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 SAR (Segmentation and Reassembly). Very few approaches to this problem have been proposed in the literature, and 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 trusted ATM LAN and untrusted ATM WAN. It is also easy to manage and flexible to use.

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 SAR (Segmentation and Reassembly), which incurs a huge overhead. Second, the filtering bandwidth of a traditional firewall is generally below 100 Mbps, which is 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 [16] proposes that access control decisions be made at connection establishment time based on the higher layer information (e.g., source and destination IP addresses, ports, etc.) that is contained in the ATM *signaling message* as *information elements*. 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 he wants to access at the connection establishment time, and there is no way to check the contents of a connection once the connection is established. Pierson [14] tries to fix this problem by proposing that "a new SVC (Switched Virtual Connection) 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 [7]. 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 [17]. A centralized encrypting gateway, called *encrypting ATM firewall* in the literature, is employed between a trusted ATM LAN and untrusted WAN to perform encryption operation on behalf of all protected hosts [15]. However, authentication and encryption do not automatically ensure proper access control. Even when a connection is authenticated, we may still want to look into its contents 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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
5th Conference on Computer & Communications Security
San Francisco CA USA
Copyright ACM 1998 1-58113-007-4/98/11...$5.00

product [6]. ATLAS is a line filter that scans an ATM physical link to perform the packet-level filtering at the rate of OC-3c (155 Mbps). StorageTek claimed that next-generation ATLAS is going to support the filtering rate of OC-12c (622 Mbps) in the near future. Two performance boosting strategies are used in ATLAS, which correspond to the two factors that render traditional firewalls unsuitable for an ATM environment. First, to avoid SAR, for each packet it only checks the first cell, which contains the IP header, protocol, TCP/UDP ports, and TCP flags (if applicable), to determine whether or not the packet is "safe". If the packet is considered safe, all following cells that belong to it are passed or otherwise dropped. Second, they use a *policy cache architecture* [11] to dramatically speed up the process of deciding whether or not a packet header is safe. The core unit of this architecture is a cache block called *policy cache*. Each entry of the policy cache is a combination of VPI/VCI, source and destination IP addresses, and source and destination TCP/UDP ports that is 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 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 CAM (Content Addressable Memory), which enables simultaneous searches of all memory locations to find a match with the pattern being 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 IP option fields because IP option can be as large as 40 bytes and may "push" the TCP header to the second cell. This may become a severe limitation in the future internetworking environments where certain IP options like AH (Authentication Header) [9] are used frequently. Second, it is not friendly to those who have to manage and administer it. Whenever a new PVC (Permanent Virtual Connection) 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 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 Our ATM Firewall

We designed a high-performance switch-based ATM firewall architecture. It implements our novel firewall design concept called *Firewalling Quality of Service* (FQoS), which employs security measures of different strength on traffic associated with different risk levels in order to achieve a nice tradeoff between performance and security. In terms of performance, it achieves a higher throughput and lower latency than ATLAS. In the next section, we present the design philosophy and logical structure of the proposed ATM firewall architecture. Section 3, 4, 5 present its physical design. Section 6 concludes the paper.

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 to connections which are more "dangerous", typically at the cost of a longer processing time per packet. Four FQoS classes are defined, namely, class A, B, C and D, which are 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 can be secured by packet-level filtering[1]. The risk level of 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 filtering; 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 filtering* and is almost as safe. Traffic inside class D connections is dangerous and will involve complex protocols that are 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 [2].

This classification of FQoS is not ad hoc; instead, it corresponds naturally to different types of data traffic in future networking environments. 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 such a 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 untrusted, e.g., an arbitrary Internet surfer.

The motivation to classify traffic into different FQoS classes is to optimize the effort to make connections secure. Safer connections are screened using faster but less aggressive checking methods while less safe connections are screened using more aggressive but more time-consuming checking methods. There are two challenges in implementing the concept of FQoS. The first is to design firewalling schemes to secure each FQoS class (except class A) efficiently while satisfying the security level required by that class. The other challenge is to design an architecture that integrates all these firewalling schemes. How we address both challenges in our ATM firewall is shown 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 an ATM *signaling message* requesting a new SVC arrives at the firewall, the *call screening service* decides whether it is safe

[1]This includes stateful filtering as used in FTP.

