Network Working Group
Request for Comments: 5062
Category: Informational

R. Stewart
Cisco Systems, Inc.
M. Tuexen
Muenster Univ. of Applied Sciences
G. Camarillo
Ericsson

September 2007

Security Attacks Found Against the Stream Control Transmission Protocol (SCTP) and Current Countermeasures

Status of This Memo

Abstract

This document describes certain security threats to SCTP. It also describes ways to mitigate these threats, in particular by using techniques from the SCTP Specification Errata and Issues memo (RFC 4460). These techniques are included in RFC 4960, which obsoletes RFC 2960. It is hoped that this information will provide some useful background information for many of the newest requirements spelled out in the SCTP Specification Errata and Issues and included in RFC 4960.

Table of Contents

1. Introduction

Stream Control Transmission Protocol, originally defined in [RFC2960], is a multi-homed transport protocol. As such, unique security threats exists that are addressed in various ways within the protocol itself. This document describes certain security threats to SCTP. It also describes ways to mitigate these threats, in particular by using techniques from the SCTP Specification Errata and Issues memo [RFC4460]. These techniques are included in [RFC4960], which obsoletes [RFC2960]. It is hoped that this information will provide some useful background information for many of the newest requirements spelled out in the [RFC4460] and included in [RFC4960].

This work and some of the changes that went into [RFC4460] and [RFC4960] are much indebted to the paper on potential SCTP security risks [EFFECTS] by Aura, Nikander, and Camarillo. Without their work, some of these changes would remain undocumented and potential threats.

The rest of this document will concentrate on the various attacks that were illustrated in [EFFECTS] and detail the preventative measures now in place, if any, within the current SCTP standards.

2. Address Camping or Stealing

This attack is a form of denial of service attack crafted around SCTP’s multi-homing. In effect, an illegitimate endpoint connects to a server and "camps upon" or "holds up" a valid peer’s address. This is done to prevent the legitimate peer from communicating with the server.

2.1. Attack Details

```
+----------+            +----------+           +----------+
| Evil     |            |  Server  |           | Client   |
|     IP-A=+------------+          +-----------+=IP-C & D |
| Attacker |            |          |           | Victim   |
+----------+            +----------+           +----------+
```

Figure 1: Camping

Consider the scenario illustrated in Figure 1. The attacker legitimately holds IP-A and wishes to prevent the 'Client-Victim' from communicating with the 'Server'. Note also that the client is multi-homed. The attacker first guesses the port number our client will use in its association attempt. It then uses this port and sets up an association with the server listing not only IP-A but also IP-C in its initial INIT chunk. The server will respond and set up the association, noting that the attacker is multi-homed and holds both IP-A and IP-C.

Next, the victim sends in an INIT message listing its two valid addresses, IP-C and IP-D. In response, it will receive an ABORT message with possibly an error code indicating that a new address was added in its attempt to set up an existing association (a restart with new addresses). At this point, 'Client-Victim' is now prevented from setting up an association with the server until the server realizes that the attacker does not hold the address IP-C at some future point by using a HEARTBEAT based mechanism. See the mitigation option subsection of this section.

2.2. Analysis

This particular attack was discussed in detail on the SCTP implementors list in March of 2003. Out of that discussion, changes were made in the BSD implementation that are now present in [RFC4960]. In close examination, this attack depends on a number of specific things to occur.

1) The attacker must set up the association before the victim and must correctly guess the port number that the victim will use. If the victim uses any other port number the attack will fail.

2) SCTP’s existing HEARTBEAT mechanism as defined already in [RFC2960] will eventually catch this situation and abort the evil attacker’s association. This may take several seconds based on default HEARTBEAT timers but the attacker himself will lose any association.

3) If the victim is either not multi-homed, or the address set that it uses is completely camped upon by the attacker (in our example if the attacker had included IP-D in its INIT as well), then the client’s INIT message would initiate an association between the client and the server while destroying the association between the attacker and the server. From the servers’ perspective, this is a restart of the association.

2.3. Mitigation Option

[RFC4960] adds a new set of requirements to better counter this attack. In particular, the HEARTBEAT mechanism was modified so that addresses unknown to an endpoint (i.e., presented in an INIT with no pre-knowledge given by the application) enter a new state called "UNCONFIRMED". During the time that any address is UNCONFIRMED and yet considered available, heartbeating will be done on those UNCONFIRMED addresses at an accelerated rate. This will lessen the time that an attacker can "camp" on an address. In particular, the rate of heartbeats to UNCONFIRMED addresses is done every RTO. Along with this expanded rate of heartbeating, a new 64-bit random nonce is required to be inside HEARTBEATs to UNCONFIRMED addresses. In the HEARTBEAT-ACK, the random nonce must match the value sent in the HEARTBEAT before an address can leave the UNCONFIRMED state. This will prevent an attacker from generating false HEARTBEAT-ACKs with the victim’s source address(es). In addition, clients that do not need to use a specific port number should choose their port numbers on a random basis. This makes it hard for an attacker to guess that number.

