CS261 Lecture 2007-10-09

Network Security

Next two weeks – focus on network security.

Network security is a funny topic. Two kinds of network attacks:

1. Attacks that use the network as a conduit, e.g. traverse the network to exploit a buffer overflow. This is not really an attack on the network, since it is analogous to using a highway to rob a bank – not an attack on the highway.

2. Focus on attacks on the network infrastructure, i.e. misuse of network features to attack other users.

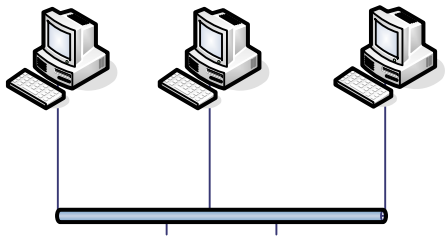
The end goal of network attacks is not to "own" the network, but to attack users using the network, e.g. commit highway robbery, instead of robbing the asphalt.

We will look at several attacks on the network.

1. Ethernet LAN Attacks

1.1. Shared Ethernet

A, B, C connected to shared medium. A communicates to B. C is another host.



A sends a packet to B. Ethernet datagram as follow.

src MAC addr	dst MAC addr	Data payload

C could

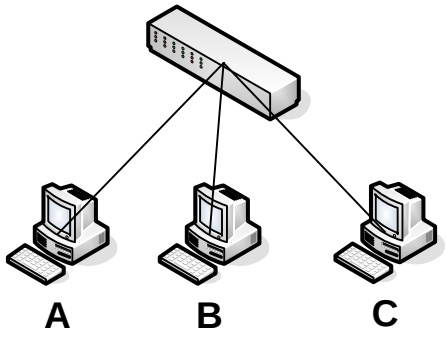
Eavesdrop	by	Receiving and examining packets intended for B
Do denial of service (DoS) attack	by	Keep on sending packets and disrespecting Ethernet random back off after collision
Spoof packets	by	Injecting packets with src other than C. Possible because Ethernet protocol has no in-built authentication.
Replay packets	by	Saving packets seen on the wire, and resend them later
Delete packets	by	Look at packets on the wire, transmit to cause collision and corrupt last few bits of packet
Replace packets	by	Similar to delete, just add resend packets with replaced content

Another example: The “Ping of Death Attack”

C does permanent DoS on A by sending the “ping of death”. Ping of death exploits vulnerability in the IP stack. There is implementation error that results in integer overflow on the “length” field for a ping. After permanent DoS, C can spoof data from A to B.

1.2. Switched Ethernet

A, B, and C connected to common switch.



Topology used for better performance. When A sends a lot, B is not prevented from sending – unlike the shared Ethernet, when packets from A and B would cause congestion on the shared medium.

Switch forwards packets to ports based on the MAC address associated with each port and the MAC address in the Ethernet datagram headers.

ARP Review

ARP = Address Resolution Protocol

ARP allows A to resolve B's MAC address based on B's IP address.

Sample exchange:

A broadcasts ARP request to Ethernet broadcast address.
B receives ARP request, and sends ARP reply unicast to A with his own MAC address.

Eavesdropping through ARP poisoning

A broadcasts ARP request to Ethernet broadcast address.
B receives ARP request, and sends ARP reply unicast to A with his own MAC address.
C also sends ARP reply to A saying his own MAC address correspond to B.

Race condition between B and C – whichever reply gets to A first will be in effect.

If ARP reply from C gets to A first, A will think B's IP address is located at C's MAC address. Thereafter, whenever A sends packets to B, A will use C's MAC address in Ethernet header, and the switch will forward the packet to C.

ARP does not have built-in mechanism to prevent this attack.

Eavesdropping through ARP flooding

Again, the idea is to spoof MAC address to confuse the switch in forwarding packets

Attacker sends lots of ARP request and fake ARP replies onto the network.
Switch sees all the request and replies.
Switch caches all ARP replies seen to help with datagram forwarding.
ARP request and reply flood causes switch's ARP cache entries to be evicted.
Attacker can now fake new ARP request/reply pair for new entry in ARP cache.

Switch will now forward packets to the attacker based on ARP entries s/he supplied.

Defense against ARP attacks

Have fixed association of port with MAC address – i.e. only this particular computer will be allowed to plug into this particular port.

Any attacks would only have originated from users of the network – network administrators can walk over and physically smack the offending users.

This defense is in place for the EECS network – need to look for port number on the wall when registering computer to connect to wired Ethernet.

2. IP Vulnerabilities

Hosts A connects to B through the network cloud.