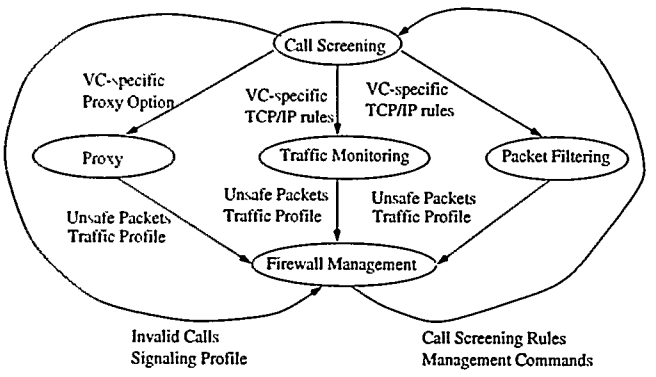

Figure 1: Logical Design of Our ATM Firewall

to establish the connection by checking the identity (e.g., ATM 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 service* assigns a FQoS to the connection 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 class is 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 provide the necessary information (e.g., packet filtering rules) to the security service which corresponds 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 interval) and packets that violate the security policy will be recorded and sent to *firewall management service*, which controls and coordinates other security services. The details of these security services are presented in the following sections.

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 five fields:

1. Source Identity
2. Destination identity
3. Authentication information (e.g., a digital signature)
4. FQoS of the new connection to be established
5. 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, the signaling message is blocked and access denied. The number of such rules may be large; however, the search can be performed fast using hashing techniques. The source ATM address of the Class A and B connections needs to be authenticated by call screening service using cryptographic measures as discussed in [17] as its spoofing may allow dangerous traffic flow through the firewall without checking. The third field contains information for authenticating the source end point such as a digital signature. The signaling message is blocked if the authentication fails. If the connection is allowed to be established, the fourth field denotes the FQoS of the new connection.

