Real-time Risk Analysis on the Internet

A Prototype

H.S. VENTER (heins@icon.co.za)
L. LABUSCHAGNE (ll@na.rau.ac.za)
J.H.P. ELOFF (eloff@rkw.rau.ac.za)

Department of Computer Science
Rand Afrikaans University
PO Box 524
AUCKLAND PARK
2006
South Africa
March 1999
Tel: +27 11 489-2847 Fax: +27 11 489-2138

Key words: Internet security, Real-time Risk Analysis (RtRA), network, firewalls, TCP/IP packet, RtRA prototype

Abstract: In current times, sending confidential data over the Internet is becoming more commonplace every day. The process of sending confidential data over the Internet is, however, concomitant with great effort: encryption algorithms have to be incorporated and encryption key management and distribution have to take place. Wouldn't it be easier, more secure and faster if only technology could be introduced to do risk analysis in real time? The objective of doing risk analysis in real time is to find a method through which dynamically to determine the vulnerability of, for example, a TCP/IP packet in terms of generic threat categories such as interception and fabrication. Once the vulnerability of the packet has been determined, the appropriate countermeasures can be activated to secure the packet before it is sent off to its original destination. The countermeasures are activated according to certain data that is found in and extracted from the TCP/IP packets. In order to be able to obtain this data, each TCP/IP packet flowing through a certain point in a network is intercepted and analysed.

1. INTRODUCTION

A paradigm shift has taken place in the commercial sectors of most first-world countries during the past decade. With the advent of the Internet, most organisations are rethinking their business strategies to exploit the biggest single quantum leap in

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35575-7_19

J. H. P. Eloff et al. (eds.), *Information Security Management & Small Systems Security*

technology for many years. In the race to find new competitive advantages, some important issues are, however, slipping through the proverbial cracks. One such issue is Internet security.

Most security professionals agree that it is very difficult to detect an attack until it is almost too late. As soon as a message leaves the organisational domain, all control is lost over it. It is not possible with absolute certainty to predict what attacks will be launched at the message while in transit over the Internet. Security measures can only be applied to the message based on a wide range of potential security threats, for example, eavesdropping or interception. It is for this reason that the vulnerability of a message, rather than the potential threat thereto, is used to determine the security services required to protect it.

The vulnerability of a message comprises all the factors that influence it, for example, its origin, content and destination. Determining the threat to a message has, however, become more difficult, as there are many unknown factors beyond the boundaries of the organisational domain. Examples of such unknown factors are the route the message will follow, the people who might benefit from attacking the message and whether or not the message has been compromised in any way.

Internet security used to manifest itself in a form that could only be described as rather "static". This means that it still lacks a network-security paradigm in terms of which real-time enhancements can be made to its network security. Is there perhaps not an easier, faster and more secure way than that provided by current security technologies? Although the ultimate network-security solution still is far from a reality, this article will be devoted to an attempt at showing that a technology, called *Real-time Risk Analysis*, can go a long way towards finding the answer to this question.

A method is, therefore, required dynamically to determine the vulnerability of a message according to its generic threat categories such as interception and fabrication [PFLE 89]. Once the vulnerability of the message has been determined, the appropriate countermeasures can be activated, in real time, to secure it during its transmission to its destination. This entire process must, however, take place in real time and must be absolutely transparent to the user. For the process to be transparent to the user, it must execute at the network level. One possible method that can be employed to meet this requirement, is *Real-time Risk Analysis* (RtRA) [LABU 98].

The remainder of this article will be structured as follows: Section 2 will be devoted to defining the concept "*Real-time Risk Analysis*" in contrast with conventional (existing) firewall technology and how TCP/IP (Transmission Control Protocol/Internet Protocol) packets can be analysed. In addition, it suggests a way in which to develop an add-on to conventional firewall technology by combining the technology of RtRA and that of firewalls. Section 3 will introduce a prototype that strengthens the argument for the implementation of RtRA technology. The article will culminate in Section 4 with a discussion on the benefits to be derived from, further research to be undertaken in and future expectations of RtRA technology.

2. REAL-TIME RISK ANALYSIS (RTRA) AND FIREWALLS

Because of the simplicity of TCP/IP and the clamant need for more secure networks, additional security measures must be implemented – security measures that will enhance network security dynamically and in real time. In the light of the lack of such security analysis technologies, the concept "RtRA" had to be developed. Another reason why RtRA had to be introduced is the dynamic and ever-changing nature of computer networks and technical aspects, such as new developing architectures. Although conventional risk analysis can, therefore, be effected on computer networks, it will never be implemented exactly as planned, owing to the changeable, dynamic nature of computer networks and architectures.

RtRA constitutes that process through which dynamically and in real time to determine the vulnerabilities, threats or risks that may possibly be incurred when sending data over the Internet, as well as that process through which to find ways in which, at best, to prevent or, at worst, to minimise these vulnerabilities, threats or risks [LABU 98]. The closest conventional solution to RtRA until now is that technology which is generally encompassed by the term "firewalls". RtRA technology could, however, constitute a more revolutionary solution. RtRA technology differs from the firewall approach in the sense that firewalls will treat two successive TCP/IP packets in exactly the same manner, because of its predefined rules. The characteristics of two successive TCP/IP packets may, however, differ vastly from each other, since packets do not necessarily arrive in a set order at a certain point in a network. In addition, the packets flowing through a certain point in the network will most likely consist of a mixture of various messages sent from different places at different times. It is in the latter respect that RtRA technology is set significantly to facilitate the real-time packet-analysing effect. Following, a detailed description of existing firewall technology.

