

[NEWS] Multiple IPSEC Implementations Do Not Adequately Validate Authentication Data (DoS)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-10/0117.html>

From: support@securiteam.com

Date: 10/24/02

From: support@securiteam.com

To: list@securiteam.com

Date: 24 Oct 2002 02:52:40 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

Multiple IPSEC Implementations Do Not Adequately Validate Authentication Data (DoS)

SUMMARY

IPSEC implementations from multiple vendors do not adequately validate the authentication data in IPSEC packets, exposing vulnerable systems to a denial of service.

DETAILS

Systems Affected:

(For an up to date version see:

<<http://www.kb.cert.org/vuls/id/459371#systems>>

<http://www.kb.cert.org/vuls/id/459371#systems>)

Vulnerable systems:

- * Apple Computer Inc.
- * eSoft
- * FreeBSD
- * FreeS/WAN
- * Global Technology Associates
- * Internet Initiative Japan
- * KAME Project
- * NEC Corporation
- * NetBSD

Securiteam: [NEWS] Multiple IPSEC Implementations Do Not Adequately Validate Authentication Data (DoS)

Immune systems:

- * Lucent
- * Intoto
- * Borderware
- * Alcatel
- * Cisco Systems Inc.
- * Clavister
- * Hewlett–Packard Company
- * Microsoft Corporation
- * Montavista
- * Hitachi
- * Cray
- * NetScreen
- * Network Appliance
- * Openwall
- * SafeNet
- * Sun Microsystems Inc.

Background:

* <<http://www.ietf.org/rfc/rfc2401.txt>> RFC 2401 Security Architecture for the Internet Protocol

* <<http://www.ietf.org/rfc/rfc2402.txt>> RFC 2402 IP Authentication Header

* <<http://www.ietf.org/rfc/rfc2406.txt>> RFC 2406 IP Encapsulating Security Payload

IPSEC supports integrity and authentication for IP traffic by including a cryptographic checksum in each IPSEC datagram. This authentication data is compared to the Integrity Check Value (ICV) that is calculated by the recipient. If the values match, the datagram is considered valid.

Technical details:

IPSEC is a set of security extensions to IP that provides authentication and encryption. It includes specification for two types of packets, ESP and AH, using IP protocols 50 and 51, respectively. This advisory is based only on ESP on IPv4, but the same issues are likely be present in the AH and/or IPv6 cases, and implementers are advised to check those, when applicable.

ESP (Encapsulating Security Payload) is the IPSEC protocol which provides packet level encryption. In most cases, it also includes some level of authentication. In this scenario, ESP packets look like this:

```
+-----+
| IP header (proto 50) |
+-----+
| SPI | 4 bytes
+-----+
| Sequence number | 4 bytes
+-----+
| ...data... | M bytes of payload
+-----+
```

| Auth data | N bytes, depending on algorithm
+-----+

SPI is a security parameter index, which, along with the IP destination address, identifies the SA (security association) to which this packet belongs. SAs (and their SPIs) are either created manually or dynamically through key negotiation with IKE/ISAKMP. The sequence number is a monotonically increasing number used to prevent replays. The size of the Auth data depends on the particular authentication algorithm used by the SA, but is usually at least 12 bytes.

When such a packet arrives, the IPSEC implementation looks up the relevant SA based on the SPI, checks the sequence number for replay, and then attempts to verify the authentication data is correct.

There is a common error in IPSEC implementations in that they either have no checks that such auth data is actually present, or that the checks are insufficient and/or incorrect. As a result, spoofing very short ESP packets with known source, destination, SPI and high sequence number (only 8 bytes of IP payload) can cause a kernel panic.

