"IPSec VPN Advanced Troubleshooting Guide"

by Darren Hartman
Senior Engineer, IPSec Testing and Certification Lab
ICSA Labs, a division of TruSecure Corporation

November 3, 2003

Advanced Troubleshooting of Interoperability Problems with IPsec Products

Introduction

Due to the complexity of the IKE and IPsec protocols, achieving interoperability in a multi-vendor environment has always been a significant challenge to VPN administrators. In 1998, ICSA Labs addressed this issue by establishing an IPsec Product Developers Consortium and an IPsec Product Certification Testing Program with interoperability as the primary focus. During the past five years, ICSA Labs analysts have been identifying interoperability problems and determining whether those problems were configuration issues or flaws in the IPsec implementation of the products. The process has helped to develop the troubleshooting skills and techniques that the analysts rely on every day.

The purpose of this document is to describe troubleshooting techniques to aid VPN administrators in resolving interoperability issues with IPsec products. It is not intended to be a configuration guidance or a best practices document and is vendor neutral. This document builds on other information available from ICSA Labs and the IPsec product vendors as described in the next section. This paper was written assuming the reader will use the documentation listed in the next section; therefore, details covered in those documents will not be repeated in this text.

Documentation

Product-specific information is covered in the documentation supplied by the IPsec product vendor as well as ICSA Labs Lab-notes. The ICSA Labs VPN Product Technical Configuration Guidelines provides an overview of the IKE and IPsec protocols and related terms as they pertain to configuring IPsec devices. The details of the IKE and IPsec protocols are specified in the Internet Engineering Task Force (IETF) Request for Comments (RFC) documents at the IETF IPsec Working Group web page, www.ietf.org/html.charters/ipsec-charter.html.

Vendor

Before turning on the power for an IPsec device, the administrator should read the available documentation. Begin with the vendor supplied documentation. Be sure to read Release Notes, Errata information, Configuration Notes, etc. in addition to the Administrator's Guides and User's Manuals. Such references are often overlooked but give valuable information that may prevent having misconfiguration or interoperability problems. It is important to become familiar with the vendor's support area of their web-site also.

ICSA Labs

ICSA Labs is the best source for interoperability information regarding ICSA Labs certified IPsec products (www.icsalabs.com/html/communities/ipsec/index.shtml). At the ICSA Labs web-site, you will find Lab-notes for each IPsec product that is certified. The Lab-notes contain

the configuration steps taken by ICSA Labs analysts during the testing of the products in the Lab. Also, there are notes for any special or undocumented steps that were necessary to make specific products inter-operate. Additionally, the Lab-notes provide details for any problems that were found and patches needed to resolve those problems. It is important to pay close attention to the version numbers and patches when comparing the certified product with your product. If you have a different version or build or do not have the same patches applied, your device may behave differently. If there is any question regarding the version of your product, contact the vendor technical support center. Another helpful feature in each of the Lab-notes is a table of the other certified IPsec products that were used in testing. This table has links to the other products' Lab-notes for easy reference.

In addition to the Lab-notes, another source of helpful information for IPsec VPN products from ICSA Labs is the VPN Product Technical Configuration Guidelines which can be downloaded at www.icsalabs.com/html/communities/ipsec/IPsec_Technical_Config_Guidelines.pdf
This reference is a configuration guide that aids administrators in achieving interoperability when using ICSA Labs certified IPsec products. It gives a brief description of the IKE and IPsec protocols with respect to configuring IPsec devices, talks about what to consider when configuring parameters for each phase of the IKE negotiation, and describes how to avoid common configuration errors. As a summary, the VPN Product Technical Configuration Guidelines provides a checklist of the points to remember when configuring IPsec products.

Where to Begin

Installation and Network Connectivity

After reading the documentation and installing the product, some steps should be taken to simplify troubleshooting interoperability problems later. Network connectivity should be verified, both on the private and public side. In cases where the product is software based and installed on a standard O.S., e.g. Windows, Solaris, or Linux, network connectivity should be verified first after the O.S. is installed but before the VPN software is installed.

If after the VPN software is installed there is a connectivity problem the VPN software may have default filter rules in place denying traffic. If applicable, check the rule base and use the documentation to ascertain the default posture of the product. Using hosts connected to both the private and public networks, ping to and from the VPN product. The results may indicate whether filter rules are interfering. Using a sniffer will be helpful to determine where traffic may be blocked.

Other network configuration options to be aware of are IP forwarding, and routing vs. bridge mode configurations. These are general networking concepts and definitions will not be provided here, but be sure to follow the vendor supplied documentation closely when configuring these options as they vary from product to product.

IPsec Configuration for Troubleshooting

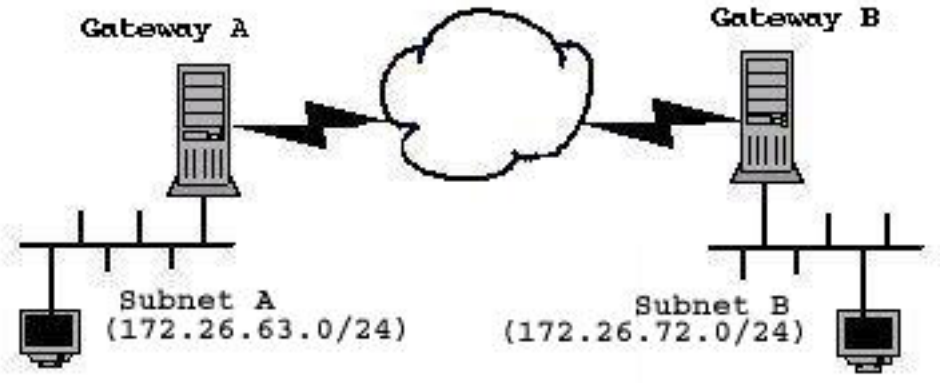
After network connectivity and filter rules have been verified, it is time to configure the VPN products to establish an IPsec tunnel. Start with a simple configuration. For example, use single

sets of parameters for the two IKE phases. (Some products can be configured to send or respond to multiple proposals.) A good set of parameters to start with would be 3DES (encryption) and SHA-1 (authentication/integrity) for both Phase 1 and Phase 2.

For the other settings in Phase 1, use Main Mode, Pre-Shared Key (PSK) authentication and Diffie-Hellman Group 2 (key exchange - KE). Even if you will be using certificates for authentication, use PSKs initially. This step will greatly simplify troubleshooting IKE problems. After the tunnel, initially configured to use PSKs, has been operating successfully, reconfigure to use certificates.

Entering PSKs can be tricky. For security reasons, the PSK should be a long hexadecimal value or string of characters that unfortunately will make typos more likely. To make matters worse, some products hide what is typed and do not have confirmation boxes, so be careful entering PSKs on both IPsec devices. Later in this guide, we will discuss what to look for when troubleshooting a PSK mismatch problem.

For Phase 2 settings, use ESP Tunnel-mode only, not AH, and disable Perfect Forward Secrecy (PFS). If no problems occur, then enable PFS for added security and configure AH if necessary. The other concern for Phase 2 is the traffic selectors (TS). These settings describe what traffic may be dropped, passed, or protected by the IPsec tunnel. The traffic is defined using the addresses (IP address, IP subnet, or range of IP addresses) of the protected entities, both local and remote, and the protocol and port number of the traffic. Initially, use a single subnet for both the local and remote addresses. For example, if VPN gateway A protects subnet A (172.26.63.0/24) and the remote peer, VPN gateway B, protects subnet B (172.26.72.0/24), the settings on VPN gateway A would be local entity: 172.26.63.0/24 and remote entity: 172.26.72.0/24. VPN gateway B would be configured with the opposite local and remote entity settings. Set both protocol and port to "any" on both gateways.



