Design of A High-Performance ATM Firewall *

Jun Xu and Mukesh Singhal
Department of Computer and Information Science
The Ohio State University
Columbus, OH 43210
{jun,singhal}@cis.ohio-state.edu

Abstract

A router-based packet-filtering firewall is an effective way of protecting an enterprise network from unauthorized accesses. However, it will not work efficiently in an ATM network because it requires the termination of end-to-end ATM connections at a packet-filtering router, which incurs huge overhead of reassembling and dissembling packets. Few approaches to this problem have been proposed in the literature; however, none of these approaches is completely satisfactory. In this paper, we present the hardware design of a high-speed ATM firewall that does not require the termination of an end-to-end connection in the middle. Compared with the traditional firewalls, this ATM firewall performs exactly the same packet-level filtering without compromising the performance and has the same "look and feel" by sitting at the chokepoint between the private and public ATM networks. It is also easy to manage and flexible to use.

Keyworks: High-performance ATM firewall, TCP/IP security, Packet filtering.

*This work was partially supported by NSA Grant MDA904-96-1-0111.

1 Motivation and Previous Works

1.1 Motivation for ATM firewalls

ATM is a promising cutting-edge technology aiming at integrating data, voice and video services in the same underlying communication infrastructure. It is expected that legacy TCP/IP data traffic will be carried over ATM networks thanks to the popularity of TCP/IP based Internet applications. Therefore, ATM networks are subject to all security problems that exist in router-based TCP/IP networks. As a packet filtering firewall is a very effective way of preventing a TCP/IP network from unauthorized accesses, it is desirable to apply it to ATM networks. Unfortunately, a traditional router-based firewall will not be able to deliver the filtering throughput that is comparable to even low-end ATM rate of OC-3c (155 Mbps) due to two factors. First, a packet-filtering router needs to terminate the end-to-end ATM connections in the middle in order to extract IP packets for inspection. This involves the process of reassembling incoming cells into packets and disassembling packets into outgoing cells, which incurs a huge overhead. Second, the filtering bandwidth of a traditional firewall is generally below T3 (45 Mbps), far less than the typical ATM rate of OC-3c and OC-12c (622 Mbps).

ATM Forum favors the avoidance of packet filtering by exerting discretions at the connection establishment time. Based on this principle, two alternative access-control schemes have been proposed. Smith [2] proposes that access control decisions be made at connection set up time based on the higher layer information (e.g., source and destination IP addresses, ports, etc.) contained in the information elements (IEs) of the signaling message. After the connection is established, the access control device "gets out of the way". Obviously, this is unacceptable because an intruder can always lie about the service it wants to access at the connection establishment time, and there is no way to check the content of a connection once the connection is established. Pierson [3] tries to fix this problem by proposing that "a new SVC is requested when each new service is started". However, this requires the ATM layer be notified whenever a new socket is opened, which entails considerable change to the whole TCP/IP stack and existing applications [6]. Even worse, this requires a new SVC for each transport layer flow, which leads to the problem of VC explosion. Therefore, replacing the packet level filtering with call screening will not solve the problem of access control in an ATM network.

Arguably, another alternative is to apply cryptographic measures end-to-end so that packet filtering in the middle can be avoided [7]. A centralized encrypting gateway, called encrypting ATM firewall in the literature, is employed between a protected ATM Network and untrusted WAN to perform encryption operation on behalf of all protected hosts [13]. However, authentication and encrypting do not automatically ensure proper access control. Even when a connection is authenticated, we may still want to look into its content if the parties involved do not trust each other completely. Moreover, many connections need to be established between parties who do not trust each other at all, e.g., between an Internet surfer and an HTTP server of a company. In such cases, an encrypting firewall does not solve the problem of access control.

Therefore, packet-level filtering is an indispensable access control scheme in an ATM network. As traditional firewalls can no longer perform the packet level filtering in an ATM network, a new ATM firewall architecture is called for.

1.2 Existing Approaches

At the time of writing, only one design of packet filtering ATM firewall is available, which is StorageTek's ATLAS product [5]. ATLAS is a line filter that scans an ATM physical link to perform the packet-level filtering at the rate OC-3c (155 Mbps). We are told by StorateTek that next-generation ATLAS is going to support the filtering rate of OC-12c in the near future. Two performance boosting strategies are used in ATLAS, which corresponds to the two factors that render traditional firewalls unsuitable for an ATM environment. First, to avoid reassembling and reassembling of packets, for each packet it only checks the first cell, which contains the IP header and possibly TCP ports information to determine whether or not the packet is "safe". If the packet is considered safe, all following cells that belong to it is passed or otherwise, they are dropped. Second, they use a policy cache architecture to dramatically speed up the process of deciding whether a packet header is safe or not. The core unit of this architecture is a cache block called "policy cache". Each entry of the policy cache is a combination of TCP/IP source/destination addresses/ports and option fields that are considered "safe". When the first cell of a packet arrives, it is compared with each entry in the policy cache. If cache hit occurs, cells of the packet are forwarded. Otherwise, the first cell will go through a software screening process and the other cells will be buffered in a queue. If the cell is found unsafe, the whole packet is dropped. Otherwise, the packet is allowed to pass and an entry that contains the header pattern of the packet is now added to the policy cache. The policy cache is implemented using content addressable memory (CAM), which enables simultaneous searches of all memory locations to find a match with the pattern that is searched for. Therefore, ATLAS can decide whether a packet header hits the policy cache very quickly.

However, ATLAS product does have its limitations and drawbacks. First, it does not accept IP packets with options because IP option fields can be as large as 40 bytes and may "push" the TCP header to the second cell. It may become a severe limitation in the future internetworking environments where certain options like AH [14] are used frequently. Second, it is not friendly to those who have to manage and administer it. Whenever a new PVC or SVC is established, TCP/IP rules for that VC will have to be manually configured. This is acceptable in an environment where the number of connections is small and most of the them are PVC. However, in the future ATM networks, a large number of SVCs will be established on-the-fly in a short period of time. Therefore, it is impossible for network manager to manually configure TCP/IP rules for each of them on demand.

1.3 Objectives of Our ATM Firewall Design

In this paper, we will present the logical and the physical designs of our ATM firewall. The objectives of our ATM firewall design are:

- Introduce a new ATM firewall design concept that employs security measures of different strength on traffic associated with different risk levels, in order to achieve a better tradeoff between performance and security than the packet filtering alone.
- Further improve the performance of packet filtering that is achieved by the state of the art ATM firewall – ATLAS using a novel scheme call "Last Cell Hostage".
- Filter IP packets that contain option fields efficiently using the "Last Cell Hostage" scheme.
- Make it easy for network managers to configure and administer the ATM firewall.