2.1 Existing firewall technology

A firewall provides a blockade between a secure, internal and private network and an **insecure** one, such as the Internet [IBMC 97]. It protects the internal network from other networks in the Internet, while at the same time allowing TCP/IP services (such as e-mail, HTTP and FTP) to access hosts outside the network – on the Internet. This type of firewall will henceforth be referred to as a "conventional firewall".

Currently, three different types of conventional firewalls are being distinguished:

- Monitoring – This type of firewall simply logs traffic going into and out of a server.
- Packet-filtering (sometimes referred to as "screening") – This type of firewall filters packets by using various protocols that, in turn, determine which incoming and outgoing IP addresses, domains, names or passwords are acceptable. This operation, in effect, blocks out undesirable or unrecognised incoming traffic and limits the extent and routing of outgoing traffic.
- Stateful inspection (also called "proxy servers" or "application-level gateways") – This type of firewall controls all traffic with strict protocols, including levels of

access or maintaining regular checks of all data trials or communications. This type of firewall is the most advanced and, when strictly enforced, it can exert a strict level of control over and knowledge about the use of the organisational server.

Examples of conventional firewall products are CheckPoint's Firewall-1 [FIR1 99], TIS's Firewall Toolkit [FWTK 99] and Raptor Firewall [RAPT 98].

Although a conventional firewall is a highly effective way of effecting network security, conventional firewall technology does not determine risk values for each TCP/IP packet or message in real time. This implies that a new generation of firewall technology is required.

Before looking at this new generation of firewall technology, we need, however, first to consider conventional firewall technology, as depicted in fig. 2.1 below. This figure represents a complete scenario, in terms of which the TCP/IP packet travels between two workstations.

Frames I and II in fig. 2.1 below represent the **potentially secure** network protected by the firewall **FW** in frame II, while frame III represents the TCP/IP packet (**DG**) that flows between **Network 1** and **Network 2**. Note, in this respect, that **DG** also is present in frames I, II, IV and V. Frame IV represents the **Internet** – the insecure path along which the TCP/IP packet travels to get from **Network 1** to **Network 2**, for example. Frame V represents an **insecure** network with no firewall.

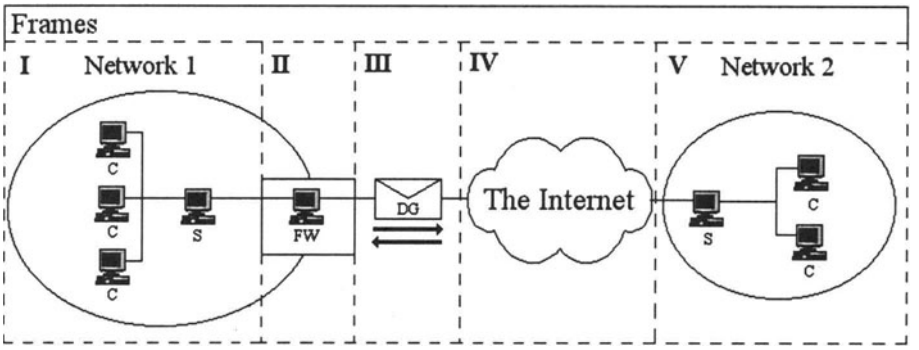


Figure 2.1: Conventional firewall technology

The functionality offered by conventional firewalls can be ascertained by looking at an example of one of the most recent developments in conventional firewall technology, namely the Raptor Eagle Firewall 5.0 [RAPT 98]. Raptor Eagle Firewall 5.0 is a high-performance, full-security enterprise firewall [RFAQ 98].

Based on application-level proxies, the firewall combines a high level of perimeter security. Some of its specific functions are as follows [EAGL 98]:

- Use of rules at the application level: Raptor denies any connections not explicitly allowed by a rule. Each rule can incorporate a range of criteria, including source and destination addresses, type of service, strong user or group authentication and the exact time of the access attempt.
- Automatic Suspicious Activity Monitoring (SAM): Raptor performs SAM on all connections throughout the firewall. SAM works by keying on thresholds established for connection rates when formulating authorisation rules. Raptor applies these thresholds on a rule-by-rule basis. In creating rules, their thresholds are specified, based on anticipated levels of access for each of them.
- Transparent access through the firewall: Raptor can support transparent connections between internal and external hosts. The term "transparency" refers to a user's awareness of the firewall. The firewall can be configured in such a way that users can connect through it to a destination system, still subject to existing authorisation rules, without being aware of the presence of the firewall.

This is the most important and recent functionality in conventional firewall technology. How will this scenario change, however, if the new generation of firewall technology were to be added? The concept "new generation firewall" entails the expansion of conventional firewall technology with additional modules to implement RtRA. Before considering the RtRA process, however, a brief discussion on how TCP/IP packets and sessions are analysed in terms of this new generation firewall technology.

2.2 Analysing a TCP/IP session

A communication session between two hosts consists of different levels, with the *communication session* itself acting as the **first level**. Furthermore, a communication session could consist of a few messages that could each be broken up into TCP/IP packets. A *message* serves as the **second level** and a *TCP/IP packet* serves as the **third level** in a communication session, as depicted in fig. 2.2 below.

There basically are two approaches to analysing a communication session, namely that at **message level** and that at **packet level**. Both approaches have certain advantages and disadvantages, however. When analysing the communication session at *message level*, the vulnerability and security countermeasures of the message will be determined in terms of the entire message. This, in turn, means that the message will only be analysed before commencement of transmission to determine the countermeasures, just before the message is broken up into packets. This also means that the same countermeasures will be applicable to all of the TCP/IP packets into which the message has been broken up, because all the packets will relate to the same message. The advantage of following the **message-level approach** is that it is faster than the **packet-level approach**, as the countermeasures have already been determined for the entire message and only need to be applied to each packet of the message. A disadvantage or shortcoming of the message-level approach, however, is the fact that limited potential ground is won towards RtRA. In fact, conventional firewall technology follows almost a similar approach to accomplish this. The basis on which a conventional firewall functions is also founded on predefined countermeasures and rules. A conventional firewall, in other

words, applies security more at user level (for a specific IP address), and not at message level.