Begin with lifetimes on both sides set to 20 minutes for Phase 1 and 10 minutes for Phase 2. After the IPsec tunnel has been established for several hours to confirm re-keying has operated properly, reset lifetimes appropriately. Proper lifetime settings will depend on achieving a good balance between performance and security in accordance with your organization's policy. Note that most products also allow lifetimes to be set based on the amount of data passing through the IPsec tunnel.

Be aware that a product's configuration may include settings for acceptable ranges for parameters such as lifetimes that the peer may send in a proposal. Again, check the documentation to learn what the defaults are to verify that the peer's settings are configured accordingly. More details about lifetimes and troubleshooting are given later. Also, refer to ICSA Labs' Lab-notes, as the parameters described above are very similar to those used in certification testing, and the Lab-notes detail the parameter settings used by the analysts.

Initial Checks for Configuration Problems
If you cannot establish a tunnel, the first thing to do is to confirm all settings. If you are working with ICSA Labs certified products, then the most likely problem is a configuration mismatch.

Be sure:
- PSKs are entered correctly,
- all IKE and IPsec parameters match on each device,
- ranges for what is accepted from the peer matches what the peer will send,
- default ranges are known,
- firewall/filter rules are simple (start with "allow all", but only in a test environment).

If problems occur after the tunnel is established, e.g. traffic stops passing through the tunnel, note the time elapsed since the tunnel was initiated. If the problem happens at a 10 minute interval (based on the lifetimes given above), the cause is most likely a re-keying problem. Reset both sides and verify that the tunnel can be initiated from the peer that was originally the responder. If the tunnel can be established all parameters are probably set correctly. In this case, check the ICSA Labs Lab-notes to see if there are any known issues with either product. Do not give up yet. Read the rest of this guide and gather as much information as possible. Even if the problem is not resolved, this information will be very valuable when talking to the vendor's technical support representatives.

Tip: after making configuration changes it is a good idea to reset the IPsec devices. Some products have the capability for an administrator to clear all Security Associations (SAs). Do this on both sides before attempting to re-establish a tunnel. If a product does not have the capability to easily clear SAs it may be necessary to reboot.

Logging

Logging is an extremely important tool in troubleshooting interoperability problems. Unfortunately, all logs are not created equal. Many are inadequate, while others are very verbose and cryptic. In both cases, it is very frustrating to administrators who are under the gun trying to resolve issues. So, it is very important to include logging in the evaluation of VPN products before purchasing them. Logs vary greatly from product to product. This section does not describe how to use any specific product’s log but provides guidance that can be applied in most cases.

Log Types

For logs that are helpful, there are generally two types. One type strikes the right balance between providing enough information and being easy to understand. The other type, usually debug output, gives very verbose information including hex dumps of IKE messages. The hex dump of IKE payloads is very useful in finding out exactly what is wrong during an IKE negotiation, but it requires a very thorough understanding of IKE and the format of the IKE messages. Refer to the IETF RFCs for IKE and the related protocols. Be advised, if you are not familiar with reading RFCs starting with those for IKE and IPsec may be overwhelming. It is highly recommended to learn about the IKE and IPsec protocols before you begin. You will also find that with a better understanding of IKE and IPsec, configuring these devices will not be so intimidating. There are many books available for IKE/IPsec that you can find by searching the on-line book sellers.

Using the Logs

The first step to using the logs is to enable logging and then find where the information is stored. This varies greatly from product to product, so once again, you must refer to the documentation. Also look for how to get more verbose information. Be aware though, a product may not perform well with verbose logging or debug mode enabled. Enable verbose logging only if needed for troubleshooting.

Use the log output to help determine at what point during IKE negotiation problems are occurring. In some cases, the log will identify the problem precisely, "Potential pre-shared key mismatch". But with other logs, it may not be as clear, "BAD_MESSAGE-failed its integrity check". However in the latter case, the logs, used for this example, also indicated that Main Mode was not completed which narrowed down the problem to Phase 1 settings such as the PSK. (In both examples above the log message was a result of a mismatched PSK.) So, even if the logs do not clearly identify the problem, they can be useful to zero in on the problem, especially when used with the troubleshooting techniques described in the next section.

Tip: When you have the IPsec devices establishing tunnels properly, capture the log entries or debug information of successful IKE negotiations for future reference.

Troubleshooting with tcpdump/Ethereal

In addition to using logs and verifying configuration information, the best way to identify interoperability problems is to monitor the traffic between the two IPsec peers. To do this you will need a sniffer. The sniffer most prevalently used by ICSA Labs analysts is tcpdump. In addition to tcpdump, Ethereal, a GUI application, is very useful to view captured traffic. These tools usually come with Linux and BSD operating systems, but refer to the web-sites for the latest versions at www.tcpdump.org and www.ethereal.com.

Using tcpdump

When you have your sniffer installed, you will need to connect it to the network of the VPN device's external interface. This is where you'll be able to see the IKE negotiations and the IPsec

traffic. Below are some different ways to invoke tcpdump depending on what traffic you want to see.

The simplest way to use tcpdump is to view all traffic (on the network your sniffer is connected to) by entering the following at a shell prompt:

```
tcpdump -i eth0 -n -vvv
```

"eth0" is the interface of the sniffer, set yours accordingly. "-n" prohibits the sniffer from trying to resolve IP addresses, and "-vvv" gives the most verbose output.

To limit the output of tcpdump to just the traffic to and from the VPN gateway, add the gateway's external IP address as follows:

```
tcpdump -i eth0 -n -vvv 'host 172.26.60.3'
```

Of course, you must use the proper IP address for your system. The examples below show how to limit output to only IKE and IPsec traffic, but it may be useful to see other traffic to and from your gateway for certain situations. For example, when trying to determine network fragmentation issues, ICMP messages may be sent to your gateway, and you will only be able to see these messages if you do not limit the tcpdump output to IKE and IPsec traffic only. (There will be more on fragmentation later.)

In cases where you are troubleshooting IKE problems, use the following:

```
tcpdump -i eth0 -n -vvv -s 1514 'host 172.26.60.3 && udp port 500'
```

Here, "-s 1514" has been added to the command line. By default, tcpdump does not capture the entire IP datagram. With this setting, tcpdump will capture the entire IP datagram ensuring that all the information in each IKE message is displayed. Also, "&& udp port 500" further limits the tcpdump output to only UDP traffic on port 500 to and from the gateway. IKE uses UDP on port 500 as the transport protocol. Network Address Translation (NAT) will be discussed briefly later, but if there is a NAT device between the two IPsec gateways, NAT Traversal (NAT-T) may be employed by the gateways. In this case, the IKE traffic may be on another port such as UDP port 4500. If you suspect NAT is involved, start by viewing all traffic to and from the gateway to verify what ports are being used.

To limit the output to IPsec traffic only, enter:

```
tcpdump -i eth0 -n -vvv 'host 172.26.60.3 && ip proto 50'
```

This limits the output to Encapsulating Security Payload (ESP) traffic only. Use "ip proto 51" for Authentication Header (AH) traffic. Go to the ICSA Labs' VPN Product Technical Configuration Guidelines mentioned in the Documentation section for more information about ESP and AH.

By using the following expression for the tcpdump command, you can limit the output to view both IKE and IPsec traffic:

```
'host 172.26.60.3 && (udp port 500 || ip proto 50)'
```

In all these examples so far, tcpdump displays the traffic in real time. Traffic can be captured to a file and viewed later by tcpdump or Ethereal. For example, to capture all traffic to and from the gateway to a file called, "t1.cap", enter:

```
tcpdump -i eth0 -s 1514 -w t1.cap 'host 172.26.60.3'
```

This file can be loaded and viewed with Ethereal, or read with tcpdump using the "-r" switch:

```
tcpdump -r t1.cap -nvvv
```

You can use any of the expressions above to limit the output when using the "-r" switch.

