Firewall Traversal authorization system

Michael Richardson and Kai Martius v1.1, 9 July 1997

This document describes a public key certificate mechanism to authorize traversal of multiple security gateways (firewalls). This work is independant of transport layer in concept, and could apply to IPsec, TLS, or SecSH. It is applied here to IPsec. The SPKI certificate format is used here.

Contents

1 Introduction

This document is a result of recent discussions in the IETF ipsec, IETF secsh and IETF mobileip working groups about how to trust security gateways (aka firewalls) with end-to-end (host to host) encryption and authentication keys.

This document describes the problem, some solutions which have been suggested in the past, and then goes on to describe a system of public key signed certificates that would allow a series of appropriate connections to automatically be setup.

$Gupta97-1$ gives an expanded view of the problems, and other solutions. Unlike $Gupta97-1$, this document does not limit itself to firewalls that are known apriori, or under the same administrative control.

For a solution to be scalable to the entire internet, the policy must either be learnt dynamically from correspondant nodes (e.g. arrived at through negotiation), or must be available in some pre-existing global database such as the domain name system.

1.1 Definition of terminology

Here is a network of two security gateways, a client node and a server node.

```
          C---{G1}---{G2}---S

    C is the client.
    G1/G1 are gateways.
    S is the server.
```

Since there are potentially more than one transport or network layer connection, we define some terms to describe the different end points.

C

is the transport layer originator. `TLO`

S

: is the transport layer target. `TLT`

C/G1

: is a network layer originator/target pair. `NLO/NLT/`

G1/G2

: is a network layer originator/target pair.

G2/S

: is a network layer originator/target pair.

If discussing application layer protocols (e.g. SSH, TLS) through security gateways, then the transport layer designations above should be replaced with the session layer designations (e.g. URL, hostname), and the network layer designations with transport layer designations (e.g. TCP endpoints, SSH/TLS host keys).

The end points of the different types of connections will be denoted with a subscript. So, when `C` is used in an `TLO` context, the symbol `C_s` will be used (s for Server) . When `C` is used in a NLO context, the symbol `C_g1` or `C_g2` will be used.

2 Introduction to the problem

The problem is not as some say, merely a question of allowing security protocols to pass unexamined through a security gateway. Many environments have very strong auditing requirements, and encrypted traffic is by design, intended to thwart attempts for third parties to eavesdrop.

Further, this policy relies on the security of target hosts being perfect. Were this the case in practice, for all vendors, for all releases, both new and old, the firewall might not be required at all.

2.1 Key Sharing methods

It has been suggested by several people XXX that a key sharing protocol will solve the problem. The firewall(s) would be provided with the encryption keys in order to examine the traffic. Alternately, a copy of the authentication keys would allow the firewall to verify the origin of the packets, thus allowing it to apply its access policy.

```
C {\(<\)}------+----------+------{\(>\)} S
               G1         G2
```

This solution is not appropriate because the problem is more complicated. In general there will be a combination of network address translating firewall, topology hiding firewalls, and particularities of various protocols.

A host behind a network address translation may have an address that is not available, or worse: illegal, to its correspondant node. The firewall must therefore be involved during all the key exchange protocol because the firewall (or the address that the `TLx` is translated to) is the logical end point for the encryption, not the actual `TLx`.

When topology is hidden by a firewall, there must be some mechanism to map the connection to the intended `TLT`. That is, despite the topology hiding, there is a need to name selected pieces of the internal topology, and communicate that name to the firewall.

The difficulty of doing this in general for all protocols in first generation application layer firewalls, even for outbound connections, is what caused the development of protocols like SOCKS.

A firewall that supports protocols that use more than a single logical connection also has a requirement to see the contents of the "control" or "nameserver" connection. The typical example is FTP, but CuSeeMe, RealAudio, SunRPC (portmap is the nameserver connection), and most multicast protocols have this problem.

Furthermore, sharing of authentication keys leads to the problem that receiver can only verify a group of senders which in fact isn't an authentication anymore.