If the FQoS class of the connection is C, its contents are subject to packet-level filtering. In an ATM firewall, filtering rules for various connections are different and are generated on-the-fly as follows. First, the ATM firewall 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 address to IP address/prefix mapping table. Then, from the fifth field of the security rule the ATM firewall determines 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 the source is a router for subnet 164.107.*.*, and the 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       SP     DP  Action
164.107.*.*  192.41.245.*  >1023  80  Allow
164.107.*.*  192.41.245.*  >1023  23  Allow
(All other packets are blocked)
```

The filtering rules to secure the return traffic are generated in the same fashion. 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 the spoofing of source IP addresses. Since only those IP addresses/prefixes that might be associated with the source ATM endpoint will be honored, packets that do not match these IP addresses/prefixes will be dropped.
- Network managers 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 trusted outsiders.
- 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 the FQoS class of a connection is B, the call screening service generates the traffic monitoring rules, which have the same format as packet filtering rules, for traffic monitoring service. If a connection is of class D, the sixth field in the signaling message specifies the options for the protocol proxyed on that connection. For example, if the protocol to be proxyed on that connection is FTP, one such option is to grant the source the privileges of "put" and "mput", but not "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), are used in our ATM firewall. If a packet is fragmented, only the first packet is filtered[2]. 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.

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 cause corruption. The introduction of LCH has 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. Let us explain this by an example. A recent survey on the packet size in WAN shows that the average packet size is around 348 [13], which will occupy 8 cells if AAL5 is used [5]. If we assume that cells belonging to different packets interleave, in a T3 rate (45 Mbps) connection on a OC-3c line, the average lapse between arrival time of the first cell and the last cell of an average-sized packet will be 22 cell time. On the other hand, today's software-based firewall technology can perform a header checking with 6 cell times (50,000 packets/second). In this case, there is a 99.8% chance (assuming exponential distribution for inter-arrival time between any two consecutive cells) that the inspection finishes even before the last cell of the packet arrives. In addition, the size of a packet is going to increase due to improvement of WAN technology to accommodate large-size packet without fragmentation, which makes this scheme more attractive.

- LCH allows us to process IP packets with IP option fields efficiently. The technical challenge in filtering IP packets with IP option is that the decision may not be made until the second cell arrives. ATLAS does not process such packets because it would have to consider each of such a 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, only the first cell is kept as the state information, and the software-based filtering process is postponed until the second cell arrives. Since all of the cells except the last one can still be passed without delay, the IP packets with IP option can be processed as fast as those without.

[2] *Tiny fragment attack* can be prevented by checking packet length, and *overlapping fragment attack* can be prevented by checking fragment offset [19]. Both are trivially implementable in hardware.

Even though the LCH scheme only blocks the last cell of a packet while unconditionally allowing other cells to 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.

Currently, 16-byte-long IPv6 is not supported in ATLAS because the width of CAM is limited to 128 bits long due to technology and marketability concerns. Our ATM firewall can support IPv6 by hashing a 32-byte-long source and destination IPv6 address pair into a 8-byte-long hash value to be stored into the policy cache. When the first cell of an IPv6 packet arrives, the hash value of its IPv6 address pair will be calculated (in hardware) and be compared against the policy cache like IPv4 packets. This approach is justified by the fact that the number of IPv6 addresses used is much less than the number of possible hash values so that the probability for a hashing collision to be exploited by intruders is negligible. We are still investigating suitable hashing algorithms for this purpose.

