Computer Security, PART IV: Host Identity Protocol

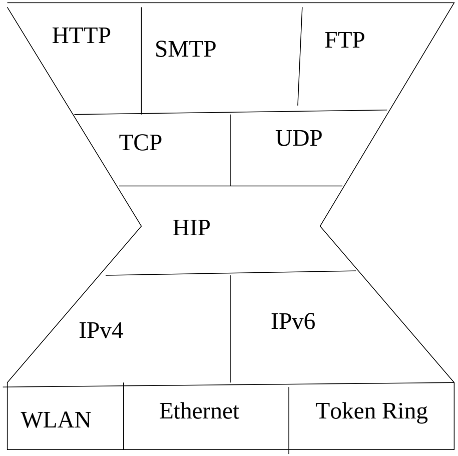
Timo Karvi

11.2010

HIP in the Internet Architecture I

- A major problem in the origal Internet architecture is tight coupling between networking and transport layers. As an example, TCP uses a pseudoheader including IP addresses in checksum calculations. Therefore, independent evolution of these two layers is not possible. Introduction of a new transport or networking layer requires changes to the other layer.
- One problem in the current TCP/IP-architecture is the control of the waist in the protocol stack. Previously, IPv4 took care of this, but nowadays we have dual-stacks networks, where both IPv4 and IPv6 are run parallel.
- Host Identity Protocol, HIP, is meant to be a new waist in the protocol stack:

HIP in the Internet Architecture II



HIP in the Internet Architecture III

- Another major problem of the present Internet is Denial-of-Service attacks. The SYN attack against TCP is a good and unsolved example of these attacks.
- When a TCP connection is established, an initiator sends the SYN packet to a recipient. The recipient replies with a SYN-ACK packet and allocates TCP control block structure.
- At that stage, the recipient has no assurance that the SYN has arrived from the same host as stated in the SYN source address.
- Using this exploit, a moderate number of hosts can swamp the recipient with SYN messages, exhausting the recipient's memory or other system resources.
- HIP makes such attacks more difficult by connecting the identity of the initiator to SYN packets and, furthermore, using cryptographic puzzles to prevent the initiator generating connection attempts at an overly fast rate.

HIP in the Internet Architecture IV

- The puzzle is basically aasking the initiator to reverse a hash function of a small length that statistically requires significant computational resources. On the other hand, verifying the answer at the initiator is a short operation.

Future Internet I

HIP is one alternative for the future Internet. There are many other suggestions:

- Packet Level Authentication (PLA)
    - PLA is a novel network-level solution where any node in the network is able to verify the authenticity and integrity of a packet without any prior communication with the originator of the packet.
    - Hence PLA can prevent different classical security attacks such as Denial of Service (DoS) attacks efficiently by being able to detect any modified or forged packet in the network.
    - PLA uses its own header together with public key cryptography in order to provide hop by hop authenticity and integrity protection.

Future Internet II

- In order to achieve minimal packet overhead and high speed signature verification, Elliptic Curve Cryptography (ECC) is used in PLA. Field Programmable Gate Arrays (FPGAs)-based implementation of ECC-based cryptographic algorithms in PLA is able to verify 166,000 packets per second with 114 s latency per verification. An accelerator based on dedicated ASIC would achieve significantly higher performance making PLA scalable to high speed core networks.
- However, despite the use of elliptic curve cryptosystems, the cryptographic settings and the primitives of PLA still involve CPU-intensive computations. Therefore, without the use of any dedicated hardware for accelerating the cryptographic operations, the required operations pertaining to the signature generation and verification scheme of PLA in a general purpose processor introduces a performance penalty

- Timed Efficient Stream Loss-Tolerant Authentication, TESLA
  - RFC 4082, 2005.

Future Internet III

- TESLA is a broadcast authentication protocol based on Message Authentication Codes (MACs) that requires time synchronization between the sender and the receiver.
- In TESLA, a MAC is embedded in each packet to provide authentication. The corresponding MAC keys are disclosed to the receiver after a time delay. The delay before the disclosure is chosen long enough so that they cannot be used to forge packets.
- Although this scheme is robust against packet loss and scalable, it requires that the sender and receiver synchronize their clocks within a certain margin. In settings where time synchronization is difficult to achieve, TESLA might not work.

