IPSec:
Manageable & Interoperable Implementations

Leon Towns-von Stauber
Seattle SAGE Group, March 2002

http://www.occam.com/ocr/security/

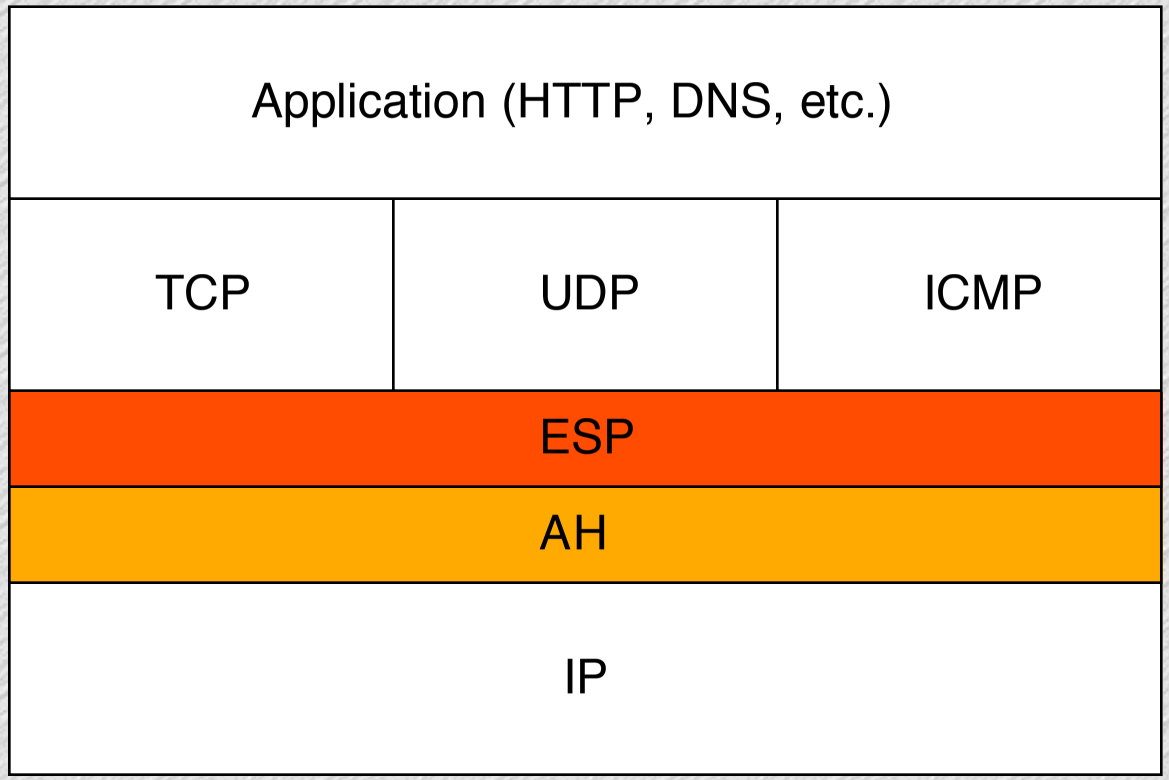
Contents

- Introduction
- Goals
- Solaris 8 IPSec
- Linux IPSec (FreeS/WAN)
- BSD IPSec (KAME)
- Interoperable IPSec
- Service Protection
- Key Generation Tool

Introduction

- What is IPSec?
  - Framework for securing TCP/IP communications
    - Secure authentication of hosts
    - Encryption of network traffic
  - Operates “between” Internet and Transport layers of networking stack (layers 3 & 4 of OSI model)
    - Transparent to upper-layer protocols
      - Unlike SSL, SSH, Kerberos, etc.
      - Simpler to implement
      - Unable to take advantage of upper-layer data (e.g., user information)
    - Encapsulates headers of Transport layer and above
    - Encapsulates parts of IP header
      - Can’t encapsulate mutable fields, such as TTL
  - Required in IPv6

Introduction (cont'd.)



IPSec in the TCP/IP Network Stack

Introduction (cont'd.)

