February 9, 1996

A High Speed Firewall Architecture for ATM/OC-3c

James Hughes
Network Systems Corporation
Research and Advanced Development
7600 Boone Ave. No
Brooklyn, Park, MN 55428, USA
hughes@network.com
http://www.network.com/~hughes

Problem

The goal of ATM is high-performance wide area connectivity both site to site and end-point to end-point. Moving ATM from the labs and LANs to the MANs and WANs is not possible for many because of a lack of data stream policy, audit, commercial privacy, data integrity and authentication within existing ATM switches, end-points or network offerings. This paper describes a device whose intended purpose is to add these security features while maintaining the end-to-end high-performance nature of ATM and to create the general direction that high-performance network security may take in the future.

While the stated goals of many organizations is the ability to establish end-to-end ATM network connectivity, the traditionally used places to add security (Routers and Firewalls) do not exist in end-to-end ATM. Routers and Firewalls have also shown in the past that they are not capable of supporting the needed security at the 300Mb/s (150Mb/s FDX) rates required to protect ATM at full rate OC-3c.

Some experts say that network security is an endpoint host problem. If end-point computers can also provide policy enforcement and thus be trusted to be secure, then an ATM Firewall is not needed. Taking this assumption further, today, if *all* critical computers can be made secure and *kept* secure, and *all* the audits are all constantly watched, then Firewalls (ATM or otherwise) are not needed. History has shown that creating a secure computer is hard enough, but keeping hundreds (if not tens of thousands) of them continually secure has proven to be a significant (or impossible) problem[1]. Furthermore, managing to make sure that the audit trails of these secure machines are watched can be even more difficult. The advantages provided by localized (centralized) policy control (security) are significant.

Public ATM networks have the same security problems as the internet in general[13].

Introduction

This paper describes a continuing project to create a device to support data stream policy, audit, commercial privacy, data integrity and authentication.

There are four major parts to an ATM Firewall. They are:

1. High-performance Encryption,
2. High-performance packet filtering,
3. Proxies and
4. Trusted servers.

This paper concentrates on the first two, that is, high-performance encryption and filtering. Proxies and trusted servers are very important parts of any security system, but are not high-performance nor specific to ATM networks.

It is assumed that the reader is already familiar with the concepts of ATM[6], TCP/IP internetworking[7], access control (firewalls)[2] and encryption technology[3].

This document describes methods for employing access control and encryption technology over ATM circuits. It also predicts how this may possibly interact with future ATM security standards.

Access Control

Access control is the implementation of policy applied to data traffic accessing a given network. Firewalls and filtering routers provide the ability to control what information (in terms of packets, streams, sockets and other abstractions) flows through a given point. Typically this is performed on IP networks and IP protocols, but more frequently this is being applied to any and all protocols. In addition, in the ATM environment there are more filtering requirements, there are ATM call access control (call filtering) as well as ATM Network Service Access Points (NSAP - ATM address) to protocol address (IP address) binding.

ATM Call Admission Policy

ATM call setup messages need to be analyzed to provide connectivity policy enforcement. It is expected that “call screening” will be available in switches, but a security device can be made “non-addressable” thus potentially making the unit less susceptible to attack. Non-addressability implies that the device can interact with the switches on either side without either switch (or the ATM configuration in general) knowing that the security device exists. (Frustrating the attacker by making the security device invisible is a convenient side effect).

ATM Firewall devices that are not addressable can be used between switches (of different network providers) to monitor the call setup messages and look at these messages for content. If the content is determined to be dangerous, then it is possible for the unit to override the signalling request “in-flight” between the switches.

