Exercises Lectures 1-4

Basic Network Security Terminology and Network Attacks

(Solutions)

NOTE: The solutions given above are sketches of the full solutions, including pointers to appropriate sections in the lecture slides. The full answer in the exam would need to expand on these sketches. However, the solution sketches given are examples of the main points that the expanded solution would need to mention, and these are the points we would be looking for, in scoring the solutions.

Basic Network Security Terminology

1. What is the difference between security policy and security mechanism

Security realised through Policy + Mechanism Policy codifies “desired” behaviour.
Mechanism is the means of enforcing policy.

2. Name 3 different kinds of security policies
Confidentiality: protection of content (information) from unauthorised parties Integrity: protection of content from modification by unauthorised parties
Availability: prevent deliberate overload; keep system/resource usable by legit users

3. Explain the following kinds of mechanisms: deter, deny, detect, delay and defend

Deter = make it “too difficult” or “not worthwhile” to attack

Detect = monitor for attacks
Deny = Prevent unauthorised access
Delay = slow down users (more suited for physical security)

Defend = Take remedial steps after attack

4. Explain the following kinds of attacks: jamming, spoofing, hijacking, sniffing and poisoning.

Jamming involves affecting availability for legitimate packets by ‘talking’ too much.
Spoofing is pretending to be someone else on the Internet.
Hijacking is taking over a (network) resource which belongs to someone else.

Sniffing involves listening to network conversation by parties who are not the intended recipients.

Poisoning is corrupting a store of information (e.g., a cache) that is relied upon for future communications.

5. Describe the functions of the transport, network and link layers in the network stack.

Link layer is responsible for forwarding packets within a single network.
Network layer routes packets across different networks.
Transport layer provides end-to-end connectivity, correcting errors, lost packets, and potentially reordering packets to maintain relative order of packets.

Network Attacks

1. Briefly explain how smurf attacks work

Smurfs are little blue cartoon creatures, some of whom may play practical jokes on others. A smurf attack is one, where a computer S can get all other computers in the network to send an IP packet to an intended victim computer V . S can do this by sending an ICMP Ping request to the broadcast address of the network. All computers think it is destined for them because it uses the broadcast address. Ping protocol requires a response to be sent back. If S spoofs V 's address in the source field, these Ping responses get sent to V – a smurf attack.

2. Explain the function of ARP and RARP, why ARP results are cached, and show how this can be used to spoof other computers.

ARP or Address Resoution Protocol is used to map from a logical address such as IP address to the physical address or Ethernet Address. Reverse ARP is the reverse operation.

ARP Functions as follows:
- Get IP address of target.
- Create a request ARP message including:
    - Fill sender physical address
    - Fill sender IP address
    - Fill target IP address
    - Target physical address is filled with 0
- The ARP message is passed to data link layer where it is encapsulated in an Ethernet frame with:
    - Source address: physical MAC address of the sender.
    - Destination address: broadcast address.
- Every host or router on the LAN receives the frame.
    - All stations pass it to ARP.
    - All machines except the one targeted drop the packet.
- Target machine replies with ARP message that contains its physical address.
    - A unicast message (with Destination MAC address=previous Source address)
- Sender receives the reply message and knows physical address of the target

(RARP is the reverse of the above, since the source knows its physical address and wants to know what logical address to use within the network.)

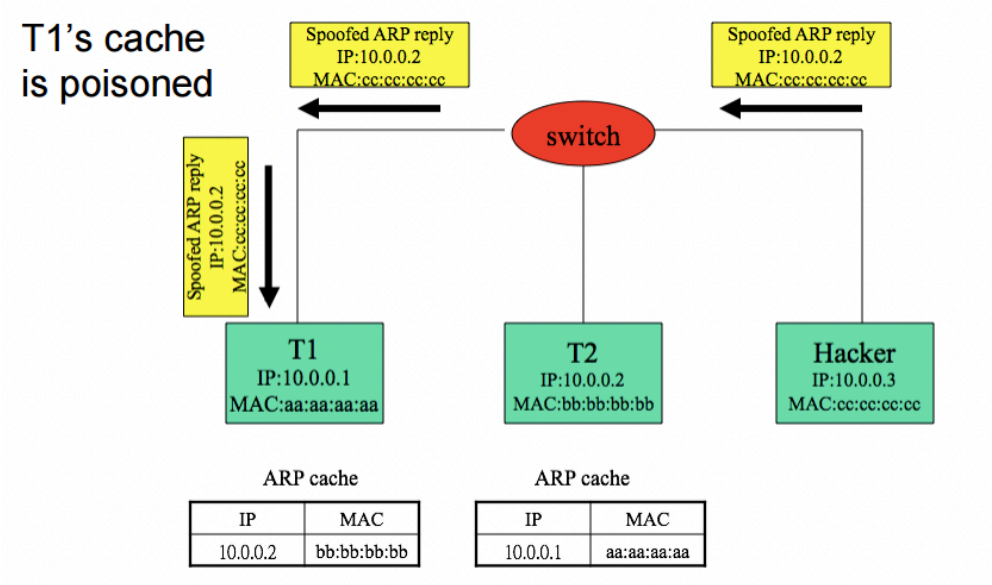
Need for ARP cache: To avoid having to send an ARP request packet each time, a host can cache the IP and the corresponding host addresses in its ARP table (ARP cache).When host receives ARP reply, it updates its ARP cache ARP is a stateless protocol, so most operating