- Protocols
  - Authentication Header (AH, protocol 51)
    - Provides source authentication, integrity assurance, and replay protection
    - RFC 2402
  - Encapsulating Security Payload (ESP, protocol 50)
    - Provides same services as AH, plus confidentiality
    - Initially did only encryption; authentication functions were added after problems pointed out by Bellovin (USENIX Security 1996)
    - RFC 2406
- Authentication & Encryption Algorithms
  - Key Exchange (authentication): Manual, IKE, ...
  - Hashes (integrity): NULL, HMAC-MD5, HMAC-SHA1, ...
  - Encryption (confidentiality): NULL, DES, 3DES, ...

Introduction (cont'd.)

- Security Association
  - Defines protocol, authentication algorithm and key, encryption algorithm and key (if any) between source and destination
  - Identified by Security Parameters Index (SPI)

Goals

- Minimal number of host-independent configuration files
  - No host-dependent config files
- Minimal number of unique configuration lines
  - More precisely, want to keep number of config lines to O(n) or less, i.e. no more than linear growth with number of hosts
    - O(1) (static) is ideal, but unattainable
    - Want to avoid $O(n^2)$, or worse
- Secure as much traffic as is manageable, given above constraints
- Secure certain traffic regardless of manageability
- Robustness
  - Tolerant of configuration errors

Solaris 8 IPSec

- Overview
  - Built into Solaris 8
  - Not heavily advertised, or fully documented
- Features & Limitations
  - Can specify use of IPSec by source and destination addresses and ports, and by transport protocol
  - Don't need to specify source in security association
    - Single SA can be applicable to any source
  - More flexible use of SPIs
    - Can share between different source/destination pairs
    - Can split between AH & ESP
  - Manual keying only, no automatic keying

Solaris (cont'd.)

- Installation
  - Install the files from the SUNWcryr package
    - Download it from Sun's Web site
    - `/kernel/strmod/encrdes`, `/kernel/strmod/encr3des`
  - Add the following to `/etc/init.d/inetinit` before the point where `ipsecconf` is run:

```
if [ -f /etc/inet/ipseckey.conf ] ; then
     /usr/sbin/ipseckey -f /etc/inet/ipseckey.conf
fi
```

  - Reboot to get the encryption modules loaded properly
    - Using `modload` without rebooting won't work

Solaris (cont'd.)

- Example Solaris-to-Solaris Configuration
  - Setup: Three hosts (10.0.0.1, 10.0.0.2, 10.0.0.3)
  - All traffic between hosts secured
    - Symmetric keys
    - SHA1 authentication with AH & ESP
    - 3DES encryption with ESP
    - SSH excepted (for secure management in case of configuration error)

Solaris (cont'd.)

- Example Solaris-to-Solaris Configuration (cont'd.)
  - /etc/inet/ipseckey.conf

```
add ah spi 0x101 dst 10.0.0.1 authalg sha authkey
edc95ba7331030c96f9d4dbae7bf1c6dffaa2ca7

add esp spi 0x101 dst 10.0.0.1 authalg sha authkey
d3a7a5088400ce164c9c5d43fbe9c0f0c0a562b3 encralg 3des encrkey
1b1252838a5a958bac426934a94642c2fecc17c9c61df2cb

add ah spi 0x102 dst 10.0.0.2 authalg sha authkey
fad2a00ba4105f951f152d41edeb83d2c797ee1f

add esp spi 0x102 dst 10.0.0.2 authalg sha authkey
883bf21450b7f6235a14a23da1d764a320f43eed encralg 3des encrkey
be0b18afd10dc7e5c35499befddd2f63d1212fe8e8400ba3

add ah spi 0x103 dst 10.0.0.3 authalg sha authkey
e3c3897b23669159c0b87fb9dac76230dc689f65

add esp spi 0x103 dst 10.0.0.3 authalg sha authkey
0c971f10849415969e58f4cd4e47ef155554e250 encralg 3des encrkey
1690567ebd5639f6c4e900abbecc7ab1ac5a0b22e65e7eae
```

Solaris (cont'd.)

- Example Solaris-to-Solaris Configuration (cont'd.)
  - /etc/inet/ipsecinit.conf

```
# Don't use IPSec on local connections, as it seems to cause
# problems (despite documentation to the contrary).
{ daddr 127.0.0.1 } bypass { dir out }
{ saddr 127.0.0.1 } bypass { dir in }

# Disable IPSec requirements for broadcast traffic.
{ daddr 10.20.1.255 } bypass { dir in }

# Don't use IPSec on SSH connections. In case something goes
# wrong with IPSec, we still need a means of secure remote
# administration.
{ sport 22 ulp tcp } bypass { dir out }
{ dport 22 ulp tcp } bypass { dir out }
{ sport 22 ulp tcp } bypass { dir in }
{ dport 22 ulp tcp } bypass { dir in }
```

Solaris (cont'd.)