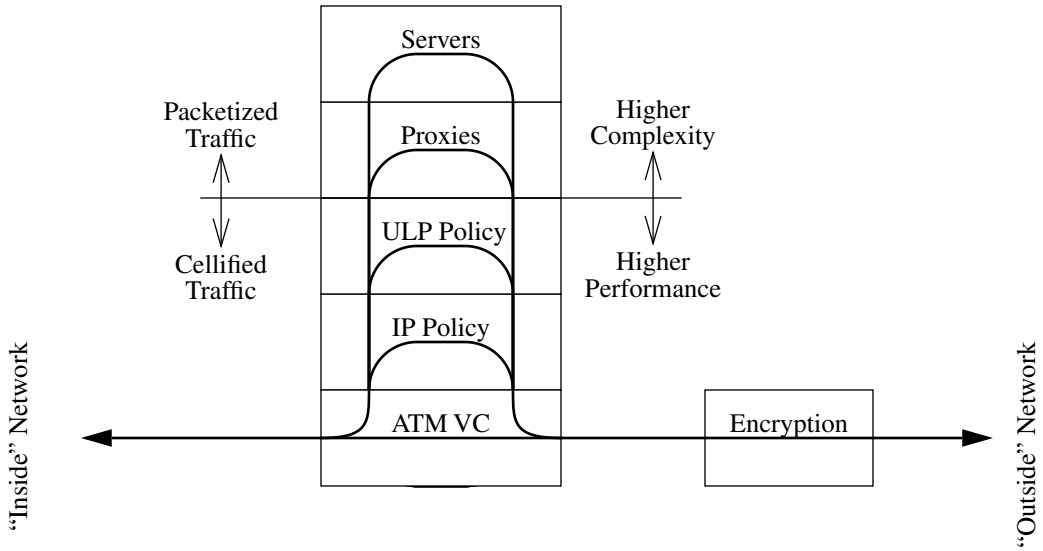

FIGURE 1. High-performance policy abstraction. Cells only go as far up the stack as the security policy dictates. Encryption adds privacy, authentication and integrity for and to the traffic flowing to and through the outside network.

Call setup policy is also key to triggering access control on the data carried within the ATM VC. This is called "data stream policy". Calls from certain NSAPs are expected to contain certain packet types and addresses. Data stream policy checks are required to ensure that the data flowing over that connection is of the proper type, contains the proper source and destination addresses, and that applications or even the application layer data does not violate some arbitrary policy when coming from a given ATM location.

Call setup policy can also be used to be determine the security (privacy, authentication, integrity) requirements of the call setups and data itself. Policy can state that traffic from certain sources and destinations will only be accepted when they are protected by certain levels of encryption, etc.

Data Stream Policy

Data stream policy is a necessary part of any network security policy. Several protocols are not secure and will never be[5]. Eliminating these protocols because they can be misused from outside an organization (even though these protocols are useful within an organization) may be counterproductive. Firewalls provide methods of segmenting communication domains so that different protocols with different vulnerabilities can be separated. Firewalls can also be used to eliminate known vulnerabilities of certain protocols[4].

Certain ATM streams are not manageable by policy checks (yet). For instance, doing policy checks on the contents of non-packetized virtual circuits (video, audio, telemetry) may not be possible. Doing a real time "dirty gesture search" in a video stream is a bit beyond the current computational techniques.

Source Based

In addition to securing protocols, it is important to provide unique policy based on source. Today, the typical Firewall treats all traffic from the internet the same, every source is equally untrusted because the data may be from someone pretending to be someone else. In the future when secure authenticated links are available the concept of trust changes. Once you know that the source is who they say they are, and the data has not been modified in transit, you can let this source do things that "just anyone" on the internet would not be allowed to do. On the flip side, even though you have authenticated a destination (say a business partner) you still do not want to allow that partner free run of your site. For example, if a business partner is engaged in Electronic Data Interchange for engineering information, allowing that partner to access the personnel records would, at least, be suspicious.

Policy flexibility is necessary to meet the diverse protocols and policy needs. This also has to be balanced against the performance impact of the specified policy. An existing example could be that of an FTP connection. The FTP protocol uses a control channel (socket) that can be thought of as a user at a terminal and a separate channel to transfer the data. Policies applied to the FTP control channel can be quite complex, like not being able to delete files or get /etc/passwd, while there may not need to be any policy applied data to the data transfer channel[1]. In this case the actual file transfer can be allowed to occur at media rates as long as the socket (and the VC it belongs to) can be verified. A flexible policy of allowing full speed for some portion of the ATM data and detailed analysis of other data is a desirable feature.

Socket verification

Data stream policy control using socket verification is the ability to ensure that only certain sockets are being used at any one time. The advantage is that certain sockets can, in certain situations, be trusted. Things like FTP data transfers (after the control channel has been validated), ICMP (Internet Control Message Protocol), NTP (network time protocol), RIP (Route Updates), DNS (Domain Name Server) and others are generally safe to let through. Other protocols like NNTP (Network News Transfer protocol) and SMTP (Simple Mail Transfer Protocol) can be re-routed or allowed when destination host can be trusted to process this protocol.

Access Control Implementation

This implement employs low level hardware assists for performing ATM VCI/VPI validation, IP header processing, security label validation and TCP and UDP port number validation as well as other processing. The purpose of these assists are to allow real time forwarding of cells without taking them out of the ATM cell stream if the policy allows.

