CAN SIP USE HIP?

Tom Henderson
thomas.r.henderson@boeing.com

ABSTRACT

Now that experimental implementations of HIP are maturing, it is important to move beyond the abstract benefits of separating identifier from locator to consider how HIP might specifically help the performance or security of popular applications or middleware. The Session Initiation Protocol (SIP) is an important emerging middleware protocol that, upon first glance, might seemingly benefit from HIP support in the network, due to its reliance on redirection and proxy servers in the network, need to support mobility, and its web of trust relationships. This position paper argues that a closer look be given to applications and protocols like SIP to consider whether and how much HIP might aid them, and performs a first-cut qualitative examination of the possible combination.

INTRODUCTION

The Session Initiation Protocol (SIP) [Ros02] is an emerging middleware call control protocol for multimedia flows. The SIP architecture uses URIs (user@domain) to identify endpoints, and includes various infrastructure components, including registrars (which map URIs to IP addresses or host names), redirect servers (signaling indirection pushed back to the client), and proxy servers (either stateful or stateless forwarding of requests). SIP permits various techniques for securing the protocol, including IPsec and TLS, HTTP Digest authentication, and S/MIME. SIP can also implement mobility management at the application layer [Sch00].

The Host Identity Protocol (HIP) [Mos04] introduces a new host name space based on public keys, thereby allowing the network and transport layers to be decoupled. The HIP architecture suggests that a new name service for host identities might be needed, also allows for indirection (rendezvous servers), and provides mobility management. Additional proposals to integrate HIP with overlay network architectures are under active consideration.

Although there are common architectural elements between SIP and HIP, SIP has no dependencies on HIP. SIP has implemented certain mechanisms also found in HIP, but at a higher layer (session or application). In light of the similarities, it seems important to ask whether HIP, were it to be deployed, would offer benefits to SIP. The use of HIP may also have negative implications for SIP, such as heavyweight mechanisms or additional round trips. This paper performs an initial exploration of these questions, based on an initial cursory (non-expert) review of SIP.

BINDING SIP URIS TO HIS

The SIP registration mechanism binds a SIP URI to a temporary IP address or host name. If host identifiers were available, it seems natural that SIP could instead bind the URI to a host identifier (HIT or HI). This interposes an additional level of indirection to obtain the addresses, so it is important to consider what is gained or lost by this new layer.

Consider an INVITE from bob@biloxi.com to carol@chicago.com (Figure 2 of Section 10 of [Ros02]). Bob's SIP proxy queries a location service for the address of Carol. This usually can be accomplished by DNS SRV records. If SIP URIs bind to host identities, one of two things is possible at this stage: either the location service returns a HI or a HIT, and the proxy needs to invoke an additional lookup to find the IP addresses, or the IP addresses are stored with the HI or HIT. Once it receives the address of Carol or the next hop proxy, Bob's proxy continues to try to invite carol@chicago.com. Upon successful negotiation, Bob's proxy returns OK to bob@biloxi.com, and Bob has the necessary information to contact Carol directly and set up media flows.

Considering this basic example, it is possible that host identity information can be stored and returned in tandem with IP addresses at each step of the resolution process, or that an additional resolution step is needed (including the possibility of multiple resolutions if multiple proxies are traversed). Further exploration is needed of whether the HIP DNS work already ongoing in the HIP working group can be leveraged for SIP resolution as well, and what additional trust relationships are needed to establish that the URI to host identity binding is secure.

Assuming that SIP infrastructure elements and user agents (UAs) obtain host identities that are bound to URIs or SIP infrastructure addresses, what benefit or drawback might SIP have in obtaining and using host identities instead of IP addresses? Both call control and user data plane can be considered, initially without considering mobility.

User plane

The SIP specification does not discuss the securing of media streams set up between clients and servers. IPsec or higher layer (e.g. application) means are suggested. The use of IPsec without HIP may open a potential man-in-the-middle vulnerability. HIP could potentially be used to set up BEET IPsec security associations, although it may be the case that HIP and IPsec are considered too heavyweight from a client device perspective.

Control plane

There are many possible attacks on SIP agents and infrastructure, but SIP reuses HTTP and SMTP techniques as much as possible to plug these attack vectors. In general, UAs authenticate themselves to servers with a Digest username and password, while servers authenticate themselves to UAs or other servers with a site certificate delivered by TLS [Ros02]. End-to-end authentication can be transitive, or else based on S/MIME if the network is not trusted. If strict security is desired, UAs can use a special form of the URI, the "SIPS" URI, to explicitly force each hop on the path to use TLS. If TLS is being used end-to-end, it is not clear that hop-by-hop HIP-enabled IPsec security offers that much more benefit to SIP, other than potentially reducing DoS attack vulnerability on the TCP connection setup.

In summary, in the non-mobile case, it is not clear at first glance how much HIP will aid SIP, other than providing a facility to set up end-to-end IPsec security associations that are not vulnerable to man-in-the-middle attacks

It should be noted that SIP itself has a facility to exchange public keys, as described in Section 23.2 of RFC 3261, although it is susceptible to attacks as might be found in ssh when obscure certificate authorities or self-signed certificates are used in place of a well-known authority [Ros02]. But using SIP to exchange HIP identities (for subsequent use by the data plane) may be worth considering.