- Example Solaris-to-Solaris Configuration (cont'd.)
  - `/etc/inet/ipsecinit.conf` (cont'd.)

```
# Set up IPSec between Solaris hosts.
{ daddr 10.0.0.1 }     apply    { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 sa shared }
{ saddr 10.0.0.1 }     permit   { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 }
{ daddr 10.0.0.2 }     apply    { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 sa shared }
{ saddr 10.0.0.2 }     permit   { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 }
{ daddr 10.0.0.3 }     apply    { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 sa shared }
{ saddr 10.0.0.3 }     permit   { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 }
```

  - `ipseckey -f /etc/inet/ipseckey.conf`
  - `ipsecconf -q -a /etc/inet/ipsecinit.conf`

Solaris (cont'd.)

- Example Solaris-to-Solaris Configuration (cont'd.)
  - Test network connectivity
    - Try pinging from one IPSec-enabled host to another
    - Use a packet sniffer to verify IPSec encapsulation
      - `snoop` *`IP_address`* `ah,icmp`
      - You should see ESP packets, not ICMP
    - If the pings are successful and the packets are encapsulated, you're done
    - If not:
      - Check `syslog` messages on either end for problems
      - Watch the counters in the output of `ndd /dev/ipsecah ipsecah_status` and `ndd /dev/ipsecesp ipsecesp_status`

Solaris (cont'd.)

- Number of unique configuration files
  - Two, used on all hosts
- Number of unique configuration lines per host
  - Two in `ipseckey.conf`, to determine security associations
  - Two in `ipsecinit.conf`, to determine access policies
  - f(n) = 4n
    - Linear, which is acceptable
    - Can eliminate host-specific lines in `ipsecinit.conf` if subnets can be split off for IPSec

Linux IPSec (FreeS/WAN)

- Overview
  - www.freeswan.org
  - Project goal to secure all Internet traffic
    - Focuses on IPSec gateways (“tunnel mode”), rather than host-to-host security (“transport mode”)
    - Opportunistic IPSec eliminates need for explicit prior negotiation; similar in spirit to StartTLS
  - Good documentation
- Features & Limitations
  - Automatic keying available
  - Opportunistic IPSec uses DNS lookups for keys
  - Cannot specify use of IPSec by port or by protocol

Linux (cont'd.)

- Installation
  - Compile a new kernel
  - Download and unarchive the FreeS/WAN source
  - From the FreeS/WAN source directory:
    - `make oldgo`
    - `make kinstall`
    - `/etc/ipsec.secrets` created by `ipsec rsasigkey 2048` (run automatically during install)
  - `lilo`
  - Reboot
- Example Linux-to-Linux Configuration
  - Setup: Three hosts (10.0.1.1, 10.0.1.2, 10.0.1.3)
  - All traffic between hosts secured
    - Asymmetric keys
    - MD5 authentication with ESP
    - 3DES encryption with ESP

Linux (cont'd.)

- Example Linux-to-Linux Configuration (cont'd.)
  - /etc/ipsec.conf

```
config setup
        interfaces=%defaultroute
        klipsdebug=none
        plutodebug=none
        plutoload=%search
        plutostart=%search
        uniqueids=yes

conn %default
        left=%defaultroute

conn opportunistic
        right=%opportunistic
        authby=rsasig
        compress=yes
        auto=route
```

Linux (cont'd.)

- Example Linux-to-Linux Configuration (cont'd.)
  - DNS

```
1.1.0.10.in-addr.arpa  IN TXT   "X-IPSec-Server(10)=10.0.1.1"
1.1.0.10.in-addr.arpa  IN KEY   HOST|NOCONF IPSEC RSAMD5
a8fb75e3e6be131072f6493a6d7659b0
2.1.0.10.in-addr.arpa  IN TXT   "X-IPSec-Server(10)=10.0.1.2"
2.1.0.10.in-addr.arpa  IN KEY   HOST|NOCONF IPSEC RSAMD5
2cc0513b3b2dac6ec3140edf6a30d09f
3.1.0.10.in-addr.arpa  IN TXT   "X-IPSec-Server(10)=10.0.1.3"
3.1.0.10.in-addr.arpa  IN KEY   HOST|NOCONF IPSEC RSAMD5
a6576c8662aa7a6703ac08d84b59d057
```

  - Appropriate KEY records can be found in `ipsec.secrets`

Linux (cont'd.)