- Lightweight Hop-by-hop Authentication Protocol, LHAP
  - LHAP, 2003, is a scalable lightweight hop-by-hop authentication protocol specially designed for ad hoc networks.
  - It uses two keys: the TRAFFIC key and TESLA key. The TRAFFIC key is used to authenticate packets and the TESLA key is used to achieve trust maintenance by an authenticating KEYUPDATE message.

Future Internet IV

  - The KEYUPDATE message is sent periodically to guarantee that the current released key is valid so that a malicious node will not be able to use an obsolete key to forge a packet.
  - This method increases the computational efficiency by avoiding the hash computation over the entire message. This makes the procedure more vulnerable against message tampering attacks, replay attacks, and wormhole attacks
- The Adapted and Lightweight Protocol for Hop-by-hop Authentication, ALPHA, 2008.
  - ALPHA is a lightweight scheme for hop-by- hop as well as end-to-end authentication and integrity protection in multi-hop wireless networks.
  - It is based on the interactive signatures and Merkle Trees (MT) that are used to protect the communication path between two arbitrary network nodes. Any intermediate node in this secured path is able to verify the integrity of the transmitted data packets.

Future Internet V

- ALPHA consists of three different modes of operation, namely the Base protocol, ALPHA-C with cumulative transmissions, and ALPHA-M combined with MTs.
- These modes complement each other and allow for a fine-grained and dynamic adaptation to different communication scenarios ranging from transmission of infrequent control traffic to sending large amounts of data.
- The downside of ALPHA is that it is network path dependent, which means that all packets must travel along the protected path to ensure end-to-end authenticity and integrity. Therefore ALPHA is not suitable for environments that use multi-path routing.

HIP: Name space I

- HIP adds a new name space to the TCP/IP protocol stack. Network hosts are identified with new identifiers, host identifiers, (HIs). The HIs are public cryptographic keys.
- Therefore, peer hosts authenticate directly by their HI. HIP has been designed to be backwards compatible, i.e. no changes to the network infrastructure or to the applications are needed.
- Using public keys as host identifiers in packets and application interfaces is inconvenient due to large and variable size. Instead of HIs, host identity tags, HITs are used in packets. HIT is a 128-bit hash of the public key and is thus of the same length as the IPv6 address. HITs have a prefix 2001:0010::/28 that enables to distinguish them from currently allocated IPv6 addresses.
- In order to bind other names to the HIT, a variety of mechanisms, like DNSSEC, SPKI/SDSI, and X.509 certificates, are available.

HIP: Name space II

- If an entity A wants to start a HIP connection with B, A first makes a DNS query in order to get B's IP address and B's host identity. In this query, A uses human-friendly host names.
- It is essential that A can trust the DNS response. If the data in the response is incorrect, then it might be possible that A communicates with a malicious entity without noticing it.
- There can be only one HIP association between a pair of HITs. Therefore, the only way to support multiple associations between two hosts is to have several HITs per host. This leads to a situation where a host may have several RSA or DSA public/private key pairs.

HIP:Base Exchange I

The HIP connection is negotiated by performing the Base Exchange protocol, see the figure below. An initiator I starts the negotiation with a responder R. Message I1 is sent without signature, messages R1, I2, R2 are signed (sig). R1 and I2 contain Diffie-Hellman parameters ($DH(r)$, $DH(i)$), a puzzle (in R2), its solution (in I2), and a puzzle difficulty parameter K.

$$\begin{array}{rl}
\text{I1} & I \longrightarrow R\text{: } \mathrm{HIT}(i),\ \ \mathrm{HIT}(r) \\
\text{R1} & R \longrightarrow I\text{: } \mathrm{HIT}(r),\ \ \mathrm{HIT}(i), \mathrm{puzzle}, DH(r), K(r), \mathrm{sig} \\
\text{I2} & I \longrightarrow R\text{: } \mathrm{HIT}(i),\ \ \mathrm{HIT}(r), \mathrm{solution}, DH(i), K(i), \mathrm{sig} \\
\text{R2} & R \longrightarrow I\text{: } \mathrm{HIT}(r), \mathrm{HIT}(i), \mathrm{sig}
\end{array}$$

Figure: Base Exchange

HIP:Base Exchange II

The key agreement method used in the Base Exchange is based on the so-called SIGMA key agreement (Krawczyk: SIGMA: The 'SIGn-and MAc' approach to authenticated Diffie-Hellman and its use in the IKE protocols. CRYPTO 2003.)

