Network Attacks

Peter Chapin

IP Spoofing

- *Generate IP packets with a source address different than that of the sending machine*.
  - Requires direct access to the link layer (bypass normal network layer)
  - This (normally) requires administrative access, but is usually supported by OS
- Why?
  - Circumvent IP based filtering rules
    - A firewall that filters based on source address can be convinced to pass packets from bad sources that bear a spoofed source address (that is allowed).
  - Defeat IP based authorization
    - A server that only provides services to specific users based on the source address can be convinced to provide a service on behave of an attacker.

IP Spoofing Mitigation

- Techniques for reducing risks
  - Filter out packets with unexpected source address
    - e.g., packets with internal source addresses arriving on an external interface or packets with external source addresses leaving the internal network.
  - Use cryptographic authentication methods (e.g., certificates) rather that IP based methods

IP Spoofing and TCP

- IP Spoofing is hard to use with TCP, but not impossible
- Consider:
  - A TCP service uses IP authorization. Attacker wishes to connect to the service and issue a command that causes trouble. Attacker spoofs their source address so the service authorizes the the attacker.
- Problem:
  - Replies will go back to the spoofed address. Attacker never sees them
    - May not matter for the command. Just issuing the command might be bad enough.
    - But… what about creating the TCP connection it the first place?

IP Spoofing and the Three-Way Handshake

- Normal TCP…
  - Client sends SYN segment with client’s initial sequence number (ISN).
  - Server replies with SYN segment with server’s ISN.
  - Client must ACK the server’s ISN.
- If the client never sees the server’s ISN, how can it ACK it?
  - Guess!
  - Client blind ACKs what it *thinks* the server’s ISN will be.
  - Connection is “established”
  - Client issues dangerous command

IP Spoofing and Guessing the Server ISN

- How to guess the server’s ISN?
  - Make several legitimate connections to the server and note the ISN
  - Based on what the server is doing, make your guess.
    - For example, if the server always uses an ISN of 1, that is your guess
- Servers should…
  - *NEVER* use an ISN of 1 (at least not consistently)
  - Server ISNs should be randomly selected using a cryptographic random number generator (so the next ISN generated isn’t predictable)
  - That takes a lot of time!

IP Spoofing and RST

- Another problem for the attacker…
  - The server's replies are going to the spoofed address which might be a real machine.
  - If that machine is on, it will likely send a TCP RST segment back to the server when it receives unknown TCP traffic. That will cause the server to abandon the connection.
  - Not what the attacker wants!
  - Attacker must spoof an IP address that will have the desired effect of bypassing filtering rules, etc., yet not be the IP address of a real system.

Smurf Attacks

- A class of denial of service (DoS) attack that uses spoofing.
- Concept (by way of example):
  - Attacker broadcasts an ICMP echo request message on a link using the victim’s address as the (spoofed) source address.
  - Every machine on the link sends an ICMP echo reply to the victim, overwhelming the victim.
  - Attacker broadcasts the ICMP echo requests as quickly as possible, generating a huge number of replies to overwhelm the victim.
- Mitigation in this case can be done if systems refuse to respond.
  - But many variations on this; don’t route packets to the broadcast address!

TCP Connection Hijacking

- Concept:
  - Attacker lets legitimate partners connect *and authenticate* normally, observes traffic to learn sequence numbers.
  - Attacker sends an RST segment to one end (say, the client) causing that end to abandon the connection (in effect, the attacker pushes the client aside)
  - Attacker steps in as the client and continues conversing with the server. The server thinks it is still talk to the (authenticated) client.
- Mitigation:
  - Use encrypted and/or authenticated security protocols (e.g., TLS)
  - Note that attacker probably needs to assume the client’s IP address as well which can be difficult in some topological situations.

