An Architecture for QoS-capable Integrated Security Gateway to Protect Avionic Data Network

M. Mostafa, A. Abou El Kalam, C. Fraboul

Université de Toulouse, INPT-ENSEEIHT, IRIT-CNRS, 2 rue Camichel Toulouse – France
{firstname.lastname}@enseeiht.fr

Abstract: While the use of Internet Protocol (IP) in aviation allows new applications and benefits, it opens the doors for security risks and attacks. Many security mechanisms and solutions have evolved to mitigate the ever continuously increasing number of network attacks. Although these conventional solutions have solved some security problems, they also leave some security holes. Securing open and complex systems have become more and more complicated and obviously, the dependence on a single security mechanism gives a false sense of security while opening the doors for attackers. Hence, to ensure secure networks, several security mechanisms must work together in a harmonic multi-layered way. In addition, if we take QoS requirements into account, the problem becomes more complicated and necessitates in-depth reflexions. In this paper, we present the architecture of our QoS-capable integrated security gateway: a gateway that highly integrates well chosen technologies in the area of network security as well as QoS mechanisms to provide the strongest level of security for avionic data network; our main aim is to provide both multi-layered security and stable performances for critical network applications.

Keywords: Integrated security gateway; session table; QoS; IPSec; VPN; firewall; NAT; IDS.

1. Introduction

Current commercial aircraft data networks are IP based [1]. These e-enabled networks not only allow passengers to access the Internet but also they have a lot of other benefits. The conversion toward e-enabled aircraft encourages the use of new applications on-board the aircraft such as Electronic Flight Bag (EFB), enhanced In-Flight Entertainment (IFE), video surveillance and VoIP. However, the introduction of IP/TCP/UDP, SNMP and TFTP protocols inside the aircraft network exposes this sensitive system to new types of attacks [2]. Many security mechanisms and solutions have evolved to deal with the ever increasing number of network security attacks. Although these conventional solutions have solved some security problems, they also leave some security holes [3], especially when they are ineffectively combined. Besides that, the dependence on a single security mechanism such as firewalls will not provide a real secure network. Worst, Hackers have devised sophisticated attacks that easily circumvent traditional security mechanisms. Dealing with this issue, this paper presents the architecture of a QoS-capable integrated security gateway: a gateway that merges well-chosen technologies in the areas of network security and QoS to provide multi-layered security and ensure in-depth defence strategy that respect performance requirements and ensure availability of the critical traffic.

The remainder of this paper is organized as follows: in Section 2, we provide a survey of current network security mechanisms and highlight their strength and weaknesses. Section 3 presents the most relevant QoS mechanisms; Then, Section 4 presents our QoS-capable integrated security gateway. In Section 5, we have adapted a case study to show how our gateway can protect avionic network. Afterwards, Section 6 briefly describes the related works. Finally, we draw some conclusions and perspectives for future work in Section 7.

2. Conventional security mechanisms

In this section we present some existing conventional security mechanisms. For each one, we provide a brief description and present the benefits and limitations.

2.1 Firewalls

Firewalls are network devices that filter network traffic at one or more of the ISO seven layers network model, most commonly at the network, transport, and application levels [4]. Basically, there are four basic types of firewalls: packet filtering firewalls (**PFs**), circuit proxy firewalls (**CPFs**), application proxies firewalls (**APFs**) and, the most widely used type, stateful inspection packet filtering firewalls (**SPFs**). We will briefly describe these techniques and present their pros and cons.

Packet Filtering Firewall (PF): PFs were the first generation of firewalls. They are basically screening routers [5] that control the flow of data in and out of a network by looking at certain fields in the packet header: *Source IP Address, Destination IP Address, Protocol identifier, Source port number,* and *Destination port number*.

The **PF** inspects all incoming and outgoing packets and applies the specified policy (e.g., drop, reject or accept the packets).

PF was considered as an efficient, fast, and cost effective solution since a single router can protect an entire network. However, **PF** has a lot of limitations: it is based on IP addresses without any authentication; it cannot defend against man in the middle attacks and forged packets with spoofed IP addresses; it depends on port number for identifying communicating applications and this is not a reliable indicator because many current protocols such as network file system (NFS) use varying port numbers. But the most important limitation is the difficulty of writing correct filters [6] for complex and permanently evolving systems. Generally, filtering rules are far from providing perfect security against holes in the **PF**.

Stateful Inspection Packet Filtering Firewall (SPF): While **PF** works by statically inspecting each packet against the rule set, **SPF** works not only by inspecting the packet headers but also by correlating the incoming traffic to the earlier outgoing requests [7]. Basically, **SPF** builds dynamic session table to record relevant information of each communication to trace the validity of each packet in these connections. **SPF** dynamically opens and closes ports according to the connection needs. In this way it makes filtering and network management easier.

While **SPF** can protect against some attacks that exploit weaknesses in the network level protocols, it cannot provide protection at the application level. Application defence require more awareness of the payload content.

Circuit Proxy Firewall (CPF): These types of firewall work as relaying agents between the internal and the external hosts [8]. The idea is to protect internal hosts from being directly exposed to the outside world. The **CPF** accepts requests from internal hosts for connections to the outside world, destroys the original IP and transport layer header, encapsulates the payload in newly constructed headers with their own IP addresses and finally sends the request with the changed IP to the outside server. In addition, **CPF** requires authentication before establishing connections. While CPFs can support large numbers of protocol as they do not have to understand application level protocols, they open security holes as they cannot provide defenses against some application level attacks. Moreover, they can allow malicious contents to be passed without filtering.