The hardware assist does not preclude more complex policy, it just precludes processing these more complex policies at media speed. Policies that are not implementable within the assist facility force data, that needs to be checked to be taken out of the stream to be processed as complete packets. If they pass the policy checks they are then inserted back into the ATM cell stream.

ATM traffic is cellified usually in AAL5. In AAL5 there are 48 bytes of data in the first cell. The first cell is guaranteed to have the IP source, destination and IP protocol. This device checks these fields in hardware. Other fields like security labels, port numbers, TCP flags are not at a fixed offset and may even stretch into second or later cells of the packet. AAL5 does not have a start of cell flag, thus the state of the Virtual Circuit must be monitored for end of packet (user bit) indications.

1. As long as the Firewall is not doing virus searches or other detailed analysis.

Access Control

The low level hardware assists implements what we have named a Policy Cache™. This operates similar to a processor cache by placing certain policies close to the data path if they are being actively used. 192 bit wide Content Addressable Memories (CAMs[1]) are used to hold the information that describe cached policies. The cache is in three parts, the VCI/VPI CAM, the IP CAM and the Transport layer control blocks.

The first hardware assist is the VCI/VPI CAMs that ensure only authorized VCI/VPI addresses are used. This also determines if the VC is traffic should be subjected to further analysis. VCs that are not packetized or for streams that data policy is not desired, then the cells are forwarded without delay.

The second hardware assist is IP policy cache. This is a set of CAMs that initially start out empty (similar to processor cache). As new cells arrive that are not in the IP policy CAMs, these cells are reassembled into a packet and checked for validity. If a packet meets policy constraints an valid entry is added to the cache and the packet is put back into the cell stream.

Finally, transport layer control blocks contain the information about ports, flags and other information that is necessary to ensure socket policies are valid.

Cells that match all CAM entries and are valid according to the Transport Layer control blocks are guaranteed to be valid to the larger policy that the system is trying to guarantee.

If the cells are (packet is) permissible according to the data stream policy, then the cells are (obviously) forwarded. If the cells that comprise the packet are not permissible, the cell that begins the violation and all subsequent cells of the packet are deleted from the stream and a zero length end of packet indication is sent into the VC to keep the host at the far side in sync with the stream. In addition to deleting the incorrect policy, alarms and audit events can be sent to the appropriate network management centers.

Scalability in terms of hardware CAM sizes and transport control block area are an issue for further research. The initial devices can handle 1024 VC and IP policies. This is not a system limit, but rather a limit of the policy cache. The same “Least Recently Used (LRU)” concepts that are used with processor caches can also be used with policy caches.

Scalability in terms of performance is also an issue for future research. This architecture is designed to be capable of operating at rates exceeding OC-12c (622Mb/s fdx). Filtering the traffic while it is still cellified is certainly easier than reassembling, filtering and then segmenting back into cells.

1. CAMs implement an extremely fast search by comparing the contents against all locations in a memory and returning the address of match(s) if they exist.

Encryption

ATM encryption has a significant number of complexities that do not exist in the world of leased lines. In ATM, one outgoing link could be communicating with many hundreds of partners (for security reasons must not share the same key), cells can be lost due to congestion, data can be inserted into the virtual circuit, and data can be reinserted (replayed) into the VC. These are significant complexities that ATM shares in common with the Internet, Frame Relay and even X.25 networks.

As a part of this project, NSC will be creating an encryption ASIC with UTOPIA[8] interfaces, implementing key-agile encryption, authentication, integrity, and replay prevention.

Call Setup Authentication (and key exchange)

Authentication provides the means of determining who is on the other end of the circuit for the call setup messages. This is similar to “Caller-ID” that exists today, but ensures that the identification can not be forged. Cryptographic authentication is typically done with public key algorithms that check authentication using third party key certification system.

Once identity has been established, a key is exchanged (through the use of key distribution algorithms like Diffie-Hellman) to ensure that the two ends share the same secret key which will be used to encypher the data with a symmetric key algorithm like DES. Public-key algorithms are not useful for bulk encryption because they are not fast enough[3].

Privacy

ATM encryptors have used the term Key-Agile encryption to explain how multiple streams each are encyphered using a pair-wise unique key. This ensures that the possibility of third parties sharing the same key is close to zero. This is important since one “compromised” node can not compromise any other node which the data was not sent to. Given the complexities of key-agile encryption, single key systems can be contemplated, but their security is minimal because stealing one unit or key provides the eavesdropper with everyone’s key. Key-agile systems limit a compromise to a single node.