MOBILITY

SIP can offer terminal mobility, as described in [Sch00]. Prior to a call, mobility is handled by reregistration with the home registrar. For mid-call mobility, the moving node sends a re-INVITE directly to the correspondent host, or via the SIP proxies if during the initial call setup, the proxy inserted a Record-Route header. To secure these handoffs, SIP uses HTTP-style digest authentication or public-key cryptography. Double moves or network partition events are recovered at the session layer by both parties resending the INVITE request to the home proxy of the other side. A SIP session timer mechanism automatically causes a refresh of the session at user-configurable intervals [Sch00].

The basic SIP mobility has two main limitations as described above. The first is the inability to move TCP sessions to new addresses. This could be accomplished by TCP extensions such as TCP-Migrate [Sno02] or M-TCP [Hui95]. Alternatively, a session-oriented checkpoint mechanism for TCP applications might be used [Zan02]. The second limitation is the speed of handoffs. Schulzrinne states that mobile-IP-like micromobility approaches could be adapted for SIP as well, or SIP proxies could be used as points of indirection [Sch00].

Were HIP to be available, HIP has the advantage that it can preserve all transport sessions. As presently defined, however, HIP mobility, like that of SIP, does not offer micromobility solutions or handling of the double-jump problem. Although proposals to use rendezvous servers to provide additional mobility mechanisms have been circulated, they are not presently available. In summary, HIP mobility mechanisms as presently defined do not offer a lot of additional features or security over what SIP can already provide, especially given that reliable transport protocols are not the dominant ones used in SIP user data flows.

MIDDLEBOX TRAVERSAL

SIP has been extended recently to allow direct UDP and indirect TCP connectivity between hosts behind NATs and firewalls, based on the ICE (Interactive Connectivity Establishment) extension, which in turn relies on the STUN and TURN protocols [Guh04]. A recent paper to SIGCOMM FDNA workshop [Guh04] suggested the NUTSS architecture (NAT, URI, Tunnel, SIP, and STUNT-- STUN for TCP) to allow SIP to be used as a general call control protocol for P2P applications, including behind NATs and firewalls. STUNT is self-described as a hack that permits TCP connections to traverse NATs and firewalls when the end-hosts are not necessarily aware of the global address and ports used by the public face of the NAT, and STUNT gets around the problem of no NAT inbound connections by having both ends initiate connections to a proxy, which bridges them together.

HIP is, in theory, better suited to NAT traversal for TCP connections because of the decoupling of transport from network layer, so it might offer an approach like NUTSS some benefit. In particular, some of the NAT traversal hacks might be obviated by middlebox support of HIP. However, HIP-aware middleboxes do not exist yet, and it is not clear whether things would be much cleaner when trying to have HIP traverse legacy middleboxes using UDP tunneling or other (unspecified) techniques.

SUMMARY

Now that basic HIP experimental RFCs are finalizing and several interoperable implementations exist, it is time for the touted benefits of identifier/locator split to be translated into deployable benefits for Internet middleware and applications. This paper has initiated an exploration as to whether SIP is such a “killer app.”

On first glance, it does not appear that basic HIP offers much benefit to the SIP call control plane. SIP uses mechanisms like S/MIME, HTTP Digest, and TLS to authenticate and protect the signaling traffic-- it is not clear, beyond DoS protection, what more HIP might benefit SIP signaling exchanges. In the data plane, SIP mobility management is similar to that of HIP, with the main difference being HIP’s ability to rehome a TCP connection-- but, at present, most SIP flows do not use TCP. It seems that the main benefit that a HIP-enabled network infrastructure would offer SIP is better mobility management (such as micro-mobility management), possible integration of rendezvous servers with SIP proxy and redirection servers, and NAT traversal. However, HIP needs to work out more details in these areas before the answer becomes clear. Perhaps SIP can be a motivating application for these HIP extensions.

REFERENCES

[Guh04] S. Guha et al., “NUTSS: A SIP-based Approach to UDP and TCP Network Connectivity,” SIGCOMM FDNA Workshop, 2004.

[Hui95] C. Huitema, “Multihomed TCP,” Internet-draft (expired), May 1995.

[Mos04] R. Moskowitz et al., “Host Identity Protocol,” Interent-draft draft-ietf-hip-base-00, June 11, 2004.

[Ros00] J. Rosenberg et al., “SIP: Session Initiation Protocol,” IETF RFC 3261, June 2002.

[Sch00] H. Schulzrinne and E Wedlund, “Application-Layer Mobility using SIP,” ACM MC2R, July 2000.

[Sno02] A. Snoeren, H. Balakrishan, and M.F. Kaashoek, “Migrate: An End-to-End Approach to Mobility,” MIT LCS Technical report, March 2002.

[Zan02] V. Zandy and B. Miller, “Reliable Network Connections,” Proceedings of MOBICOM 2002.