Application Proxy Firewall (APF): **APF**s (also called *application-level gateways*) operate on Layer 7 of the OSI model. Like **CPF, APF** acts as an intermediary between internal and external hosts [9]. **APF** provides all the services provided by **CPF**. In addition, **APF** is application level protocol aware. In this way, **APF** can inspect application level commands and discard malformed ones. The main drawback is that a separate application-proxy must be written for each application type being proxied. In addition, the applications must be modified to work with **APF.** Finally **APF** is not efficient against several malwares.

2.2 Network Address Translation (NAT)

NAT is an IETF [10] standard that enables a local area network (LAN) to modify network IP addresses and ports numbers in headers of datagram packets (in transit across a traffic routing device) for the purpose of remapping a given address space into another. One of the main objectives of NAT is to solve the scalability problem when the number of IP addresses allowed to access the external network is limited. From the security point of view, NAT more or less hides internal private network addresses from outsiders, enforces control over outbound connections, and restricts incoming traffics. However, even for these objectives, NAT is sometimes not very efficient and cannot provide defenses against malformed packets, application level attacks and malwares.

2.3 Virtual Private Network (VPN)

A VPN [11] is a means of connecting to a private network using a tunnel over a public network, such as the Internet. VPNs can use authenticated links to ensure that only authorized hosts can connect to the private network, and can use encryption to ensure the confidentiality of the transmitted data. VPNs can be built at different layers with different protocols, e.g., L2TP (level 2), MPLS (level 2.5), IPSec (level 3), SSL/TLS (level 4), SSH (level 5). In this section we will talk about IPSec and SSL/TLS, the most commonly used security protocols to carry out VPNs at layers 3 and 4 respectively.

IPSec: IPSec is the de-facto standard for network security [12]. It can be seen as a framework that provides the following security services: (1) access control, (2) data origin authentication, (3) anti-reply integrity, (4) connectionless integrity, and (5) data confidentiality.

However, with IPSec, it is difficult to exercise control on a per user basis on a multi-user machine as IPSec is implemented at the network layer. Moreover, IPSec cryptographic algorithms add overhead to the network and application traffic; hence the use of hardware accelerators is usually required. Furthermore, while IPSec can prevent some DoS attacks, it can't stop all of them. Finally, IPSec protects the packet regardless of its content; this means that it does not protect against malformed headers or malicious contents.

Transport Layer Security (TLS / SSL): **SSL** stands for Secure Sockets Layer, though IETF has renamed it **TLS** (Transport Layer Security) [13].

TLS / SSL is a protocol that provides security services such as authentication, integrity and confidentiality on top of the Transport Control Protocol (TCP).

TLS needs to maintain the context for a connection and is not currently implemented over the User Datagram Protocol (UDP), as UDP does not maintain any context. As this security mechanism is transport protocol specific, security services such as key management may be duplicated for each transport protocol. Another limitation of transport layer security is that the applications still have to be modified to request security services from the transport layer. Also, some TLS uses and configurations can be subject to some attacks such as man in the middle attacks.

2.4 Network Intrusion Detection System (NIDS)

A NIDS is a packet sniffer that passively inspects all inbound and outbound network traffic and notifies network administrators when it identifies suspicious patterns that may indicate an attack [14]. The two main categories of IDS detection algorithms are signature-based and anomaly-based. Signature-based IDS examines packets for well known attack signatures, while anomaly-based IDS detects unusual behavior, usually based on statistical methods. Signature-based IDSes are in general not capable of detecting new attacks, while anomaly-based IDSes suffer form high rate of false alarms. Correlation techniques try to overcome these kinds of limitations but do not completely resolve IDSes limitations and issues. Note that IDSes are processing intensive. For this reason, they are generally placed in parallel with the data stream. The major limitation of IDSes is their inability to instantly drop suspicious packets or stop attacks as it is not placed in-line with the data stream.

2.5 Intrusion prevention system (IPS)

Unlike its predecessor IDS, IPS is placed in-line with the data stream [15]. The main aims are to actively analyze traffics and take protective actions like dropping suspicious packets and closing some ports (sources of attacks). The application specific integrated circuits (**ASIC**) [16] technology makes IPS processing much faster. IPS can reassemble the traffic stream and look at application commands to detect suspicious fields. However, IPS does not perform detailed analysis at the file level, which is required to detect the large number of malwares.

2.6 Anti-Malware Gateway (AMG)

A malware is a general term for software programs that have been designed with or can be used for malicious intent [17]. **AMG** is used to scan files, e-mail attachments, and web pages objects against a large database of malware signatures. Generally, AMGes are time consuming especially if they are used to defend against a variety of malwares such as viruses, worms and Trojans. As with IDS / IPS, the main disadvantage of **AMGes** is that generally they cannot protect against 0-day and newly developed malwares and attacks which do not have a known and integrated signature.

After discussing the state of the art of the commonly used security mechanisms, let us now deal with QoS-related mechanisms.

3. Quality of Services (QoS) mechanisms

QoS is a heavily loaded term with many different meanings depending upon the specific context. IETF [18] has defined QoS as nature of the packet delivery service provided, as described by parameters such as achieved bandwidth, packet delay, and packet loss rates. In this paper, we focus on QoS mechanisms such as priority classification, rate limiting as well as queuing and scheduling.