3. Association Hijacking 1

Association hijacking is the ability of some other user to assume the session created by another endpoint. In cases of a true man-in-the-middle, only a strong end-to-end security model can prevent this. However, with the addition of the SCTP extension specified in [RFC5061], an endpoint that is NOT a man-in-the-middle may be able to assume another endpoint’s association.

3.1. Attack Details

The attack is made possible by any mechanism that lets an endpoint acquire some other IP address that was recently in use by an SCTP endpoint. For example, DHCP may be used in a mobile network with short IP address lifetimes to reassign IP addresses to migrant hosts.

```
     IP-A                 DHCP-Server's       Peer-Server
       |
       |
    1  |-DHCP-Rel(IP-A)---->|
    2  |------ASCONF(ADD-IP(IP-B), DEL-IP(IP-A)---->XXlost
      time
       |
       |-DHCP-new-net------>|
    3  |<---Assign (IP-A)
       |
    4  |<------------Tag:X-DATA()------------------
       |
       |-------------INIT()------------------------>
    5  |<------------INIT-ACK()---------------------
       |
    6  |----ASCONF(ADD-IP(IP-Z),DEL-IP(IP-A))------>
```

Figure 2: Association Hijack via DHCP

At point 1, our valid client releases the IP address IP-A. It presumably acquires a new address (IP-B) and sends an ASCONF to ADD the new address and delete to old address at point 2, but this packet is lost. Thus, our peer (Peer-Server) has no idea that the former peer is no longer at IP-A. Now at point 3, a new "evil" peer obtains an address via DHCP and it happens to get the re-assigned address IP-A. Our Peer-Server sends a chunk of DATA at point 4. This reveals to the new owner of IP-A that the former owner of IP-A had an association with Peer-Server. So at point 5, the new owner of IP-A sends an INIT. The INIT-ACK is sent back and inside it is a COOKIE. The cookie would of course hold tie-tags, which would list both sets of tags that could then be used at point 6 to add in any other IP addresses that the owner of IP-A holds and thus acquire the association.

It should be noted that this attack is possible in general whenever the attacker is able to send packets with source address IP-A and receive packets with destination address IP-A.

3.2. Analysis

This attack depends on a number of events:

1) Both endpoints must support the SCTP extension specified in [RFC5061].

2) One of the endpoints must be using the SCTP extension for mobility specified in [RFC5061].

3) The IP address must be acquired in such a way as to make the endpoint the owner of that IP address as far as the network is concerned.

4) The true peer must not receive the ASCONF packet that deletes IP-A and adds its new address to the peer before the new "evil" peer gets control of the association.

5) The new "evil" peer must have an alternate address, aside from the IP-A that it can add to the association, so it can delete IP-A, preventing the real peer from re-acquiring the association when it finally retransmits the ASCONF (from step 2).

3.3. Mitigation Option

[RFC4960] adds a new counter measure to this threat. It is now required that Tie-Tags in the State-Cookie parameter not be the actual tags. Instead, a new pair of two 32-bit nonces must be used to represent the real tags within the association. This prevents the attacker from acquiring the real tags and thus prevents this attack. Furthermore, the use of the SCTP extension specified in [RFC5061] requires the use of the authentication mechanism defined in [RFC4895]. This requires the attacker to be able to capture the traffic during the association setup. If in addition an endpoint-pair shared key is used, capturing or intercepting these setup messages does not enable the attacker to hijack the association.

4. Association Hijacking 2

Association hijacking is the ability of some other user to assume the session created by another endpoint. In cases where an attacker can send packets using the victims IP-address as a source address and can receive packets with the victims’ address as a destination address, the attacker can easily restart the association. If the peer does not pay attention to the restart notification, the attacker has taken over the association.

4.1. Attack Details

Assume that an endpoint E1 having an IP-address A has an SCTP association with endpoint E2. After the attacker is able to receive packets to destination address A and send packets with source address A, the attacker can perform a full four-way handshake using the IP-addresses and port numbers from the received packet. E2 will consider this a restart of the association. If and only if the SCTP user of E2 does not process the restart notification, the user will not recognize that the association just restarted. From this perspective, the association has been hijacked.

4.2. Analysis

This attack depends on a number of circumstances:

1) The IP address must be acquired in such a way as to make the evil endpoint the owner of that IP address as far as the network or local LAN is concerned.

2) The attacker must receive a packet belonging to the association or connection.

3) The other endpoint’s user does not pay attention to restart notifications.