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 first checks whether the sender spoofs an internal address by comparing the source IP with a list of internal IP addresses/prefixes. This operation can be performed fast using CAM. 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., terminating the connection and blacklisting the source).

Traffic monitoring service achieves almost the same level of security as packet filtering service for TCP traffic as it 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 sent from the internal host to prevent the TCP connection from being established. Additionally, the traffic monitoring service will tear down the connection to prevent the foreign host from making further attacks. However, there is still two loopholes to be addressed. The first loophole is the *sequence number prediction attack* [1]. By predicting the sequence number of the SYN+ACK packet sent from the internal host, the foreign host can forge the final ACK packet in the 3-way handshake without receiving the SYN+ACK packet from the internal host. The forged packet is sent along with the SYN packet. If the prediction turns out to be correct, the receiver will assume that the TCP connection is established. This ACK packet may contain a BSD "r" command that requests the deletion of some files. Since such commands will only be honored if they are sent from an internal host, the external host has to spoof the address of the internal host. As has been explained, traffic monitoring service effectively detect and block such IP spoofing. The second loophole is *SYN floods*, in which the

attack floods the victim host with SYN packets. However, the traffic monitoring service will track down such attacks and tear down such half-open TCP connections by sending the victim hosts RST packets.

Since UDP does not require the three-way handshake, the traffic monitoring service can not guarantee the security of UDP traffic inside Class B connections. Therefore, it is safe to block all UDP traffic inside Class B connections. This can be performed very fast by checking the protocol field of the first cell of a packet in hardware. In the same way, ICMP packets can also be kicked out of Class B connections.

In conclusion, traffic monitoring is almost as safe as packet filtering for TCP traffic. Due to its after-the-fact nature, it does not incur any latency while packet filtering service does when the packet size is small and its header misses the policy cache. Using the policy cache to boost the performance, the throughput of traffic monitoring service is no less than that of the packet filtering service. Therefore, traffic monitoring service at least doubles the throughput that is achievable by packet filtering alone. 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. (With packet filtering service, in order to preserve cell order in an ATM connection, parallel processing may incur significant delay due to synchronization.)

2.2.5 Proxy Service

Proxy service serves as the proxy server for a number of Internet protocols. A proxy server for a protocol acts as the middleman between the client processes and the server processes [2, 4]. It interprets every step of the client-server interaction and blocks any unsafe steps. For example, an proxy server for FTP protocol can be configured to allow "get" and "mget", but disallow "put" and "mput", from inside to outside (letting file in but not out). Unlike packet filtering service which looks only at the header of the packet, proxy service monitors the execution of the protocol and filters at the application level [2, 4]. However, since proxy service may need to look into the contents of a packet to understand the protocol and requires SAR, it can only be performed at the rate of a traditional firewall. Fortunately, most protocols (including FTP) can be secured by stateful packet-level filtering, protocol proxying can thus be used sparingly.