3.1 Priority classification

The main goal of QoS is to provide priority including dedicated bandwidth, controlled jitter and latency, and improved loss characteristics. A QoS policy should identify what priority level to be given for each traffic flow. After that, classification algorithms [19] can be used to inspect each packet and mark it with its associated priority level. For example, high priority traffic such as VoIP should be served before non-priority one such as e-mail or FTP packets.

3.2 Queuing and Scheduling

Queues represent locations where packets may be held (or dropped). Packet scheduling refers to the decision process used to choose which packets should be serviced or dropped. Buffer management refers to any particular discipline used to regulate the occupancy of a particular queue. Packets will be placed in different queues according to their priority levels. Afterwards, schedulers will pick packet to be served according to their priorities. The most important objectives of scheduling are computational efficiency and fairness [20]. In this work we depend mainly on two scheduling algorithms: CBQ (class-based queuing) and LLQ (low latency queue). CBQ provides fine granularity of bandwidth sharing and traffic priority control [21]. It divides a network bandwidth among multiple queues or classes. Each queue has a traffic assigned to it, based on source

or destination address, port number, protocol, etc. Queues are also given a priority, so that, those containing high priority traffic are processed ahead of queues containing low priority traffic. LLQ [22] creates a low latency queue dedicated to real time applications traffic. LLQ serves highly delay sensitive real time traffic faster than the other queues.

3.3 Rate limiting

Rate limiting is used to control the rate of traffic sent or received over a network interface [23]. Traffic that is less than or equal to the specified rate is sent, whereas traffic that exceeds the rate is simply dropped or delayed.

Variations of token bucket algorithms are used to control network traffics. A token bucket algorithm [24] uses the analogy of a bucket to control the rate of requests. A bucket has a size (B), which corresponds to the number of tokens that may fit into the bucket. The size of the bucket also corresponds to the maximum burst size. Each request that comes in consumes some tokens from the bucket. When no more tokens are left in the bucket, the request cannot be processed currently and must either be rejected or queued. Tokens regenerate at a rate (R), which corresponds to the maximum average sustained rate of traffic. In any interval (t), a maximum of (B + Rt) tokens may be consumed. This allows for bursty output behavior while constraining the average throughput.

4. QoS-CAPABLE INTEGRATED SECURITY GATEWAY ARCHITECTURE

As cited above, each of the previously mentioned security mechanism has its own limitations. Since we cannot depend on a single mechanism to protect our networks, the need for an integrated solution has become a must. Besides that, most of the existing applications such as embedded avionic networks have both security and QoS requirements, while current solutions only deals with one of these requirements. In this section, we present the architecture of our QoS-capable integrated security gateway; a security gateway that homogeneously and efficiently merges the most relevant security and QoS mechanisms to provide in-depth layered security solutions with appropriate QoS mechanisms ; the aim is to ensure not only availability, integrity, confidentiality of critical network traffics but also achieve temporal constraints. In fact, this integrated solution satisfies easy and cost effective security and QoS management, and fast response.

Figure 1 represents the architecture of QoS-capable Integrated Security Gateway. Solid arrows represent data flow while doted arrows represent control flow. The architecture of our secure and QoS-aware gateway consists of the following units:

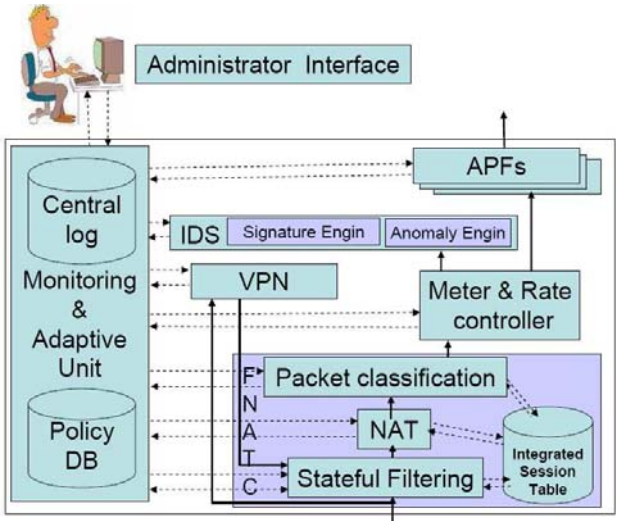

Figure 1: QoS-capable Integrated Security Gateway Architecture.

4.1 Filtering, Network Address Translation, and Classification (FNATC) unit

Linked to the network interface, this component can be considered as the first layer of defense in our architecture. It receives all incoming packets and offers three main functions: firewall filtering, NAT translation, and QoS classification. Traditionally, theses functionalities are achieved separately, which causes redundant steps and hence more processing time. Conversely, in our view, since these three functions and routing depend heavily on searching and classification algorithms to identify which flow the packet belongs to, we merged them into a single unit to improve the performance. In this way, we will not repeat the classification task for each function; we only classify one time and the result of our classification is used for firewall filtering, NAT translation, QoS classification and routing. To achieve this goal, we built an integrated session table.

Integrated session table: Figure 2 shows the architecture of the integrated session table; due to column wide space limitation the table is divided into two parts.