- Example Linux-to-Linux Configuration (cont'd.)
  - Test network connectivity
    - Try pinging from one IPSec-enabled host to another
    - Use a packet sniffer to verify IPSec encapsulation
      - `tcpdump -pq -i eth0 -s 100 host` *`IP_address`*
      - You should see `ip-proto-51` packets, not ICMP
    - If the pings are successful and the packets are encapsulated, you're done
    - If not:
      - Check `syslog` messages on either end for problems
      - Use `ipsec look` to study the configuration

Linux (cont'd.)

- Number of unique configuration files
  - One used on all hosts
  - One used per host (`ipsec.secrets`)
  - f(n) = n + 1
    - Not too bad, especially since `ipsec.secrets` is generated only once
- Number of unique configuration lines per host
  - None in `ipsec.conf`
  - Two in DNS
  - f(n) = 2n
    - Linear, which is acceptable

BSD IPSec (KAME)

- Overview
  - www.kame.net
  - IPv6 and IPSec implementation for *BSD
- Features & Limitations
  - Automatic keying available
  - Cannot specify use of IPSec by port or by protocol?

Interoperable IPSec

- Example Solaris-to-Linux Configuration
  - Setup: Four hosts
    - Two Solaris (10.0.0.1, 10.0.0.2)
    - Two Linux (10.0.1.1, 10.0.1.2)
  - All traffic between hosts secured
    - Symmetric keys between Solaris and others
    - Asymmetric keys between Linux
    - SHA1 authentication with AH & ESP between Solaris and others
    - MD5 authentication with ESP between Linux
    - 3DES encryption with ESP
    - SSH excepted between Solaris

Interoperable IPSec (cont'd.)

- Example Solaris-to-Linux Configuration (cont'd.)
  - /etc/inet/ipseckey.conf

```
add ah spi 0x101 dst 10.0.0.1 authalg sha authkey
edc95ba7331030c96f9d4dbae7bf1c6dffaa2ca7

add esp spi 0x101 dst 10.0.0.1 authalg sha authkey
d3a7a5088400ce164c9c5d43fbe9c0f0c0a562b3 encralg 3des encrkey
1b1252838a5a958bac426934a94642c2fecc17c9c61df2cb

add ah spi 0x101 src 10.0.0.1 dst 10.0.1.1 authalg sha authkey
edc95ba7331030c96f9d4dbae7bf1c6dffaa2ca7

add esp spi 0x101 src 10.0.0.1 dst 10.0.1.1 authalg sha
authkey d3a7a5088400ce164c9c5d43fbe9c0f0c0a562b3 encralg 3des
encrkey 1b1252838a5a958bac426934a94642c2fecc17c9c61df2cb

add ah spi 0x101 src 10.0.0.1 dst 10.0.1.2 authalg sha authkey
edc95ba7331030c96f9d4dbae7bf1c6dffaa2ca7

add esp spi 0x101 src 10.0.0.1 dst 10.0.1.2 authalg sha
authkey d3a7a5088400ce164c9c5d43fbe9c0f0c0a562b3 encralg 3des
encrkey 1b1252838a5a958bac426934a94642c2fecc17c9c61df2cb
```

Interoperable IPSec (cont'd.)

- Example Solaris-to-Linux Configuration (cont'd.)
  - `/etc/inet/ipseckey.conf` (cont'd.)

```
add ah spi 0x102 dst 10.0.0.2 authalg sha authkey
fad2a00ba4105f951f152d41edeb83d2c797ee1f

add esp spi 0x102 dst 10.0.0.2 authalg sha authkey
883bf21450b7f6235a14a23da1d764a320f43eed encralg 3des encrkey
be0b18afd10dc7e5c35499befddd2f63d1212fe8e8400ba3

add ah spi 0x102 src 10.0.0.2 dst 10.0.1.1 authalg sha authkey
fad2a00ba4105f951f152d41edeb83d2c797ee1f

add esp spi 0x102 src 10.0.0.2 dst 10.0.1.1 authalg sha
authkey 883bf21450b7f6235a14a23da1d764a320f43eed encralg 3des
encrkey be0b18afd10dc7e5c35499befddd2f63d1212fe8e8400ba3

add ah spi 0x102 src 10.0.0.2 dst 10.0.1.2 authalg sha authkey
fad2a00ba4105f951f152d41edeb83d2c797ee1f

add esp spi 0x102 src 10.0.0.2 dst 10.0.1.2 authalg sha
authkey 883bf21450b7f6235a14a23da1d764a320f43eed encralg 3des
encrkey be0b18afd10dc7e5c35499befddd2f63d1212fe8e8400ba3
```