Another use of proxy service is to "oversee" ISAKMP (Internet Security Association Key Management Protocol) [12]. ISAKMP is used to exchange necessary *security association* information between two communication parties such as encryption algorithms, keys, and initialization vectors [10]. The security association will be used by ESP (IP Encapsulating Security Payload) [8] to provide confidentiality and/or by AH (Authentication Header) [9] to provide authentication. If two parties intend to communicate with each other using IP security protocols, they will indicate it in the signaling message. It can be arranged that the first several packets which are used to exchange security association information go through the proxy service. The proxy service will oversee the protocol steps to make sure that a valid security association is successfully established. Once the proxy service finds out that the security association is successfully established, it can instruct the ATM firewall to make a shortcut so that cells in that connection will be passed without inspection. Afterwards, the firewall keep monitoring (in hardware) the security association index field contained in the packet header. Once it indicates that the security association is terminated, the firewall will turn the connection back to the "proxying mode".

2.2.6 Firewall Management Service

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 such as specifying or updating call screening rules, monitoring abnormal user activities, and disabling connections that violate the security policy.

The firewall management system logs two types of events forwarded from other security services. The first type of event 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 proxy 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 event 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 transported during its life cycle. With such information, an ATM firewall can have an expectation of access pattern from each source. This expectation allows it 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 transported within a certain time interval. As such abnormal activities may indicate intrusion attempts, network managers should be brought to attention when they are detected.

3 Physical Components of Our ATM Firewall

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

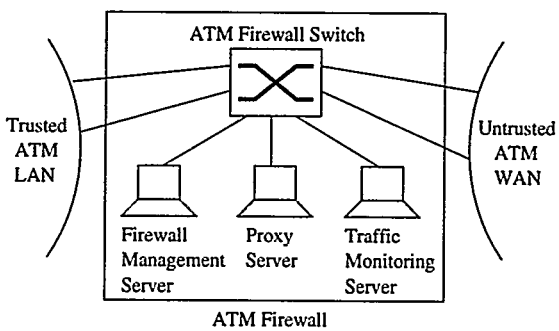

Figure 2: Physical Components of ATM Firewall

Fig. 2 depicts the physical components of our ATM Firewall. It consists of an *ATM firewall switch*, a *proxy server*, a *traffic monitoring server*, and a *firewall management server*. The proxy server, the traffic monitoring server and the firewall management server are attached to the ATM firewall switch through high-speed links. The 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. The firewall management server implements the firewall management service. It is a general-purpose workstation equipped with firewall management software, the design of which is out of the scope of this paper.

4 ATM Firewall Switch

The ATM 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

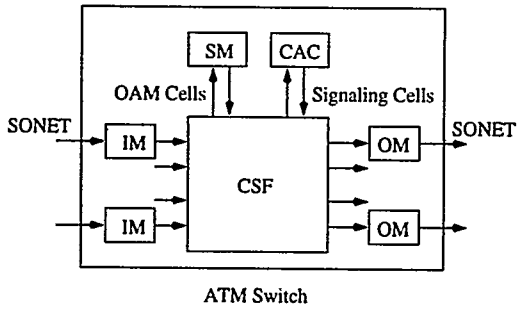

Figure 3: Internal Structure of an ATM Switch[3]

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 just logical modules, 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 types of cells are identified by IM. They are 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 tags. CAC assembles the signaling cells into a 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 generates new signaling cells. SM processes the OAM cells, performs switch management functions, and generates 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 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 structures of these modules and show the modifications a firewall switch makes to each of them.

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 firewall functions, its internal structure is explained in detail. 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.

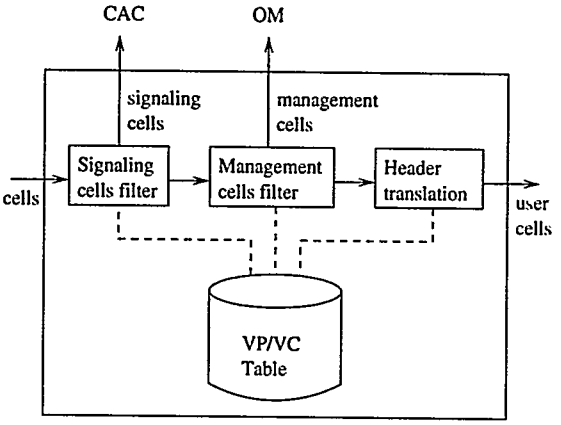

Figure 4: Cell Processing in a Standard Switch[3]

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 table*. *Signaling cells filter* 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 table 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 table 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 table.
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.

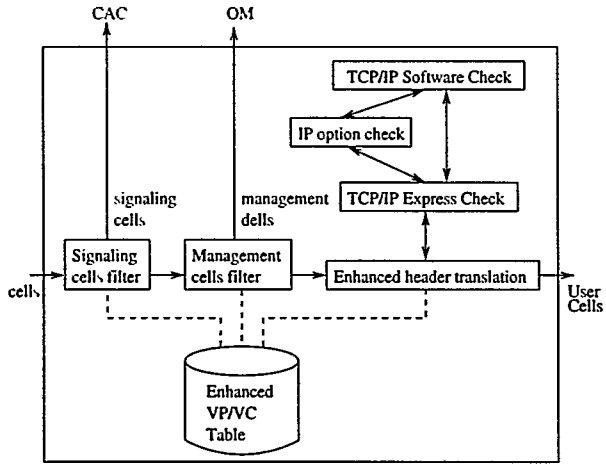

Figure 5: Cell Processing in Our ATM Firewall Switch