Session ID					**NAT info.**	
Lan addr	Lan port	Ext addr	Ext port	Proto	Gwy addr	Gwy port

Stateful filtering info.		**QoS**	**Routing info.**	
State …	Time	DSCP	Ext- next-hop	Lan-next-hop

Figure 2: Integrated session table architecture.

The first five fields of the table constitute session ID. Lan-addr and lan-port are the internally private source IP address and source port number while ext-address and ext-port are external communicating host destination IP address and destination port number. Finally, proto field is the protocol identifier that is used to refer to the transport layer protocol in use. These five fields are used to identify traffic flow.

The second part of the table contains NAT information necessary to perform mapping between LAN private address and Gateway public address. gwy-addr and gwy-port are the NATed publicly available address and port number.

For stateful filtering, session state (such as sequence number and TCP flags) and session time out are stored in state and time fields.

For QoS classification, the QoS DSCP [25] priority value will be stored in the DSCP field.

For routing, ext-next-hop field will be used to send the packet to the next destination -outside the protected network. While lan-next-hop, field will be used to send the packet to the next destination inside the protected network.

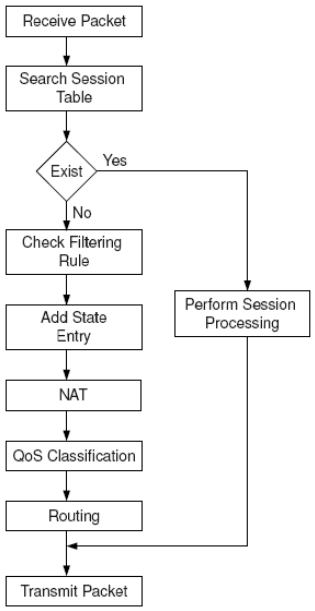

Figure 3: Integrated session table processing.

Figure 3 represents the integrated session table processing. For incoming packet, the session table will be searched. If there is no entry in the session table, this means that it is the first packet in this session. So first, the packet will be validated against the filtering rules. If it is allowed to pass, an entry in the session table will be added; then NAT translation will be performed to make the necessary mapping between external and internal addresses. After that, QoS classification will be performed and the DSCP will be added to its field in session table.

Finally routing table will be searched to obtain the next hop values.

If an entry is found in the session table (ongoing packet) the packet will be inspected to ensure its conformance to the session state and all the needed session processing (NAT, QoS priority classification, and routing decision) will be performed in one shot without further research overhead, as all the needed information is available from the single lookup in the short session table. In addition, hashing function is used to accelerate session table lookup; this is clearly a great enhancement which save processing time and increase performance.

The worst case session table look up time is *O(log s)*; where s is the number of session table entries. It is very low if compared with the worst case linear search time needed to perform the four functions, $T= O(f) + O(n)+ O(c)+ O(r)$; where f, n, c, and r are the lengths of filtering, NAT, classification and routing tables and rules entries. This clearly gives processing gain and enhances the performance.

Packet filtering sub-unit: The Stateful Inspection Packet Filtering Firewall (SPF) sub-unit inspects all incoming and outgoing packets; if it is identified as the first packet in the connection, SPF will be obligated to search a large set of access control rules (stored in the centralized policy database) to decide if the packet is allowed to enter the protected network. The SPF keeps state information (such as TCP sequence number, window size and flags) for all current connections and least recently rejected connections in the integrated session table. For any packet that belongs to existing connection, only the session table will be searched to ensure that the state of the packet is in accordance with a legal connection. SPF is responsible for detecting all types of network level attacks (i.e., flooding, spoofing, etc.). SPF performs IP defragmentation and protocol normalization to detect and remove ambiguity [26]. Actually, Packet defragmentation is also needed for proper functioning of NAT component. If a suspicious packet is detected, it will be dropped and in some cases the connection will be blocked and logged.

NAT sub-unit: In this architecture, NAT will be provided as optional service to protect internal network addresses from exposure to external world. NAT will translate internal private IP addresses to their associated public addresses and vice-versa. Clearly, any incoming packet that has not internally associated address will be dropped and logged in the central log (another component of the architecture) for further auditing. After establishing a connection, the NAT associated internal and public addresses are kept in the integrated session table to make ongoing processing of current connections faster.

Classification sub-unit: Each accepted packet must be classified and given a QoS priority level. This priority level will affect the treatment of the packet. Basically, high priority packets such as VoIP packets will be served faster than low priority packets such as e-mail packets. Different queuing and scheduling algorithms (such as CBQ and LLQ) are used to achieve this goal. As we mentioned before, the QoS priority value will be stored in the integrated session table to make its retrieval performed in single search process as described above.

4.2 VPN unit

If the packet is cryptographically protected by IPSec, it will be sent to the IPSec VPN specific module for cryptographic processing. In our architecture, to allow adequate classification of IPSec encrypted packets; we had designed and implemented Q-ESP (QoS-friendly Encapsulated Security Payload) protocol [27]. Q-ESP is a security protocol that, not only provides the same security services provided by IPSec ESP and AH simultaneously, but also allows QoS classification. QoS classification is not allowed by the IPSec ESP protocol as it encrypts transport layer header, which contains necessary fields needed for QoS classification. To overcome this limitation, and to give network control devices the ability to classify IPsec encrypted traffic, the Q-ESP protocol copies the source and destination port number from the transport layer protocol header and place them in clear (without encryption) in its own header. Figure 4 represents Q-ESP packet format in IPv6.