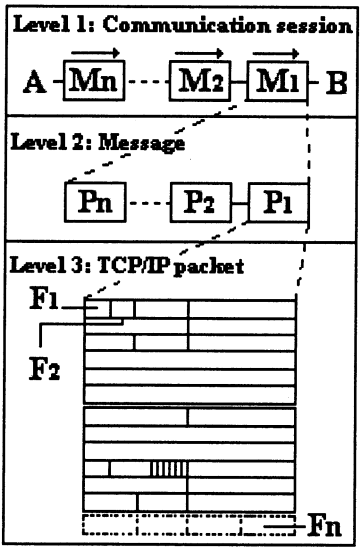


Figure 2.2: Layering in a communication session

By following the *packet-level* approach, the workload increases, as each TCP/IP packet can be individually encrypted while being transmitted [FORD 97]. This is owing to countermeasures that are determined for and applied to each TCP/IP packet as an entity before the next packet is analysed, thus resulting in a slower process. Despite the cost of an increased workload, however, some benefits may be derived, such as the fact that analysing packets is not bound to a session any more. In this way, any packet can be analysed, regardless of its type, the connection from which it is coming or the order in which it arrives. A packet-filtering firewall functions in almost the same way in the sense that packets are also analysed in this manner.

Only the packet-level approach will, however, be considered for the purposes of this article, so that more emphasis could be placed on the real-time effect of RtRA and its advantages, while keeping the explanation as simple as possible. The authors are, however, aware of the fact that, in following this approach, they did not opt for the best way in which to implement RtRA. A combination of the message-level and the packet-level approaches would, for instance, offer a better solution. By implementing both these approaches, room would be left for optimisation. An example of how optimisation could play a prominent part in this scenario is the following: if it were known that a message was very long, it would make more sense rather to follow the message-level approach, as it would save more time (a new countermeasure would not need to be determined for each packet). If, however, the real-time effect and higher security were considered to be of greater importance, it would be better to follow the packet-level approach. Following, a discussion on the RtRA process and its modus operandi.

2.3 The RtRA process

The modus operandi of the RtRA process is as follows: a risk value is determined for each TCP/IP packet travelling through a monitored point in a network in real time. This risk value is determined as follows: certain fields are extracted from the TCP/IP packet as they are intercepted at the monitored point, for example, the *source-address* field, the *destination-address* field and the *time-sent* field. First, a risk value for each of these fields is determined and consolidated into an overall risk value for the TCP/IP packet. Based on this risk value, certain security services are activated dynamically to reduce the vulnerability of the packet. This has the effect that two consecutive packets with the same *source* and *destination* fields can have different risk values, because their *time-sent* fields indicate that they have been sent at different *times*. The first packet could, for example, have been sent a fraction before a time threshold value change. As soon as this threshold value changes, it causes the risk level of the time-sent field to change for each packet to be sent from that point onwards. The second packet could have been sent a fraction after this threshold value has changed, thereby causing the second packet either to have a higher or a lower risk value. This will also cause the overall risk value determined for the TCP/IP packet to change. The other packet is, therefore, treated differently.

The RtRA process proposes that all network traffic is analysed, but the appropriate security level is only applied to the TCP/IP packets according to the packets' determined vulnerability level. This means that not all packets are necessarily secured (encrypted) since not all packets are vulnerable or sensitive packets. This does not seem normal since people mostly secure their systems in advance. However, it simply is impossible to secure every TCP/IP packet travelling through a network, because the overhead and processing power acquired would simply be too high since millions of packets can travel throughout a network in only fractions of time.

RtRA may seem to relate to *intrusion detection*. Shortly, an *intrusion detection system (IDS)* monitors a system or network constantly with the goal to report intrusion attacks. This is done by monitoring users' whereabouts in a system, for example, number of attempted logins, activities by users, system resource usage etc. This data forms a footprint of network and system usage over time. From this footprint, the IDS will compute metrics about the system's overall state and decide whether an intrusion is currently occurring [PRIC 98]. Although RtRA also monitors the network, it takes the process a little further. An IDS only detects a possible intrusion attack as soon as the attack occurs. RtRA, however, attempts to secure communications even before a potential attack can occur by applying appropriate countermeasures in advance to the appropriate TCP/IP packets and sessions in real time. In addition, RtRA's effectiveness level is constantly at a maximum as soon as it is installed, but that of an IDS increases with time.

The approach followed for the implementation of RtRA is that two new additional modules be added to conventional firewall technology (**FW**), implementing RtRA as follows. These two additional modules are called the "Gateway module" (**GW**) and the "Countermeasure Executor module" (**CE**) respectively. Together, they are referred to as the "Gateway Bridge" (**GB**), as

depicted in fig. 2.3. In addition, combining **FW**, **CE** and **GB** forms the "New Generation Firewall" (**NGF**). Note that frames I and II of fig. 2.1 change as follows in fig. 2.3. Based on the above, a more detailed explanation of what RtRA is will be given by means of a prototype in the next section.