SYN Flooding

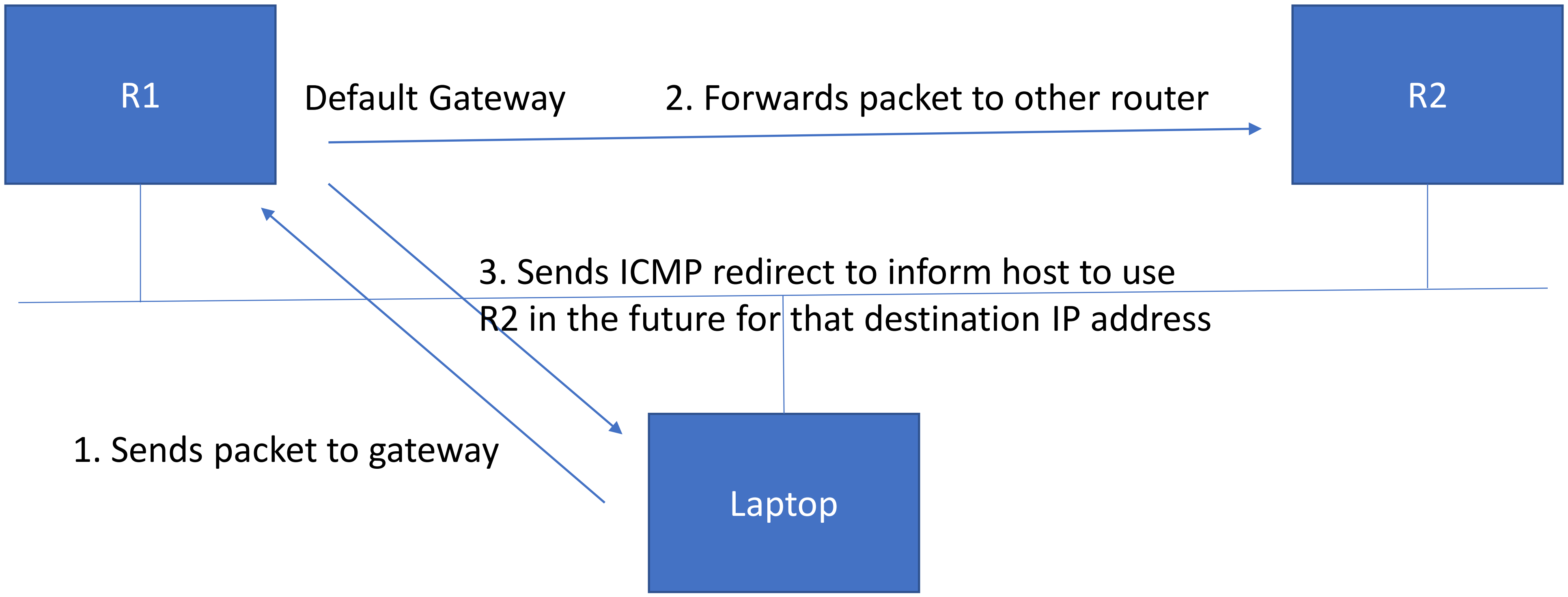
- A DoS attack…
  - Attacker sends a TCP SYN segment to victim
  - Victim replies with SYN-ACK
  - Attacker ignores this… sends a new SYN segment (instead of final ACK)
  - Victim replies again, but builds up a mass of half-open connections
- Once the maximum number of half-open connections is reached…
  - Victim unable to make any more TCP connections!
  - Half-open connections will time-out…
  - … but attacker keeps sending SYN segments
- Tricky to mitigate; how to tell between legitimate traffic and flooding?

SYN Flooding

- One mitigation method is to use “SYN Cookies”
  - The idea: Use the initial sequence number to encode all necessary information that would have been stored in the half-open connection queue.
  - Then… don’t store any half-open connection information (so no queue to flood)
  - If the client ACKs normally, use the acknowledged sequence number to reconstruct the necessary state for the initial connection.
  - See: https://en.wikipedia.org/wiki/SYN_cookies

ICMP Redirects

- How ICMP redirect messages are supposed to work…



R1

Default Gateway

2. Forwards packet to other router

R2

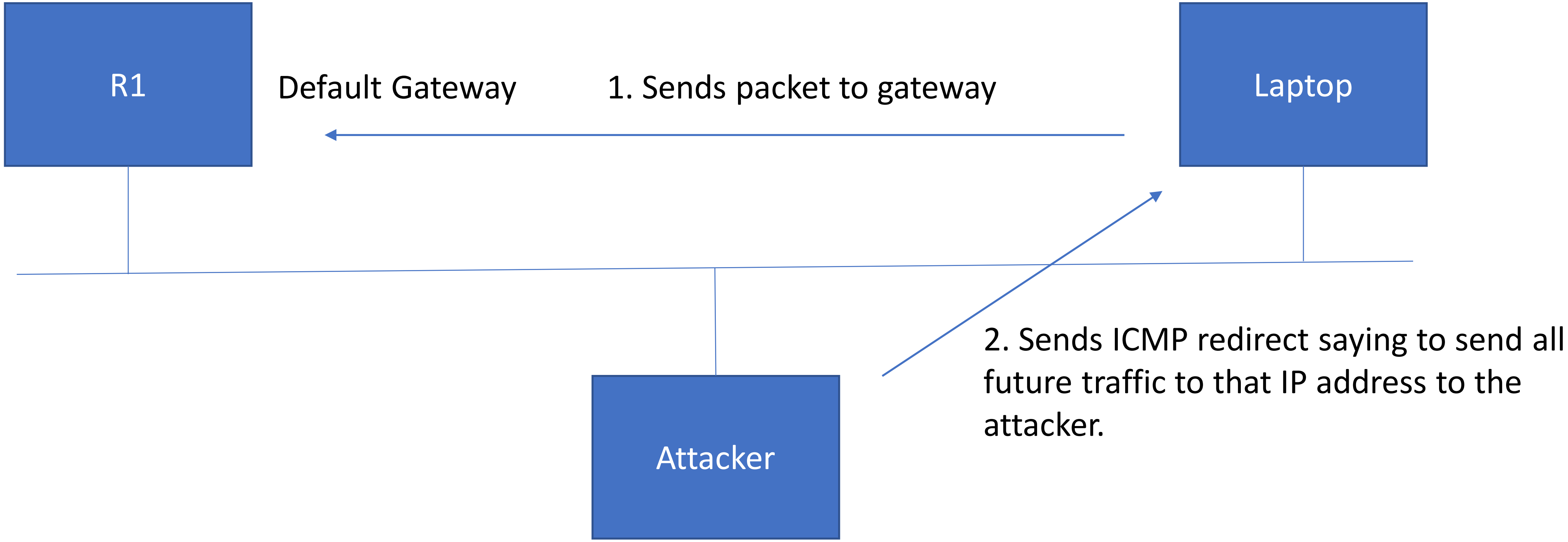
3. Sends ICMP redirect to inform host to use R2 in the future for that destination IP address

1. Sends packet to gateway

Laptop

ICMP Redirects

- How ICMP redirect messages can be abused…



R1
Default Gateway
1. Sends packet to gateway
Laptop
Attacker
2. Sends ICMP redirect saying to send all future traffic to that IP address to the attacker.