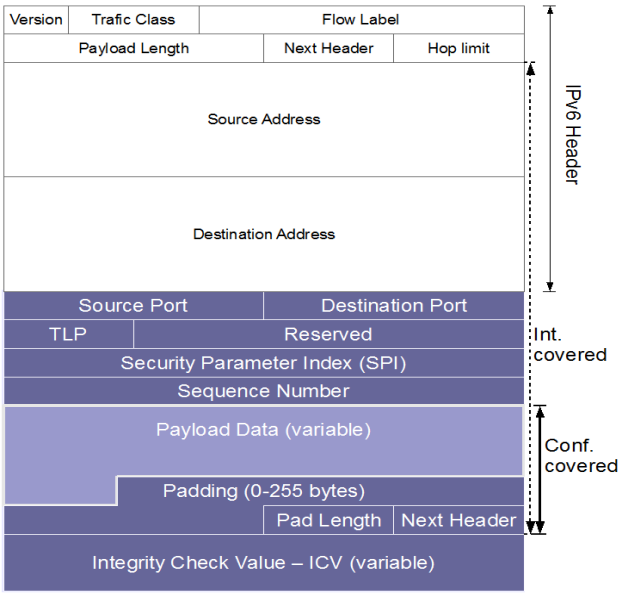

Figure 4: Q-ESP packet format in IPv6.

Moreover, in order to ensure a higher level of availability, we expect using a hardware support where the Q-ESP processing will be carried out in full line speed.

After applying the concerned cryptographic processing and extracting the protected packet. The packet will be sent to the FNATC unit for processing.

4.3 Meter and rate controller unit

To ensure stable state of the traffic and to guarantee the fair sharing of the network bandwidth, we suggest using a meter and rate controlling unit to monitor and control the traffic. This component is placed just after FNATC unit, as it will monitor and control the flow of accepted, NATed and classified packets. Basically, a “rate controller” is a policer which controls and limits the throughput rate to exactly what the flow can send and receive. The meter makes the necessary accounting process and notifies the monitoring unit about the state of the whole traffics or specific flow. In this way, we call on a QoS mechanism to satisfy the availability security property; in fact, with meter and rate controller, we can detect some DoS traffic that behaves abnormally (trying to consume network resources) early and more easily. More precisely, while the meter can detect such abnormal traffic, the rate controller can prevent it.

To illustrate this, let us take the case where a mal-behaving unit (due to system failure) starts sending traffic that may saturate the network bandwidth. Even if it may succeed in consuming the bandwidth allocated for this traffic flow, it will not be able to affect the bandwidth allocated to other traffic. Hence, this can effectively protect the overall network infrastructure. Note that the level of granularity in applying bandwidth management is a critical factor. Finer assignment of bandwidth will naturally help to ensures more control over the network resources but may incur more management overhead. Actually in avionic networks per flow specification is performed, calculated and simulated offline to ensure safe and secure operation of the flight.

4.4 Application Proxy Firewall (APF)

By contrast to IDS, APF uses a positive security model. Rather than trying to define every possible bad thing that could occur, they “learn” legitimate application behavior on the fly, ensuring that every user request conforms to expected application usage and that only valid traffic is passed to backend servers. In addition, APF performs user authentication to prevent non-authorized users from accessing protected applications.

APFs act as a proxy between users and applications; each sent request will be intercepted and analyzed to ensure its proper conformance with the pre-specified rules and command signatures. Malformed commands will be rejected.

APF examines control and data fields within the application flow to verify that the actions are allowed by the security policy and do not represent a threat to end systems. By understanding application-level commands and primitives, they can identify content out of the norm and content that represents a known attack or exploit.

4.5 Intrusion detection System (IDS)

This system performs deep packet inspection. The IDS uses two inspection engines: signature-based and anomaly-based. Both engines have access to the central log file and can analyze its content to make correlations. Both engines log their activities and eventually notify administrators for some specific and critical attacks. In the avionic context, we can imagine an architecture where the IDS sensors are located within the Passenger network (PN), crew network (CrN), and the control network (CoN). In addition, another sensor would be located within the aircraft access network to ascertain if malicious traffic is introduced into the CrN and/or the CoN.
SPADE (Statistical Anomaly Detection Engine) is an example Snort pre-processor plug-in that is adapted to avionic networks [28]. The anomaly detection engines supplement the classical Snort attack signatures. SPADE assigns anomaly scores to packets. The anomaly score is based on the observed history of the network. The fewer times that a particular kind of packet has occurred in the past, the higher its anomaly score will be. Moreover, SPADE maintains a probability table that reflects the occurrences of different kinds of packets in history, with higher weight for more recent events. It calculates a raw anomaly score directly from the probability of anomaly.

4.6 Monitoring and adaptive unit

This unit monitors the internal state of the gateway and enforces adaptive actions if necessary. The monitoring unit contains the central log and the integrated policy database. This unit intercepts all logged activities and analyses them. It may also receive information about the state of the protected network. It manages and enforces the cooperation between the different units in the gateway. For example, when IDS detects an attack, it logs the action and notifies the monitoring and adaptive unit. Based on this notification, the unit will modify SPF rules to drop all packets from this connection. Furthermore, the adaptive unit can dynamically change the VPN cryptographic key lengths to shorter keys, in case of heavy traffic load, to gain some processing time; inversely, it may change the keys to be longer to give more security protection. Hence, to provide dynamic control with adaptive solutions, the central policy database has been constructed in this unit. It is an integrated policy repository which contains QoS, NAT, VPN, and all security.