systems will update their cache if a reply is received, regardless of whether they have sent out an actual request.

To spoof other computers, the Hacker needs to poison the ARP cache of computers who want to correspond with the victim's computer. This is done as follows:
- Construct spoofed ARP replies.

- A target computer could be convinced to send frames destined for computer A to instead go to computer B.

- A will have no idea that this redirection took place.
This process of updating a target computer's ARP cache is referred to as "ARP poisoning" and is depicted in the figure below.



3. What are SYN flood attacks? How can IP spoofing be used to make them more effective?

A SYN flood attack exploits how the TCP three-way handshake works by not responding to the server with the expected ACK code once a SYN message is sent. The malicious client can either not send the expected ACK, or more effectively spoof the source IP address in the SYN, causing the server to send the SYN-ACK to a falsified IP address - which will not send an ACK because it "knows" that it never sent a SYN. The server will wait for the missing

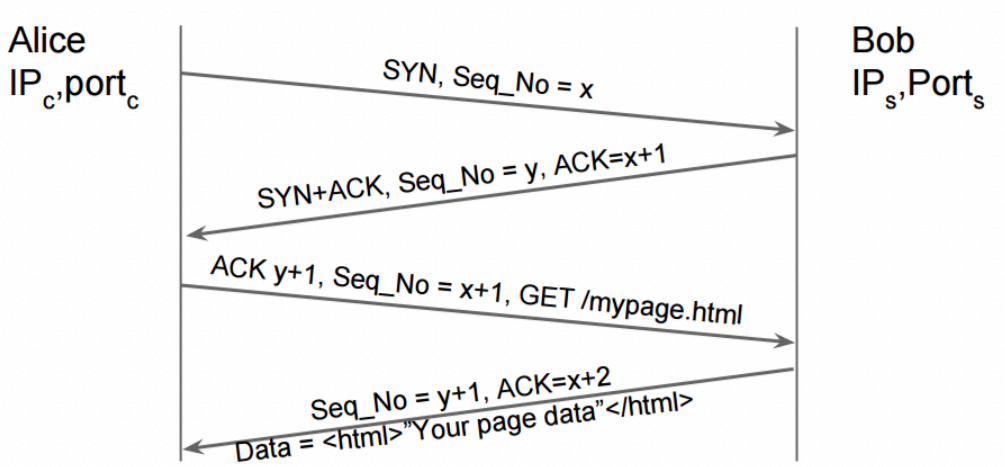
ACK for a bit. However, the resources bound on the server may eventually exceed the resources available and the server cannot connect to any clients.

4. Discuss the advantages and limitations of on-path adversaries over off-path adversaries

On-path adversaries are more powerful. On-path adversary can use connection state since it can intercept connection. On-path adversary can use connection state since it can intercept connection. However, this is difficult to do at scale! Inserting false data is very difficult if off-path, as you cannot see any ongoing connection and you need to guess mostly blindly important parameters of a connection, including IP address of the target and TCP sequence numbers.

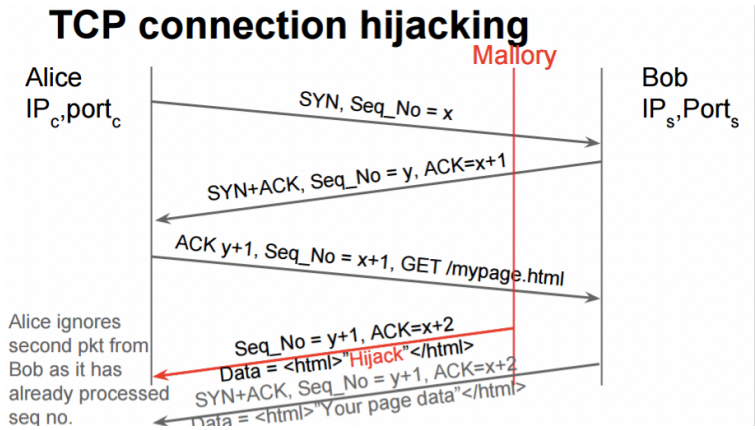
5. Explain how TCP connections are setup through a 3-way handshake, and discuss TCP session hijacking by an on-path adversary.