Interoperable IPSec (cont'd.)

- Example Solaris-to-Linux Configuration (cont'd.)
  - /etc/inet/ipsecinit.conf

```
# Don't use IPSec on local connections, as it seems to cause
# problems (despite documentation to the contrary).
{ daddr 127.0.0.1 } bypass { dir out }
{ saddr 127.0.0.1 } bypass { dir in }

# Disable IPSec requirements for broadcast traffic.
{ daddr 10.20.1.255 } bypass { dir in }

# Don't use IPSec on SSH connections. In case something goes
# wrong with IPSec, we still need a means of secure remote
# administration.
# Since SSH to the Linux boxes needs to be encapsulated, we
# need to specifically except Solaris boxes.
{ daddr 10.0.0.1 sport 22 ulp tcp } bypass { dir out }
{ daddr 10.0.0.1 dport 22 ulp tcp } bypass { dir out }
{ saddr 10.0.0.1 sport 22 ulp tcp } bypass { dir in }
{ saddr 10.0.0.1 dport 22 ulp tcp } bypass { dir in }
{ daddr 10.0.0.2 sport 22 ulp tcp } bypass { dir out }
{ daddr 10.0.0.2 dport 22 ulp tcp } bypass { dir out }
{ saddr 10.0.0.2 sport 22 ulp tcp } bypass { dir in }
{ saddr 10.0.0.2 dport 22 ulp tcp } bypass { dir in }
```

Interoperable IPSec (cont'd.)

- Example Solaris-to-Linux Configuration (cont'd.)
  - `/etc/inet/ipsecinit.conf` (cont'd.)

```
# Set up IPSec between Solaris hosts.
{ daddr 10.0.0.1 }     apply    { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 sa shared }
{ saddr 10.0.0.1 }     permit   { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 }
{ daddr 10.0.0.2 }     apply    { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 sa shared }
{ saddr 10.0.0.2 }     permit   { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 }
```

Interoperable IPSec (cont'd.)

- Example Solaris-to-Linux Configuration (cont'd.)
  - `/etc/inet/ipsecinit.conf` (cont'd.)

```
# Set up IPSec between Linux hosts and Solaris hosts.
{ daddr 10.0.1.1 saddr 10.0.0.1 }   apply    { auth_algs sha1
encr_algs 3des encr_auth_algs sha1 sa shared }
{ saddr 10.0.1.1 daddr 10.0.0.1 }   permit   { auth_algs sha1
encr_algs 3des encr_auth_algs sha1 }
{ daddr 10.0.1.1 saddr 10.0.0.2 }   apply    { auth_algs sha1
encr_algs 3des encr_auth_algs sha1 sa shared }
{ saddr 10.0.1.1 daddr 10.0.0.2 }   permit   { auth_algs sha1
encr_algs 3des encr_auth_algs sha1 }
{ daddr 10.0.1.2 saddr 10.0.0.1 }   apply    { auth_algs sha1
encr_algs 3des encr_auth_algs sha1 sa shared }
{ saddr 10.0.1.2 daddr 10.0.0.1 }   permit   { auth_algs sha1
encr_algs 3des encr_auth_algs sha1 }
{ daddr 10.0.1.2 saddr 10.0.0.2 }   apply    { auth_algs sha1
encr_algs 3des encr_auth_algs sha1 sa shared }
{ saddr 10.0.1.2 daddr 10.0.0.2 }   permit   { auth_algs sha1
encr_algs 3des encr_auth_algs sha1 }
```

Interoperable IPSec (cont'd.)