4.7 Administrator interface

It is a graphical user interface that enables administrators to set policies and perform auditing tasks. It should be scalable and flexible enough to allow administrators to add custom rules or new signatures. The added policies should be verified off-line to be conflict free. Policy conflict occurs when the objectives of two or more policies cannot be simultaneously satisfied. For more details about conflict detection and resolution refer to [29]. In addition, the interface should include analysis tools (such as SEC [30]) to enable powerful auditing of the central log.

5. Case study: Securing Avionic data network

In this section we present a case study in which our solution has been adapted to protect the avionic data network

5.1 Aircraft Data Network Architecture

As shown in figure 5, the on-board avionic network is divided into four main domains [31]:

1. Aircraft Control Domain: contains the flight and embedded control system where the aircraft is controlled from the flight deck, and cabin core system, which provides environmental functions dedicated to cabin operation, such as environmental control, passenger address, smoke detection, etc.
2. Airline Information Service Domain: provides operational and airline administrative information to both the flight deck and cabin. It also provides information to support the passengers.
3. The Passenger Information and Entertainment Service Domain: provides In Flight Entertainment (IFE), Internet access and support for passenger owned devices.
4. Passengers owned Devices Domain (PDD): contains the passenger's owned laptops and digital personal assistant devices.

In some next generation and future avionic architectures, each of these networks may require connectivity with each other and with ground-based computing networks. The Common Network Interconnects is used to connect the different sub-networks with Ground Networks. These links can be shared with appropriate care to QoS and security considerations. Furthermore, it is clear that inter-domain connections require a security perimeter at the border of the connected domains, incorporating network routing, QoS and security services.

5.2 The proposed solution

First, it is worth noting that, avionic network uses VLAN between the aircraft domains and the smart switch; the aim is to provide each domain access only to those links that can support the associated class of traffic. The aircraft control domain uses "Avionics Full-Duplex Switched Ethernet" (AFDX), further limiting the ability for intrusion by a non-configured user.

Second, we propose to place our QoS capable integrated security gateway at the border of each sub-network domain. Our aim is to protect each domain from external attacks while respecting the QoS requirements. Figure 5 shows the location of our gateway inside the avionic network.

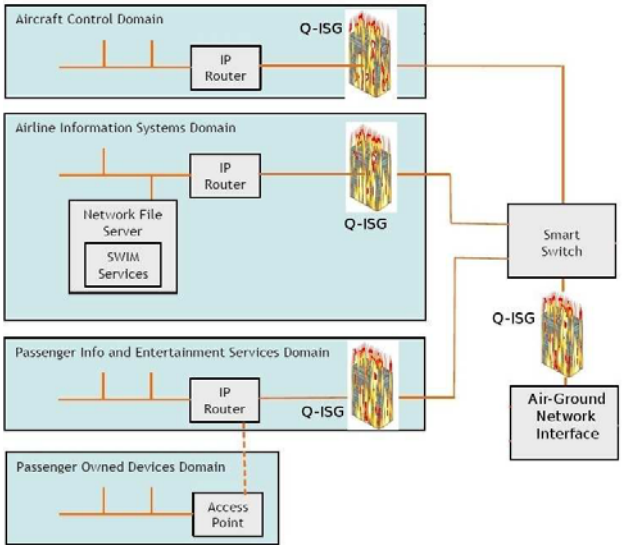

Figure 5: The use of QoS-capable integrated security gateway (Q-ISG) in the avionic network

Figure 6 illustrates the complete packet processing flows of the QoS-capable integrated security gateway. Incoming packets will be processed as follow:

When the FNATC unit receives the packet, the rule set will be verified to see if this connection is accepted, If the packet is part of a VPN (i.e., is protected by Q-ESP or AH) it will be dispatched to the VPN module for cryptographic processing. In our case the VPN must use a hardware accelerator that works in the full line speed to prevent the possibility of denial of service and to reduce processing delays. After extracting the protected packet, the VPN module sends it to the FNATC unit.

Non-cryptographically protected packet will be handled as follow: the state table will be searched to see if there is an entry for this connection or not, if there is no entry in the state table, this is the matter with the first packet in a connection; thus the rule set will be verified to see if this connection is accepted, if the packet is accepted, an entry will be added in the state table for this connection and the need information to perform NAT will be added. After that, the QoS priority level will be identified and added to the same entry in the state table. If the packet is not allowed, it will be dropped and logged in the central log. For example, if the packet does not have internally associated address, it simply will be dropped and logged in the central log for further auditing.

If the packet is not the first one in the connection, an entry will be found in the state table and all the needed information to perform the state inspection, NAT mapping, QoS priority assignment and routing will be available. Note that contrarily to existing solutions, these information are available only after four different steps while (in our solution) they are extracted in only one step; which is a great saving in processing time and space. SPF will inspect the packet to ensure that the state of the packet is in accordance with a legal connection. If a suspicious packet is detected, it will be dropped and logged. In our case packet fragmentation is not allowed so that all communicating parties must respect the announced maximum transmission unit MTU. In this way, we prevent all attacks based on fragmentation and save the de-fragmentation time.

According to the associated priority level the packet will be placed in the suitable queue for further processing. For example, high priority traffic such as flight control command will be placed in the LLQ to be served before any other traffic as it has the highest priority.