Troubleshooting IKE/IPsec with tcpdump

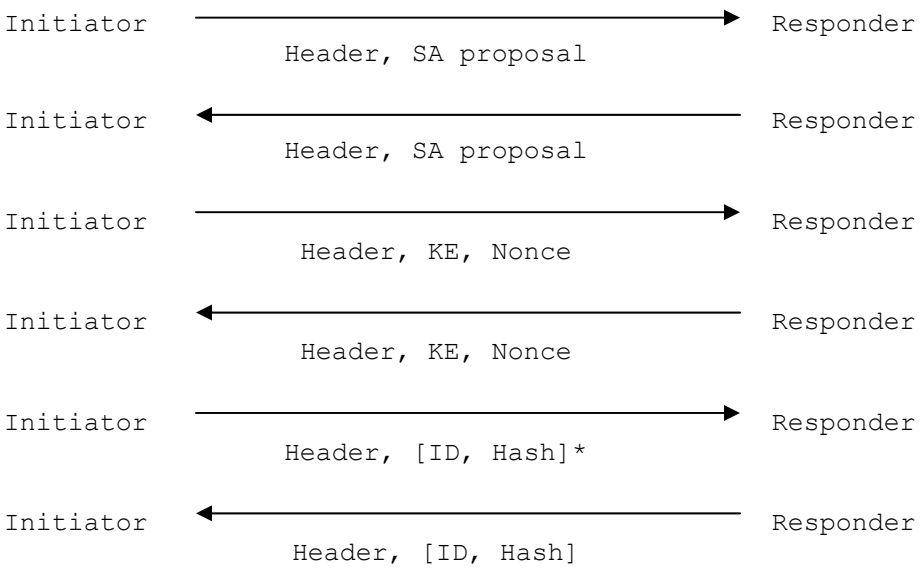
Before troubleshooting with a sniffer, it is important to know what a successful IKE exchange looks like. The following example shows a tcpdump output of communication between two IPsec gateways, A (172.26.60.3) and B (172.26.70.2). The gateways were configured with the settings listed in the "Where to Begin" section. In this example, Gateway A initiates IKE negotiation to Gateway B due to a host on A's private network sending traffic (a ping request) to a host on B's private network. The output shows a total of nine IKE messages exchanged between the gateways followed by the IPsec (ESP) traffic (4 ping request/response pairs).

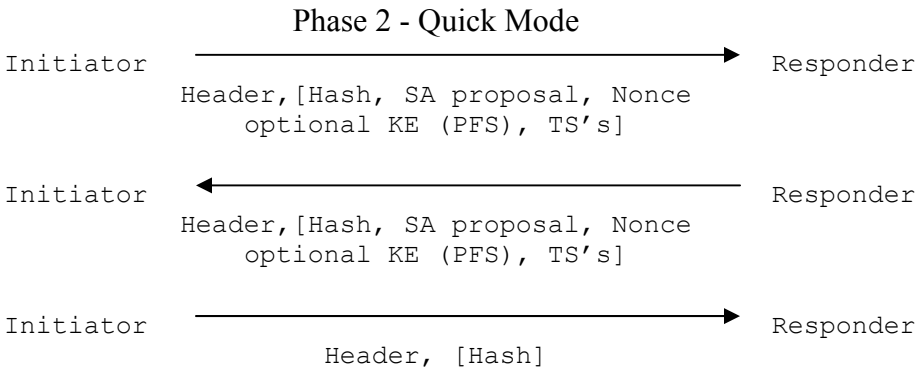
```
14:08:30.930276 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie a34dfc557f331e0d->0000000000000000: phase 1 I ident:
    (sa: doi=ipsec situation=identity
        (p: #1 protoid=isakmp transform=1
            (t: #0 id=ike (type=enc value=3des)(type=hash value=sha1)
            (type=auth value=preshared)(type=group desc value=modp1024)
            (type=lifetype value=sec)(type=lifeduration value=2328))))
 (ttl 64, id 33731, len 108)

14:08:30.943050 172.26.70.2.500 > 172.26.60.3.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie a34dfc557f331e0d->f972df49d26ac5d3: phase 1 R ident:
    (sa: doi=ipsec situation=identity
        (p: #1 protoid=isakmp transform=1
            (t: #1 id=ike (type=enc value=3des)(type=hash value=sha1)
            (type=group desc value=modp1024)(type=auth value=preshared)
            (type=lifetype value=sec)(type=lifeduration value=2328))))
 (ttl 63, id 1235, len 164)

14:08:31.040908 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie a34dfc557f331e0d->f972df49d26ac5d3: phase 1 I ident:
    (ke: key len=128
fc72e907edb31ad18520dc3162aec5f3d409d87d432a850a52841db503a6745fcfc086430905c8dff1059c962bae2d5b7
77287a0c821fe9765299ba43818b1dffa6e2143aade07178dcf9d51abdc68eed62d14fa75316bfa5e6adfe5d704e98f39
14cc4130b7771e1f5412101c03223cb1cb8799dd24a94be3dc5887e44e9f76)
    (nonce: n len=16 fc50497ec5ac2c79a949e19eba858641) (ttl 64, id 65169, len 208)

14:08:31.056529 172.26.70.2.500 > 172.26.60.3.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie a34dfc557f331e0d->f972df49d26ac5d3: phase 1 R ident:
    (ke: key len=128
658c09c873f7201ccb221e4abf39c0cb63e0b7d3ba732c40a03226d5ce942812473d29e6a079019913d74245e8372d125
6c8806e998e3460e39dbcb2a92e37225c2125379cba3a0fb4ccbe49c4cbf0d2954ae1022fa5f2155a1fa87cc1083f9d34
a79cbc29fad308b8c4848440f5e8dfab1d026b14db82bcfa1001b9902be6be)
    (nonce: n len=20 c89e3e4f8f8c6af1292632e675a292b383303438) (ttl 63, id 1236, len 212)

14:08:31.111684 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie a34dfc557f331e0d->f972df49d26ac5d3: phase 1 I ident[E]: [encrypted id]
 (ttl 64, id 59997, len 120)
```

```
14:08:31.126757 172.26.70.2.500 > 172.26.60.3.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie a34dfc557f331e0d->f972df49d26ac5d3: phase 1 R ident[E]: [encrypted id]
 (ttl 63, id 1237, len 96)

14:08:31.165969 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid ebf8b702
 cookie a34dfc557f331e0d->f972df49d26ac5d3: phase 2/others I oakley-quick[E]:
      [encrypted hash] (ttl 64, id 59331, len 320)

14:08:31.187398 172.26.70.2.500 > 172.26.60.3.500: [udp sum ok] isakmp 1.0 msgid ebf8b702
 cookie a34dfc557f331e0d->f972df49d26ac5d3: phase 2/others R oakley-quick[E]:
      [encrypted hash] (ttl 63, id 1238, len 360)

14:08:31.251738 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid ebf8b702
 cookie a34dfc557f331e0d->f972df49d26ac5d3: phase 2/others I oakley-quick[E]:
      [encrypted hash] (ttl 64, id 40694, len 80)

14:08:39.951498 172.26.60.3 > 172.26.70.2: ESP(spi=0xa6d66b86,seq=0x1) (DF)
 (ttl 64, id 52701, len 136)

14:08:39.962248 172.26.70.2 > 172.26.60.3: ESP(spi=0xf95344cf,seq=0x1)
 (ttl 63, id 1240, len 136)

14:08:40.972092 172.26.60.3 > 172.26.70.2: ESP(spi=0xa6d66b86,seq=0x2) (DF)
 (ttl 64, id 43100, len 136)

14:08:40.972336 172.26.70.2 > 172.26.60.3: ESP(spi=0xf95344cf,seq=0x2)
 (ttl 63, id 1241, len 136)

14:08:41.982002 172.26.60.3 > 172.26.70.2: ESP(spi=0xa6d66b86,seq=0x3) (DF)
 (ttl 64, id 34938, len 136)

14:08:41.982246 172.26.70.2 > 172.26.60.3: ESP(spi=0xf95344cf,seq=0x3,)
 (ttl 63, id 1242, len 136)

14:08:42.990236 172.26.60.3 > 172.26.70.2: ESP(spi=0xa6d66b86,seq=0x4,) (DF)
 (ttl 64, id 55203, len 136)

14:08:42.991322 172.26.70.2 > 172.26.60.3: ESP(spi=0xf95344cf,seq=0x4,)
 (ttl 63, id 1243, len 136)
```

The following diagram shows the IKE information exchanged in the example above. This diagram illustrates the typical payloads contained in an IKE exchange (Main Mode with PSKs).