In a firewall switch, *cell processing* 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* 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 including 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 to 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 be 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[3].

[3]A cell time is around 800ns in an OC-12c line.

TCP/IP express check (TEC) block employs StorageTek's policy cache scheme [11]. 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. Otherwise, if the packet contains IP option or the packet misses the policy cache, TEC will instruct the EHT to put the packet into LCH. In the meantime, the cell(s) is forwarded either to IP option check block or to TCP/IP software check block. 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 StorageTek's policy cache to speed up the filtering process. The detailed process is also 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 also contains 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 lookup 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 standard switch decides whether an incoming signaling request can be accommodated based on whether the requested QoS (Quality of Service) 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, CAC will set up an entry in VP/VC table for the new connection and initiate the firewall-related fields such as FQoS. If the FQoS class 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 conducted in parallel with normal call admission control process and will not introduce any extra delay.

4.5 System Management (SM)

In a standard switch, SM handles all the management plane functions including fault management, performance management, configuration management, accounting management, security management, and traffic management [3]. In a firewall switch, we need to add a management function called firewall management in order to manage the firewall functions that need to be performed by the switch. It performs following functions:

- Maintaining firewall-related *managed objects* such as call screening rules.
- Executing the commands sent from firewall management server such as updating call screening rules.
- Monitoring 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 modification 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 is an ATM-attached workstation 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 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 in such a way that each packet received 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 offered a viable solution for high-speed packet-level filtering in ATM networks. We designed 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" (LCH) scheme to minimize the delay incurred when the cell misses the policy cache and to enable the ATM firewall process packets with IP option. We presented an implement-ready hardware design of the ATM firewall to demonstrate 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 includes a number of firewall-related fields to keep track of cell-level and packet-level state information. FQoS indicates the Filtering Quality of Service (FQoS) of the connection. EXP-1st and EXP-2nd remembers whether the first and the second cell of the packet is expected, respectively. DROP, PASS, and LCH records the decision on the current packet. SEL_NO records the serial number of the packet. The internal tag will be extended to contain a number of firewall-related fields (starts with T-). T-MONITOR 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 field is turned on. T-SEL_NO, T-LCH, T-OMCHK carries the value of SEL_NO, LCH, and OMCHK in VP/VC table to OM, respectively.

The flow diagram of the enhanced header translation (EHT) is shown in Fig. 6. When a cell arrives at this block, 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"). If the cell belongs to a class B connection ("Class B Traffic?"), the first two cells of the packet should be duplicated to traffic monitoring server. In the internal tags of the first two cells of a packet, the T-MONITOR bits are turned on. CSF will duplicate the cells with the T-MONITOR bit turned on to the traffic monitoring server. If the cell belongs to a class C connection, it needs to be filtered by the firewall switch. If it is not the first cell of the packet ("LCH?", "DROP?", and "PASS?"), the decision on the current packet has already been made and it is carried out accordingly. Otherwise, it will be forwarded to the TCP/IP express check block for the decision. When the decision arrives, corresponding fields will be modified to record the decision.

