Data Plane Security - Notes

EP2500/3200: Networked Systems Security

Panos Papadimitratos
Networked Systems Security Group
www.ee.kth.se/nss

November 25, 2015

Contents

1 Introduction (slides: 3-10)

The forwarding plane, sometimes called the data plane, defines the part of the router architecture that decides what to do with packets arriving on an inbound interface. Routers perform two basic functions:

- Forward packets between interfaces (switching/forwarding).
- Exchange reachability information between routers (route discovery).

Conceptually, switching (packet forwarding) is performed in the data (forwarding) plane. Routing (exchange of routing information) is performed in the control plane. The information collected with routing protocols is used to build topology databases. Then, each router knows how it should forward packets.

In addition to forwarding, there are some other tasks that a router performs in the data plane:

Drop packets due to congestion, security (filtering), Quality of Service (QoS)

Delay packets for QoS

Transform Packets for encapsulation, tunneling

In the secure routing module, we investigated what happens when routers exhibit faulty behavior in terms of their control plane functionality.

In this module, the question at hand is: What happens if a malicious router abuses functionality relating to the data plane? *Data Plane Security* deals with such situations and seeking to ensure correct reception of data at their respective destination. While Routing Security tries to find correct routes in a network in the presence of adversaries (attackers). It is possible for a router to insert, drop, or modify a packet; thus, an adversary who takes control of a router can mount a *Denial of Service (DoS)* attack against any flow of data across the given the router.

Recall the distinction between a DoS attack and the *clogging DoS* attacks we saw in module 2: the outcome is the same, the mean differ; a clogging DoS is a special type of DoS that exhausts resources of the victim. There are many attacks against route discovery and data forward that are not clogging DoS; although it is possible to use some characteristics of some types of routing protocols to do that too.

Recall that an attack against route discovery can either be outright a DoS (e.g., the data traverse part of a network and are lost over an incorrect/inexistent route), or it can influence the route selection/calculation, so that the attacker attracts data flows, i.e., finds itself as part of many routes; then, once traffic is intercepted, the DoS can be mounted against data. Less important, perhaps, if there is end-to-end encryption, is the interception of sensitive information from the attracted flows.

Reference: "Routing: Data vs Control Plane" by Jeffrey Shafer.

Differences of classic Internet and ad hoc networks: Recall that the route discovery provides one (or more) route, enabling any host to communicate with any other host (or gateway). In the case of Internet, the routing functionality is undertaken by a set of dedicated machines. Their topology discovery, or essentially, route (communication path) discovery reflects in some cases the quality of the route, e.g., its reliability. But any failure, which can be detected in an end-to-end manner (e.g., by detecting successive TCP time-outs) does not imply that any route change. In fact, the end-host has essentially no control on the choice of the route. One can imagine an unwanted situation, with a compromised switch/router/gateway tampering with en route data and the end host(s) experiencing consistent loss - without the routing protocol choosing an alternative route that does not include the wrong-doer. For classical Internet, reacting to persistent loss of data, other than contacting the network operations group, or tracing the dropping of packets, might be doable to leverage some route diversity, as that provided by SCTP (caution: only in the case of multi-homed networks/hosts). Recall the original Internet failure model: packets are dropped mainly due to congestion (queue/buffer overflow at some router or the destination).

On the other hand, in ad hoc networks, the end hosts (source and destination nodes) can be themselves intermediate nodes: that is, assist other nodes in discovering paths and, in this case, forward their data packets. There is no dedicated infrastructure for routing (neither for discovery nor for forwarding). This, in spite the additional overhead, esp. in the light of mobility and wireless link failures, has an advantage: the detection of a failing route (for many of the ad hoc routing and data communication protocols) can be followed by an explicit choice (computation) and use of an alternative route that is disjoint or at least differs from the previously deemed compromised (non-operational) route. In other words, delivery (reliable) can leverage such control on the path choice, often at the source nodes.

This difference will become apparent in the two parts of this module (lecture): first, secure data communication for classic Internet is considered, and then the case of ad hoc networking protocols.

2 IPsec (slides: 11-23)

IPsec is a protocol suite at the network layer of the Internet Protocol (IP) Suite, used to encrypt and authenticate IP packets. There are two methods of authentication and/or encryption. The AH can be implemented to authenticate IP packets, while ESP can encrypt packets and optionally offer authentication. The two protocols are typically used independently.

IPSec includes the Internet Key Exchange (IKE) protocols to set up the Security Associations (SAs), which are the established security parameters between the entities to support secure communication.