2 Philosophy and Logical Design of Our ATM Firewall

2.1 Firewalling Quality of Service (FQoS)

We observe that in an ATM network, IP traffic inside different connections is associated with different risk levels determined by identities of source and destination parties and Internet services requested/rendered. Based on this observation, we propose the concept of Firewalling Quality of Service (FQoS). Informally speaking, better FQoS will be applied on connections which are more "dangerous", typically at a higher time cost per packet/cell. Four FQoS classes are defined, namely, class A, B, C and D, ordered from the "safest" to "the most dangerous". Class A connections are exempt from any kind of inspections as they are considered absolutely safe. Class C connections are those that need to undergo packet-level filtering like in traditional firewalls. The risk level Class B connections is between those of class A and C. They are secured by a scheme called "traffic monitoring". This scheme is slightly different than packet-leveling; packet filtering determines whether the packet is safe before forwarding it while the traffic monitoring reverses this order. We show in Section 2.2.4 that it incurs much less latency than packet-level filtering and is almost as safe. Traffic inside Class D connections is unsafe and will involve complex protocols that is hard to be secured merely through packet-level filtering. They are secured through proxying, a scheme used in traditional firewalls to secure complex protocols such as TALK.

This classification of FQoS is not ad hoc; instead, it corresponds naturally to different types of data traffic in future networking environments. Here we assume that a connection is established between an internal party, which is located within the ATM firewall, and a foreign party, which is located without. A Class A connection typically involves a foreign party that is trusted and authenticated, e.g., a host at another site of the same company. A Class B connection typically involves an authenticated cooperating party that can be held responsible if he intentionally violates the security policy, e.g., a host that belongs to a business partner. In this case, monitoring and responding promptly is enough to track down the bad guy and guard against further attacks. A Class C or D connection typically involves a foreign party that is totally untrusted, e.g., an arbitrary Interent surfer.

The motivation to classify traffics into different FQoS classes is to optimize the effort to make connections secure. Safe connections are screened using fast but less aggressive checking method while unsafe connections are screened using aggressive but more time-consuming checking method. There are two challenges in implementing the concept of FQoS. The first is to design efficient schemes to secure each type of FQoS class (except Class A, which does not need any check) efficiently while satisfying the security level required by that FQoS. The other challenge is to design an architecture that determines FQoS for each new connection based on security policy and screen the traffic inside it using the scheme corresponding to the FQoS. We show how we address both challenges in our ATM firewall in the following section.

2.2 Logical Design of Our ATM firewall

2.2.1 General Scenario

Fig. 1 depicts the logical design of our ATM firewall. It consists of five security services that have interactions among them. Let us briefly explain its overall operation from the perspective of the life cycle of a connection. When a signaling message requesting a new SVC arrives at the firewall, the *call screening service* decides whether it is safe to establish the connection by checking the identity, address, and authentication information contained in the signaling message against the security policy governing call admission. If the connection is allowed to be established, the *call screening*

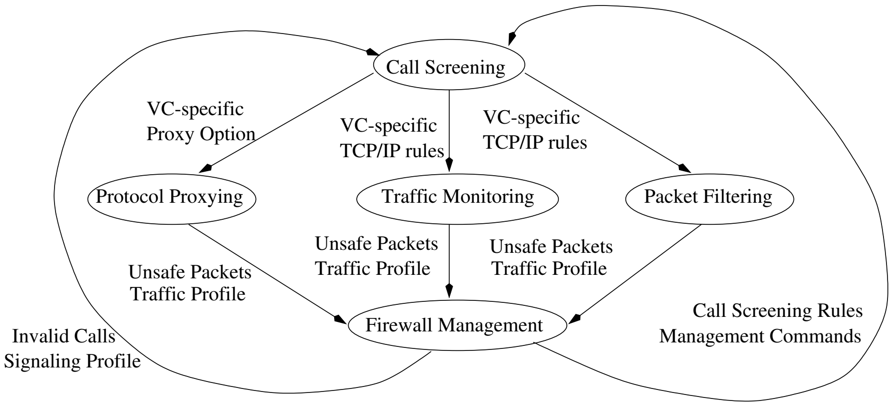


Figure 1. Logical Design of Our ATM Firewall

service assigns the connection a FQoS based on the security policy. If the FQoS of the connection is Class A, its contents will be exempt from packet-level inspection. Otherwise, depending upon whether its FQoS is Class B, C, or D, the connection will be screened by *traffic monitoring service*, *packet filtering service*, and *proxy service*, respectively. Call screening service will also supply the necessary information (e.g., packet filtering rules) to the security service corresponding to the FQoS of the connection. While the traffic inside the connection is screened, its profile information (e.g., amount of traffic within a certain time window) and packets that violate the security policy will be recorded and sent to *firewall management service*, which controls and coordinates the other security services. In the following sections, we present these services in detail.

2.2.2 Call Screening Service

Call screening service inspects whether two communicating parties are allowed to establish a connection. This is checked when a signaling message arrives at the firewall. A signaling message contains fields about source and destination identity, higher layer information such as the port(s) to be accessed, and/or a digital signature that authenticates the origin of the message. The identity of an endpoint is typically its ATM address, however, it can also be the name of the end user. The firewall keeps a set of call screening rules, each of which includes but is not limited to following six fields:

1. Source Identity
2. Destination identity
3. Decision (pass or block)
4. Authentication information (keys, cryptographic algorithms)
5. FQoS of the new connection to be established
6. Information needed for packet-level inspection

When a signaling message arrives at the ATM firewall, call screening service compares source and destination identities in the message with the first and second fields of call screening rules for a match. If no match is found or if a match is found and the value of the third field (decision) is to block the connection, the signaling message is blocked and access denied. The number of such rules may be large, however, it can be reduced by the use of wild cards. If the source identity needs to be authenticated, the call screening service expects to find the authentication information such as a digital signature in the message and checks it using the information contained in the fourth field. The signaling message is rejected if the authentication fails. If the connection is allowed to be established, the fifth field denotes the FQoS of the new connection.

If the FQoS of the connection is C, its contents are subject to packet-level filtering. In traditional firewalls, each incoming packet from the Internet is independently filtered against the same set of filtering rules because we have no clue from which IP address and TCP port does the packet come until the filtering process ends. However, in an ATM network, when a packet from a foreign source arrives over a certain connection identified by a VPI/VCI, we can figure out the IP address/prefix of this packet from the VPI/VCI even before examining the header of that packet. The reason is that when the connection is established, we can figure out which IP address/prefix the foreign ATM endpoint is bound to. The foreign ATM endpoint is bound to an IP address if it is a host and is bound to an IP prefix if it is a router. Therefore, in an ATM firewall, filtering rules for different connections are different. Moreover, they are generated on-the-fly as follows. First, we can determine source and destination IP addresses/prefixes either from the higher layer IEs (information elements) contained in the authenticated signaling message or from the source and destination ATM addresses by querying a preconfigured ATM-to-IP mapping table. Then, from the security policy we determine the source and the destination TCP ports. Finally, these two pieces of information are glued to generate a set of filtering rules. For example, if we know that the source is a router for subnet 164.107.*.*, and destination is a router for subnet 192.41.245.*, and the destination subnet only allows HTTP (port 80) and TELNET (23) from the source subnet. Then the TCP/IP filtering rules are:

```
Src IP       Dest IP         Src Port   Dest Port    Action
164.107.*.*  192.41.245.*    ANY        80           Allow
164.107.*.*  192.41.245.*    ANY        23           Allow
(All packets that do not match either of these rules are blocked.)
```

Such automatic configuration of TCP/IP filtering rules is vital in future ATM environments where a large number of connections will be established within a short period of time. It has three major advantages:

- It is effective against IP spoofing. Since only those IP addresses/prefixes that might be associated with the source ATM endpoint will be used, packets that do not match these IP addresses/prefixes will be dropped.
- The network manager need not worry about setting up rules for each SVC when it is established. They just need to configure TCP/IP rules and cryptographic information for source endpoints.
- The number of TCP/IP rules that needs to be checked in each connection is much less than needed in traditional firewalls because the possible source and destination IP addresses/prefixes of packets in this connection are limited to a small set in an ATM connection.