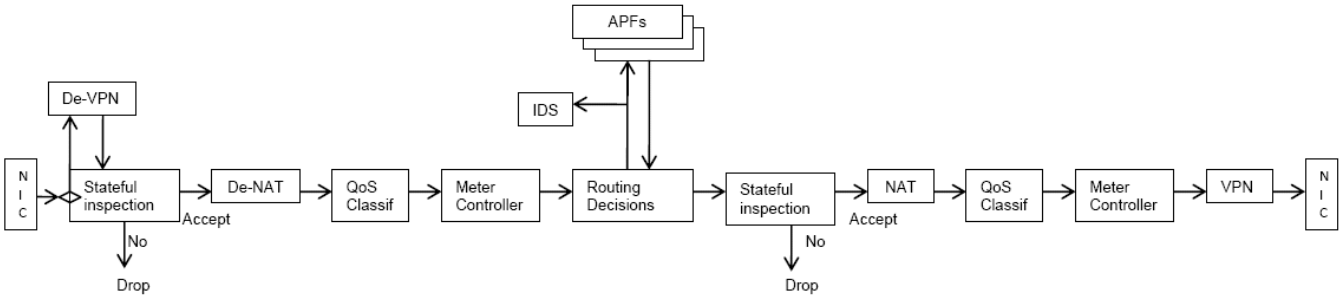

Figure 6: complete packet processing flows of the QoS-capable integrated security gateway.

Accepted packets will be forwarded to the meter and rate control unit to measure it and to ensure that it is within the specified limit. Flow that exceeds the allocated limits will be dropped, in this way; misbehaving units that send out of limit traffic or perform denial of service attacks will not affect other traffic. After that the packet will be directed to the concerned APF and a copy of the same packet will sent in the same time to the IDS. In our case, IP routing between interior and exterior networks is completely disabled, and all desired services (e.g. SNMP, TFTP and other applications) are provided by APFs. They will first authenticate users trying to connect to these services. Furthermore, the APFS will inspect applications commands and data to ensure their safety. All types of files that may contain malicious contents such as .exe, .com and .doc are prohibited. Only files that known to be saver such as .txt and .bmp are allowed finally, if all the previous steps succeeded, the packet is forwarded to its destination.

Our IDS is protocol specification aware. First, the application level protocol is identified, and then, the packet is inspected against the set of rules and signatures associated with this protocol. Moreover, both anomaly and signature based engines could inspect the packets and the central log to make correlations. Any signs of attacks will be alarmed. Trace out mechanism is used to identify the source of attack.

In our case the policy database will contain different policies; each one is suitable for a specific context. For example, the aircraft has different operational mode: some activities which are allowed when the aircraft is in the flight mode may not be allowed when the aircraft is in the maintenance mode. Consequently, the monitoring subunit will monitor the state of the aircraft and notify the adaptive subunit of each change in the aircraft state. Based on the state information, the adaptive subunit will enforce the suitable policy in each context.

The administrator interface will be used to set policies and modify rules. The system can detect any conflict between policy rules and could suggest other alternatives to resolve this conflict. It is a helpful tool to assist the administrator in setting conflict-free policies.

Furthermore, the central log will be available to the administrator for further analysis and investigation with the aid of the log analysis tool.

While flying, a synchronized copy of the log file will be sent to the flight controller in the airport. This will give the ability for corrective actions and make e-enabled aircraft safer.

6. Related work

While a lot of research have been done in the area of security mechanisms, little research have been carried out in the area of integrated security gateway with QoS requirements. In our context, we deal with both security and QoS requirements and mechanisms.

AEEC [31] has suggested reference security model for communication between the networks of different security levels in the aeronautical networks. The described security gateway contains packet filtering firewall, IPSec VPN and application proxy firewalls. Like our model all internally protected services are provided only through APFs. In comparison to our work this model does not consider the use of QoS mechanisms (i.e., meter and rate controller) to prevent DoS attacks and ensure availability. Although they considered the uses of packet filtering firewall, it is not clear if it is a stateful one or not, while in our work we show the importance of this type of filtering firewall and extended its session table to accelerate the processing of other important functions (NAT, QoS classification, and routing). This actually gives a great chance to enhance performance. In addition, we suggested the use of IDS to the purpose of on line detection and trace out of attacks and their sources. More over, we used monitoring and adaptive unit to manage the overall gateway security and QoS policy in an adaptive way and to provide central log for further auditing.

7. Conclusion

In this paper we presented the conventional network security mechanisms and we have identified their limitations. Since we cannot depend on a single security mechanism to protect our network, we have shown the architecture of our QoS capable integrated security gateway. This architecture provides multi-layered protection and implements QoS mechanisms to ensure proper functioning of the gateway. This architecture can benefits greatly from advancement in hardware technology such as ASIC and coprocessors. Actually, we had started a kernel based implementation of the architecture; both IPSec Q-ESP VPN and FNATC components were implemented successfully. Currently we are completing the implementation phase and optimizing our code. The next step is to perform security and QoS performance evaluation experiments to show the soundness of this architecture. As additional perspectives, we will search how to integrate the QoS capable integrated security gateway with distributed intrusion detection sensors. Further work is also needed to integrate the gateway QoS and security policies with the overall system policy.