When a cell is passed from the enhanced header processing block to the TEC, it must be either the first cell or the second cell of a packet. If it is the second cell, it must be expected by the IP option check block (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. If the answer is yes, TCP/IP check will advise the enhanced cell processing block to put the packet into LCH. In the meantime, the cell will be forwarded to the IP option check block and is buffered there. If the cell does not contain IP option, TEC will try to match the packet header

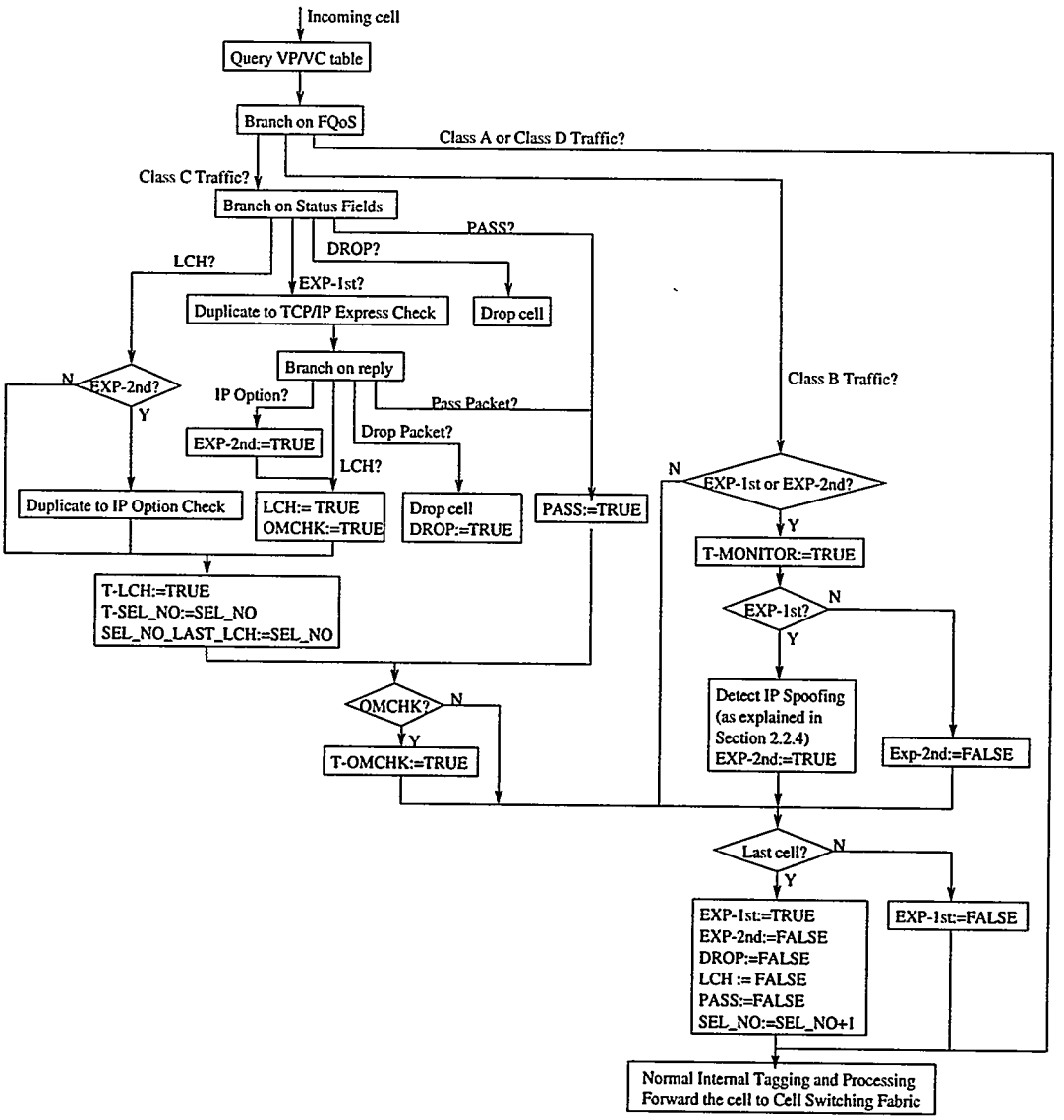

Figure 6: Flow Diagram of Enhanced Header Translation Block

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. If a cache miss happens, TCP/IP check block will advise the enhanced cell processing block to put the packet into LCH. In the meantime, the cell is forwarded to the TCP/IP software check block for further inspection.

When the IP Option Check block or TCP/IP Software Check block has finished inspecting the packet header , it will forward the decision to the TEC, and then to the enhanced cell processing block. The enhanced cell processing block will further forward the decision 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 into LCH.

IP option check block works almost in the same way as the TCP/IP express check 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 express check block immediately. Otherwise, the two cells will be handed over to TCP/IP software check block. When the decision comes back from TCP/IP software check block, it is in turn forwarded to the TEC. 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 other packets that follow the LCH packet in the same connection arrive. The third process is invoked when the decision on a LCH packet 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 packet occurs 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 any, 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, packets 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 of multicast cells because both involve a VPI/VCI table lookup. The third process can be accomplished efficiently by ASIC (Application Specific Integrated Circuit).

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 becomes empty, the OMCHK bit is cleared in the corresponding VP/VC entry in IM. This is accomplished by OMCHK Clearance Protocol, which is described in [18].

Acknowledgment: This research was partially supported by NSA Grant MDA904-96-1-0111.

References

[1] S. Bellovin, "Defending Against Sequence Number Attacks", *RFC 1948*, May 1996.

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

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

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

[5] D. Ginsburg, ATM: Solutions for Enterprise Internetworking, Addison Wesley, London, England, 1996.

[6] J. Hughes, "A High Speed Firewall Architecture for ATM/OC-3c", StorageTek Corp., MN, 1996, Available URL: http://www.network.com/.

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

[8] S. Kent, and R. Atkinson, "IP Encapsulating Security Payload (ESP)", IPSEC Working Group, May 1998.

[9] S. Kent, and R. Atkinson, "IP Authentication Header", IPSEC Working Group, May 1998.

[10] S. Kent, and R. Atkinson, "Security Architecture for the Internet Protocol", Network Working Group, May 1998.

[11] B. Kowalski, "ATLAS Policy Cache Architecture", StorageTek Corp., MN, 1997, Available URL: http://www.network.com/.

[12] D. Maughan, et al., "Internet Security Association and Key Management Protocol (ISAKMP)", IPSEC Working Group, March 1998.

[13] National Laboratory for Applied Network Research (NLANR), "FIX-West statistics data summaries", Available URL: http://www.nlanr.net/.

[14] L. Pierson, and T. Tarman, "Requirements for Security Signalling", ATM Forum/95-0137.

[15] Secant Network Technologies Inc., "Encrypting ATM Firewall", Secant Network Technologies Inc., NC, 1997, Available URL: http://www.secantnet.com/.

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

[17] T. Tarman, "Phase I ATM Security Specification", ATM Forum BTD-SECURITY-01.13, July, 1997.

[18] J. Xu, and M. Singhal, "Design of A High-Performance ATM Firewall", OSU Technical Report (OSU-CISRC-4/98-TR13), April, 1998.

[19] G. Ziemba, et al., "Security Considerations for IP Fragment Filtering", *RFC 1858*, Network Working Group, Oct. 1995.