Similarly, if a connection is of Class B, the call screening service generates the traffic monitoring rules, which has the same format as packet filtering rules, to be used by traffic monitoring service. If a connection is of Class D, the sixth field in the signaling message specifies the option for the protocol proxyed on that connection. For example, if the protocol to be proxyed on that connection is FTP, we can set the option to grant the source the privileges "put" and "mput", but deny "get" and "mget".

2.2.3 Packet Filtering Service

Packet filtering service inspects the header of IP packets to block "unsafe" packets while allowing the "safe" packets to pass. Two performance boosting schemes invented by ATLAS, namely, filtering only on the first cell and hardware caching of the security policy (see Section 1.2), will be used in our ATM firewall. In addition, we introduce a new scheme, called "Last Cell Hostage" (LCH), into our ATM firewall to further reduce the latency incurred by packet filtering, which works as follows.

In ATLAS, if the first cell misses the policy cache, the whole packet has to be blocked to wait for a slow software inspection process to finish. In contrast, with our LCH scheme, all cells of a packet except the last one is allowed to pass even if a cache miss occurs. Only the last cell is kept as "hostage" before this process is finished. The last cell is passed (dropped) if the inspection finds out that the packet is safe (unsafe). When the last cell is dropped, it will be substituted with a pseudo last cell whose payload is generated randomly so that CRC failure of the whole packet at the receiver is guaranteed. This prevents the drop of one packet from affecting the following packet, which will otherwise be mixed with the previous packet and thus corrupted. The introduction of LCH brings us two advantages:

- Even when a packet header misses the policy cache, if the packet is reasonably long, the software-based filtering process can be finished when the last cell arrives. This means that no delay is incurred even when a cache miss occurs. A recent survey on the packet size in a WAN shows that the average packet size is around 257 [12], which will occupy 6 cells if AAL5 is used [1]. If we assume that cells of different packets interleave with each other, on an OC-3c line, a connection of 25 Mbps, the average lapse between arrival times of the first cell and the last cell of a packet will be 36 cell time, enough for the inspection to finish. The size of a packet is going to increase due to increased use of VPN (Virtual Private Network) and improvement of WAN technology to accommodate large-size packet without fragmentation, which makes this scheme more attractive.
- Naturally, LCH allows us to process IP packets with option fields efficiently. The technical challenge in filtering IP packets with option fields is that the decision can not be made until the second cell arrives. ATLAS does not process such packets because otherwise it would have to consider each such packet as missing the policy cache. When the percentage of such packets is high, the processing overhead would be unbearable in ATLAS. However, with LCH, we just keep the first cell as the state information, and postpone the software-based filtering process until the second cell arrives. Since all cells but the last of a packet can stilled be passed without delay, in this way, we can process the IP packets with option field as fast as those without.

Even though the LCH scheme only blocks the last cell of a packet while unconditionally allowing the other cells pass, it is still a safe approach. If an attacker sends a "bad" packet to an internal host, since its last cell will be "corrupted", the packet will be discarded by the receiver.

2.2.4 Traffic Monitoring Service

Traffic monitoring service monitors the headers of IP packets contained in Class B connections. When a packet in a Class B connection arrives from the untrusted WAN, traffic monitoring service will first check whether the sender spoofs an internal address by comparing the source IP with a list of internal IP addresses. If a match occurs, which indicates an IP spoofing attack, the packet is dropped. Using Content Addressable Memory (CAM) that allows simultaneous look up, this operation can be performed very fast and will not incur any latency. If the sender does not claim to be an internal host, the packet will be allowed to pass but the first two cells that contain the TCP/IP header are duplicated and forwarded to the traffic monitoring service. The traffic monitoring service checks the packet headers against the traffic monitoring rules (generally the same as the packet filtering rules). When any attack attempt is detected, traffic monitoring service will immediately react to the attack so that the "reply" from the receiver will be blocked if there is any. In addition, actions are taken to prevent the same source from initiating further attacks (e.g., terminate the connection and blacklist the source).

Traffic Monitoring Service achieves the same level of security as Packet Filtering Service for both TCP and UDP packets:

- For TCP, this scheme effectively blocks the 3-way handshake for establishing unsafe sessions. Suppose a foreign host sent a "dangerous" SYN packet to an internal host, it will be passed because of the "after-the-fact" nature of the traffic monitoring service. However, this service is guaranteed to block the ACK+SYN packet from the internal host, and thus, the connection will not be established. Besides, the service will tear down the connection to prevent the foreign host from making further attacks. However, there is still one loophole to be addressed, which is the sequence number prediction attack [18]. By predicting the sequence number of the SYN+ACK packet sent from the internal host, the foreign host can forge final ACK packet along with the first SYN packet without receiving the SYN+ACK packet from the internal host. If the prediction turns out to be correct, the receiver will consider the connection be established. This ACK packet may contain a BSD "r" command that requests for the deletion of some files. Since such commands will only be honored if it is sent from an internal host, the external host has to spoof the address of the internal host. However, as we have explained, traffic monitoring service effectively detect and block such type of attack.
- NFS, SNMP, and DNS are three UDP-based protocols that have security holes [16]. With DNS, since we only have one DNS server in the internal network, we can always make it secure by only exchange information with authenticated DNS servers outside. With SNMP and NFS, we only run it inside a private network (with VPN, we can use encrypting gateway). So a SNMP and NFS packet can only be honored if the sender is an internal address. As we have explained above, external hosts are not able to spoof an internal address. Therefore, no SNMP and NFS attack can be launched from outside the firewall.

ICMP is shown to be pretty safe [15] though exploitation of its security weakness can lead to the denial of service. However, denial of service attack of this type is not effective in ATM networks because our ATM firewall can immediately trace down the attack and block the connection and/or blacklist the source ATM address to guard against further attacks.

In conclusion, traffic monitoring is as safe as packet filtering. Due to its after-the-fact nature, it does not incur any latency in any circumstances while packet does when the packet size is small and the header misses the policy cache. Therefore, its throughput is better than packet filtering service. Also, due to its "after-the-fact" nature, there is no technical difficulty in employing many such servers to perform traffic monitoring service in parallel.