An SA is basically instantiated by a unidirectional secure connection through a triplet of: (i) a Security Parameter Index (SPI), (ii) an IP destination address, and (iii) an IPsec protocol identifier, "selecting" between AH and ESP, only one security protocol can be used per SA. The SPI is a 32-bit value that distinguishes multiple SAs terminating at the same destination and using the same IPsec protocol. The SA also determines the cryptographic algorithms, the parameters of the related cryptographic keys the two entities utilize.

IPsec can be used in *Transport* or *Tunnel Mode*. In transport mode, the end-to-end secure connections are achieved by encapsulating the IP packet payload. Transport mode is usually deployed from host to host (or host to gateway) within the same network but it can also be across networks. Transport mode requires IPsec support at each host. In tunnel mode the whole IP packet is encapsulated. Sending gateway appends a new IP header to the encapsulated packet .This means, Tunnel mode protects both the original IP header as well as packet payload. This is a suitable technique for Virtual Private Networks (VPNs) to send packets over the insecure Internet. Tunnel mode is usually deployed from gateway to gateway (or host to gateway) owned by the same organization with insecure network in the middle. Tunnel mode (gateway-to-gateway) requires IPsec support only at the Gateways.

2.1 Authentication Header (AH) (slides: 14-19)

When only packet authentication is needed, the *Authentication Header* protocol can be used. It can guarantee the integrity and origin of the packet (No Confidentiality), by computing a hash of the IP header fields. This prevents attackers from capturing packets, (manipulating) and re-injecting them to the wire in order to perform certain attacks. Time To Live (TTL), service type, fragment offset and header checksum are not included in the end-to-end integrity check because they change in transit (mutable fields).

Important authenticated parts of the IP header include:

(A) Next hdr: protocol of the following payload (TCP/UDP payload)

(B) The sequence number of the packets, which is a monotonically increasing number signifying the order at which the packets were transmitted, to protect against relay message attacks by using the sliding window method.

(C) Authentication Data: which is the integrity check value of the entire IP packet. The recipient recomputes the hash value of the IP packet and checks its validity according to this value.

2.1.1 AH in Transport Mode

In transport mode a header (AH) is introduced, accompanying the IP header. The IP header (except for the mutable fields) is authenticated, preventing the adversary from changing any values.

2.1.2 AH in Tunnel Mode

In tunnel mode, the full original IP header is encapsulated, as well as the payload, in new IP packets with new IP headers. This allows the IP source and destination addresses to be different from those of the IP packets encompassing the original one. This technique allows the formation of secure tunnels and finds applications in VPNs. When the destination receives a packets, it goes through an authentication check, stripping off its IP and AH headers. The packet can then be forwarded to its final destination in the private network. The tunnel mode is used between gateways and not for end-to-end connections.

2.2 Encapsulating Security Payload (ESP) (slides: 20-22)

The ESP encrypts the packet and optionally provides authentication. The main difference from AH, is that the payload is surrounded (encrypted) rather than preceded with an authentication header as in the AH protocol. By surrounding the payload, ESP protects it and thus, offers additional security. In Transport mode, the original IP header is not included in the encryption or authentication. In Tunnel mode the original IP header is included in ESP payload, therefore an adversary cannot see the IP addresses of the end hosts communicating.

3 TCP security (slides: 24-27)

The TCP Options are located at the end of the Header. Thanks to the TCP Options field we have been able to enhance the protocol by introducing new features (or *addons*), defined by their respective Request for Comments (RFC)'s[1].

3.1 TCP-MD5

The RFC2385 describes a TCP extension to enhance security for Border Gateway Protocol (BGP). It defines a new TCP option for carrying an Message Digest v5 (MD5) digest in a TCP segment. This digest acts like a signature for that segment, incorporating information known only to the connection end points. Since BGP uses TCP as its transport, using this option in the way described in this paper significantly reduces the danger from certain security attacks on BGP.

Every segment sent on a TCP connection to be protected against spoofing will contain the 16-byte MD5 digest produced by applying the MD5 algorithm to these items in the following order:

1. the TCP pseudo-header (in the order: source IP address, destination IP address, zero-padded protocol number, and segment length)
2. the TCP header, excluding options, and assuming a checksum of zero
3. the TCP segment data (if any)
4. an independently-specified key or password, known to both TCPs and presumably connection-specific

Upon receiving a signed segment, the receiver must validate it by calculating its own digest from the same data (using its own key) and comparing the two digest. A failing comparison must result in the segment being dropped and must not produce any response back to the sender.

TCPMD5's use of a simple keyed hash for authentication is problematic because there have been escalating attacks on the algorithm itself. It also lacks both key-management and algorithm agility.