For illustration, here is the relevant code from the FreeBSD 4.6 implementation, and why it is wrong.

esp_input.c: esp4_input ()

Various code segments looking up the SA based on SPI and replay checking. Then, starting on line 219:

```
{
    u_char sum0[AH_MAXSUMSIZE];
    u_char sum[AH_MAXSUMSIZE];
    const struct ah_algorithm *sumalgo;
    size_t siz;
    figure out how big the signature should be (siz)

    sumalgo = ah_algorithm_lookup(sav->alg_auth);
    if (!sumalgo)
        goto noreplaycheck;
    siz = (((*sumalgo->sumsiz)(sav) + 3) & ~(4 - 1));
    if (AH_MAXSUMSIZE < siz) {
        ipseclog((LOG_DEBUG,
            "internal error: AH_MAXSUMSIZE must be larger than
%lu\n",
            (u_long)siz));
        ipsecstat.in_inval++;
        goto bad;
    }
}
```

Next, it simply copies the number of bytes the signature should be from the end of the packet, without checking that such data exists. This is not

Securiteam: [NEWS] Multiple IPSEC Implementations Do Not Adequately Validate Authentication Data (DoS)

actually fatal because if the signature isn't present, it will simply copy part of the IP header and then the authentication should fail later.

```
m_copydata(m, m->m_pkthdr.len - siz, siz, &sum0[0]);
```

Next it tries to verify the signature. 'off' was calculated earlier and is the offset into the packet that the ESP data starts at. Assuming a 20 byte IP header and ignoring any link-layer data, off will be 20. If we sent an esp packet with just an SPI and sequence number, m_pkthdr.len will be 28. If the authentication data is supposed to be 96 bits, size will be 12. Then, the calculation m->m_pkthdr.len - off - siz should be -4. However, 'siz' is of type size_t, which is unsigned, so (assuming 32bit ints) it's actually 4294967292 which is passed to esp_auth

```
if (esp_auth(m, off, m->m_pkthdr.len - off - siz, sav, sum)) {
    ipseclog((LOG_WARNING, "auth fail in IPv4 ESP input: %s
%s\n",
            ipsec4_logpacketstr(ip, spi), ipsec_logsastr(sav)));
    ipsecstat.in_espauthfail++;
    goto bad;
}
```

In esp_core.c:

```
int
esp_auth(m0, skip, length, sav, sum)
    struct mbuf *m0;
    size_t skip; /* offset to ESP header */
    size_t length; /* payload length */
    struct secasvar *sav;
    u_char *sum;
```

'length' is also of type size_t, so it's still 4294967292. There are some sanity checks, but they fail to catch this.

```
/* sanity checks */
if (m0->m_pkthdr.len < skip) {
    ipseclog((LOG_DEBUG, "esp_auth: mbuf length < skip\n"));
    return EINVAL;
}
if (m0->m_pkthdr.len < skip + length) {
    ipseclog((LOG_DEBUG,
            "esp_auth: mbuf length < skip + length\n"));
    return EINVAL;
}
```

The second sanity check 'should' catch this, but doesn't, because the 'skip + length' expression is again unsigned, so we add 20 to 4294967292 and get 16. Subsequently, the code tries to checksum 'length' bytes of memory and causes a kernel panic.

Much the same thing happens with the Free/SWAN code. There are other suspicious things in both code bases, as well. Neither implementation

Securiteam: [NEWS] Multiple IPSEC Implementations Do Not Adequately Validate Authentication Data (DoS)

checks that the ESP packet isn't totally empty before grabbing the SPI from it, e.g.

Workarounds:

If your firewall is capable of filtering based on packet payload length that may be an effective workaround.

Recommendations:

Install the appropriate patch from your vendor.

ADDITIONAL INFORMATION

For the original BindView RAZOR advisory go to:

<http://razor.bindview.com/publish/advisories/adv_ipsec.html>
http://razor.bindview.com/publish/advisories/adv_ipsec.html

For the original CERT advisory go to:

<<http://www.kb.cert.org/vuls/id/459371>>
<http://www.kb.cert.org/vuls/id/459371>

The information has been provided by <<mailto:tsabin@razor.bindview.com>>
Todd Sabin of BindView RAZOR and CERT.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

-
- **Previous message:** support@securiteam.com: "[NT] IBM WebSphere Edge Server Caching Proxy Cross-Site Scripting Issues"
 - **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)