I1 Packet

- When a I1 packet is sent, a retransmission timer is started. After a timeoout, the host retransmits the I1 packet and restarts the timer.
- The same packet can be sent in parallel up to three different IP addresses of a peer host to reduce the latency of the association establishment.
- All HIP control packets are transmitted after a basic IPv4 or IPv6 header.

R1 Packet

- When a responder receives the I1 packet, it does not store yet any information (called state) about the initiator.
- The R1 packet is signed with a private key and the corresponding public key is sent in the R1 packet.
- The current standard does not define how certificates are used to validate public keys. However, there is an optional field CERT in R1 and I2 packets, where a certificate can be sent. Moreover, it is possible to send a list of certificate authorities which can be used to verify the certificate.
- The signature is based on RSA or DSA and it is calculated over the whole R1 packet.

I2 Packet

- Packet I2 has two authentication fields. The first field is HMAC. Its value is calculated over the entire I2 packet excluding the signature and the secret parameter used in the calculation is computed from the Diffie-Hellman value.
- HMAC is faster to verify than the signature and it is checked before the signature verification as an additional protection against replay attacks. The mere signature does not prevent replay attacks.
- The signature is calculated using the initiator's private key and the corresponding public key is sent in the I2 packet.

R2 Packet and After That

- The R2 packet contains the HMAC value in addition to the signature.
- The data can flow on a HIP association after the R2 packet.
- The data can be encapsulated in IPSec ESP packets. The standard says that the support for ESP is mandatory.
- The common secret needed to establish an IPSec association is got from the Diffie-Hellman value. Thus no IKE is needed.
- The data transfer uses typically TCP. To set up a TCP connection, the three-way handshake takes an additional round-trip time after two round-trips of the base exchange.
- To reduce the delay observed by applications, it may be possible to transmit upper-layer data already in I2 and R2 packets. Earlier transmission is not possible due to security concerns.

Other HIP Control Packets

- UPDATE: A host sends an UPDATE packet when it wants to update some information related to a HIP association, e.g. mobility management (new IP address) and rekeying of an existing ESP security association.
- NOTIFY can be used to inform of protocols errors.
- CLOSE closes the HIP connection. It is acknowledged by CLOSE-ACK packet.

Security Mechanisms in HIP I

- Puzzle Mechanism:
  - The purpose of the HIP puzzle mechanism is protection against denial-of-service threats. It delays state creation and it uses a fairly cheap calculation to check that the HIP Base Exchange initiator is sincere.
  - Puzzle calculation is based on the initiators HIT and therefore the responder can remain stateless and drop most spoofed I2 packets.
  - The responder can set the puzzle difficulty for the initiator. The difficulty is based on the level of trust in the initiator. The responder uses heuristics to determine whether it is the target of a denial-of-service attack and to set the puzzle difficulty.
  - RFC 5201.
- Security Associations:
  - IPSec with its security associations is used in HIP connections. A Security Association (SA) is needed by security services for the data traffic.

Security Mechanisms in HIP II

- A SA is uniquely identified by a triple consisting of a Security Parameter Index (SPI), an IP Destination Address, and a security protocol identifier.
- Because the SAs are not bound to IP addresses, a host node is able to receive HIP created ESP SA packets from any IP address. A node can change its topological location and continuously send and receive packets to and from its peers.
- When a host is authenticated, the Base Exchange sets up two SAs, one in each direction, to enable the end-to-end security of ESP enveloping.
- The origin of a HIP control packet can be verified by the responder by verifying the packet signature. An alternative is that the two end-points of an active HIP association verify the message authentication code.
- When a data packet is received, the host locates an ESP SA based on the SPI carried in the packet, verifies packet integrity, and places the source and destination HITs into the packet. These HITs are stored in the ESP BEET-mode (bound end-to-end tunnel) SA.

- HIP Replay Protection:

Security Mechanisms in HIP III

- The HIP protocol includes mechanisms to protect against malicious replays.
- Responders are protected against replays of I1 packets with presigned R1 messages and against false I2 packets by the puzzle mechanism and optional use of opaque data.
- Initiators are protected against R1 replays by a generation counter. The value of this counter is kept across system reboots and invocations of the HIP base exchange.
- Protection against replays of R2s and UPDATEs is handled by HMAC verification preceding HIP signature verification. A cryptographic HMAC and a signature are located at the end of the message packet. The peer host can use Diffie-Hellman session keys to verify the HMAC. The signature is meant for middle boxes. Typically, these boxes do not have access to the Diffie-Hellman key but may have access to the senders public key.