Phase 1 - Main Mode





* [] denotes encrypted payloads

The first six packets are Phase 1 (Main Mode) messages which establish the IKE SA. All further IKE messages are protected by this IKE SA. The first two Phase 1 messages exchanged between the gateways are Gateway A's proposal for the IKE SA and Gateway B's acceptance of this proposal. If there is a mismatch for Phase 1 settings the problem will be evident at this point (except with a mismatched PSK which is described below).

```
14:28:20.032075 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie 0b47e3d3f56281ff->0000000000000000: phase 1 I ident:
    (sa: doi=ipsec situation=identity
        (p: #1 protoid=isakmp transform=1
            (t: #1 id=ike (type=enc value=3des)(type=hash value=sha1)
            (type=group desc value=modp1024)(type=auth value=preshared)
            (type=lifetype value=sec)(type=lifeduration value=0384))))
(ttl 64, id 1897, len 164)

14:28:20.046367 172.26.70.2.500 > 172.26.60.3.500: [udp sum ok] isakmp 1.0 msgid 1e607652
 cookie 0b47e3d3f56281ff->0000000000000000: phase 2/others R inf:
    (n: doi=0 proto=1 type=NO-PROPOSAL-CHOSEN) (DF) (ttl 254, id 3841, len 68)
```

In the example above, Gateway A's proposal was not acceptable to Gateway B. The cause could be any mismatched Phase 1 parameters, but in this case, it was a mismatched Diffie-Hellman Group setting (`type=group desc value=modp1024`).

Tip: To verify the Phase 1 mismatch using tcpdump, first initiate from one peer then the other. You can compare the two tcpdump outputs to see which parameters are different.

With the next two messages, message 3 and 4, the gateways exchange Diffie-Hellman public values and nonces (random data). It is very unusual to see negotiations stop at this point. If there are problems at this point, check ICSA Labs Lab-notes and the device's log information. Try initiating from both sides to verify that problem occurs in both directions. Verify the configurations as outlined in the "Initial Checks for Configuration Problems" section. If the problem persists, contact the vendor technical support center.

The final two Main Mode messages are the first messages to be encrypted, and therefore tcpdump cannot decode and display the payloads within these messages. With PSKs, the payloads are ID payloads containing the gateways' public IP address and hash values. The PSK

is one of the inputs the IPsec devices use to compute the hash value, so if there is a PSK mismatch a problem will be evident after the fifth Main Mode message as shown below.

```
14:32:52.374301 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie 6b7325612f3c0df1->0000000000000000: phase 1 I ident:
    (sa: doi=ipsec situation=identity
        (p: #1 protoid=isakmp transform=1
            (t: #1 id=ike (type=enc value=3des)(type=hash value=sha1)
            (type=group desc value=modp1024)(type=auth value=preshared)
            (type=lifetype value=sec)(type=lifeduration value=0384))))
 (ttl 64, id 1967, len 164)

14:32:52.378445 172.26.70.2.500 > 172.26.60.3.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie 6b7325612f3c0df1->c3aa7eab58898556: phase 1 R ident:
    (sa: doi=ipsec situation=identity
        (p: #1 protoid=isakmp transform=1
            (t: #1 id=ike (type=enc value=3des)(type=hash value=sha1)
            (type=group desc value=modp1024)(type=auth value=preshared)
            (type=lifetype value=sec)(type=lifeduration value=0384))))
 (DF) (ttl 254, id 3846, len 108)

14:32:52.383433 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie 6b7325612f3c0df1->c3aa7eab58898556: phase 1 I ident:
    (ke: key len=128
ad422c125e68121435a2368df0a4016f5c02ea19cd1ca7f958c3130e2cc45fa183c67edf196e7d71b58adb2cd46fc3d39
94f3e1018fd51384942fbbbcea4019df0540f8837101b9fd13cee19faf0f5d59192e7dbdf84ac8cd49a33a4bc81c8a3b2
2b34abec1b4f815d4aad760508ce742f19dc09299f1525a0dc5bc27b4f4602)
    (nonce: n len=20 7b93021e90d6aff641f29a39a8a3d049ee31fddc) (ttl 64, id 1968, len 212)

14:32:52.397147 172.26.70.2.500 > 172.26.60.3.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie 6b7325612f3c0df1->c3aa7eab58898556: phase 1 R ident:
    (ke: key len=128
ebb0614ef3cb0acb0c30efec70aac7b450d119de754ea4236960261e18679602b349b253455ff7b332abe24e2a921f4a3
ca94fa781299bfcc6bc81c6d2db89afcb35cea7c040f5466377dc3415d61ba46af51e750a7b4a845f6e0a680fcb4442e0
86489044db12b74c709d688fc151153780b2dd9ce6e9dea3f0a8b936608955)
    (nonce: n len=20 e7540513dbe3446adcc4e45b13f17bc26f182f88) (DF) (ttl 254, id 3847, len 212)

14:32:52.410196 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie 6b7325612f3c0df1->c3aa7eab58898556: phase 1 I ident[E]: [encrypted id] (ttl 64, id 1969,
len 96)

14:32:52.467292 172.26.70.2.500 > 172.26.60.3.500: [udp sum ok] isakmp 1.0 msgid df679e18
 cookie 6b7325612f3c0df1->c3aa7eab58898556: phase 2/others R inf:
    (n: doi=0 proto=1 type=PAYLOAD-MALFORMED) (DF) (ttl 254, id 3848, len 68)
```

In this case, there is a PAYLOAD-MALFORMED response from Gateway B after Main Mode message #5 from Gateway A. Some products may not respond at all. However, along with the gateways' logs and the tcpdump output, it should be apparent that this problem is due to a mismatched PSK. Note, it is not likely the problem is due to the ID payload because that would require that the local gateway's (Gateway A in this example) IP address was configured differently than it's actual address. That problem would become apparent during configuration, before attempting to establish a tunnel. Also, if Gateway B was configured with the wrong remote address (Gateway A's address), negotiations would not get as far as the fifth Main Mode message as in the example above. Communication would stop after the first Main Mode message. (Remember, these scenarios are based on using Main Mode and PSKs, with certificates it is possible to have an ID mismatch.)