3.2 TCP Authentication Option (AO)

The RFC5925 specifies the TCP AO, which obsoletes the TCP MD5 Signature option of RFC 2385. TCP-AO specifies the use of stronger Message Integrity Codes (MICs), protects against replays even for long-lived TCP connections, and provides more details on the association of security with TCP connections than TCP MD5.

[1] A full list is available at iana.org

This new option supports the use of other, stronger hash functions, provides replay protection for long-lived connections and across repeated instances of a single connection, coordinates key changes between endpoints, and provides a more explicit recommendation for external key management. The result is compatible with IPv6, and is fully compatible with proposed requirements for a replacement for TCP MD5.

4 Stream Control Transmission Protocol (SCTP) (slides: 28-34)

The SCTP (IETF RFC 2960) is a Transport Layer protocol, serving in a similar role to the popular protocols Transmission TCP and User Datagram Protocol (UDP). It provides some of the same service features of both: it is message-oriented like UDP and ensures reliable, in-sequence transport of messages with congestion control like TCP.

SCTP applications submit their data to be transmitted in messages (groups of bytes) to the SCTP transport layer. SCTP places messages and control information into separate chunks (data chunks and control chunks), each identified by a chunk header. A message can be fragmented over a number of data chunks, but each data chunk contains data from only one user message. SCTP chunks are bundled into SCTP packets. The SCTP packet, which is submitted to the Internet Protocol, consists of a packet header, SCTP control chunks when necessary, followed by SCTP data chunks when available.

SCTP may be characterized as message-oriented, meaning it transports a sequence of messages (each being a group of bytes), rather than transporting an unbroken stream of bytes as does TCP. As in UDP, in SCTP a sender sends a message in one operation, and that exact message is passed to the receiving application process in one operation. In contrast, TCP is stream-oriented protocol, transporting streams of bytes reliably and in order. However TCP does not allow the receiver to know how many times the sender application called on the TCP transport passing it groups of bytes to be sent out. At the sender TCP effectively simply appends more bytes to a queue of bytes waiting to go out over the network, rather than having to keep a queue of individual separate outbound messages which must be preserved as such.

The term **multi-streaming** refers to the capability of SCTP to transmit several independent streams of chunks in parallel, for example transmitting web page images together with the web page text. In essence, it is the bundling of several connections into a single SCTP *association*, operating on messages (or chunks) rather than bytes.

TCP preserves byte order in the stream by assigning a sequence number to each packet. SCTP, on the other hand, assigns a sequence number to each message sent in a stream. This allows independent ordering of messages in different streams. However, message ordering is optional in SCTP; a receiving application may choose to process messages in the order they are received instead of the order they were sent.

Although encryption was not part of the original SCTP design, SCTP was designed with features for improved security, such as 4-way handshake (compared to TCP 3-way handshake) to protect against SYN flooding attacks (DDoS module), and large "cookies" for association verification and authenticity.

Reliability was also a key aspect of the security design of SCTP. It supports multi-Homing that means multiple IP addresses per host is supported. This enables an association to stay open even when some routes and interfaces are down.

5 Data Communication (slides: 31-34)

Consider an adversary who wants to attack a network. First, he tries to be a part of a route in the routing phase, i.e. he pretends that he has the best way (the shortest path in hops, delay, or any other metric) to the desired destination. Also, it is possible for him to mount other type of attacks against routing protocol. After that, he can decide what to do with the packets. He may drop some packets, modify some packets or inject some packets into the network.

Note that even if we secure the routing protocol, ensuring loop-free, fresh, and accurate routes against any possible attack, it is possible for an adversary to be a part of network and to mount such type of attacks. Securing the routing protocols only decreases the chance of adversary to place in a path. Authenticating all nodes in a path does not remove the risk totally either.

5.1 Attacks (slides: 35-42)

Blackhole attack : The adversary is part of the route and drops *all* the packets. In some cases, called sinkhole; inspired by wireless sensor networks for which nodes try to send data to a special node, e.g., a base station/sink, and thus having one node getting all traffic is an acceptable pattern of communication.

Grayhole attack : The adversary is part of the route and drops some packets. A smarter version of the previous attack, where the adversary discards only a portion of in-transit packets. Also called Selective Forwarding. If all packets are dropped, the attack detection likelihood, e.g., by the end-nodes increases.