- Example Solaris-to-Linux Configuration (cont'd.)
  - /etc/ipsec.conf

```
config setup
        interfaces=%defaultroute
        klipsdebug=none
        plutodebug=none
        plutoload=%search
        plutostart=%search
        uniqueids=yes
        manualstart="sun1 sun2"

conn %default
        left=%defaultroute
```

Interoperable IPSec (cont'd.)

- Example Solaris-to-Linux Configuration (cont'd.)
  - `/etc/ipsec.conf` (cont'd.)

```
conn sun1
     also=manual
     spi=0x101
     right=10.0.0.1
     ahkey=0xedc95ba7331030c96f9d4dbae7bf1c6dffaa2ca7
     espauthkey=0xd3a7a5088400ce164c9c5d43fbe9c0f0c0a562b3

espenckey=0x1b1252838a5a958bac426934a94642c2fecc17c9c61df2cb

conn sun2
     also=manual
     spi=0x102
     right=10.0.0.2
     ahkey=0xfad2a00ba4105f951f152d41edeb83d2c797ee1f
     espauthkey=0x883bf21450b7f6235a14a23da1d764a320f43eed

espenckey=0xbe0b18afd10dc7e5c35499befddd2f63d1212fe8e8400ba3
```

Interoperable IPSec (cont'd.)

- Example Solaris-to-Linux Configuration (cont'd.)
  - `/etc/ipsec.conf` (cont'd.)

```
conn manual
     type=transport
     ah=hmac-sha1-96
     esp=3des-sha1-96

conn opportunistic
     right=%opportunistic
     authby=rsasig
     compress=yes
     auto=route
```

  - DNS

```
1.1.0.10.in-addr.arpa  IN TXT   "X-IPSec-Server(10)=10.0.1.1"
1.1.0.10.in-addr.arpa  IN KEY   HOST|NOCONF IPSEC RSAMD5
a8fb75e3e6be131072f6493a6d7659b0
2.1.0.10.in-addr.arpa  IN TXT   "X-IPSec-Server(10)=10.0.1.2"
2.1.0.10.in-addr.arpa  IN KEY   HOST|NOCONF IPSEC RSAMD5
2cc0513b3b2dac6ec3140edf6a30d09f
```

Interoperable IPSec (cont'd.)

- Number of unique configuration files
  - Two, used on all Solaris hosts
  - One, used on all Linux hosts
  - One used per Linux host (`ipsec.secrets`)
  - $f(n_L) = n_L + 3$
    - Still not too bad
- Number of unique configuration lines/stanzas per host
  - Two per Solaris host in `ipseckey.conf`
  - Two per Solaris/Linux pair in `ipseckey.conf`
  - Six per Solaris host in `ipsecinit.conf`
  - Two per Solaris/Linux pair in `ipsecinit.conf`
  - Two per Solaris host in `ipsec.conf` (counting `manualstart` line)
  - Two per Linux host in DNS
  - $f(n_S, n_L) = 10n_S + 2n_L + 4n_Sn_L$
    - Quadratic -- Yuck!

Interoperable IPSec (cont'd.)

- Number of unique configuration lines/stanzas per host (cont'd.)
  - $f(n_S, n_L) = 10n_S + 2n_L + 4n_Sn_L$
    - $f(n_S + 1, n_L) - f(n_S, n_L) = 4n_L + 10$
      - For every Solaris host added, we need to create $4n_L + 10$ unique configuration entries
    - $f(n_S, n_L + 1) - f(n_S, n_L) = 4n_S + 2$
      - For every Linux host added, we need to create $4n_S + 2$ unique configuration entries

IPSec for Individual Service

- Securing LDAP was my original motivation for exploring IPSec
- Example Service Protection Configuration
  - Setup: Four hosts
    - One Solaris LDAP (port 389) server (10.0.0.1)
    - One Solaris LDAP client (10.0.0.2)
    - Two Linux LDAP clients (10.0.1.1, 10.0.1.2)
  - All traffic between like hosts secured, all LDAP traffic secured
    - Symmetric keys between Solaris and others
    - Asymmetric keys between Linux
    - SHA1 authentication with AH & ESP between Solaris and others
    - MD5 authentication with ESP between Linux
    - 3DES encryption with ESP
    - SSH excepted between Solaris

Service Protection (cont'd.)