After the six Phase 1 messages, a Phase 2 (Quick Mode) exchange occurs. These Quick Mode messages are also encrypted and tcpdump cannot decode and display the payloads within the messages. However, knowing the contents you can narrow down the possible causes at this point. The first message from Gateway A contains the proposal for the IPsec SAs, nonce,

optional Diffie-Hellman value if PFS is on, and local and remote ID payloads which contain the traffic selector information. If there is a mismatch in any of the Phase 2 settings, it will be evident in the tcpdump output after the first Quick Mode message.

```
14:40:20.190005 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie 5ef1eba502066c72->0000000000000000: phase 1 I ident:
    (sa: doi=ipsec situation=identity
        (p: #1 protoid=isakmp transform=1
            (t: #1 id=ike (type=enc value=3des)(type=hash value=sha1)
            (type=auth value=preshared)(type=group desc value=modp1024)
            (type=lifetype value=sec)(type=lifeduration len=4 value=00000384))))
 (DF) (ttl 254, id 3857, len 156)

14:40:20.195320 172.26.70.2.500 > 172.26.60.3.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie 5ef1eba502066c72->3137a7ee2c20903c: phase 1 R ident:
    (sa: doi=ipsec situation=identity
        (p: #1 protoid=isakmp transform=1
            (t: #1 id=ike (type=enc value=3des)(type=hash value=sha1)
            (type=group desc value=modp1024)(type=auth value=preshared)
            (type=lifetype value=sec)(type=lifeduration value=0384))))
 (ttl 64, id 2199, len 164)

14:40:20.210299 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie 5ef1eba502066c72->3137a7ee2c20903c: phase 1 I ident:
    (ke: key len=128
2dedee59877c0adeb6125e42e0b550d7cd8750d6344f06ac5bc1892d60179508778d4ea7077b21c53fd559946c5e0cb02
64a9c95c49f6609a43308cc40ac07756ca1aaa66ab21ca2e32c98fbd729cd3374bba35b2163aa5b105ac58471c45944df
65959d49477f635c7ee28d9bcc0d85ea97ce201d17afefc0a959bc97af3ef6)
    (nonce: n len=20 9a9ab48fdfeba86ec3207f18c7a43b592633935c) (DF) (ttl 254, id 3858, len 212)

14:40:20.222762 172.26.70.2.500 > 172.26.60.3.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie 5ef1eba502066c72->3137a7ee2c20903c: phase 1 R ident:
    (ke: key len=128
7a5753ec5f03e68ae92e613fac75752c4e0cbaa45bdea68309b6e78e09e7442d88bc2d6de6033e791359d47fad8f93abc
2bc5561a484c5e0cc1953567e8857ccbacacfe3c036872f03c3badb8ffb6956931687ae3f8d32bacf6095fe5a90b90c78
acc2e70d063aa6187df0489f7ada6bdcb358f7117db15281a1ceb89503a114)
    (nonce: n len=20 9c2c38780013598541aa22c0b6dc6f51e223dfec) (ttl 64, id 2200, len 212)

14:40:20.280129 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie 5ef1eba502066c72->3137a7ee2c20903c: phase 1 I ident[E]: [encrypted id] (DF) (ttl 254, id
3859, len 96)

14:40:20.285246 172.26.70.2.500 > 172.26.60.3.500: [udp sum ok] isakmp 1.0 msgid 00000000
 cookie 5ef1eba502066c72->3137a7ee2c20903c: phase 1 R ident[E]: [encrypted id] (ttl 64, id 2201,
len 96)

14:40:20.292677 172.26.60.3.500 > 172.26.70.2.500: [udp sum ok] isakmp 1.0 msgid ed4d654b
 cookie 5ef1eba502066c72->3137a7ee2c20903c: phase 2/others I oakley-quick[E]: [encrypted hash]
(DF) (ttl 254, id 3860, len 328)

14:40:20.302701 172.26.70.2.500 > 172.26.60.3.500: [udp sum ok] isakmp 1.0 msgid 247f7f75
 cookie 5ef1eba502066c72->3137a7ee2c20903c: phase 2/others R inf[E]: [encrypted hash] (ttl 64, id
2202, len 120)
```

In the example above, Gateway A is configured for PFS, but Gateway B is not. Gateway A sends the first Quick Mode message including an optional KE payload for PFS, and Gateway B rejects the proposal and sends an encrypted Informational message (`inf[E]`). (The message is an encrypted NO-PROPOSAL-CHOSEN message.) The tcpdump output above does not identify exactly which Phase 2 setting has been configured incorrectly, but with verbose log data it should be apparent which setting is wrong. If the log information does not help, it will be necessary to go over each Phase 2 setting on each gateway verifying the configuration.

A Note about Lifetimes

In all these Phase 1 and Phase 2 misconfiguration examples, mismatched lifetimes would only be a problem if a product has a restriction on what it will accept. For example, by default a product may have a maximum acceptable lifesize (lifetime based on volume of data) of 1 GB, and a peer may include in its proposal, by default, a lifesize of 2 GB. This would cause the responder to reject the sender's proposal. It is important to know if a product has limits on what it accepts, as well as the default values for both sides. However, based on experience in the lab, few products have these restrictions. Look for configuration notes in the ICSA Labs Lab-notes where this type of configuration was observed to be an issue. Otherwise, lifetimes do not need to match on each gateway. In fact, it is a requirement of the ICSA Labs IPsec Product Certification criteria that products inter-operate successfully with mismatched lifetimes. This allows either gateway to enforce its own lifetime policy for the life of the SAs.

IPsec Traffic

After IKE has completed successfully, there still may be other problems. IPsec traffic may not pass properly. In this situation, the tcpdump output will show no IPsec traffic or maybe traffic in only one direction. Try pinging from the private host on both sides to verify which direction, if any, the IPsec traffic flows. Assuming network connectivity has been verified, the cause of this is very likely to be filter rules on the gateways. Either inbound traffic, outbound traffic, or both are being blocked by the gateway. Check the logs and filter rule configurations. Refer to the vendor supplied documentation if necessary.

Sometimes it may be helpful to see the traffic within the tunnel. This may help to identify a filter rule problem. This traffic can be seen with tcpdump by configuring each side to use ESP-NULL for the Phase 2 encryption setting. Actually, with NULL there is no encryption so the inner traffic can be seen in the tcpdump capture. (Note, if using ESP-NULL as an alternative to AH, be sure to use authentication, e.g. SHA-1.)

Below is a capture of an IPsec packet from Gateway A to Gateway B using ESP-NULL. The inner traffic is an ICMP echo request from Gateway A's private host (172.26.63.1) to Gateway B's private host (172.26.72.1). Using the "-x" switch with tcpdump will display a hex dump of the data:

```
  tcpdump -i eth0 -n -vvv -s 1514 -x 'host 172.26.60.3'

14:45:03.560251 172.26.60.3 > 172.26.70.2: ESP(spi=0xf3f9b1d7,seq=0x32)
 (ttl 64, id 2297, len 128)
                        4500 0080 08f9 0000 4032 9717 ac1a 3c03
                        ac1a 4602 f3f9 b1d7 0000 0032 4500 0054
                        0000 4000 3f01 5a72 ac1a 3f01 ac1a 4801
                        0800 2c21 fc49 2f00 996a 633f b3e7 0500
                        0809 0a0b 0c0d 0e0f 1011 1213 1415 1617
                        1819 1a1b 1c1d 1e1f 2021 2223 2425 2627
                        2829 2a2b 2c2d 2e2f 3031 3233 3435 3637
                        0102 0204 ae99 ccbd cd23 bf4f 511e f4a9
```

The hex dump can be broken down as follows:

The outer IP header:

```
4500 0080 08f9 0000 4032 9717 ac1a 3c03
 ac1a 4602
```

ESP header information (SPI and Sequence #):

```
f3f9 b1d7 0000 0032
```

The inner IP datagram containing ICMP echo request:

```
                                4500 0054
0000 4000 3f01 5a72 ac1a 3f01 ac1a 4801
0800 2c21 fc49 2f00 996a 633f b3e7 0500
0809 0a0b 0c0d 0e0f 1011 1213 1415 1617
1819 1a1b 1c1d 1e1f 2021 2223 2425 2627
2829 2a2b 2c2d 2e2f 3031 3233 3435 3637
```

ESP trailer information (padding, pad length, next header, authentication data):

```
0102 0204 ae99 ccbd cd23 bf4f 511e f4a9
```

In the inner IP datagram, the IP addresses of private host A (ac1a 3f01 = 172.26.63.1) and private host B (ac1a 4801 = 172.26.72.1) can be seen. This can help determine if any address translation has occurred on either private network which sometimes can cause problems.

tcpdump and Ethereal are invaluable tools for troubleshooting IKE/IPsec as well as other network problems. Always have a laptop handy with these tools installed, and keep the tools current as the developers are continually adding functionality.