4.3. Mitigation Option

It is important to note that this attack is not based on a weakness of the protocol, but on the ignorance of the upper layer. This attack is not possible if the upper layer processes the restart notifications provided by SCTP as described in section 10 of [RFC2960] or [RFC4960]. Note that other IP protocols may also be affected by this attack.

5. Bombing Attack (Amplification) 1

The bombing attack is a method to get a server to amplify packets to an innocent victim.

5.1. Attack Details

This attack is performed by setting up an association with a peer and listing the victims IP address in the INIT’s list of addresses. After the association is setup, the attacker makes a request for a large data transfer. After making the request, the attacker does not acknowledge data sent to it. This then causes the server to re-transmit the data to the alternate address, i.e., that of the victim.

After waiting an appropriate time period, the attacker acknowledges the data for the victim. At some point, the attackers address is considered unreachable since only data sent to the victims address is acknowledged. At this point, the attacker can send strategic acknowledgments so that the server continues to send data to the victim.

Alternatively, instead of stopping the sending of SACKs to enforce a path failover, the attacker can use the ADD-IP extension to add the address of the victim and make that address the primary path.

5.2. Analysis

This attack depends on a number of circumstances:

1) The victim must NOT support SCTP, otherwise it would respond with an "out of the blue" (OOTB) abort.

2) The attacker must time its sending of acknowledgments correctly in order to get its address into the failed state and the victim’s address as the only valid alternative.

3) The attacker must guess TSN values that are accepted by the receiver once the bombing begins since it must acknowledge packets it is no longer seeing.

5.3. Mitigation Option

[RFC4960] makes two changes to prevent this attack. First, it details proper handling of ICMP messages. With SCTP, the ICMP messages provide valuable clues to the SCTP stack that can be verified with the tags for authenticity. Proper handling of an ICMP protocol unreachable (or equivalent) would cause the association setup by the attacker to be immediately failed upon the first retransmission to the victim’s address.

The second change made in [RFC4960] is the requirement that no address that is not CONFIRMED is allowed to have DATA chunks sent to it. This prevents the switch-over to the alternate address from occurring, even when ICMP messages are lost in the network and prevents any DATA chunks from being sent to any other destination other then the attacker itself. This also prevents the alternative way of using ADD-IP to add the new address and make it the primary address.

An SCTP implementation should abort the association if it receives a SACK acknowledging a TSN that has not been sent. This makes TSN guessing for the attacker quite hard because if the attacker acknowledges one TSN too fast, the association will be aborted.

6. Bombing Attack (Amplification) 2

This attack allows an attacker to use an arbitrary SCTP endpoint to send multiple packets to a victim in response to one packet.

6.1. Attack Details

The attacker sends an INIT listing multiple IP addresses of the victim in the INIT’s list of addresses to an arbitrary endpoint. Optionally, it requests a long cookie lifetime. Upon reception of the INIT-ACK, it stores the cookie and sends it back to the other endpoint. When the other endpoint receives the COOKIE, it will send back a COOKIE-ACK to the attacker and up to HB.Max.Burst HEARTBEATS to the victim’s address(es) (to confirm these addresses). The victim responds with ABORTs or ICMP messages resulting in the removal of the TCB at the other endpoint. The attacker can now resend the stored cookie as long as it is valid, and this will again result in up to HB.Max.Burst HEARTBEATs sent to the victim(’s).

6.2. Analysis

The multiplication factor is limited by the number of addresses of the victim and of the endpoint HB.Max.Burst. Also, the shorter the cookie lifetime, the earlier the attacker has to go through the initial stage of sending an INIT instead of just sending the COOKIE. It should also be noted that the attack is more effective if large HEARTBEATs are used for path confirmation.

6.3. Mitigation Option

To limit the effectiveness of this attack, the new parameter HB.Max.Burst was introduced in [RFC4960] and an endpoint should:

1) not allow very large cookie lifetimes, even if they are requested.

2) not use larger HB.Max.Burst parameter values than recommended. Note that an endpoint may decide to send only one Heartbeat per RTT instead of the maximum (i.e., HB.Max.Burst). An endpoint that chooses this approach will however slow down detection of endpoints camping on valid addresses.

3) not use large HEARTBEATs for path confirmation.

7. Association Redirection

This attack allows an attacker to wrongly set up an association to a different endpoint.

7.1. Attack Details

The attacker sends an INIT sourced from port 'X' and directed towards port 'Y'. When the INIT-ACK is returned, the attacker sends the COOKIE-ECHO chunk and either places a different destination or source port in the SCTP common header, i.e., X+1 or Y+1. This possibly sets up the association using the modified port numbers.

7.2. Analysis

This attack depends on the failure of an SCTP implementation to store and verify the ports within the COOKIE structure.