Wormhole attack : Wormhole is a physical layer tunnel in a network that allows signals from nodes near its one end to travel faster than normal (than what they would across the legitimate network) to nodes near its other end. This is similar to the definition of a wormhole in space which allows faster space travel. A wormhole attack is an attack done using one or more wormholes in a network. A successful attack may result in a disruption or breakdown of a network: again, any route that includes a wormhole would look more attractive than actual routes.

Securing data communication comprises methods that guarantee reliable and low-latency delivery of data to the desired destination. One method is to detect and avoid using compromised and failing paths. But, this is difficult in a highly dynamic network system. Another approach is to use multiple routes. The sender can divide the message into some pieces and forward pieces via different paths in a way that if some of the pieces lost, the reconstruction of the original message would be possible yet.

6 Secure Single Path (SSP) (slides: 43)

Secure Single Path (SSP) is a lightweight secure data delivery protocol. It relies on end-to-end security association. It operates on top of a secure route discovery protocol (SRP, ARIADNE etc). SSP aims at making data delivery reliable. It has a route rating mechanism which increases the rating of a route when data sent on that route is ACKed by destination. SSP abandons a non reliable route when its rating drops below a threshold; choosing another route or invoking new route-discovery in case no reliable route exists. This simple mechanism makes SSP robust to any attack that causes packets to be dropped.

7 Secure Message Transmission (SMT) (slides: 43-49)

Secure Message Transmission (SMT) is a lightweight, yet very effective, protocol that can operate solely in an end-to-end manner. It exploits the redundancy of multi-path routing and adapts its operation to remain efficient and effective even in highly adverse environments.

The SMT protocol safeguards pair-wise communication across an unknown frequently changing network, possibly in the presence of adversaries. It combines four elements: end-to-end secure and robust feedback mechanism, dispersion of the transmitted data, simultaneous usage of multiple paths, and adaptation to the network changing conditions. Its goal is to promptly detect and tolerate compromised transmissions, while adapting its operation to provide secure data forwarding with low delays.

SMT requires a Security Association (SA) only between the two end communicating nodes, the source and the destination. Since a pair of nodes chooses to employ a secure communication scheme, their ability to authenticate each other is indispensable.

With SMT, at any particular time, the two communicating end nodes make use of a set of diverse, preferably node disjoint paths that are deemed valid at that time.

With a set of routes at hand, the source disperses each outgoing message into a number of pieces. At the source, the dispersion introduces redundancy and encodes the outgoing messages. At the destination, a dispersed message is successfully reconstructed, provided that sufficiently many pieces are received. In other words, the message dispersion ensures successful reception even if a fraction of the message pieces is lost or corrupted, either due to the existence of malicious nodes, or due to the unavailability of routes (e.g., breakage of a route as a result of nodes' mobility). Each dispersed piece is transmitted across a different route and carries a MIC, so that the destination can verify its integrity and the authenticity of its origin. The destination validates the incoming pieces and acknowledges the successfully received ones through a feedback back to the source.

The feedback mechanism is also secure and fault tolerant: It is cryptographically protected and dispersed as well. This way, the source receives authentic feedback that explicitly specifies the pieces that were received by the destination. A successfully received piece implies that the corresponding route is operational, while a failure is a strong indication that the route is either broken or compromised.

For extra reading, please see: Secure message transmission in mobile ad hoc networks.

8 CASTOR (slides: 50-51)

Continuously Adapting Secure Topology-Oblivious Routing (CASTOR) is a routing protocol that addresses simultaneously three aspects: security, scalability and adaptability to changing network conditions. It does not use any control messages except simple packet acknowledgements, and each node makes routing decisions locally and independently without exchanging any routing state with other nodes. Its novel design makes CASTOR resilient to a wide range of attacks and allows the protocol to scale to large network sizes and to remain efficient under high mobility. CASTOR achieves a high packet delivery rate. Besides, it is able to survive more severe attacks and recovers from them very fast.

CASTOR uses End-to-End (EtE) and Neighbor-to-Neighbor (NtN) security associations and has EtE and NtN authentication only.

There is a local learning mechanism from failures (of any type) and each node has an estimation of its communication link to each of its neighbors. This estimation is different for each flow of data. Using such estimation, each node unicasts each packet to the best neighbor. In a few cases and for a few nodes, a broadcast message is required. Broadcasts occur when there is no reliability history and no clear choice among neighbors.

Each node estimates each of its neighbors per flow by sending the packets and waiting for the acknowledge. The ACK follow the reverse path of corresponding PKT. If PKT was broadcast, the corresponding ACK is broadcast too. A failure detects when there is no valid ACK for a PKT.

For extra reading, please see: Castor: Scalable Secure Routing for Ad Hoc Networks.