Encryption algorithms for ATM must not only be key-agile, but they must also be scalable in terms of performance. ATM is already commercially available at OC-12c and there is work going on OC-48c and OC-192c as well as talk about OC-768c running at 40Gb/s. At these rates, the speed of the encryption algorithm must scale to low cost end-points in the megabit range up to a level 4.5 orders of magnitude faster. Existing CBC (Cipher Block Chaining) methods are not scalable in this manner.

One method of providing parallelization and pipelining is operating a cipher in counter mode[3]. This mode allows a simple transformation of the IV to allow many cipher operations to occur at the same time.

Performance of the algorithm in terms of mm^2 of silicon and cipher speed is important. Some algorithms are much faster than DES, but the silicon required to create a similar speed is significantly larger. DES was designed in 1974 to be simple to implement in a chip of that day[9]. If the algorithm can be parallelized to one or many complete cells, then the size of a single cipher unit (in mm^2) is a significant factor.

Encryption

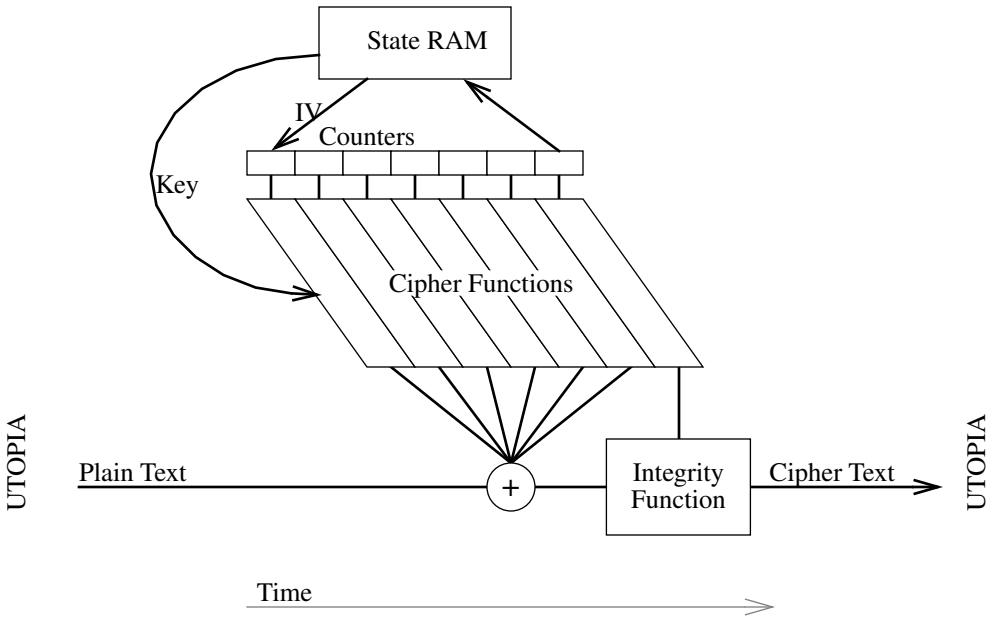

FIGURE 2. Diagram of single chip, feed-forward privacy and integrity chip for key-agile ATM encryption.

Integrity

The integrity of a cell stream is also an important issue. The encryption ASIC will include a keyed cell hashing function after the encryption function that mixes the results so that the malicious attacker can not reliably change the packet in flight, insert or rearrange the packets without causing the bad cell(s) to be completely randomized. If this cell is a part of an AAL5 packet, then the probability of that packet having a correct CRC is significantly less than 2^{-32} [10].

In addition to the malicious attacker, the issue of “normal” Cell loss in ATM is an issue. Traffic multiplexing (of anything other than Constant Bit Rate [CBR] traffic) presents the possibility that there will be a statistical collision so that traffic may overflow a queue and cells are lost. Reassembling AAL5 packets containing lost cells will cause length errors and the CRC of the packet to be in error[10].

Periodic re-sync cells (<1%) will be added so that out of sync conditions will not last a significant amount of time.

Authentication

Authentication is provided to the cell stream through the combination of the privacy transform and the AAL5 CRC. If the sender does not posses the same key as was established at call setup, then the sender has significantly less than a 2^{-32} probability of getting a packet through.