- Example Service Protection Configuration (cont'd.)
  - /etc/inet/ipseckey.conf

```
add ah spi 0x101 dst 10.0.0.1 authalg sha authkey
edc95ba7331030c96f9d4dbae7bf1c6dffaa2ca7

add esp spi 0x101 dst 10.0.0.1 authalg sha authkey
d3a7a5088400ce164c9c5d43fbe9c0f0c0a562b3 encralg 3des encrkey
1b1252838a5a958bac426934a94642c2fecc17c9c61df2cb

add ah spi 0x101 src 10.0.0.1 dst 10.0.1.1 authalg sha authkey
edc95ba7331030c96f9d4dbae7bf1c6dffaa2ca7

add esp spi 0x101 src 10.0.0.1 dst 10.0.1.1 authalg sha
authkey d3a7a5088400ce164c9c5d43fbe9c0f0c0a562b3 encralg 3des
encrkey 1b1252838a5a958bac426934a94642c2fecc17c9c61df2cb

add ah spi 0x101 src 10.0.0.1 dst 10.0.1.2 authalg sha authkey
edc95ba7331030c96f9d4dbae7bf1c6dffaa2ca7

add esp spi 0x101 src 10.0.0.1 dst 10.0.1.2 authalg sha
authkey d3a7a5088400ce164c9c5d43fbe9c0f0c0a562b3 encralg 3des
encrkey 1b1252838a5a958bac426934a94642c2fecc17c9c61df2cb
```

Service Protection (cont'd.)

- Example Service Protection Configuration (cont'd.)
  - `/etc/inet/ipseckey.conf` (cont'd.)

```
add ah spi 0x102 dst 10.0.0.2 authalg sha authkey
fad2a00ba4105f951f152d41edeb83d2c797ee1f

add esp spi 0x102 dst 10.0.0.2 authalg sha authkey
883bf21450b7f6235a14a23da1d764a320f43eed encralg 3des encrkey
be0b18afd10dc7e5c35499befddd2f63d1212fe8e8400ba3
```

Service Protection (cont'd.)

- Example Service Protection Configuration (cont'd.)
  - /etc/inet/ipsecinit.conf

```
# Don't use IPSec on local connections, as it seems to cause
# problems (despite documentation to the contrary).
{ daddr 127.0.0.1 } bypass { dir out }
{ saddr 127.0.0.1 } bypass { dir in }

# Disable IPSec requirements for broadcast traffic.
{ daddr 10.20.1.255 } bypass { dir in }

# Don't use IPSec on SSH connections. In case something goes
# wrong with IPSec, we still need a means of secure remote
# administration.
# Since SSH to the Linux boxes needs to be encapsulated, we
# need to specifically except Solaris boxes.
{ daddr 10.0.0.1 sport 22 ulp tcp } bypass { dir out }
{ daddr 10.0.0.1 dport 22 ulp tcp } bypass { dir out }
{ saddr 10.0.0.1 sport 22 ulp tcp } bypass { dir in }
{ saddr 10.0.0.1 dport 22 ulp tcp } bypass { dir in }
{ daddr 10.0.0.2 sport 22 ulp tcp } bypass { dir out }
{ daddr 10.0.0.2 dport 22 ulp tcp } bypass { dir out }
{ saddr 10.0.0.2 sport 22 ulp tcp } bypass { dir in }
{ saddr 10.0.0.2 dport 22 ulp tcp } bypass { dir in }
```

Service Protection (cont'd.)

- Example Service Protection Configuration (cont'd.)
  - `/etc/inet/ipsecinit.conf` (cont'd.)

```
# Require use of IPSec for LDAP.
{ sport 389 } apply    { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 sa shared }
{ dport 389 } permit   { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 }

# Set up IPSec between Solaris hosts.
{ daddr 10.0.0.1 }     apply    { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 sa shared }
{ saddr 10.0.0.1 }     permit   { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 }
{ daddr 10.0.0.2 }     apply    { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 sa shared }
{ saddr 10.0.0.2 }     permit   { auth_algs sha1 encr_algs 3des
encr_auth_algs sha1 }
```

Service Protection (cont'd.)