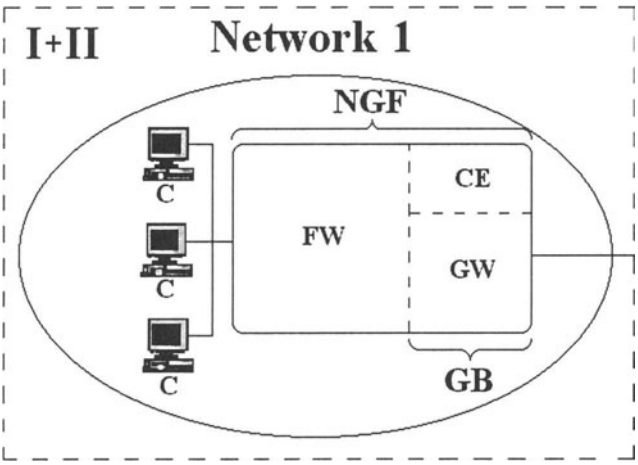


Figure 2.3: Conventional firewall expanded to the NGF

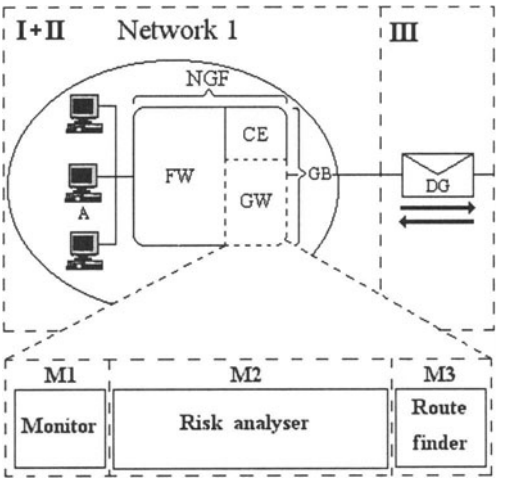
3. DEMONSTRATING RTRA: A PROTOTYPE

This section will be devoted to an attempt at demonstrating the essential RtRA concepts, for which purpose a prototype has been constructed.

Referring to RtRA as a "new generation" in firewall technology (**NGF**), implies that the concept forms part of an already existing concept, namely that of firewalls. As was mentioned earlier, the concept of a firewall has, however, been expanded into a more intelligent version of a firewall by merely adding certain intelligent modules that will be able not only to analyse the communication session in real time but also to do so in an intelligent fashion. The two main modules are **GW** and **CE**. **GW** can, in turn, be broken up into three smaller modules:

- **Module 1**: Monitor
- **Module 2**: Risk analyser
- **Module 3**: Route finder

The output of one module serves as the input for the next consecutive module. In fig. 3.1, a basic configuration of the prototype is given. Frames **M1**, **M2** and **M3** show where the respective activities of **Module 1**, **Module 2** and **Module 3** take place.



A: Sender client
B: Receiver client
GW: Gateway
CE: Countermeasure Executor
GB: Gateway Bridge
M1: Module 1 – Monitor
M2: Module 2 – Risk analyser
M3: Module 3 – Route finder

Network 1: This is a private network that makes use of RtRA
Network 2: This is an ordinary private network with no network security
S: Server without firewall
FW: Conventional firewall
DG: TCP/IP packet

Figure 3.1: GW expanded

Module 2 is the more comprehensive module, because it is in this module that the critical processing of the prototype is performed. Although the functionality and processing of **Module 1 (M1)** and **Module 3 (M3)** are less comprehensive than that of **Module 2 (M2)**, all three modules are equally important for the execution of the prototype. Following, a more detailed discussion of the three modules.

3.1 Module 1 – the monitor

The principal aim of this module is to analyse a TCP/IP packet in a bid to obtain the information necessary to perform RtRA. This will be achieved by intercepting each TCP/IP packet in a single session and by extracting the required fields from each packet. A session is the duration for which two workstations are connected to and the period during which they exchange data with each other. The latter data is sent to **Module 2** for further processing.

What exactly is a TCP/IP packet? It is a unit or "package" of information that contains a portion of a greater chunk of data. Furthermore, data such as that contained in e-mail messages and Web pages is sent across the Internet using packets [INTE 98].

The inputs for **Module 1** are the TCP/IP packets, while its outputs are the appropriate extracted fields. A graphic illustration of how a TCP/IP packet (which is the input to **Module 1**) is composed, is presented in fig. 3.2. This is a single

TCP/IP packet that travels between two workstations on the Internet. The fields extracted for the prototype (which constitute the input to **Module 2**) appear in bold print and are considered the most important fields (for the purpose of the prototype). The reason for only using them in the prototype will be explained next [PABR 96].