Replay prevention

Replay prevention is a significant threat when using Cipher Block Chaining (CBC) mode encryption. In CBC mode, the algorithm will sync up to the data stream so that if the stream changes to other properly encrypted data (possibly older data) then the stream will come into sync after a brief “burp” of random bits[3].

Counter mode is a method of creating a key stream that moves in only one direction. Old data can not be decrypted if it is presented out of order to the key stream. The most significant feature of counter mode is that the re-sync cells themselves must not be replayable, that is, they must always move forward in some known sequence space.

Standards

Standardization is the task of creating safe and interoperable systems. One analogy that can be used is the electrical outlet. If the plug and socket both conform to the same standard, then they will be compatible. The encryption technology described here is an example of something that would be valuable if standardized so that each end of a system could be manufactured by different companies and still be compatible.

The are a significant number of standardization groups such as the ATM Forum, (national organizations such as) ANSI, and (internationally organizations such as) the ITU. Each of these organizations have their own security experts. There are even countries which have limits on encryption technology usage within their borders, or for export. The definition of a common standard in this area will be a significant and long term task.

While it is possible to ignore these groups, and create a single or ad hoc solution, the benefit of standardization, that of peer reviewed compatible solutions from many vendors, outweigh the single or ad hoc solution advantages.

Taking the electrical outlet analogy further, the concept of multiple standards bodies coming up with multiple solutions in the encryption area is a foregone conclusion (thus a market for universal outlet converters or universal power supplies). Even if there were only one method for end-to-end ATM encryption, there is always the possibility that users will be able to use other privacy techniques, such as IPSEC[11], 802.10[12], etc., to create a secured system.

Access control is a different standardization model than encryption protocols. Access control, or the enforcement of a one sided policy, is an example of a task that does not need to cooperatively communicate with the entity whose access is being policed (and better off if it does not).

Access control may be considered similar to the line cord attached to a electrical plug. While there may (someday) be minimum safety standards for Firewall devices (as there are for electrical cords), this is not an area that effects the compatibility of the socket and plug.

While the outlook for standards in the area of ATM encryption and access control may not be impendent, further work to create appropriate standards is warranted.

Conclusion

This document has described a device that will be able to provide flexible network security for commercial off the shelf end-to-end ATM devices and networks. The device has the ability to monitor the call setup and IP traffic itself for customer specific policy, while running at full OC-3c rates. This device will also have the ability to extend trust and privacy, at full OC-3c rates, across unsecure (public) infrastructures. By combining the encryption and access controls, it is possible to allow a trusted party to communicate while still maintaining the ability to police their traffic.

Acknowledgments

I wish to thank the ATM Firewall team of Rod Pullis, Rodney McJimpsey, Nancy (RodNot) Golio, and Steve Timm for their hard work and creativity that allows me to stand up at conferences and take credit for their work.

Thanks go to Sarah Wheeler, BJ Kowalski, Nancy Golio and Ted Doty with help editing the paper.

About the Author

James Hughes is an Fellow at Network Systems Corporation for 15 years specializing in High Speed Networking and Information Security. He is a current or former member of ANSI X3T11 (HIPPI), T1S1 (ATM), T1X1 (SONET), The ATM Forum, IETF and is a member of the Transport Area IESG.

References

1. Cheswick, Bellevin
2. Ranum, “Thinking about Firewalls”
3. Schneier, “Applied Cryptography”
4. Chapman, Zwicky, “Building Internet Firewalls”
5. Bellovin, “Problems in the TCP/IP Protocol Suite”
6. McDysan, Spohn, “ATM”
7. Comer, “Internetworking with TCP/IP”
8. ATM Forum, “UTOPIA interface definition”
9. Coopersmith, “DES and its Strength Against Attack”, IBM Journal of Research and Development. Vol. 38, No. 3. <http://eagle.almaden.ibm.com/journal/rd38-3.html#two>
10. Partridge, Hughes, Stone, “Performance of Checksums and CRCs over Real Data”, ACM SIGCOMM ‘95, Cambridge. <http://www.network.com/~hughes/AAL5_CRC.ps>

References

11. Atkinson, “Security Architecture for the Internet Protocol, RFC 1825”, IETF, <ftp://ds.internic.net/rfc/rfc1825.txt>
12. IEEE, “802.10”, <http://www.c3i.wsoc.com/sils/silsarti.html>
13. The ATM Report, ISSN 1072-981X, Broadband Publishing Corporation. security issue was Vol. 3, No. 7