2.2.5 Proxy Service

Proxy service serves as the proxy server for a number of Interent protocols as well as IP security protocols such as [11, 14, 9, 10]. A proxy server for a protocol acts as the middleman between the client processes and the server processes of the protocol [16, 17]. It interprets every step of the client-server interaction and block any unsafe steps. For example, an proxy server for FTP protocol can be configured to allow "get" and "mget", while disallow "put" and "mput" from inside to outside (lets file in but not out). Unlike packet filtering service, which only looks at the header of the packet, proxy service monitors the execution of the protocol and filters at the application level [16, 17]. However, since proxy service may need to look into the content of packets to understand the protocol; reassembling and disassembling of packets are required for this service. Therefore it can only be performed at the rate of a traditional firewall. Fortunately, most protocols (including FTP) can be secured by packet-level filtering, protocol proxying can thus be used sparingly.

A major use of proxy service is to proxy ISAKMP (Internet Security Association Key Management Protocol) [10]. ISAKMP is used to exchange necessary security association information between two communication parties like encryption algorithms, keys, and initialization vectors [11]. The security association will be used by ESP (IP Encapsulating Security Payload) [9] to provide confidentiality and/or used by AH (Authentication Header) [14] to provide authentication. If two parties intend to communicate with each other using IP security protocols, they will indicate it in the signaling message. We can arrange that the first several packets used to exchange security association information go through the proxy service. Once the proxy service finds out that the security association is successfully established, it can instruct the ATM firewall to allow packets in that connection pass without check.

Proxy service gets proxy option from the call screening service when a new Class D connection is established. Such option is protocol specific, which, for instance, can specify whether "get" and "mget" are allowed in a FTP proxy.

2.2.6 Firewall Management

Firewall management service controls and manages other security services in the ATM firewall and provides user-friendly administration tools to network managers. These tools allow network managers to perform operations like specifying or updating call screening rules, monitoring abnormal user activities, and disable connections that violate the security policy.

The firewall management system will log two types of events forwarded from other security services. The first type of information is the violation of security policy. This includes unsafe signaling messages detected by call screening service and unsafe packets detected by packet filtering service, traffic monitoring service and protocol proxying service. Such information will be used to analyze what type of intrusion is involved in a violation instance. Security policy may need to be updated in response to the intrusions. The other type of events is the profile information on each connection such as the identities of the communicating parties, start and end time of the connection, and number of cells transfered during its life cycle. With such information, we can have an expectation of access pattern from each source. This expectation allow us to identify abnormal activities from a certain source such as unusually large number of connection requests within a certain time window, and unusually large amount of data transfered within a certain time interval. As such abnormal activities may indicate intrusion attempts, the network manager should be brought to attention when they are detected.

3 Physical Components of Our ATM Firewall

In the last section, we discussed the logical design of the firewall, which consists of five security services. In this section, we present the physical components of the firewall and show how logical functions (the security services) are mapped to physical components.

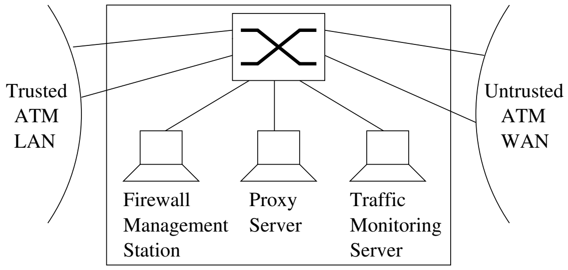


Figure 2. Physical Components of ATM Firewall

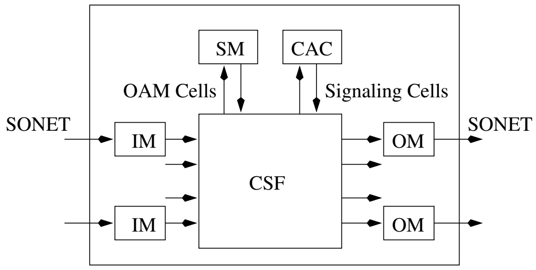


Figure 3. Internal Structure of an ATM Switch [4]

Fig. 2 depicts the physical components of our ATM Firewall. It consists of an *ATM firewall switch*, a *proxy server*, a *traffic snooping server*, and a *firewall management station*. The servers are attached to ATM firewall switch through high speed links. ATM firewall switch implements the call screening service and packet filtering service, which are discussed in the next section. The proxy server implements the proxy service and the traffic monitoring server implements the traffic monitoring service. They will be discussed in Section 5.1 and 5.2 respectively. Firewall management station implements the firewall management service. It is a general-purpose workstation equipped with firewall management software, which implements the functions described in the last section. Its design is out of the scope of this paper.

4 ATM Firewall Switch – The Heart of ATM Firewall Architecture

The firewall switch is the most complicated and important component in our ATM firewall. As the design of a firewall switch is based on a standard switch, in the following, we first briefly present the logical components inside a standard ATM switch. Then we show how they are modified in an ATM firewall switch to perform the packet filtering function, and provide support to traffic monitoring server and proxy server.

4.1 The Structure of a Standard ATM Switch

Due the complex nature of the ATM switch design and limitation of space, only those features that are relevant to our ATM firewall will be discussed. For a detailed description of the design of an ATM switch, [4] can be consulted.

A conceptual structure of a typical ATM switch is depicted in Fig. 3. An ATM switch consists of five functional modules, namely, input module (IM), output module (OM), cell switching fabric (CSF), call admission control (CAC), and system management (SM). Note that these are logical modules, and actual partitioning of functions varies from implementation to implementation. For ease of discussion, we assume that SONET/SDH is used as the physical layer.

IM extracts cells from the SONET payload envelope, determines the port or module each cell is destined for, and appends an internal tag to each cell. Three type of cells are identified by IM, namely, user cells that carry user traffic, signaling cells that carry signaling messages, and OAM cells that carry management information. For each user cell, IM looks up its destined output port or ports (for multicast traffic) from the routing table and writes this information into the internal tags. For signaling cells and OAM cells, the internal tags appended by IM indicate that they should be forwarded to CAC and SM, respectively. Cell switching fabric (CSF) routes the cells coming from IMs to their destinations designated in the internal tag. CAC assembles the signaling cells into the signaling message, processes the signaling message, decides whether or not the call should be accepted based on the availability of resources or other considerations, and generate new signaling cells. SM processes the OAM cells, performs switch management functions, and generate new OAM cells if necessary. User cells, new signaling cells generated by CAC, and new OAM cells generated by SM are forwarded to OM through CSF. OM processes and removes the internal tags appended to the cells, and puts the cells into the SONET payload envelopes for transmission.

Compared to the organization of a standard switch as shown in Fig. 3, a firewall switch makes modifications to all five functional modules while keeping its overall structure intact. In the following we explain the internal structure of each module and show the modifications a firewall switch made to it.