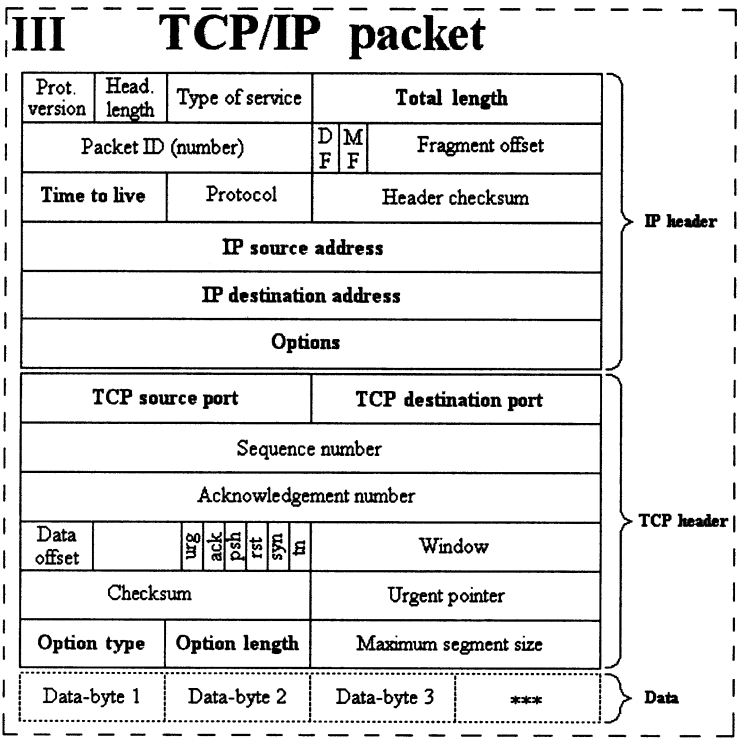


Figure 3.2: TCP/IP packet – extracted fields

The **Total-length** field indicates how many bytes the total length of the message consists of and, from this field, one can distinguish what type of message it is. A short length may, for example, indicate that it is an SMS cellular-telephone message (160 characters maximum) or, if longer, it may indicate e-mail messages or even a file or document transfer. By keeping track of Web statistics using some Web-monitoring software, one could easily determine the average length of e-mail messages sent for a specific organisation. This average can then be used as a threshold value for the **Total-length** field. If the total length of a message were significantly to exceed this average, a higher risk factor will become applicable to the message forthwith. It could, for example, be an indication that large chunks of information are in the process of being insecurely transferred. The greater the margin by which the message length, therefore, exceeds the average message length, the higher the risk that some extraordinary message is in transit (either insecurely exported by an inside employee or insecurely downloaded by an outsider).

The **Time-to-live** (sometimes referred to as the "time-sent") field indicates how long the packet has been travelling from the sender to the receiver and serves as a

timer, although the value is only incremented each time it reaches another host (or a "hop"). In this way, this field indicates how long the packet has been travelling and, from that, one could determine whether or not the packet was sent inside or outside a valid threshold **Time to live**. If, for example, the threshold value were exceeded, it could, therefore, indicate a possible packet interception or modification.

The **IP source address** and the **IP destination address**, as well as the **TCP source port** and the **TCP destination port**, provide information from and to whom the packet is sent. From the data in these fields, one could determine whether or not a packet was sent to or from the correct computer/person.

The **Options** field also contains valuable information, among which details on how loose or strict the routing of the packet is. If a packet were routed very strictly, it would carry a lower risk of being a malicious packet.

3.2 Module 2 – the risk analyser

The purpose of this module is to determine the level of risk associated with each packet in the current communication session. If RtRA were done for the entire communication session, a value would only have been obtained after the communication session has been completed – which would, naturally, have been too late! Say, for example, a time threshold value is 5:00 pm and that a connection between two workstations has been established at 4:57 pm, lasting until 5:49 pm. At 5:00 pm, the risk level for the **Time-to-live** field changes to that of a higher risk level. This will have the effect that, in the latter part of the said communication session, a more effective countermeasure will have to be activated at 5:00 pm. To determine a risk value for the first packet and then to apply it to the entire communication session would also not work, as the risk level would be appropriate for the first three minutes of the connection, but would have changed from 5:00 pm to 5:49 pm, because of the time threshold value of 5:00 pm. The risk value determined for the first packet would, therefore, not be appropriate for the greater part of the session. This, in turn, means that the real-time effect will be completely lost and that the countermeasures would be rendered ineffective.

The inputs to **Module 2** are the outputs from **Module 1** – the extracted fields from the current TCP/IP packet - that were intercepted at the Gateway (**GW**). The output of **Module 2** is a Global Risk Value (GRV). A GRV is a value determined to specify the overall risk value associated with a current TCP/IP packet in order to allocate the right degree of security to such packet. An *Inference Engine* (**IE**) determines a GRV by consolidating the extracted fields from the TCP/IP packet into a single value, namely the GRV. The consolidation method determines to what extent these extracted fields comply with characteristics in two databases, called a *Rule Base* (**RB**) and a *Knowledge Base* (**KB**).

The **RB**, **KB** and an **IE** are the three sub-modules of **Module 2**. Note, however, that although the **RB** and the **KB** form sub-modules of **Module 2**, they are "read-only" in respect of the RtRA prototype. In addition, the **RB** and **KB** are databases that are updated by a system administrator. The components of **Module 2** are depicted in fig. 3.3 as follows:

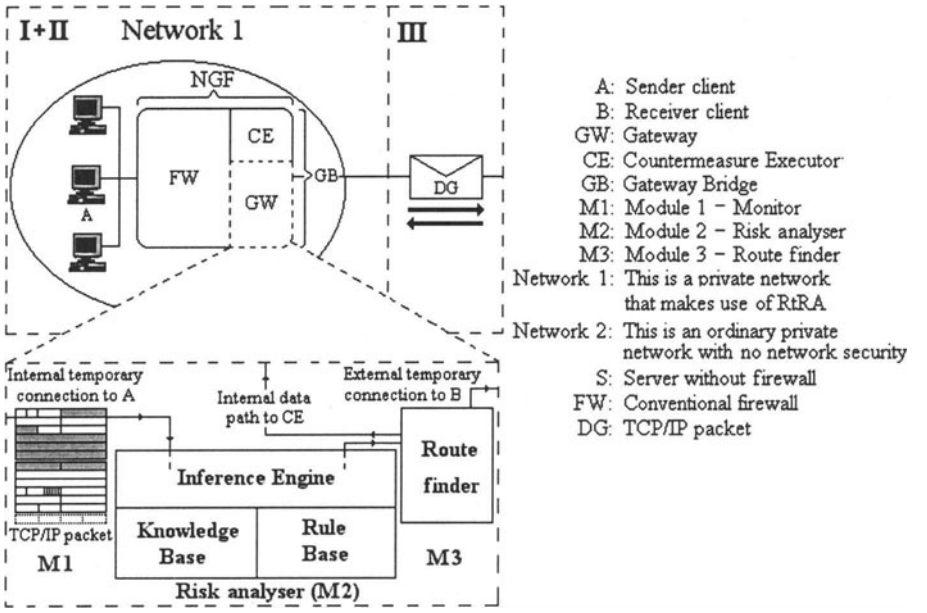


Figure 3.3: Showing the incorporation of the risk analyser (Module 2)