Using OpenBSD to View Encrypted IKE Messages

Unfortunately, there are times when a problem occurs during the exchange of encrypted messages limiting the usefulness of the tcpdump capture. And if the logs do not help, where can you go next? For analysts at ICSA Labs, there is always another product available that has verbose debugging information. These products can be set up to inter-operate with the product suspected of having a problem, and the analyst can better determine what is wrong. However, in the field it is not possible to pick another product with better logging to use for troubleshooting, but there is another option. This is where a readily available implementation like OpenBSD can be very beneficial. Although OpenBSD has not been certified by ICSA Labs, it is a very useful troubleshooting tool.

OpenBSD is an operating system that can run on PC hardware and has support for IKE and IPsec. It can be configured to inter-operate with other IPsec implementations, and it has the capability to save all the IKE messages, unencrypted, in the same file format that tcpdump uses (pcap). Therefore, tcpdump or Ethereal can be used to read the file and see everything in each IKE message, including the encrypted messages. This is possible because, as one of the IKE peers, OpenBSD must decrypt any encrypted messages. It simply writes the information within these messages to a file in the proper format, similar to writing hex output to a log file. Below is an example of an IKE exchange showing all payloads in all messages. The capture file can be loaded into Ethereal or read with tcpdump using the "-r" switch as described in the previous section. Compare this with the first tcpdump example above.

```
14:08:30.766680 172.26.60.3.500 > 172.26.70.2.500:  [udp sum ok] isakmp v1.0 exchange ID_PROT
        cookie: a34dfc557f331e0d->0000000000000000 msgid: 00000000 len: 80
        payload: SA len: 52 DOI: 1(IPSEC) situation: IDENTITY_ONLY
            payload: PROPOSAL len: 40 proposal: 1 proto: ISAKMP spisz: 0 xforms: 1
                payload: TRANSFORM len: 32
                    transform: 0 ID: ISAKMP
                        attribute ENCRYPTION_ALGORITHM = 3DES_CBC
                        attribute HASH_ALGORITHM = SHA
                        attribute AUTHENTICATION_METHOD = PRE_SHARED
                        attribute GROUP_DESCRIPTION = MODP_1024
                        attribute LIFE_TYPE = SECONDS
                        attribute LIFE_DURATION = 9000 [ttl 0] (id 1)

14:08:30.790230 172.26.70.2.500  > 172.26.60.3.500:  [udp sum ok] isakmp v1.0 exchange ID_PROT
        cookie: a34dfc557f331e0d->f972df49d26ac5d3 msgid: 00000000 len: 136
        payload: SA len: 52 DOI: 1(IPSEC) situation: IDENTITY_ONLY
            payload: PROPOSAL len: 40 proposal: 1 proto: ISAKMP spisz: 0 xforms: 1
                payload: TRANSFORM len: 32
                    transform: 1 ID: ISAKMP
                        attribute ENCRYPTION_ALGORITHM = 3DES_CBC
                        attribute HASH_ALGORITHM = SHA
                        attribute GROUP_DESCRIPTION = MODP_1024
                        attribute AUTHENTICATION_METHOD = PRE_SHARED
                        attribute LIFE_TYPE = SECONDS
                        attribute LIFE_DURATION = 9000 [ttl 0] (id 1)

14:08:30.884284 172.26.60.3.500 > 172.26.70.2.500:  [udp sum ok] isakmp v1.0 exchange ID_PROT
        cookie: a34dfc557f331e0d->f972df49d26ac5d3 msgid: 00000000 len: 180
        payload: KEY_EXCH len: 132
        payload: NONCE len: 20 [ttl 0] (id 1)

14:08:30.900196 172.26.70.2.500  > 172.26.60.3.500:  [udp sum ok] isakmp v1.0 exchange ID_PROT
        cookie: a34dfc557f331e0d->f972df49d26ac5d3 msgid: 00000000 len: 184
        payload: KEY_EXCH len: 132
        payload: NONCE len: 24 [ttl 0] (id 1)

14:08:30.957178 172.26.60.3.500 > 172.26.70.2.500:  [udp sum ok] isakmp v1.0 exchange ID_PROT
        cookie: a34dfc557f331e0d->f972df49d26ac5d3 msgid: 00000000 len: 92
        payload: ID len: 12 type: IPV4_ADDR = 172.26.60.3
        payload: HASH len: 24
        payload: NOTIFICATION len: 28
            notification: INITIAL CONTACT (a34dfc557f331e0d->f972df49d26ac5d3) [ttl 0] (id 1)

14:08:30.965573 172.26.70.2.500  > 172.26.60.3.500:  [udp sum ok] isakmp v1.0 exchange ID_PROT
        cookie: a34dfc557f331e0d->f972df49d26ac5d3 msgid: 00000000 len: 68
        payload: ID len: 12 type: IPV4_ADDR = 172.26.70.2
        payload: HASH len: 24 [ttl 0] (id 1)

14:08:31.006459 172.26.60.3.500 > 172.26.70.2.500:  [udp sum ok] isakmp v1.0 exchange QUICK_MODE
        cookie: a34dfc557f331e0d->f972df49d26ac5d3 msgid: ebf8b702 len: 288
        payload: HASH len: 24
        payload: SA len: 52 DOI: 1(IPSEC) situation: IDENTITY_ONLY
        payload: PROPOSAL len: 40 proposal: 1 proto: IPSEC_ESP spisz: 4 xforms: 1 SPI: 0xf95344cf
                payload: TRANSFORM len: 28
                    transform: 1 ID: 3DES
                        attribute LIFE_TYPE = SECONDS
                        attribute LIFE_DURATION = 4200
                        attribute ENCAPSULATION_MODE = TUNNEL
                        attribute AUTHENTICATION_ALGORITHM = HMAC_SHA
                        attribute GROUP_DESCRIPTION = 2
        payload: NONCE len: 20
        payload: KEY_EXCH len: 132
        payload: ID len: 16 type: IPV4_ADDR_SUBNET = 172.26.63.0/255.255.255.0
        payload: ID len: 16 type: IPV4_ADDR_SUBNET = 172.26.72.0/255.255.255.0 [ttl 0] (id 1)

14:08:31.026918 172.26.70.2.500  > 172.26.60.3.500:  [udp sum ok] isakmp v1.0 exchange QUICK_MODE
        cookie: a34dfc557f331e0d->f972df49d26ac5d3 msgid: ebf8b702 len: 332
        payload: HASH len: 24
        payload: SA len: 56 DOI: 1(IPSEC) situation: IDENTITY_ONLY
        payload: PROPOSAL len: 44 proposal: 1 proto: IPSEC_ESP spisz: 4 xforms: 1 SPI: 0xa6d66b86
                payload: TRANSFORM len: 32
```

```
                transform: 1 ID: 3DES
                    attribute LIFE_TYPE = SECONDS
                    attribute LIFE_DURATION = 00001068
                    attribute ENCAPSULATION_MODE = TUNNEL
                    attribute AUTHENTICATION_ALGORITHM = HMAC_SHA
                    attribute GROUP_DESCRIPTION = 2
        payload: NONCE len: 24
        payload: KEY_EXCH len: 132
        payload: ID len: 16 type: IPV4_ADDR_SUBNET = 172.26.63.0/255.255.255.0
        payload: ID len: 16 type: IPV4_ADDR_SUBNET = 172.26.72.0/255.255.255.0
        payload: NOTIFICATION len: 28
            notification: RESPONDER LIFETIMESPI: 0xa6d66b86
                attribute LIFE_TYPE = SECONDS
                attribute LIFE_DURATION = 000001ffffffa4 [ttl 0] (id 1)

14:08:31.090287 172.26.60.3.500 > 172.26.70.2.500:  [udp sum ok] isakmp v1.0 exchange QUICK_MODE
        cookie: a34dfc557f331e0d->f972df49d26ac5d3 msgid: ebf8b702 len: 52
        payload: HASH len: 24 [ttl 0] (id 1)
```

Using OpenBSD for the first time, as with any O.S., can be daunting; however, it is a very useful operating system for more than what is described here. OpenBSD is available at www.openbsd.org with a very thorough FAQ section as well as access to MAN pages, the Unix on-line manual pages. This paper will show the IPsec configuration that was used to create these examples. OpenBSD 3.3 was installed on an 800MHz P-III PC with 128MB RAM and a 2GB hard drive. The system was set up as a gateway with two NICs, one (172.26.63.254/24) connected to the private network and the other (172.26.60.3/24) to the public network. The private interface was the default gateway for the private hosts. For this example, the peer's public IP address was 172.26.70.2 and the peer's private network was 172.26.72.0/24.