7.3. Mitigation Option

This attack is easily defeated by an implementation including the ports of both the source and destination within the COOKIE. If the source and destination ports do not match those within the COOKIE chunk when the COOKIE is returned, the SCTP implementation silently discards the invalid COOKIE.

8. Bombing Attack (Amplification) 3

This attack allows an attacker to use an SCTP endpoint to send a large number of packets in response to one packet.

8.1. Attack Details

The attacker sends a packet to an SCTP endpoint, which requires the sending of multiple chunks. If the SCTP endpoint does not support bundling on the sending side, it might send each chunk per packet. These packets can either be sent to a victim by using the victim's address as the sources address, or it can be considered an attack against the network. Since the chunks, which need to be sent in response to the received packet, may not fit into one packet, an endpoint supporting bundling on the sending side might send multiple packets.

Examples of these packets are packets containing a lot of unknown chunks that require an ERROR chunk to be sent, known chunks that initiate the sending of ERROR chunks, packets containing a lot of HEARTBEAT chunks, and so on.

8.2. Analysis

This attack depends on the fact that the SCTP endpoint does not support bundling on the sending side or provides a bad implementation of bundling on the sending side.

8.3. Mitigation Option

First of all, path verification must happen before sending chunks other than HEARTBEATs for path verification. This ensures that the above attack cannot be used against other hosts. To avoid the attack, an SCTP endpoint should implement bundling on the sending side and should not send multiple packets in response. If the SCTP endpoint does not support bundling on the sending side, it should not send in general more than one packet in response to a received one. The details of the required handling are described in [RFC4960].

9. Bombing Attack (Amplification) 4

This attack allows an attacker to use an SCTP server to send a larger packet to a victim than it sent to the SCTP server.

9.1. Attack Details

The attacker sends packets using the victim’s address as the source address containing an INIT chunk to an SCTP Server. The server then sends a packet containing an INIT-ACK chunk to the victim, which is most likely larger than the packet containing the INIT.

9.2. Analysis

This attack is a byte and not a packet amplification attack and, without protocol changes, is hard to avoid. A possible method to avoid this attack would be the usage the PAD parameter defined in [RFC4820].

9.3. Mitigation Option

A server should be implemented in a way that the generated INIT-ACK chunks are as small as possible.

10. Bombing Attack (amplification) 5

This attack allows an attacker to use an SCTP endpoint to send a large number of packets in response to one packet.

10.1. Attack Details

The attacker sends a packet to an SCTP endpoint, which requires the sending of multiple chunks. If the MTU towards the attacker is smaller than the MTU towards the victim, the victim might need to send more than one packet to send all the chunks. The difference between the MTUs might be extremely large if the attacker sends malicious ICMP packets to make use of the path MTU discovery.

10.2. Analysis

This attack depends on the fact that an SCTP implementation might not limit the number of response packets correctly.

10.3. Mitigation Option

First of all, path verification must happen before sending chunks other than HEARTBEATs for path verification. This makes sure that the above attack cannot be used against other hosts. To avoid the attack, an SCTP endpoint should not send multiple packets in response to a single packet. The chunks not fitting in this packet should be dropped.

11. Security Considerations

This document is about security; as such, there are no additional security considerations.

12. References

12.1. Normative References

[RFC2960] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

[RFC4460] Stewart, R., Arias-Rodriguez, I., Poon, K., Caro, A., and M. Tuexen, "Stream Control Transmission Protocol (SCTP) Specification Errata and Issues", RFC 4460, April 2006.

[RFC4820] Tuexen, M., Stewart, R., and P. Lei, "Padding Chunk and Parameter for the Stream Control Transmission Protocol (SCTP)", RFC 4820, March 2007.

[RFC4895] Tuexen, M., Stewart, R., Lei, P., and E. Rescorla, "Authenticated Chunks for Stream Control Transmission Protocol (SCTP)", RFC 4895, August 2007.

[RFC5061] Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", RFC 5061, September 2007.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, June 2007.

12.2. Informative References

[EFFECTS] Aura, T., Nikander, P., and G. Camarillo, "Effects of Mobility and Multihoming on Transport-Layer Security", Security and Privacy 2004, IEEE Symposium , URL http://research.microsoft.com/users/tuomaura/Publications/aura-nikander-camarillo-ssp04.pdf, May 2004.

Authors' Addresses

Randall R. Stewart
Cisco Systems, Inc.
4785 Forest Drive
Suite 200
Columbia, SC 29206
USA

EMail: rrs@cisco.com

Michael Tuexen
Muenster Univ. of Applied Sciences
Stegerwaldstr. 39
48565 Steinfurt
Germany

EMail: tuexen@fh-muenster.de

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com