:

US 2011/167493 A1 (SONG YINGBO [US] ET AL) 7 juli 2011 (2011-07-07)

2 The present application does not meet the criteria of patentability, because the subject-matter of **independent claims 1 and 13** is not new.

2.1 **D1** discloses the subject-matter of **claim 1**. See paragraphs 21, 31, first sentence, 55, 64 and 69-71. The subject-matter of **claim 1** is therefore not new.

2.2 The same reasoning applies, mutatis mutandis, to the subject-matter of the corresponding independent **claim 13**, which therefore is also considered not new.

3 **Dependent claims 2-12 and 14-24** do not contain any features which, in combination with the features of any claim to which they refer, meet the requirements of novelty and/or inventive step, see document **D1** and the corresponding passages cited in the search report.