OpenBSD IPsec Configuration

The steps taken to enable and configure the IKE/IPsec functionality on the system described above are outlined here. After the system was installed and networking was operational, the next steps were:

1. Edit /etc/sysctl.conf and set the following parameters:
   ```
   net.inet.ip.forwarding=1
   net.inet.esp.enable=1
   ```
2. Edit the two configuration files used by the ISAKMPD daemon (/sbin/isakmpd), /etc/isakmpd/isakmpd.conf and /etc/isakmpd/isakmpd.policy. See the sample files at the end of this document.

Capturing IKE messages

After editing the files, the system was restarted, and the following command was used to start the IKE daemon and begin the IKE negotiations with the IPsec peer.

```
isakmpd -d -l t2.cap
```

The "-l t2.cap" switch instructs the daemon to capture all IKE messages and write to the file "t2.cap". The "-d" causes the daemon to run in the foreground. After stopping the daemon, the file "t2.cap" was viewed with the version of tcpdump included with the OpenBSD installation.

As mentioned in the beginning of this section, OpenBSD is a very useful operating system. A troubleshooting aid for IKE and IPsec is just one helpful feature in addition to others, including

routing, firewalling, NAT, and bridge configuration. OpenBSD, as well as other BSD and Linux operating systems, are versatile and well worth spending the time getting very familiar with them.

Other Potential Problems

IKE and IPsec protocols are very complex and there is much that can affect interoperability. Even when two IPsec devices can inter-operate successfully, other external factors can impede communications between the devices. In this section, we will discuss briefly some of those external factors to be aware of when deploying IPsec products. We will also talk about some things to consider when using certificates instead of PSKs, and situations where one of the IPsec peers has lost SA state information, e.g. power is recycled.

Fragmentation
IP fragmentation has been an issue with some products when they were tested by ICSA Labs. However, it is a certification requirement that IPsec products successfully operate in an environment where there is an intermediate hop between the IPsec peers that has an MTU of 576. In cases where products could not successfully send or receive IPsec traffic in a test network that introduced fragmentation, a fix was required. To handle fragmentation, IPsec products usually either clear the DF bit in the outer IP header allowing intermediate routers to fragment, or the IPsec products do MTU path discovery to avoid fragmentation. On the receiving end, the IPsec product must successfully reassemble fragmented IPsec traffic. Fragmentation as described here should not be an issue in the field with ICSA Labs certified products. However, an administrator should be aware that this is a potential problem for some non-certified products or non-certified versions of products.

A typical symptom related to fragmentation would be indicated by a ping working fine through the tunnel but not other traffic. For example, a user may be able to ping or establish an ftp connection, but when trying to send or receive data via ftp, it does not work. This would occur because the ping or ftp connection setup is small enough packets to go through without a problem. But, the larger packets, i.e. ftp data transfer, are susceptible to fragmentation. tcpdump can be used to help determine if this is a problem. Pay attention to the DF bit in the IP header of the IPsec packet, and look for fragmented packets and ICMP Fragmentation Required messages at each gateway. With this information, you should be able to determine if one of the peers is having a problem with transmission or reassembly in a fragmenting environment.

Firewalls
Firewall or filter rules have already been mentioned as potential problems on the IPsec devices themselves. But, external devices such as firewalls, especially those that do NAT, can cause problems. Initial testing should be done with the simplest configuration possible. Of course, that would be with no firewall/NAT device between the IPsec products. But when there is, an administrator needs to be aware of the potential problems.

Firewalls between the IPsec devices must allow the IKE and IPsec traffic to pass. They must allow IP protocol 50 (ESP) or 51 (AH) as well as UDP port 500 for IKE traffic to traverse the

firewall. Other UDP ports such as 4500 may be used in the case where the IPsec devices are trying to do NAT detection and traversal. Many firewall devices refer to allowing IKE/IPsec traffic through as "IPsec pass-through". This does not mean that the device can be an IPsec peer, but that it can allow that traffic through. A problem involving a firewall/NAT device would be evident by IKE/IPsec traffic not getting to one peer or the other. This should be easily determined by monitoring the traffic at each end. Also, with products that are capable of logging successfully passed traffic, the log information can be a good indicator of whether traffic is reaching it or not.

NAT

NAT devices can cause many different problems with IKE/IPsec communication. An excellent description of the different NAT related issues is provided in an Internet Draft entitled "IPsec-NAT Compatibility Requirements." The section most useful to an IPsec VPN administrator is entitled "Known Incompatibilities between NA(P)T and IPsec." The current version, at the time of this writing, is available at www.ietf.org/internet-drafts/draft-ietf-ipsec-nat-reqts-05.txt. Attain a copy of this draft if you suspect NAT is affecting your VPN.

Certificates

With authentication using certificates, configuration becomes somewhat more difficult, especially when using a third-party Certification Authority (CA) product. First, certificates must be enrolled into each IPsec device, and this process varies with each vendor. Some products support protocols which communicate with the CA for enrollment, but most all support a manual method where a request is generated and output to a file. The CA administrator can then take the file and generate a certificate that is manually installed into the IPsec device.

If possible, keep the Public Key Infrastructure (PKI), the infrastructure which supports certificate management, as simple as possible. A simple, single root-CA configuration should be relatively easy to work with. In this configuration, making sure the information for generating requests is properly input by the administrator, is the primary concern. Carefully, enter the appropriate information for the Distinguished Name (DN), contained in the Subject Name field of the certificate, as well as other identifying information, i.e. IP address, Fully Qualified Domain Name (FQDN) or an e-mail address (USER_FQDN). This additional information is contained in the Subject Alternative Name field of the certificate.

With this added identifying information there are more options for the Phase 1, Main Mode identity (ID) setting of the IPsec peers. With PSKs, IP address is the only setting for identity for the local or remote gateway; however, other options are available with certificates. If an administrator included all identifying information described above in the certificate, the ID can be configured to be any one of these, DN, IP address, FQDN, or USER_FQDN. Make sure in the configurations of each peer, that what is sent matches what is expected to be received. (You do not have to send and receive the same ID on the same device. For example, you could send IP address and receive DN as long as the peer is configured accordingly.) An ID misconfiguration will cause a problem after Main Mode message #5. Look for an entry in the receivers log indicating an ID mismatch and double check the ID information entered on each peer.

CRLs

Another problem area with the use of certificates is configuration of Certificate Revocation List (CRL) checking. Products may have the option to turn CRL checking on or off. First, try establishing the tunnel with CRL checking disabled. If successful, then enable it to see if there are any problems. Generally, the problems associated with CRL checking are related to improper configuration for the device to retrieve the CRL. CRLs are typically retrieved via LDAP or HTTP, although there are other options for CRL retrieval as well. (It's an ICSA Labs IPsec Product Certification version 1.1 requirement that the product supports either LDAP or HTTP.) Proper configuration for CRL checking will require input from the CA administrator for configuration information, e.g. server IP address, protocol and port.