8. References

[1] N Thanthry, M.S. Ali, and R Pendse, “Security, Internet Connectivity and Aircraft Data Networks,” IEEE Aerospace and Electronic System Magazine, November 2006

[2] Reinhart, Tod; Boettcher, Carolyn; Gandara, G A; Hama, Mark; “Defining a Security Architecture for Real-Time Embedded Systems”. Report of AIR FORCE RESEARCH LAB WRIGHT-PATTERSON AFB ON EMBEDDED INFORMATION SYSTEMS BRANCH. Jun 2004.

[3] Albin Zuccato, “Holistic Security Requirement for Electronic Ecommerce”, Computer Security, 23, 2004

[4] Kenneth Ingham and Stephanie Forrest, “A History and Survey of Network Firewalls”, Technical Report, TR-CS-2002-37, University New Mexico, 2002.

[5] Zwicky, E. D.; Cooper S. and Chapman D. B.: "Building Internet Firewalls", Orielly & Associates Inc., 2nd Edition, June 2000

[6] Al-Shaer, E.; Hamed, H.; Boutaba, R. and Hasan, M.: "Conflict Classification and Analysis of Distributed Firewall Policies", In IEEE Journal on Selected Areas in Communications, Volume 23, No. 10, pp. 2069 – 2084, October 2005.

[7] Siyan, Karanjit and Hare, Chris, "Internet Firewalls and Network Security", Indianapolis: New Riders Publishing, 1995

[8] Bob Stephens, “Security Architecture for Aeronautical Networks”, Proceedings of the Fourth Integrated Communications, Navigation, and Surveillance (ICNS) Conference and Workshop; August 2004; 27.

[9] Panko, “Corporate Computer and Network Security”, Prentice-Hall, 2004

[10] Egevang, K. and P. Francis, "The IP Network Address Translator (NAT)", IETF RFC 1631, May 1994.

[11] RFC 2764 A Framework for IP Based Virtual Private Networks. B. Gleeson, A. Lin, J. Heinanen, G. Armitage, A. Malis. February 2000.

[12] Kent, S., Atkinson, R., Security architecture for the Internet protocol. IETF, RFC2401,Nov. 1998. 5.

[13] Dierks, T., Rescorla, E., “The Transport Layer Security (TLS) Protocol, Version 1.2", IETF, RFC 5246, August 2008.

[14] R. Sekar, Y. Guang, S. Verma & T. Shanbhag, A high- performance network intrusion detection system, Proc. of the 6th ACM conference on Computer and communications security, 1999

[15] Xinyou Zhang, Chengzhong Li, Wenbin Zheng, Intrusion prevention system design, in: The Fourth International Conference on Computer and Information Technology (CIT’04), 2004.

[16] Konstantinos Xinidis, Kostas G. Anagnostakis, and Evangelos P. Markatos: “Design and Implementation of a High-Performance Network Intrusion Prevention System”, in Proceedings of the 20th International Information Security Conference (SEC 2005), Makuhari-Messe, Chiba, Japan, May 30-June 1.

[17] Klaus Brunnstein, “From antivirus to antimalware software and beyond: Another approach to the protection of customers from dysfunctional system behaviour”, In Proceedings of 22nd National Information Systems Security Conference, 1999.

[18] Shenker, S. and J. Wroclawski, Network Element Service Specification Template. RFC 2216, September 1997

[19] David E. Taylor, “Survey and taxonomy of packet classification techniques”, ACM Computing Surveys. 2005.

[20] John Evans and Clarence Filsfils, “Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice”; Morgan Kaufmann, 2007

[21] Risso, F. and P.Gevros; "Operational and Performance Issues of a CBQ router". ACM SIGCOMM Computer Communication Review Vol. 29, No. 5 Oct.

[22] B. Dekeris, T. Adomkus, A. Budnikas. Analysis of QoS assurance using weighted fair queueing (WFQ) scheduling discipline with low latency queue (LLQ) . ITI 2006 : Proceedings .2006. pp. 507-512.

[23] Shenker, S. and J. Wroclawski, Network Element Service Specification Template. RFC 2216, September 1997

[24] Wroclawski, J.; Specification of the Controlled-load Network Element Service, RFC 2211, IETF, September 1997.

[25] Nichols, K., S. Blake, F. Baker, and D. Black, Definition of the Differentiated Services Field in the IPv4 and IPv6 Headers. RFC 2474, December 1998.

[26] David Watson Farnam Jahanian, G. Robert Malan and Paul Howell. Transport and application protocol scrubbing. Infocom 2000.

[27] Mostafa, M.; El Kalam, A.A.; Fraboul, C.; “Q-ESP: a QoS-compliant Security Protocol to enrich IPSec Framework”, IFIP / IEEE Third International Conference on New Technologies, Mobility and Security, 20-23 December 2009.

[28] Ali, M.S. Bhagavathula, R. Pendse, “Airplane data networks and security issues. “, the 23rd Digital Avionics Systems Conference 2004, 24.-28. October 2004, Salt Lake City, Utah, USA.

[29] Hassine. MOUNGLA and. F. KRIEF; “Conflict detection and resolution in QoS policy based management”, in the IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC, September 2005.

[30] John P. Rouillard, “Real-time log file analysis using the Simple Event Correlator (SEC)”, in Proceedings of the 18th USENIX conference on System administration, November 2004.

[31] AEEC, Aircraft Data Network 664 Specification, ARINC, 2002.