2.2 Stacked or tunnelled solutions

A second proposal is to stack algorithms. This is being proposed by $Gupta97-2$ for the mobileIP group. This is best illustrated by a diagram.

```
C_s  {\(<\)}------+----------+------{\(>\)} S_c
C_g2 {\(<\)}------+--------{\(>\)} G2_c***{\(>\)}
C_g1 {\(<\)}----{\(>\)} G1_c*****{\(>\)}
```

Each arrow represents a secure connection, the upper connections being transported in the lower connections. Stars represent unencrypted (from that layer's point of view) connections.

Note: there is n+1 layers when n gateways are involved. Two gateways are typical (one at each location), but should two higher security networks (e.g. research or finance) inside the lower security organizational network need to communicate, this number rises to 4. Future topologies could further increase n.

Some systems which have implemented this method include SOCKS: one runs SOCKS inside SOCKS.

The most glaring problem, however, is that the data, if encrypted, is opaque to both gateways! The traversal problem has been solved, but the auditing and protocol requirements remain.

There is some difficulty with dealing with ICMP messages, since the gateway may not be able to do anything with them.

The repeated encapsulation (tunnelling) of one packet inside another increases the overhead, and for packet protocols, decreasing the amount of payload per packet.

This has serious efficiency and reliability implications. Node C may have difficulty getting accurate ICMP messages back, and may not be able to set its TCP MSS properly leading to excessive fragmentation. This problem is not unique to this topology.

2.3 Virtual Circuit solutions

An alternate way to set up the associations is a series of adjacent tunnels rather than a stack of tunnells. This is similar to what happens in ATM networks with virtual circuits are setup. The ATM switches negotiation frame ids between themselves and act as stateful routers.

There may still be a need for strong end to end security in this situation, so the tunnels may be used to transport an end to end security association. At most, there are two layers of security associations with this method.

```
End to end      C_s  {\(<\)}--------+----------+--------{\(>\)} S_c
Hop by hop      C_g1 {\(<\)}------{\(>\)}G1_c
                             G1_g2{\(<\)}----{\(>\)}G2_g1
                                         G2_s{\(<\)}-----{\(>\)} S_g2
```

There are three ways to arrange the two sets of security associations:

1. Delegated traversal

the gateways may provide sufficient trust in identity that an internal tunnel is not required. In that case, an upper protocol may appear immediately inside the hop-by-hop security header. The hop-by-hop SA's may provide authentication alone, or encryption and integrity. On-the-wire IPsec packets might look like:

```
IP[C_g1-{\(>\)}X] AH[C_g1-{\(>\)}G1_c TCP/UDP/ICMP]
```

or

```
IP[C_g1-{\(>\)}X] ESP[C_g1-{\(>\)}G1_c TCP/UDP/ICMP]
```

2. Audited Traversal

If the gateways do not permit unexamined (i.e. encrypted) data to pass through, then the end to end (C_s/S_c) SA must be authentication only. The hop-by-hop SA's would have to provide the privacy features. On-the-wire IPsec packets might look like:

```
IP[C_g1-{\(>\)}X] ESP[C_g1-{\(>\)}G1_c
    IP[C_s-{\(>\)}S_c] AH[C_s-{\(>\)}S_c] TCP/UDP/IP/ICMP]
```

3. Authenticated traversal

If the gateways do allow encrypted data to pass through, then the end to end SA's can include privacy features. The hop-by-hop SA could be authentication only, or might include additional privacy features to thwart traffic analysis. On-the-wire IPsec packets might look like:

```
IP[C_g1-{\(>\)}X] AH[C_g1-{\(>\)}G1_c
        ESP[C_s-{\(>\)}S_c] <whatever>]
```

or

```
IP[C_g1-{\(>\)}X] ESP[C_g1-{\(>\)}G1_c
        ESP[C_s-{\(>\)}S_c] <whatever>]
```

Some notes on above:

1. In the audited case (1) it is possible for the gateways to control what kind of data can flow through by looking at the next protocol header in the AH packet. Thus the gateway can prevent unauthorized tunnels from being formed. The gateway could allow IP without necessarily giving up the ability to audit. This is not true for the authenticated case, because once any ESP is allowed through, the gateway gives up control of what protocols get transmitted through the gateway.
2. At all times a single SA's could be used for different streams of traffic (at the same sensitivity), or multiple SA's could be used for a single stream of traffic.
3. in case 2, where there is an outer AH or integrity protected ESP, the inner ESP is NOT REQUIRED to also provide integrity protection, but may do so.
4. the X above could be either S_c or it could be G1_c. It is not clear which is more appropriate. In the NAT situation, there are reduced choices, in the absense of NAT, either is possible. (ISSUE)

2.4 Issues raised

The following questions are posed, which this document will attempt to resolve:

1. how does the client know that it can trust gateway g1 or g2?
2. how does the server know that it can trust gateway g1 or g2?
3. how to tell the real server who the real client is?

Problems 1 and 2 are related, and are solved by the same mechanism. The information as to the real client is also passed by this proceedure.

3 Firewall traversal certificates

If one makes an analogy between security perimeters and altitudes, then the end nodes can be thought of as being on plateaus, possibly with several ledges on the way up, with large amounts of plain between plateaus.

Virtual private network tunnels are then bridges that span between ledges/plateaus. Prior to building a bridge, some guide wires (symmetric keys) must be established. To establish the guide wires requires sending an ambassador out with appropriate proof of origin. The ambassador meets up with a representative of the distant plateau, and they exchange credentials. The meeting place, which occurs on the plain, at zero altitude will be referred to as "security zone 0".

The wide open Internet is a good example of such a zone and it is probably equivalent to sea level. In general, the security zone 0 is just the lowest point on the path between the two security plateaus.

The remainder of this document is therefore a description of the credentials that are provided by non-zero security zones to lower altitude security levels.

3.1 The IP-Gateway Certificate

At each downward hop, a certificate is used to delegate the identity of the `TLO` node to an lower security level gateway. At security level 0, the ambassador meets with their counterpart. The counterpart also brings a set of certificates.

Once proof of identify has been exchanged, the ambassador needs to bring that proof back to the security plateau. The ambassador does this by using the same certificate chain that was issued to it, to delegate the higher identity to the ambassador.

The certificates are SPKI format. The issuer of the certificate is ultimately the `TLO` or `TLT` node. The subject of the certificate is the node to which authority is being delegated. The authentication being delegated the list of hosts for which the gateway is authorized to speak for.

In the simplest case, there is only one certificate issued by the `TLx` nodes, and it can only delegate authority for itself. A more complicated system would have an organizational CA or ISP based CA signing a certificate that delegated a particular portion of the IP address space to a particular key.

3.2 Definition of certificate

v4-network

this is followed by the v4 network prefix and the length of the prefix. Hosts are indicated by a prefix length of 32.

v6-network

this is followed by the v6 network prefix and the length of the prefix. Hosts are indicated by a prefix length of 128.

host

this is followed by a DNS name, or by a SDSI name

3.3 An example

For example, Somecompany.com used to use the class C subnet: 192.1.2.128/26. This is further divided into several 16 address subnet that exist behind a packet filter. Further firewalls inside did network address translation as well. The network topology would look like:

```
X14---fw1----pf1---<INTERNET>
```

where `X14` has a private network address, `fw1` provides network address translation, and `pf1` provides filtering only. Also assume that this organization had a IPv6 prefix of `c0:ff:ee:04:ba:be::`

Given that, DNSSEC would delegate authority for 192.1.2.128/26 and for somecompany.com to the key `SC1`, then the following statements might be made:

```
(cert (issuer  SC1)
```

```
        (subject pf1)
        (auth   (ip-gateway
                    (v4-network 205.233.54.128 26)
                    (v6-network c0:ff:ee:04:ba:be:: 48))))

(cert (issuer (ref SC1 xterm14.somecompany.com))
        (subject fw1)
        (auth   (ip-gateway (host
                        (ref SC1 xterm14.somecompany.com)))))
```

We have to use a host name here because there is no IP address that can be legitimately used.