3.2.1 The Knowledge Base

The Knowledge Base (called "**KB**" for short) is a database consisting of entries that provide information on the organisation. The **KB** operates on information based on the extracted fields of the TCP/IP packet from **Module 1**. As an example, a risk level can be determined for each type of IP address. An executive will be allocated with a higher risk level than that allocated to an operator. It is also possible to determine a risk value based on the location of a workstation. A mobile workstation (for example, a notebook) will incur a higher risk than a protected workstation securely locked up in an office somewhere.

Risk values are allocated to each entry in the **KB**. **KB** values can be perceived to remain relatively constant for different organisations in the same market sector. **KB** values are *qualitative* values, for example, low (L), medium (M) and high (H).

Another important concept of the Knowledge Base is the concept of global risk values. Global risk values will activate the required combination of countermeasures. *Symmetric key encryption* is, for instance, used as a baseline security mechanism (*low GRV*), while *private key encryption* is applied where stronger security is required (*high GRV*). If a GRV were low (were to have a low risk value), minimum-security countermeasures would be applied to the packet. Risk levels can, however, be refined to include multiple levels, for example, low, medium-low, medium, medium-high and high.

3.2.2 The Rule Base

The Rule Base (Called "**RB**" for short) contains information that is specific to an organisation – it differs from one organisation to the next, regardless of whether two organisations are in the same market sector. The reason for this is that the **RB** reflects the current situation in an organisation; the values in the **RB** are, therefore, *quantitative.*

An example of an **RB** entry is that IP address www.xxx.yyy.zzz belongs to person X. Another example is that the IP address aaa.bbb.ccc.ddd belongs to a mobile computer. From the above, it is clear that the values www.xxx.yyy.zzz and aaa.bbb.ccc.ddd are *quantitative* values, for example, an IP address varies in quantity: 152.106.42.155, and 152.106.42.156 are two IP addresses that follow quantitatively on each other. The values in the **KB** examples above (high risk, medium risk and low risk) are *qualitative* values, because they indicate the quality of the risk in question.

Another important function of the **RB** is to specify what the quantitative value is for each appropriate risk level that has been determined. This is referred to as the **risk level activation RB** (see fig. 3.4). A GRV between 0 and 3.4 (for a scale out of 10) is, for example, is considered to be a GRV with a low risk level. A GRV between 3.5 and 5, in turn, is considered to be a GRV with a medium risk level. A GRV between 5.1 and 10, on the other hand, is considered to be a GRV with a high risk level. Should a GRV be fixed at, for example, a value of 4, the medium-level-of-risk countermeasure will be activated. This will have the effect that *symmetric key encryption* will be executed.

3.2.3 The Inference Engine

The **IE** is at the heart of the prototype. It uses the outputs from **Module 1**, together with the **KB** and **RB**, to determine an Interim Risk Value (IRV) for each field extracted by **Module 1**. An IRV is a risk value that is determined for each field extracted from the TCP/IP packet. This takes place just before the consolidation process to determine the GRV (see fig. 3.4). In other words, an IRV is, in essence, determined in the same way as a GRV, with the exception that it merely serves as an in-between process to obtain a single risk value for each extracted field. Consider, for example, the extracted field "TCP source port". It is found to be **P** in fig. 3.4. From **P**, two characteristics have been derived: **P** is a **reserved port**, resulting in a risk value of 7. In addition, **P** normally operates in an **Operating System W** environment, resulting in a risk value of 5. In order to obtain a single risk value for **P**, 7 and 5 have to be consolidated, resulting in an IRV of 6.

These IRVs are then consolidated to obtain the GRV for each TCP/IP packet that passes through **GW**. There are different and complex ways in which actually to consolidate the IRVs into a GRV, for example, the concept of *fuzzy logic* [DERU 97]. In the current version of the prototype, however, the consolidation of the IRVs into a GRV will simply be done by calculating the averages of the IRVs.

A representation of the steps to be followed in the **IE** is provided in fig. 3.4.