4.2 Input Module (IM)

The most important function that IM performs is to translate the VPI/VCI of each incoming cell to determine its output VPI/VCI/port. This is performed by a functional block called *cell processing*. As this block will be modified in a firewall switch to add security functions, we present its internal structure next. IM also performs other functions such as cell delineation, SONET functions, and UPC/NPC; however, they are not relevant to our firewall design and hence are outside the scope of this paper.

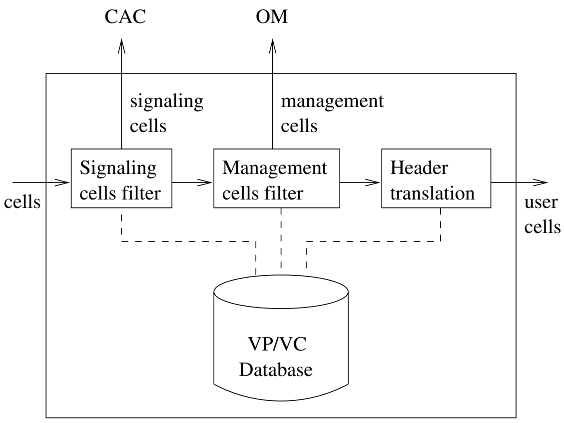


Figure 4. Cell Processing in a Standard Switch [4]

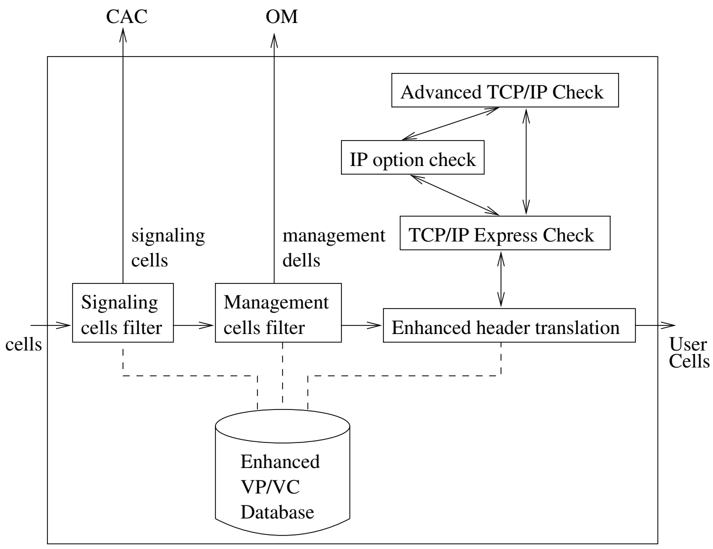


Figure 5. Cell Processing in Our ATM Firewall Switch

Fig. 4 shows the internal structure of cell processing block inside a standard switch. It consists of a *signaling cells filter* block, a *management cells filter* block, a *header translation* block, and a *VP/VC database*. Signaling cells filter block and management cells filter remove signaling cells and OAM cells from the cell stream and forward them to CAC and SM, respectively. The rest two blocks handle the processing of user cells. Each entry in VP/VC database contains at least following five fields (input VPI, input VCI, output port, output VPI, output VCI). When a user cell arrives, following three steps are performed by header translation block:

1. It looks up output VPI/VCI/port from the VP/VC database using the input VPI/VCI as the key.

2. It updates the VPI/VCI in the cell with the output VPI/VCI found in the VP/VC database.

3. It appends an internal tag to the cell indicating which output port it should be routed to, whether it is a multicast cell, and other fields used for internal housekeeping.

In a firewall switch, the cell processing block is modified to perform firewall functions, namely, distribution of traffic in different connections to different ATM firewall components based on their FQoS values, and filtering of the traffic inside Class C connections. Fig. 5 presents the modified cell processing block. Compared to its counterpart in a standard switch, it has enhanced the VP/VC table and the header translation block, and consists of three new functional blocks, namely, a *TCP/IP Express Check* (TEC) block, an *IP Option Check* (IOC) block, and a *TCP/IP Software Check* block.

The enhanced VP/VC table in a firewall switch will contain a number of firewall-related fields such as FQoS of a connection, packet-level state information such as the decision on the current packet (pass, drop, or LCH), and cell-level state information such as whether the first or second cell of a packet is expected. Enhanced Header Translation (EHT) block in a firewall switch will use the VPI/VCI of the incoming cell to look up the value of such fields, and perform firewall functions on Class B and C connections. With a Class B connection, EHT will instruct the Cell Switching Fabric (CSF) to duplicate the first two cells of each packet inside the connection to traffic monitoring server. The internal tag will be extended to contain several firewall-related fields (see Appendix A), one of which is devoted for this purpose.

The major firewall function that EHT performs is to filter the traffic inside Class C connections. In a Class C connection, when the first cell of a packet arrives at EHT, it is forwarded TCP/IP Express Check block, which decides whether the packet should be forwarded, dropped, or put into LCH. In any circumstances, the decision is sent back to EHT immediately so that the cell will not get delayed. If the decision is to pass (drop) the packet, all cells of the packet is passed (dropped). If the decision is to put the packet into LCH (due to policy cache miss or the existence of IP option field in the packet header), EHT will instruct OM to keep the last cell of the packet as hostage (indicate that in the internal tag). All these operations are implemented using simple combinatorial logic (see Appendix A) and can be performed within less than a cell time.

TCP/IP Express Check (TEC) block employs StorageTek's policy cache scheme [8]. When the first cell of a packet arrives at the TEC, it is checked against the policy cache for a match. If a match is found, the decision (forward or drop) is forwarded to EHT. However, if the packet misses the policy cache or if the packet contains IP option, TEC will instruct the packet to put the packet into LCH. In the meantime, the cell(s) is forwarded to IP option block if it contains IP option or to software check block otherwise. Once the decision whether the packet should be passed or dropped is available from either block, TEC will forward the decision to OM through CSF. The detailed process is explained in Appendix A

IP Option Check block processes packets with IP option fields. It also employs policy cache scheme to speed up the filtering process. The detailed process is explained in Appendix A. TCP/IP Software Check block checks (in software) the header of a packet to see whether the packet is safe. This process is identical to what happens in a traditional firewall.

4.3 Output Module (OM)

In a standard switch, OM processes the internal tag of each cell for housekeeping purposes and updates output VPI/VCI for multicast cells. OM has a VP/VC table, which contains fields for housekeeping information such as number of cells transported in each connection, and a field that stores output VPI/VCI for multicast connections.

In a firewall switch, OM is involved in implementing the LCH scheme. When the LCH scheme needs to be applied to a packet, IM will indicate that in the internal tag and forward the packet to OM. Decision on that packet will also be forwarded to OM from TCP/IP Express Check block as soon as it is available. The responsibility of OM is to keep the last cell of the packet hostage until the decision on the packet arrives. The detailed process is explained in Appendix B.