TCP connection set-up involves sending SYN, getting a SYN+ACK and sending back an ACK, as shown in diagram below:



An on-path adversary can hijack this connection as follows:
1. Sniff packets
2. Predict seq. no (client → server / server → client)
3. Inject data

To spoof client: authentication may happen at the beginning of connection. By hijacking the connection after authentication, Mallory could leave a record of bytes as an “Authenticated” user - e.g. HTTP POST /transfer-money £100,000
To spoof server: Insert false data from server to client (attack as shown in timing diagram above)



6. Explain BGP sub prefix Hijacking

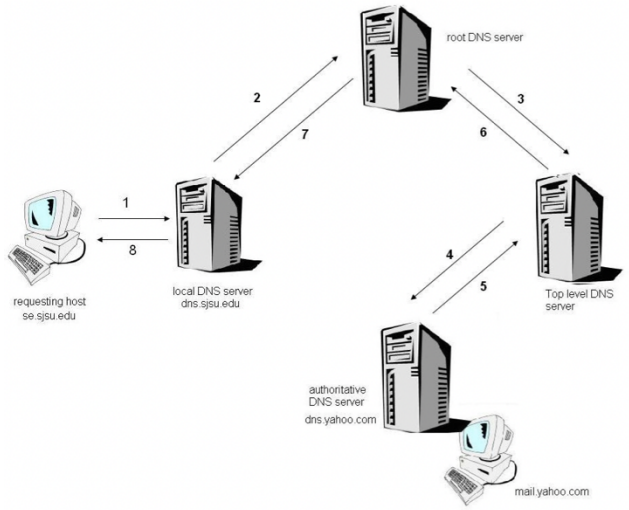
Inter-domain routing happens through the Border Gateway Protocol which controls the routes packets take through Autonomous Systems (ASes) ASes advertise prefixes that they can serve. IP addresses follow a hierarchical scheme. Packets in the Internet are routed in the direction of the longest prefix matching the destination address of the packet. Thus if x.y.z.w/16 is a route advertised on the Internet, and another AS announces a path to the sub-prefix x.y.z.w/24, the latter path would be preferred. Thus, rogue AS can announce paths to sub-prefixes and hijack parts of the Internet, creating routing black holes. E.g. 137.73.0.0/16 is KCL's list of IP addresses and imagine UCL advertised routes to 137.73.0.0/24. Real examples include the day Youtube migrated to Pakistan.

7. Explain how recursive DNS queries work and how this can be exploited to poison caches

There is a need for DNS caching as both recursive and iterative lookups require many steps, which are repeated many times across many lookups (imagine just one user browsing the Internet). In fact, DNS performance is now critical for WWW, because most links are human-friendly DNS names, not IP addresses, and because it adds an extra layer of indirection that enables load-balancing. Therefore, we need to cache DNS queries so that second and subsequent lookups are cheap and easy.

The complete flow of a recursive query is given below (and depicted the Figure below too)
1. Requesting host 'se.sjsu.edu' request its local DNS server 'dns.sjsu.edu' to solve a DNS query 'mail.yahoo.com' and to give its IP address
2. The Local DNS query asks the root DNS server for the IP address of 'mail.yahoo.com'
3. The root DNS server finds the 'com' suffix in the query and request one of the top level DNS server responsible for com
4. The com, top-level DNS server keeps track of the entire authoritative DNS server; it asks the authoritative DNS server of Yahoo (dns.yahoo.com) for the IP address of mail.yahoo.com
5. The authoritative DNS server of Yahoo returns the IP address to the com Top Level DNS server who queries the authoritative DNS
6. The Top level DNS server returns this IP address to root DNS server

7. The root DNS server in turn returns the IP address to the local DNS query.
8. The host receives the IP address of its desired query.



DNS cache poisoning makes use of the fact that DNS uses UDP. Therefore, it is connectionless. An attacker can spoof the real DNS server and if its UDP packet containing a false response beats the real DNS server, it poisons the cache of the resolver.