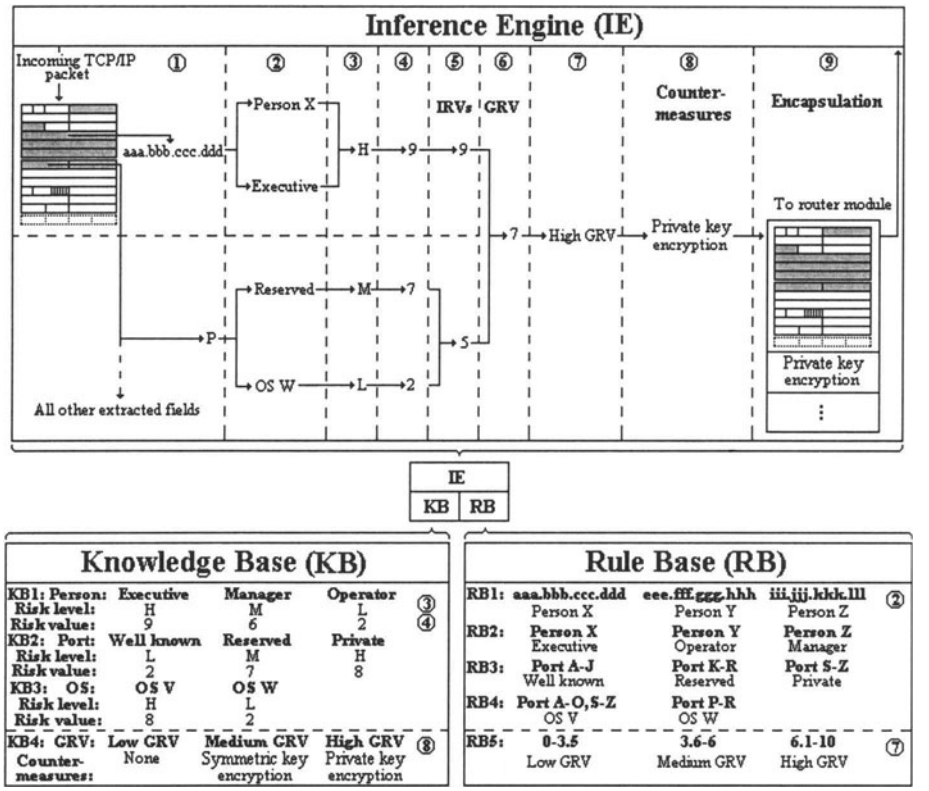


Figure 3.4: The various steps in the risk analyser (Module 2)

Fig. 3.4 illustrates the nine steps to be set out next. Each frame in the **IE** section of fig. 3.4 is numbered according to the specific step with which it is associated. In the bottom section of fig. 3.4, parts of the **KB** and **RB** are given. The numbers in each frame of the **KB** and **RB** indicate the step with which the specific **KB** or **RB** is associated. Read fig. 3.4 as follows: look at ***Step 2***, for example. This step is taken in a bid to retrieve the associated information for the IP source address (**aaa.bbb.ccc.ddd**) and the TCP source port (**P**). The IP source address **aaa.bbb.ccc.ddd** belongs to **Person X** (from **RB1**), who is an **executive** (from **RB2**). In addition, for extracted field **P**, it is learnt that **P** is a **reserved port** (from **RB3**) in an **Operating System W** (from **RB4**) environment.

Following, a discussion on these steps:

Step 1: Obtain the extracted fields from the TCP/IP packet from **Module 1**; for example, suppose the IP source address is **aaa.bbb.ccc.ddd** and the TCP source port is **P**.

Step 2: Retrieve the associated information for the current extracted fields (from the TCP/IP packet) from the **RB**, for example, IP source address **aaa.bbb.ccc.ddd** belongs to **Person X**. Person X is found to be an

executive in the **RB**. In addition, TCP source port **P** is found to be a **reserved port**. Port **P** is also found to be a port normally used inside an **Operating System W** environment.

Step 3: Link this information (**Person X, P**) to the **KB**. **Executives**, for example, incur a high level of risk (typically because they enjoy greater access rights to more sensitive information). **Reserved ports** carry a medium level of risk and **Operating System W** environments carry a low level of risk.

Step 4: Use a consolidation method to determine risk values for all the extracted fields from the TCP/IP packet, for example, the risk level for **Person X** at IP address **aaa.bbb.ccc.ddd** is **9** (out of 10, for instance). In addition, the risk level for port **P** as a **reserved port** is **7** and, as a standard port in an **Operating System W** environment, it is **2**.

Step 5: Consolidate all the risk values determined in ***Step 4*** into **IRVs**.

Step 6: Consolidate all the **IRVs** determined in ***Step 5*** into a **GRV**, for example, a **GRV** of **7** (for a scale out of 10) has been determined by means of a certain consolidation method.

Step 7: Retrieve the countermeasure information from the **risk level activation RB**. A **GRV** of **7**, for example, implies that the **GRV** falls into the **High GRV** range.

Step 8: Link the information from the **RB** to the **countermeasure activation KB**. The countermeasure value of **High GRV** in the **RB**, for example, implies that a **private key encryption countermeasure** in the **KB** must be applied to the current session from that point onwards.

Step 9: Compile a list of **countermeasures** to be executed (those found in ***Step 8***) and encapsulate them together with the original TCP/IP packet. (The term *encapsulation* is used in a different sense than usual here, though. Normally, the term **encapsulation** pertains to the idea that a packet is "wrapped" with another packet, so that the original packet is invisible/inaccessible and is usually used when a network protocol does not "understand" the packet format [COME 97]. In terms of the Internet, another header is "wrapped around" the original packet. This header can then only be removed (decapsulated) by a protocol that "knows" how the packet has been encapsulated. The purpose of encapsulation is to enhance the security of that specific packet. What is meant by the term *encapsulation* in ***Step 9***, is, however, something different. The *encapsulation* of the TCP/IP packet and the countermeasures means that it is simply concatenated into an argument. The purpose of this simply is to keep the data together when the packet and the countermeasures are passed by argument to **CE**.)

Note that the entire foregoing process is based on an outgoing packet (a packet travelling from **Network 1** to **Network 2**). The process, however, remains the same for an incoming packet (a packet travelling from **Network 2** to **Network 1**).

3.3 Module 3 – the route finder

The purpose of **Module 3** is to re-route the original TCP/IP packet according to the countermeasures determined in **Module 2**. The outputs of **Module 3** will be the specific route that the current TCP/IP packet must follow. (Refer to fig. 3.3 for a reminder as to where the **route finder** fits into the prototype.)

There are two kinds of possible outputs to be mentioned here. The first possibility is that if the GRV were so low that no countermeasure needed to be executed on the TCP/IP packet, the output would simply be the original TCP/IP packet. In this case, the original TCP/IP packet would simply be forwarded to the destination IP address. The second possibility is that if some countermeasure(s) needed to be executed on the TCP/IP packet, the encapsulated argument would first be passed to the Countermeasure Executor (**CE**). At the **CE**, the encapsulated argument would then be taken apart and the compiled countermeasure(s) would be detected and executed on the accompanied TCP/IP packet. The processed packet would then be passed back to **GW**. Only then would the processed packet be forwarded to the destination IP address.