When the last cell of a packet is put into LCH and buffered at OM, OM needs to make sure that the next packet in the same connection should also be buffered at OM in order to preserve the cell order in an ATM connection. This is accomplished through synchronization between IM and OM using a field inside the internal tag, which is explained in detail in Appendix B. With such cells, OM needs to perform a look up in the VP/VC table using its input VPI/VCI as the key, the cost of which is identical to that of processing a multicast cell. So we would like to minimize the number of such cells. This is accomplished by "OMCHK clearance protocol" as explained in Appendix B.

4.4 Call Admission Control (CAC)

CAC in a normal switch decides whether an incoming signaling request can be accommodated based on whether the requested QoS can be satisfied without compromising the QoS guarantees given to existing connections.

In a firewall switch, CAC implements the call screening service as discussed in Section 2.2.2. It compares the identities of the communicating parties contained in the signaling message with call screening rules to decide whether the connection is allowed to be established. If the connection is allowed to be established, CAC will set up an entry in VP/VC table for the new connection and initiate the firewall-related fields such as FQoS. If FQoS of the new connection is C, CAC will generate the packet filtering rules and send them to IM. Similar actions are taken if the FQoS is B or D. CAC is also equipped with fast cryptographic hardware to quickly authenticate the digital signature contained in the signaling message.

Call screening process can be done in parallel with normal call admission control process and will not introduce any extra delay.

4.5 System Management (SM)

SM handles all the management plane functions in a switch including fault management, performance management, configuration management, accounting management, security management, and traffic management [4]. In a firewall switch, we need to add a management function, called firewall management, to manage the security functions that need to be performed by the switch. It performs following functions:

- Maintenance of security-related managed objects such as call screening rules.
- Execute the commands sent from firewall management station such as updating call screening rules.
- Monitor the performance of call screening service and packet filtering service, such as the throughput of filtering operation at each port.

4.6 CSF

CSF in a firewall switch needs to examine the T-MONITOR bit (see Appendix A) in the internal tag and duplicate a copy of cells with this bit turned on to the traffic monitoring server. This can be achieved with little cost no matter what architecture (e.g., shared memory) CSF is built upon.

5 Other components of Our ATM firewall

5.1 Traffic Monitoring Server

Traffic monitoring server implements the traffic monitoring service as explained in Section 2.2.4. It is a ATM-attached workstation which is equipped with policy cache hardware to perform header checking at high speed. Its implementation is almost identical to the implementation of packet filtering service. Many servers can run in parallel (each attached to the ATM firewall switch through a separate port) to increase the monitoring throughput as explained in Section 2.2.4.

5.2 Proxy Server

Proxy server implements the proxy service as explained in Section 2.2.5. It is a traditional proxy firewall equipped with ATM interface(s). When the ATM firewall switch decides that the FQoS of a new connection is D, it will set up two connections, one between the source and proxy server, and the other between proxy server and the destination. Two connection are concatenated so that each packet receives from one connection will be proxyed by the proxy server and forwarded to the other if it is safe (blocked otherwise).

6 Summary

A router-based packet-filtering firewall is an effective way of protecting an enterprise network from unauthorized access. However, it will not work efficiently in an ATM network because it requires the termination of end-to-end ATM connections at a packet-filtering router, which incurs huge overhead of reassembling and dissembling packets. This paper offers a viable solution for high-speed packet-level filtering in ATM networks. We design an ATM firewall which is novel in concept, high in performance, easy to manage, and less restrictive in packet formats (allow IP option fields). Our ATM firewall implements our novel concept of FQoS, which can potentially increase the amount of traffic that an ATM firewall can secure per unit time dramatically. We also introduced a "last cell hostage" scheme to minimize the delay incurred when the cell misses the policy cache and to enable the ATM firewall process packets with IP option fields. We presented a hardware design of the ATM firewall to show the feasibility of these schemes.

Appendix A. Detailed Design of Input Module (IM) in a Firewall Switch

The enhanced VP/VC table in a firewall switch will contain following firewalling-related fields:

1. FQoS: This field indicates the Filtering Quality of Service (FQoS) of the connection.
2. SEL_NO: Each packet in a Class C connection is assigned a serial number, which is used by the OM in the last cell hostage scheme to match the filtering decision with the packets arriving at OM.
3. EXP-1st: This bit indicates whether the first cell of a packet is expected in the connection.
4. EXP-2nd: This bit indicates whether the second cell of a packet is expected in the connection.
5. DROP: This bit indicates whether all cells of the current packet should be dropped.
6. PASS: This bit indicates whether all cells of the current packet should be passed.
7. LCH: This bit indicates whether the last cell hostage scheme (LCH) will be applied to the current packet.
8. OMCHK (Output Module Check): When this bit is on, OM should check each cell in this connection to see whether there is a queue for this connection at OM.
9. SEL_NO_LAST_LCH: This field indicates the serial number of the last packet that has been put into LCH.

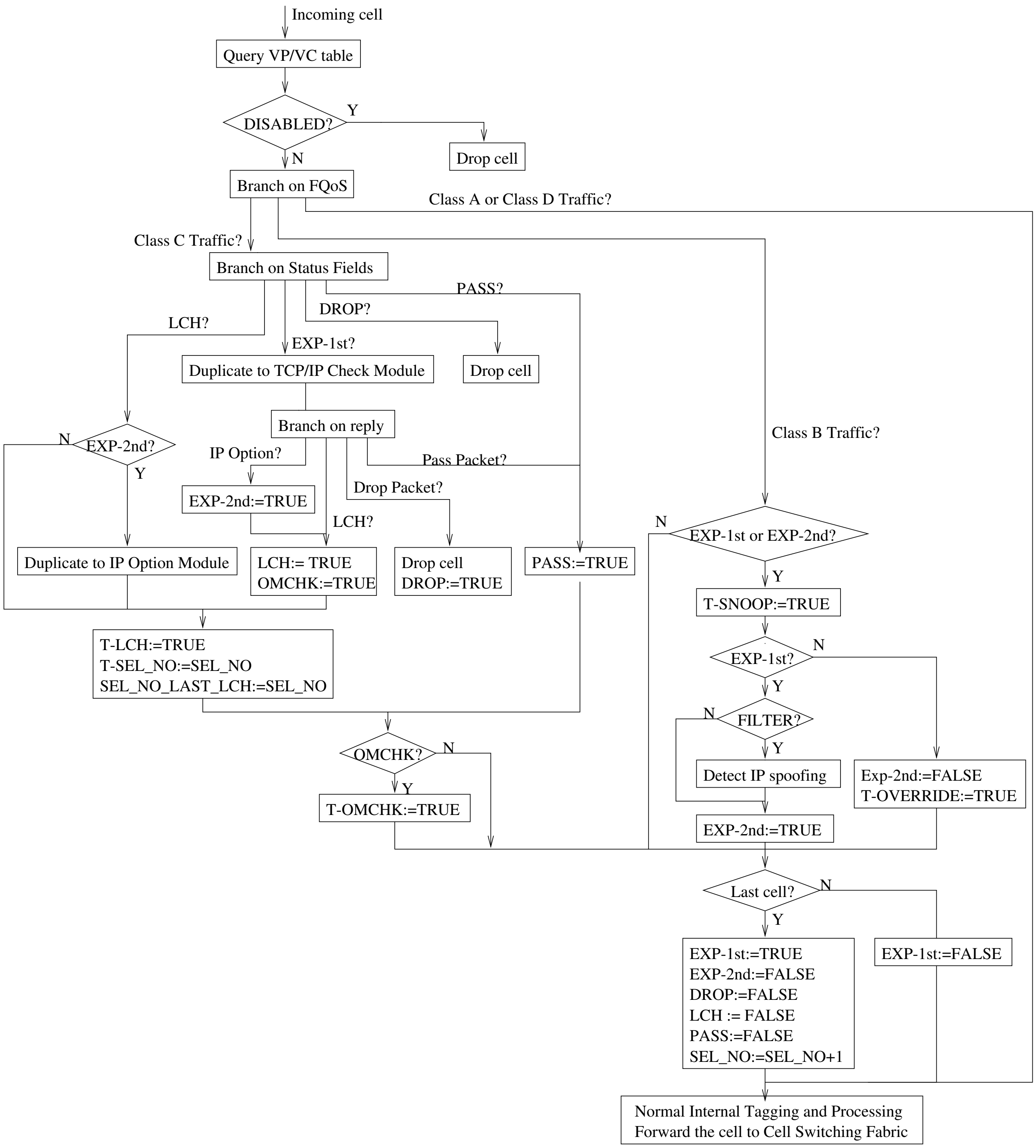
In a firewall switch, following firewalling-related fields will be added to the internal tag:

1. T-MONITOR: This bit denotes whether the cell is among the first two cells of a packet in a Class B connection. CSF will duplicate a copy of such cells to the traffic monitoring server if this bit is turned on.
2. T-SEL_NO: The semantics and value of this field is the same as the SEL_NO field in the VP/VC table. It is used by OM to match packets (actually the last cell) with the decisions on them in LCH scheme.
3. T-LCH: This bit denotes whether the last cell of the packet needs to be kept hostage at OM.
4. T-OMCHK: The semantics and value of this field is the same as the OMCHK field in the VP/VC table.

The flow diagram of the header translation module is shown in Fig. 6. When a cell arrives at this module, its VPI/VCI is used as the key to query the VP/VC table for the values of output VPI/VCI/port and firewall-related fields listed above. Different actions are taken depending on the FQoS of the connection ("Branch on FQoS").

If the cell belongs to a Class A or D connection, header processing is performed in the same way as in a standard switch ("Normal Internal Tagging and Processing"). After that, the cell is forwarded to the CSF.

If the cell belongs to a Class B connection ("Class B Traffic?"), the first two cells of the packet that the cell belongs to should be duplicated to traffic monitoring server. The first and second cells are identified through the Exp-1st and Exp-2nd bits in the VP/VC table respectively. When the last cell of a packet is encountered ("Last cell?") , which is identified through the PTI field [1] in the cell header, Exp-1st is turned on ("Exp-1st:=TRUE"). When the first cell of a packet is encountered, Exp-2nd is turned on. ("Exp-2nd:=TRUE"). In the first two cells of a packet, the T-MONITOR bits are turned on. CSF will duplicate the cells with T-MONITOR bit on to the traffic monitoring server.



If the cell belongs to a Class C connection, which needs to be filtered by the firewall switch, one of the four different actions will be taken depending on the value of four fields ("LCH?", "Exp-1st?", "DROP?", and "PASS?") in the VP/VC table ("Branch on Status Fields"). If the cell is the first cell of a packet ("Exp-1st?"), it will be forwarded to the TCP/IP Express Check module for the decision on the packet. Four different actions are taken depending on ("Branch on reply") four possible decisions ("IP Option?", "LCH?", "Drop Packet?", and "Pass Packet?"). If the decision is to drop the packet ("Drop Packet?"), the DROP bit in the VP/VC table is turned on (DROP:=TRUE) to indicate that all cells that belong to this packet should be dropped. If the decision is to pass the cell ("Pass packet?"), the PASS bit in VP/VC table is turned on (PASS:=TRUE) to indicate that all cells in this packet should be passed. If the decision is that the last cell hostage scheme should be used, the LCH bit in the VP/VC table and the T-LCH bit in the internal tag of the cell are set, and the cell is forwarded to OM through CSF. If the reply indicates that the packet contains IP option ("IP Option?"), then the second cell is needed for the decision to be made. In this case, the Exp-2nd bit in VP/VC table is turned ("Exp-2nd:=TRUE") to indicate that the second cell is expected for inspection. Since the last cell hostage scheme will also be used with this case, the LCH bit in VP/VC table should also be set. Whenever the LCH bit is set, the OMCHK bit in the VP/VC table should also be set and SEL_NO_LAST_LCH should to set to the serial number of the current packet. We will explain the use and rationale of them in Appendix B.

On the other hand, if the incoming cell is a non-first cell of a packet, the action that needs to be taken on the cell depends on the decision the firewall switch has made on first cell of the packet. As explained above, there are three possibilities, to drop (if DROP is TRUE), to pass (if PASS is TRUE), or to put it "Last Cell Hostage" (if LCH is TRUE). The first two cases are straightforward. In the case of LCH, the T-LCH bit in the internal tag of the cell will be set and the cell is passed to OM. If the Exp-2nd is TRUE, which means the incoming cell is the second cell of a packet and is needed by the IP option block, a copy of the cell is forwarded to TCP/IP Check block.

In Class B, C, and D connections, when the last cell of a packet arrives, the Drop, Pass, LCH bits should be turned off and the Exp-1st bit should be turned on, and the SEL_NO should be increased by one.

The flow diagram of the TCP/IP Check block is depicted in Fig. 7. When a cell is passed from the header processing module to the TCP/IP check module, it must be either the first cell or the second cell of a packet. If it is the second cell ("Second cell?"), it must be expected by the IP option module (explained immediately below) and is forwarded to it accordingly. Otherwise, (the cell is the first cell of a packet) TCP/IP Express Check block checks whether the packet header contains IP option field ("Contain IP option?"). If the answer is yes, TCP/IP check will advise the Enhanced Cell Processing block to put the packet into LCH and indicate to it that IP option block is expecting the second cell of the packet. In the meantime, the cell will be forwarded to the IP Option Check module and is buffered there. Otherwise (the cell does not contain an IP option field), TCP/IP check block will try to match the packet header with the cached TCP/IP filtering rules. If there is a hit, the decision, which is either DROP or PASS, is given back to the VPI/VCI instantly. Otherwise (if a cache miss happens), TCP/IP check block will advise the Enhanced Cell Processing to put the packet into LCH. In the meantime, the cell is forwarded to the Software Check block for further inspection.

When the IP Option Check block or Software Check block has finished inspecting the packet header , it will forward the decision to the TCP/IP Express Check block, which will in turn be forwarded to Enhanced Cell Processing block. Enhanced Cell Processing block will further forward the deicison to OM to either drop (if the decision is to drop) or forward (if the decision is to PASS) the cell that has been put LCH.