Typically, if there is a problem with CRL checking, a log entry or error will be generated when the product tries to retrieve the CRL. CRL retrieval occurs after saving and applying the configuration, on power up, during IKE negotiations, or some combination of these. If it occurs during IKE negotiations, it most likely will happen during Main Mode after the fifth or sixth message. To see this traffic, use tcpdump on the appropriate interface and look for the appropriate traffic, i.e. TCP port 389 for LDAP or TCP port 80 for HTTP. As suggested above, initiating IKE negotiations with both CRL checking on and off should determine if there is a CRL issue. Keep in mind though, the problem could actually be that the peer's certificate has been revoked and appears on the CRL. Verify this possibility with the CA administrator. Do not be tempted to simply turn off CRL checking permanently. It is an important part of a secure VPN implementation, and you should take the time to get it working properly.

AR and DPD

Another interoperability issue that arises with IPsec products happens when one of the peers loses track of what tunnels are established. For example, after the tunnel has been established one side loses power for some time and then reboots. The other peer is not aware of what has happened and continues to use the established SAs. In this situation, communication could be lost for a period of time, in some cases, a very long period of time. The IPsec product vendors have taken different approaches to deal with this situation. Many times these approaches are proprietary and will work only when both ends are products from the same vendor. However, there are products that provide a solution that will work in multi-vendor environments.

There really are two considerations in this scenario. One is that the peer that did not lose state, the persistent peer, is wasting resources attempting to maintain a tunnel with a peer that may be down or has lost any awareness of the SAs. The persistent peer may continually send traffic to the peer but get no response. If the persistent peer had a mechanism to detect a dead peer, then it could free up resources, and possibly attempt to re-establish a tunnel later. There is an IETF Internet Draft for such a mechanism, and the mechanism is called Dead Peer Detection (DPD). At the time of this writing, the draft is in "last call" to become an Informational RFC and is available at www.ietf.org/internet-drafts/draft-ietf-ipsec-dpd-03.txt.

DPD requires that both products support the protocol in accordance with the draft. Each peer notifies the other of support for DPD during Main Mode with the use of Vendor ID payloads. The interoperability issue with DPD is whether both products support it. Knowing whether a product uses DPD will require doing some research ahead of time. Check the vendor’s technical

specifications for the product as well as checking ICSA Labs Lab-notes. For products that support DPD, it will be tested and reported in the Lab-notes.

The other consideration with the scenario described in the beginning of this section is the behavior of the peer that recovers. Some products may take action on recovery to re-establish SAs, others will not. For those that take action, there are two basic ways of re-establishing SAs automatically after recovery (referred to as Automatic Recovery or AR). One approach is the recovering peer will immediately initiate IKE negotiations on power-up for every configured peer or for only those peers for which an administrator has explicitly configured the functionality. This behavior will re-establish communications without manual intervention from an administrator. Another approach is not to initiate IKE on power-up but initiate only after receiving messages from the persistent peer. There are some variations of these AR methods, and there is currently no draft available that specifies AR. If a vendor claims that his product supports AR, ICSA Labs will test the functionality and report it in the Lab-notes.

There may be some interoperability issues with AR. These issues will typically cause problems during IKE negotiations, Main Mode or Quick Mode. The best source of information for AR interoperability will be in the ICSA Labs Lab-notes. Determining these problems can be accomplished using tcpdump to troubleshoot the IKE negotiations on recovery. Use the same techniques that were outlined in the "Troubleshooting IKE/IPsec with tcpdump" section.

OpenBSD IPsec Configuration Files

/etc/isakmpd/isakmpd.conf

```
# A configuration sample for the isakmpd ISAKMP/Oakley (aka IKE) daemon.

[General]
Default-phase-1-lifetime=     9000,60:86400
Default-phase-2-lifetime=     4200,60:86400
Retransmits=                  5
Exchange-max-time=            120
Listen-on=                    172.26.60.3

[Phase 1]
172.26.70.2=                  ike-peer-ip

[Phase 2]
Connections=                  ipsec-peer-ip
#Passive-connections=         ipsec-peer-ip

[ike-peer-ip]
Phase=                        1
Transport=                    udp
Local-address=                172.26.60.3
Address=                      172.26.70.2
Configuration=                Default-main-mode
Authentication=               presharedkey

[ipsec-peer-ip]
Phase=                        2
ISAKMP-peer=                  ike-peer-ip
Configuration=                Default-quick-mode
Local-ID=                     net-local
Remote-ID=                    net-peer-ip

[net-peer-ip]
ID-type=                      IPV4_ADDR_SUBNET
Network=                      172.26.72.0
Netmask=                      255.255.255.0

[net-local]
ID-type=                      IPV4_ADDR_SUBNET
Network=                      172.26.63.0
Netmask=                      255.255.255.0

[Default-main-mode]
DOI=                          IPSEC
EXCHANGE_TYPE=                ID_PROT
Transforms=                   3DES-SHA

[Default-quick-mode]
DOI=                          IPSEC
EXCHANGE_TYPE=                QUICK_MODE
Suites=                       QM-ESP-3DES-SHA-PFS-SUITE
```

/etc/isakmpd/isakmpd.policy

```
KeyNote-Version: 2
Comment: This policy accepts ESP SAs from a remote that uses the right password
Authorizer: "POLICY"
Licensees: "passphrase:presharedkey"
Conditions: app_domain == "IPsec policy" &&
            esp_present == "yes" &&
             esp_enc_alg == "3des" &&
             esp_auth_alg == "hmac-sha" -> "true";
```

Acknowledgement

I would like to acknowledge the assistance provided in writing and editing this paper:
Steve Ratcliffe and Pano Ftohidis – Testing Engineers within the IPSec testing and certification program
Mark Zimmerman – IPSec and Cryptography Program Manager
Leo Pluswick – Wireless LAN Program Manager and the person who originally launched the IPSec Program in 1998
ICSA IPSec Consortium and vendor participants who took the time to read and edit the paper

Contact ICSA Labs

It is obvious that there are many influences that affect interoperability of IPsec products, some are within the products themselves and others are external influences. But, armed with the information given here as well as the other sources of information mentioned in the Documentation section, the VPN administrator should be able to resolve most of these IPsec interoperability issues. For questions or comments about this paper contact Darren Hartman at dhartman@icsalabs.com. For more information regarding the ICSA Labs IPsec Product Certification Program or IPsec Product Developers Consortium, visit www.icsalabs.com or e-mail the IPsec Program Manager, Mark Zimmerman, at mzimmerman@icsalabs.com.

About ICSA Labs

ICSA Labs, an independent division of TruSecure Corporation, offers vendor-agnostic testing and certification of security products. Hundreds of the world's top security vendors submit their products for testing and certification at ICSA Labs. The end-users of security technologies rely on ICSA Labs to authoritatively set and apply objective testing and certification criteria for measuring product compliance and reliability. The organization tests products in key technology categories such as anti-virus, firewall, IPSec VPN, cryptography, intrusion detection, PC firewall and content security.

About TruSecure Corporation

TruSecure is the leading provider of intelligent risk management products and services. TruSecure dramatically improves security and reduces risk by helping organizations make better security decisions and maximize the effectiveness of existing security people, processes, and products. Leveraging TruSecure's vast security knowledge and intelligence gathering resources—including ICSA Labs, the global leader in information security product certification—as well as innovative technology and time-tested processes, our customers can *predict* which vulnerabilities present real risk, *prioritize* remediation efforts, quickly *adapt* to changes in the security threatscape, *measure* progress in improving their security posture, and *document* compliance with applicable security policies, standards and regulations.