- NATs and Firewalls:

Security Mechanisms in HIP IV

- Network Address Translation (NAT) devices allow the use of private IP addresses on internal networks behind routers with a single public IP address facing the Internet.
- It is agreed (HIP Working Group at the IETF) that HIP control packets and ESP protected data should be embedded in UDP packets in a similar manner as IPSec IKE and ESP traffic is embedded in the IPSec traversal solution.
- Reaching a receiver located behind a NAT is problematic. Initiated communication and some mapping state in the NAT is needed. Port numbers and SPI values are not included in HIP Base Exchange packets for IPv4. This implies that Base Exchange cannot traverse HIP-unaware NATs performing port translation.
- NAT traversal of HIP data packets is similar to other IPsec applications, except in BEET IPSec mode. Modified header fields may cause checksum errors. In an "integrated" NAT, information from the HIP Base Exchange and UPDATE packets is used to determine the IP addresses, HITs, and SPI values.

Security Mechanisms in HIP V

- With this information, required address translations between public and private address spaces, and mapping of several HIP identities to a single external IP address may be performed.
- Firewalls inspect individual packets and decide whether to forward, discard or process them. Firewall middle boxes are not problematic as long as their policy rules permit HIP base exchange and IPsec traffic to traverse.
- Typically, HIP peers are not reachable outside a protected network because Base Exchange attempts from outside are blocked. The largest concern with regard to ESP traffic is the firewall configuration because ESP is not a well-known transport protocol (RFC 5207).

- Credit-Based Authorization:
    - By the use of Credit-Based Authorization (CBA), a host may securely use a new locator before the reachability test has passed, i.e. the address embedded in the locator has not yet been verified.

Security Mechanisms in HIP VI

- To prevent flooding attacks, CBA limits the data a host can send to an unverified peer address. A credit based on the amount of data received from the mobile host may not be exceeded.
- When the reachability status of the peer is verified, the credit limitation will be lifted. HIP hosts deploy a mechanism called credit aging. The mechanism prevents an attacker to build up a large credit by decreasing the credit over time.

Security Attacks I

- In the following, we discuss impersonation and DoS attacks.
- A host's IP address is updated with HIP UPDATE messages and the HIP host cryptographically verifies the sender of an UPDATE message. This makes forging or replaying of a HIP UPDATE message very difficult (RFC 5201 and RFC 5206).
- Impersonation attacks (direct conversion with the victim, man-in-the-middle) are malicious redirection of legitimate connections. This type of attack is possible only if the attacker resides on the routing path or acts as a router.
- HIP introduces an ability to redirect a connection, both before and after a HIP SA establishment. Thus, an attacker could misuse this redirection feature.

Security Attacks II

- The authentication and authorization mechanisms of the HIP Base Exchange [RFC5201], the signatures in the UPDATE message and the secure binding of a HIP SA to a HIP HI/HIT prevent this attack. Furthermore, the original owner can always reclaim a connection.
- Man-in-the-middle attacks are possible if the attacker is present during the initial HIP Base Exchange and if the hosts do not authenticate each other's identities [RFC 5201]. After the opportunistic Base Exchange, even a man-in-the-middle attacker cannot create a legitimate update. UPDATE packets use HMAC and are signed. To forge an UPDATE packet the secret keys must be known.
- A denial-of-service attack aims at the victim's network attachment (flooding attack), its memory, or its processing capacity in such a way that the victim ceases to operate correctly. During a direct flooding attack, unwanted data packets fill the available bandwidth of the victim (node or entire network).

Security Attacks III

- In a DDoS (Distributed Denial of Service) attack, viral software is distributed to as many nodes as possibly, and the infected nodes (zombies) will start to jointly send packets to the victim. By initiating a large download from a server and redirecting the download to the victim a DDoS attack can be realized without distributing viral code.
- A combination of reachability checks and credit-based authorization lowers a HIP redirection-based flooding attack to the level of a direct flooding attack (RFC 5206).
- In a simple memory-exhaustion DoS attack, when many UPDATE packets are sent, the number of not preferred flagged IP addresses associated with the attacker's HI must be limited.
- If a central server in a computational-exhaustion attack has to deal with a large number of mobile clients, it may increase the SA lifetimes to try to slow down the rate of rekeying UPDATEs. It may also increase the cookie difficulty to slow down the rate of attack-oriented connections.