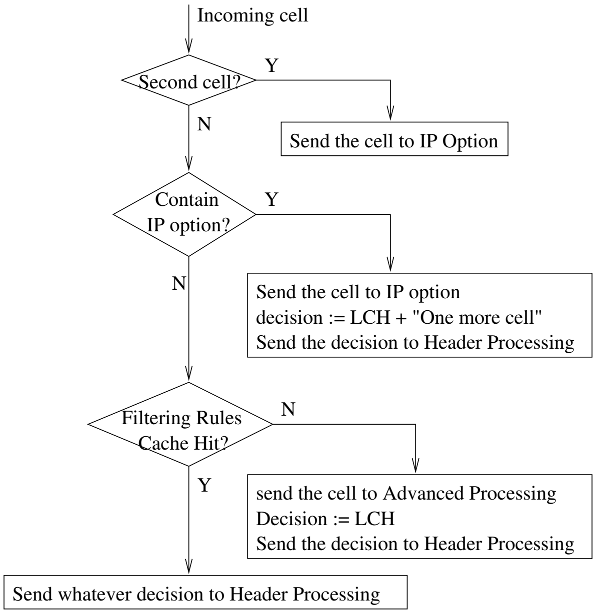


Figure 7. Flow Diagram of TCP/IP Express Check block

IP Option Check works almost in the same way as the TCP/IP block except that it buffers the first cell of a packet that contains IP option as the state information. When the first cell arrives, the offset of the TCP header can be precalculated from the IP option size field. When the second cell of a packet arrives, all the necessary TCP/IP fields will be retrieved from these two cells to match with the TCP/IP filtering rules in the policy cache. If there is a hit, the decision is given back to TCP/IP check module immediately. Otherwise, the two cells will be handed over to Software Check block. When the decision comes back from Software Check block, it is in turn forwarded to the TCP/IP Express Check block. In addition, the decision will be cached into the policy cache so that later packets with similar header information may hit the cache.

Appendix B. Detailed Design of Output Module (OM) in a Firewall Switch

In a firewall switch, OM is involved in implementing the LCH scheme in the packet filtering service. When the LCH scheme needs to be applied to a packet, IM will indicate it in the T-LCH field of the internal tag and forward the packet to OM. Decision on that packet will also be forwarded to OM as soon as it is available. The responsibility of OM is to hold the last cell of such a packet until the decision on the packet arrives. To accomplish this function, VP/VC table will be enhanced to include two new fields. One is LCH_Cell, which buffers the last cell of a LCH packet. The other is SEL_NO, which stores the serial number of the last decision received (PASS or DROP).

Following processes are involved in implementing the LCH scheme. The first process is invoked when the last cell of a LCH packet arrives. The second process is invoked when packets that follow the LCH packet in the same connection arrive. The third process is invoked when the decision on LCH packets arrives.

1. OM identifies the last cell of a LCH packet from the LCH bit in the internal tag and the PTI field in the cell header. When such a cell is identified, OM will check whether the decision on the packet has arrived (SEL_NO = T-SEL_NO?). If the decision is available, the decision is applied to the cell.

2. While the decision on the LCH packet is pending, packets that follow the LCH packet in the same connection should also be buffered in order to preserve the cell order. This is taken care of by the T-OMCHK bit in the internal tag of a cell. IM will turn T-OMCHK bit on after a LCH packets occur in the connection. When OM detects such a cell, OM will check whether there is a cell buffered in LCH_Cell. If there is not one, the cells should be allowed to pass. Otherwise the cells need to be queued up.

3. When the decision on a packet arrives, OM needs to update the SEL_NO field to the serial number contained in the decision message. Also, cells contained in the queue for that connection will now be processed by OM.

The overhead of the first process can be absorbed into standard housekeeping process of OM. The overhead incurred by the second process is equivalent to the processing multicast cells because both need to look up the VPI/VCI table for more information. The third process can be accomplished efficiently by ASIC (Application Specific Integrated Circuit) and will not incur any delay.

As each OMCHK cell results in as much overhead as a multicast cell, it is desirable to make the number of such cells as small as possible. The ideal situation is that once a queue for OMCHK cells in a VP/VC gets empty, the OMCHK bit is cleared in the corresponding VP/VC entry in IM. This is accomplished by OMCHK Clearance Protocol, which works as follows.

Once a queue gets empty, an internal cell is sent back to IM, which contains the VPI/VCI of the connection and the serial number of the last packet that is applied LCH. When IM receives the internal cell, it checks whether the last LCH matches the serial number of the last LCH packet kept. If it matches, which means that no new LCH packet has been sent to OM after that one, IM can safely take OMCHK bit off. Otherwise, this message is ignored.

References

[1] D. Ginsburg, ATM: solutions for enterprise internetworking, Addison Wesley Longman, London, England, 1996.

[2] T. Smith, and J. Stidd, “Requirements and Methodology for Authenticated Signalling”, ATM Forum/94-1213.

[3] L. G. Pierson, and T. D. Tarman, “Requirements for Security Signalling”, ATM Forum/95-0137.

[4] T. M. Chen, and S. S. Liu, ATM Switching Systems, Artech House Inc., Boston, MA, 1995.

[5] J. Hughes, “A High Speed Firewall Architecture for ATM/OC-3c”, http://www.network.com/, 1996.

[6] J. Hughes, and A. Guha, “Requirements for Secure Packet-Level Access over ATM”, ATM Forum/95-1126.

[7] T. D. Tarman, “Phase I ATM Security Specification”, ATM Forum BTD-SECURITY-01.13, July, 1997.

[8] B. J. Kowalski, “ATLAS Policy Cache Architecture”, http://www.network.com/, 1997.

[9] R. Atkinson, “IP Encapsulating Security Payload (ESP)”, RFC 1827, Aug. 1995.

[10] D. Maughan, et al., “Internet Security Association and Key Management Protocol (ISAKMP)”, IPSEC Working Group, Feb. 1997.

[11] R. Atkinson, “Security Association for the Internet Protocol”, RFC 1825, Aug. 1995.

[12] Internet posting, “WAN packet size distribution”, http://www.nlanr.net/NA/Learn/packetsizes.html.

[13] Secant Network Technologies, “Encrypting ATM Firewall”, http://128.109.127.7/articles/wink1/wink.html.

[14] R. Atkinson, “IP Authentication Header”, RFC1826, Aug. 1995.

[15] S. M. Bellovin, “Security Problems in the TCP/IP Protocol Suite”, *Communications of ACM*, Vol. 4, 1989.

[16] Cheswick, W. R., and S. M. Bellovin, Firewalls and Internet Security, Addison Wesley, Reading, MA, 1994

[17] Chapman, D. B., and E. D. Zwicky, Building Internet Firewalls, O'Reilly & Associates , Sebastopol, CA, 1995

[18] S. Bellovin, “Defending Against Sequence Number Attacks”, RFC 1948, May 1996.