Packets from A destined for B is routed through routers in the network cloud.

Routing is based on the dst address in the IP datagram.

src IP addr	dst IP addr	Other hdr stuff	Ethernet datagram

Forgery

Anyone can do it.

Only requires appropriate src and dst address fields in IP datagram.

There is movement to convince ISP to do egress filtering, i.e. only allow packets with certain IP src addresses to enter networks from certain entry points. Drawbacks include administration overhead on both CPU cost and human costs, as well as the lack of economic incentive for ISP to do extra work to increase security for other ISP. Egress filtering also conflicts with the need to have more than one ISP.

Eavesdrop

Anyone along the path of a packet can eavesdrop. Anyone on the router's shared network can also eavesdrop.

Delete/replace packets

Possible if packets are fragmented. Attacker on the path of the packet can learn fragment ID and timing. Afterwards attacker can inject packets with appropriate fragment ID at the appropriate time.

3. Transport Protocol Vulnerabilities

3.1. UDP

UDP datagram has additional headers outside the IP datagram.

src port	dst port	Other hdr stuff	IP datagram

UDP provides no reliability, with no ACK for packets received.

Thus UDP is almost the same as bare IP, except one host can have many UDP ports.

Security analysis is the same as that for IP.

3.2. TCP

TCP datagram is very similar to UDP datagram.

src port	dst port	Other hdr stuff	IP datagram

TCP provides reliable (lossless) data transfer between two hosts.

Eavesdropping

Can do on path, cannot do off path.

Forgery

Forge a new connection – Off path

Can do off path if ISN (initial sequence number) is guessable. This is blind forgery. Any TCP ACK from B to A will not be seen by the attacker. Attacker need to also do DoS on A to stop A from telling B to tear down unsolicited connection.

e.g. rlogin vulnerability: Login with host, username, use IP to authenticate. If ISN is guessed, attacker can spoof host, user, and run destructive command with privileges of the compromised user.

Forge a new connection – On path

No need to guess ISN. Look at ISN in a real connection initialization packet, discard real packet, and send on spoofed packet with the ISN seen.

Forge packets in existing connection

Can do off path if we could guess the sequence number.

Can do on path if attacker controls a router. Analogous to forging a new connection, no need to guess the sequence number, just look at the sequence number in a real packet.

An attacker on the same subnet as the router can also forge packets. The attacker can see the sequence numbers, and so can construct packets with the correct sequence number. There would be race condition between the good and the bad data. The attack would be implementation dependent – should TCP take the first version of the data that arrives and discard the other? Or take the most recently arrived data? No implementation presently rejects packets that have the same sequence number but different content, since it is not clear how packets should be rejected. This attack is a semi-blind forgery, since the attacker can reply to the sending host before the receiving host ACK the forged packet.

Example attack

A connects to NFS server. Attacker resides on the same subnet as router.

NFS server sends packet to A with sequence number 17.
The next sequence number is 18.
Attacker spoofs NFS server packet to A with sequence number 18, length 26.
A expects next sequence number is 44.
A thinks all the subsequent packets with sequence number 18 are duplicates.
ACK war results between A and NFS server (by accident in protocol design).

This is just a complicated form on DoS.

Modification:

Attacker spoofs NFS server packet to A with sequence number 18, length 100,000.
Now NFS real data is out of the range of the window – no ACK storm.
Not attacker sends data to A using sequence number 100,018 onwards.
Attacker also ACK to NFS server.
Now NFS server this A received good data.

Attacker has placed himself/herself between A and the NFS.

4. Summary

Most problems arises from

Lack of authentication, or

Unclear authorization.

4.1. Vulnerable application architectures

FTP and HTTP all vulnerable.

SSH and SSL encrypted so not as vulnerable.

Past security architecture SKey – one time password, carried around in pocket, used once then crossed out. Usability not so great …

4.2. Threat models

Crypto threat model: Attacker IS the network. Too conservative, no distinction between a broken application and a maliciously hijacked router.

More common model: Attacker can take over a host, but not a router.

Trick possible in the more common model: Server send cookie to client, and clients echo back cookies to identify themselves.

4.3. Take away lessons

1. Address and host ID are insufficient for authentication – they only hint about the best way to reach people.

2. The world changes *a lot*. Most serious threats today do not exist 10 years ago – spam, phishing, etc.

As the world change, threat models change.

e.g. Once upon a time, /etc/hosts file *was* DNS. DNS was not a threat back then, now it is a threat – DNS can be attacked.

Security requirements creep up …