Making Hip Lighter

- Current HIP, especially with certificates, may be too heavy and slow for some applications.
- The current demands support for RSA and DSA, but elliptic curve signatures are more efficient.
- HIP without certificates is vulnerable, but certificate checking adds a lot of computational costs. Therefore, also identity-based cryptography (IBC) have been thought to replace RSA-based cryptography, because this does not demand certificates. On the other hand, the key escrow property of IBC makes it hard to accept IBC globally.
- One suggestion, replacing signature-based authentication with hash chains has aroused more interest.
- We introduce this idea, because it is used in the newest suggestions, too.

Hash chains I

- Leslie Lamport suggested these already 1981.
- In order to define a hash chain, choose first pseudo-random seed value r. Then define

$$\begin{aligned} h_0 &= r, \\ h_n &= H(h_{n-1}). \end{aligned}$$

  The sequence $(h_i, h_{i-1}, h_{i-2}, ..., h_1, r)$ is called the hash chain.
- Its first element h_i is called hash chain anchor.
- In order to use a hash chain, the anchor is published while the the rest of the chain is kept secret. The elements of the chain are disclosed in reverse order of their generation beginning with the anchor value.
- Hash chains can be used for authentication in several ways:

  i) Given h_{i-1} and h_i, it is easy to verify that h_i belongs to the same hash chain as h_{i-1}.

Hash chains II

ii) It is computationally hard to find h_{i-1} if only h_i is given. This ensures that someone who is not in possession of the hash chain cannot calculate unrevealed elements from already disclosed ones.

iii) Given h_{i+1}, it is hard to find an h' with $H(h') = H(h_i) = h_{i+1}$. This ensures that nobody can forge a single unrevealed element of the hash chain.

iv) Given h_i, it is possible to verify that h_{i-k} is part of the same hash chain, if $0 < \leq i$. This ensures that hash chain values can be verified even when some values get lost due to transmission errors or attacks.

- If the sender sends h_i with a message m_1, h_{i-1} with m_2, h_{i-2} with m_3 and h_{i-3} when he is sure that all the messages have already been received, the receiver can be sure that all the messages come from the same source by calculating the hash values from h_{i-3}.

HIP without public keys

Without PKI, HIP cannot

- authenticate a host's identity,
- encrypt payload (because not even DH values are exchanged),
- protect against ma-in-the -middle attack during the base exchange.

But with hash chains it can

- authenticate succeeding messages,
- protect the integrity of the control messages,
- protect against man-in-the-middle after the base exchange.

HIP with hash chains I

- In the Lightweight HIP suggestions, hash chains are used interactively.
- Assume that Sender has hash chain $\{h_i^s\}$ and Receiver has a chain $\{h_i^v\}$. Sender sends a message with a hash signature and Receiver verifies the signature.
- The follwing diagram shows how hash chains are applied in HIP.

HIP with hash chains II

$$\begin{array}{ll}
S \longrightarrow R: & h_i^s,\ HMAC(h_{i-1}^s, \mathrm{message}) \\
R \longrightarrow S: & h_i^v,\ H(c_{ack} || secret_{ack} || h_{i-1}^v),\ H(c_{nack} || secret_{nack} || h_{i-1}^v) \\
S \longrightarrow R: & h_{i-1}^s,\ \text{message} \\
R \longrightarrow S: & h_{i-1}^v,\ secret_{ack} / secret_{nack}
\end{array}$$

- R generates two acknowledgements after it receives the first message, so called presignature. At this point it is unclear whether the verification of the signature succeedes or not. There acknowledgements for both.
- It uses a constant number c_{ack} and corresponding constant nack as messages and creates pre-signatures with the next undisclosed hash chain element.
- In order to let S to know, if the acknowledgement is ack or nack, R adds a secret number in the signatures.
- S sends the hash chain value of the presignature with the message so that R can verify the presignature.
- R sends the corresponding hash chain value plus the secret which shows whether the acknowledgement was ack or nack. R must choose new secret values for every signature.

Base Exchange in LHIP

- With I1 packet, I sends the presignature of the I2 packet.
- With R1 packet, R sends the presignature of the R2 packet.
- With I2, I sends the necessary anchors for the verification of the signature in I1 packet.
- With R2, R sends anchors for the verification of the signature in R1 packet.