- Example Service Protection Configuration (cont'd.)
  - `/etc/inet/ipsecinit.conf` (cont'd.)

```
# Set up IPSec between Linux hosts and Solaris LDAP server.
{ daddr 10.0.1.1 saddr 10.0.0.1 }   apply    { auth_algs sha1
encr_algs 3des encr_auth_algs sha1 sa shared }
{ saddr 10.0.1.1 daddr 10.0.0.1 }   permit   { auth_algs sha1
encr_algs 3des encr_auth_algs sha1 }
{ daddr 10.0.1.2 saddr 10.0.0.1 }   apply    { auth_algs sha1
encr_algs 3des encr_auth_algs sha1 sa shared }
{ saddr 10.0.1.2 daddr 10.0.0.1 }   permit   { auth_algs sha1
encr_algs 3des encr_auth_algs sha1 }
```

Service Protection (cont'd.)

- Example Service Protection Configuration (cont'd.)
  - /etc/ipsec.conf

```
config setup
     interfaces=%defaultroute
     klipsdebug=none
     plutodebug=none
     plutoload=%search
     plutostart=%search
     uniqueids=yes
     manualstart="sun1"

conn %default
     left=%defaultroute

conn sun1
     also=manual
     spi=0x101
     right=10.0.0.1
     ahkey=0xedc95ba7331030c96f9d4dbae7bf1c6dffaa2ca7
     espauthkey=0xd3a7a5088400ce164c9c5d43fbe9c0f0c0a562b3

espenckey=0x1b1252838a5a958bac426934a94642c2fecc17c9c61df2cb
```

Service Protection (cont'd.)

- Example Service Protection Configuration (cont'd.)
  - `/etc/ipsec.conf` (cont'd.)

```
conn manual
     type=transport
     ah=hmac-sha1-96
     esp=3des-sha1-96

conn opportunistic
     right=%opportunistic
     authby=rsasig
     compress=yes
     auto=route
```

  - DNS

```
1.1.0.10.in-addr.arpa  IN TXT   "X-IPSec-Server(10)=10.0.1.1"
1.1.0.10.in-addr.arpa  IN KEY   HOST|NOCONF IPSEC RSAMD5
a8fb75e3e6be131072f6493a6d7659b0
2.1.0.10.in-addr.arpa  IN TXT   "X-IPSec-Server(10)=10.0.1.2"
2.1.0.10.in-addr.arpa  IN KEY   HOST|NOCONF IPSEC RSAMD5
2cc0513b3b2dac6ec3140edf6a30d09f
```

Service Protection (cont'd.)

- Number of unique configuration files
  - Two, used on all Solaris hosts
  - One, used on all Linux hosts
  - One used per Linux host (`ipsec.secrets`)
  - $f(n_L) = n_L + 3$
- Number of unique configuration lines/stanzas per host
  - Assume LDAP server's entries have already been accounted for; only consider configuration for clients
  - Two per Solaris host in `ipseckey.conf`
  - Two per Linux host in `ipseckey.conf`
  - Six per Solaris host in `ipsecinit.conf`
  - Two per Linux host in `ipsecinit.conf`
  - Two per Linux host in DNS
  - $f(n_S, n_L) = 8n_S + 6n_L$
    - Linear -- Yay!
    - Limiting use of IPSec resulted in more manageable config

Key Generation Tool

```
$ ./ipsec_genkey.pl
usage:  ipsec_genkey.pl auth_alg encr_alg dest_addr [src_addr]

     'auth_alg' may be either 'md5' or 'sha'.
     'encr_alg' may be either 'des' or '3des'.
     'dest_addr' and 'src_addr' are either IP addresses or hostnames.

     Use 'man ipseckey' (on Solaris) or 'man ipsec.conf' (on Linux)
     for more information.
$ ./ipsec_genkey.pl  sha  3des  10.0.0.1
Solaris
-------
add ah spi 0x8e8 dst 10.0.0.1 authalg sha authkey 42952a0ab23d080d06bf5cb72366b4ef774a2
add esp spi 0x8e8 dst 10.0.0.1 authalg sha authkey 77a7fa5f10f7dbefe5699591ddbf3fcb6735
95b60a6efd7821aff5f24cda31059692b49be09dd16020f2

FreeS/WAN
---------
     spi=0x8e8
     right=10.0.0.1
     rightahkey=0x42952a0ab23d080d06bf5cb72366b4ef774a254d
     rightespauthkey=0x77a7fa5f10f7dbefe5699591ddbf3fcb67352853
     rightespenckey=0x95b60a6efd7821aff5f24cda31059692b49be09dd16020f2
```