$ISSUE : I'mnotsurehowthe(stop - at - key)etc.stuffofSPKIapplieshereyet, butIKNOWthatitdoes$

$ISSUE : awildcardforthehostnamemightalsobedesirable, butperhapsSDSIgroupswouldalsoworkhere$

3.4 Completing the certificate loop

Since all certificates chains must be rooted with a self-signed certificate, the verifier of the ip-gateway chain (for instance, the `TLx` node) must determine the origin of the `TLO`'s authority to speak for that address/hostname. The origin of its authority would come from either DNSSEC or X.500 servers that it trusts.

This initial trust relationship would appear in this node via a preconfigured root CA key, or cross certification of DNSSEC servers... the technology for doing this is not described here.

4 Security Considerations:

This entire document discussed a security protocol.

5 References:

$RFC - 1825$

R. Atkinson, "Security Architecture for the Internet Protocol", RFC-1825, August 1995.

$RFC - 1826$

R. Atkinson, "IP Authentication Header", RFC-1826, August 1995.

$RFC - 1828$

P. Metzger, W. A. Simpson, "IP Authentication using Keyed MD5",RFC-1828, August 1995.

$KSM - AH$

New AH draft.

Maughan97

D. Maughan, M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", version 7, draft-ietf-ipsec-isakmp-07.txt, work in progress: February 21, 1997

Harkins97

D. Harkins, D. Carrel, "The resolution of ISAKMP with Oakley", draft-ietf-ipsec-isakmp-oakley-03.txt, version 3, work in progress: February 1997

Ylo97 − 1

T. Ylonen, "SSH Transport Layer Protocol", draft-ietf-secsh-transport-00.txt, version 0, work in progress: March 22, 1997

Ellison97

C. Ellison, B. Frantz, B.M. Thomas, "Simple Public Key Certificate", draft-ietf-spki-cert-structure-01.txt, version 1, work in progress: March 25, 1997

SDSI

R. Rivest, B. Lampson, "SDSI - A Simple Distributed Security Infrastructure *SDSI*", <URL:http://theory.lcs.mit.edu/ rivest/sdsi11.html>, October 2, 1996

Monte96

G. Montenegro, V. Gupta, "Firewall Support for Mobile IP", draft-montenegro-firewall-mobileip-00.txt, <URL:http://skip.incog.com/drafts/draft-montenegro-firewall-mobileip-00.txt>, work in progress: Sept. 14, 1996

Doraswaymy97 − 1

N. Doraswaymy. "Implementation of Virtual Private Networks (VPNs) with IP Security", draft-ietf-ipsec-vpn-00.txt, work in progress: March 12, 1997

Gupta97 − 1

V. Gupta, S. Glass, "Firewall Traversal for Mobile IP: Goals and Requirements", draft-ietf-mobileip-ft-req-00.txt, work in progress: Jan. 20, 1997

Gupta97 − 2

V. Gupta, S. Glass, "Firewall Traversal for Mobile IP: Guidelines for Firewalls and Mobile IP entities", draft-ietf-mobileip-firewall-trav-00.txt, work in progress: March 17, 1997

5.1 Authors' Addresses

Michael C. Richardson
Sandelman Software Works Corp.
152 Rochester Street
Ottawa, ON K1R 7M4
Canada

Telephone: +1 613 233-6809
EMail: mcr@sandelman.ottawa.on.ca
http://www.sandelman.ottawa.on.ca/People/Michael_Richardson/

Kai Martius
Dresden University of Technology
Faculty of Medicine
Institute of Medical Informatics and Biometrics
Fetscherstr. 74
01307 Dresden
Germany

EMail: kai@imib.med.tu-dresden.de

5.2 Expiration and File Name

This draft expires January 9, 1998

Its file name is draft-richardson-ipsec-traversal-cert-01.txt