4. CONCLUSION

The concept of RtRA introduces a new approach to conventional firewall technology and network security. The benefits to be derived from this statement are as follows: firstly, some of the common management efforts at encrypting data are left to RtRA. RtRA determines the current level of risk when sending data over the Internet. It further activates countermeasures to safeguard the data to the appropriate level in real time.

Secondly, the fact that RtRA is done in **real time** not only makes life easier but also speeds up the process. Much time is saved, as no explicit encryption or countermeasure has to be executed on the data by the user him-/herself. Most importantly, however, the real-time effect of RtRA actually provides the key to all the foregoing benefits to be derived from RtRA.

Thirdly, the users and the workstations in a network do not need to know anything about RtRA, except that it is there and that it secures their data much more effectively than any conventional network-security system.

There are, however, a few aspects in the realm of RtRA that still warrant further research. One such aspect is the fact that RtRA should be able to deal with multiple connections, which the prototype cannot cope with at this stage. Another aspect is that the prototype should be able to deal with asynchronous communication too. Further research on implementing asynchronous RtRA communications is, therefore, sorely needed. In addition, some optimisation issues might be addressed when applying the countermeasures. Currently, for example, the **risk analyser** determines a GRV for every single packet and applies the countermeasures to every packet, with the result that it only follows the *packet-level approach* discussed under paragraph 2.2. The ability to incorporate both approaches is, therefore, still a shortcoming in the prototype. By following both approaches, the prototype and the efficiency of RtRA will be enhanced even further. Another hot spot for which further research is sorely needed is the refining and implementation of more effective countermeasures. Countermeasures such as digital certificates and the distribution of keys also are possible areas that need to be investigated. The **RB** and **KB** constitute yet another area that warrants further research. Should ways and means be found to endow the process with the intelligence dynamically to grow,

new rules could be generated automatically as risk values change. This would, in turn, have the effect that the **RB** and **KB** would not be "read-only" any more and that the system administrator's job would be minimised in maintaining the **RB** and **KB**.

Be that as it may, RtRA is expected to have a significant impact on future technologies. Some of the security problems that stand to be minimised include hacking (for instance, eavesdropping and message interception), as well as the encryption of the necessary data. The only question left is this: the theory behind RtRA proved that it could work, but would it work in practice? The prototype attempted to prove the latter. It is now up to researchers, system analysts and programmers to let RtRA come into its own and actually make its potentially significant contribution to the domain of network security.

5. LIST OF SOURCES CONSULTED

[COME 97] COMER, D.E.; 1997; *Computer Networks and Internets*; "Encapsulation"; ISBN 0-13-239070-1; New Jersey: Prentice Hall; p. 230.

[DERU 97] DE RU, W.G.; ELOFF, J.H.P.; November 1997; *Computers and Security*; "Risk-analysis modelling with the use of fuzzy logic"; Vol. 15 no. 3; pp. 239-248.

[EAGL 98] RAPTOR SYSTEMS; 1998; *Technical White Paper: The Eagle 5.0 Firewall*; "Overview of Eagle 5.0 Features"; http://www.raptor.com.

[FIR1 99] CHECKPOINT SOFTWARE TECHNOLOGIES LIMITED; 1999; *Firewall-1*; www.checkpoint.com.

[FORD 97] FORD, W.; BAUM, M.S.; 1997; *Secure Electronic Commerce*; "Packet Encryption"; ISBN 0-13-476342-4; Prentice Hill; pp. 149-150.

[FWTK 99] TRUSTED INFORMATION SYSTEMS INCORPORATED; 1999; *Firewall Toolkit*; www.tis.com.

[IBMC 97] IBM CONSULTING GROUP; 1997; *IBM Firewall Version 3.2 for AIX at a Glance*; "What is a firewall?"; International Business Machines Corporation; Second edition; pp. 5-7; http://www.computerps.com/internet/security/firewalls/.

[INTE 98] INTERNIC; 20 March 1998; *Internic 15 Minute Series*; "What is a Packet?"; http://krikkit.tss.nwu.edu/dss/training/internic/.

[LABU 98] LABUSCHAGNE, L.; ELOFF, J.H.P.; 1998; *Computers and Security*; "The Use of RtRA to Enable Dynamic Activation of Countermeasures"; Vol. 17 no. 4; pp. 347-357.

[PABR 96] PABRAI, U.O.; GURBANI V.K.; 1996; *Internet & TCP/IP Network Security*, "TCP/IP and Security"; ISBN 0-07-048215-2; McGraw-Hill; pp. 69-74.

[PFLE 89] PFLEEGER, C.P.; 1989; *Security in Computing*; ISBN 0-13-799016-2; pp. 3-4.

[PRIC 98] PRICE, K.; 1998; *Intrusion Detection*; "Characteristics of a Good Intrusion Detection System"; http://www.cs.purdue.edu/coast/intrusion-detection/.

[RAPT 98] AXENT TECHNOLOGIES; 1998; *Raptor Firewall*; "Raptor Firewall 5.0 White Paper"; http://www.axent.com/product/rsbu/firewall4nt/default.htm.

[RFAQ 98] RAPTOR SYSTEMS; 1998; *Raptor Firewall 5.0 Frequently Asked Questions (FAQ)*; "What is the Raptor Firewall 5.0 for Solaris?"; http://www.raptor.com/products/solaris5/s50faq.html.