Network Security: Attacks

Debdeep Mukhopadhyay
and Mainack Mondal

CS 60065
Autumn 2020

Outline

- Basics of computer networks
    - Or how computers talk to each other
- Basic network attacks
    - Attacking host-to-host data transmission protocols
    - Attacking network infrastructure
    - DDos, smurf attack, reflection attack
- Some mitigations

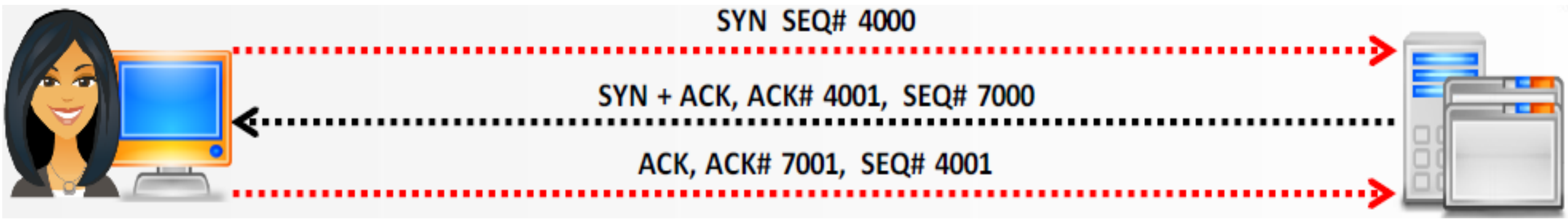
Two rules in network attacker model

Two rules in network attacker model

Two rules in network attacker model

Basics of TCP communication

- TCP 3-way handshake



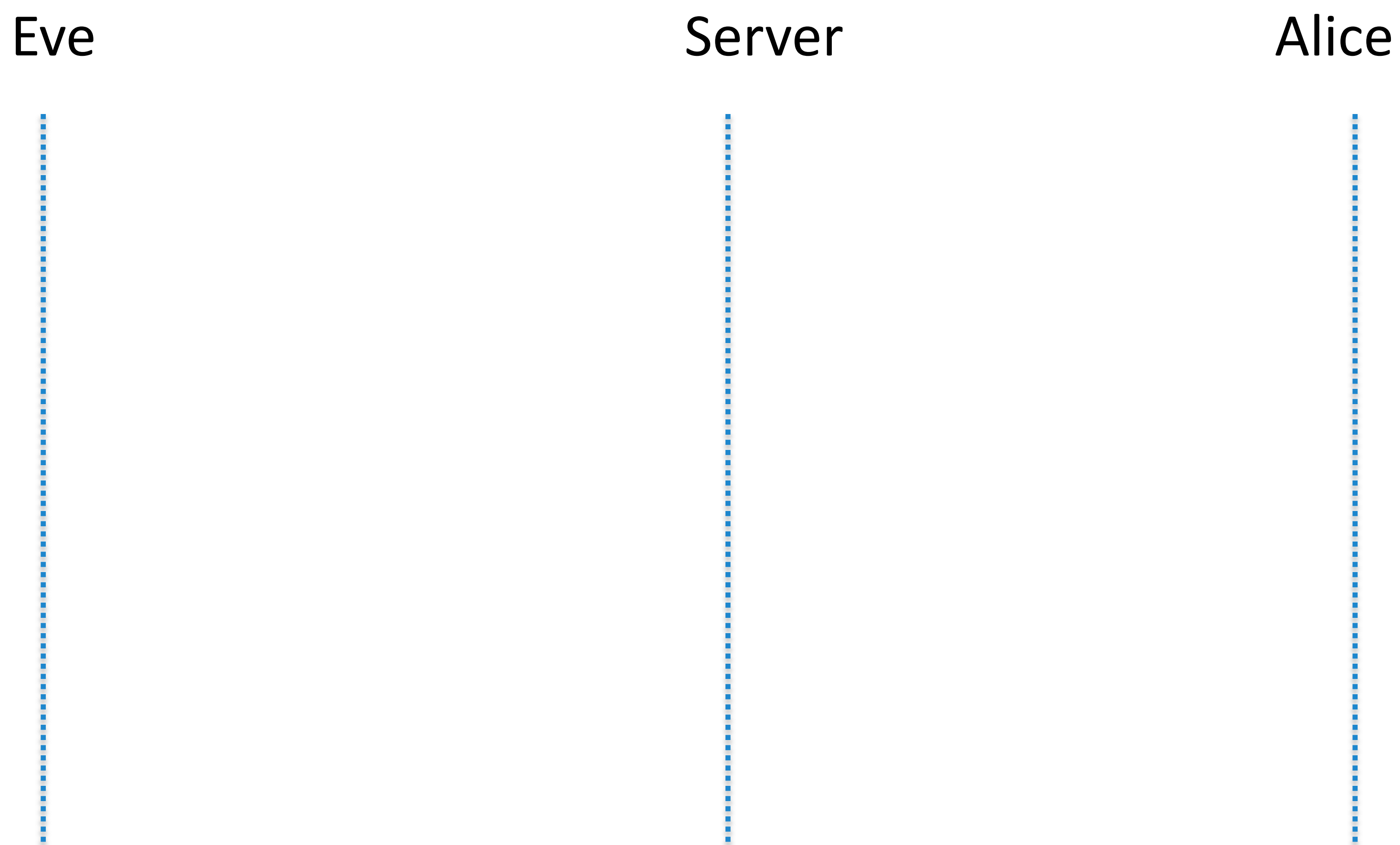
Alice

Server

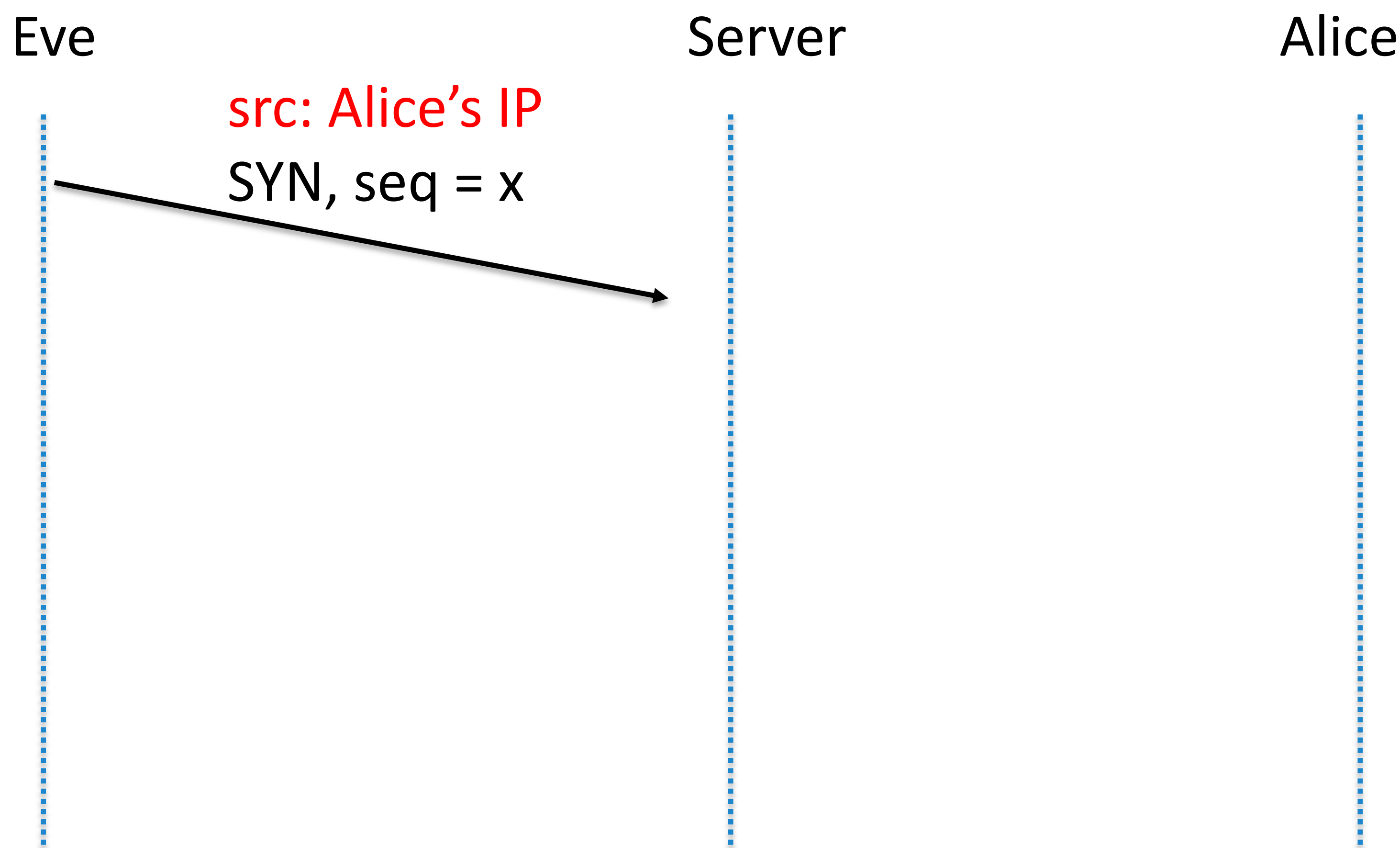
Src: https://ktflash.gitbooks.io/ceh_v9/103_The%203-Way%20Handshake.png

Attacks on TCP

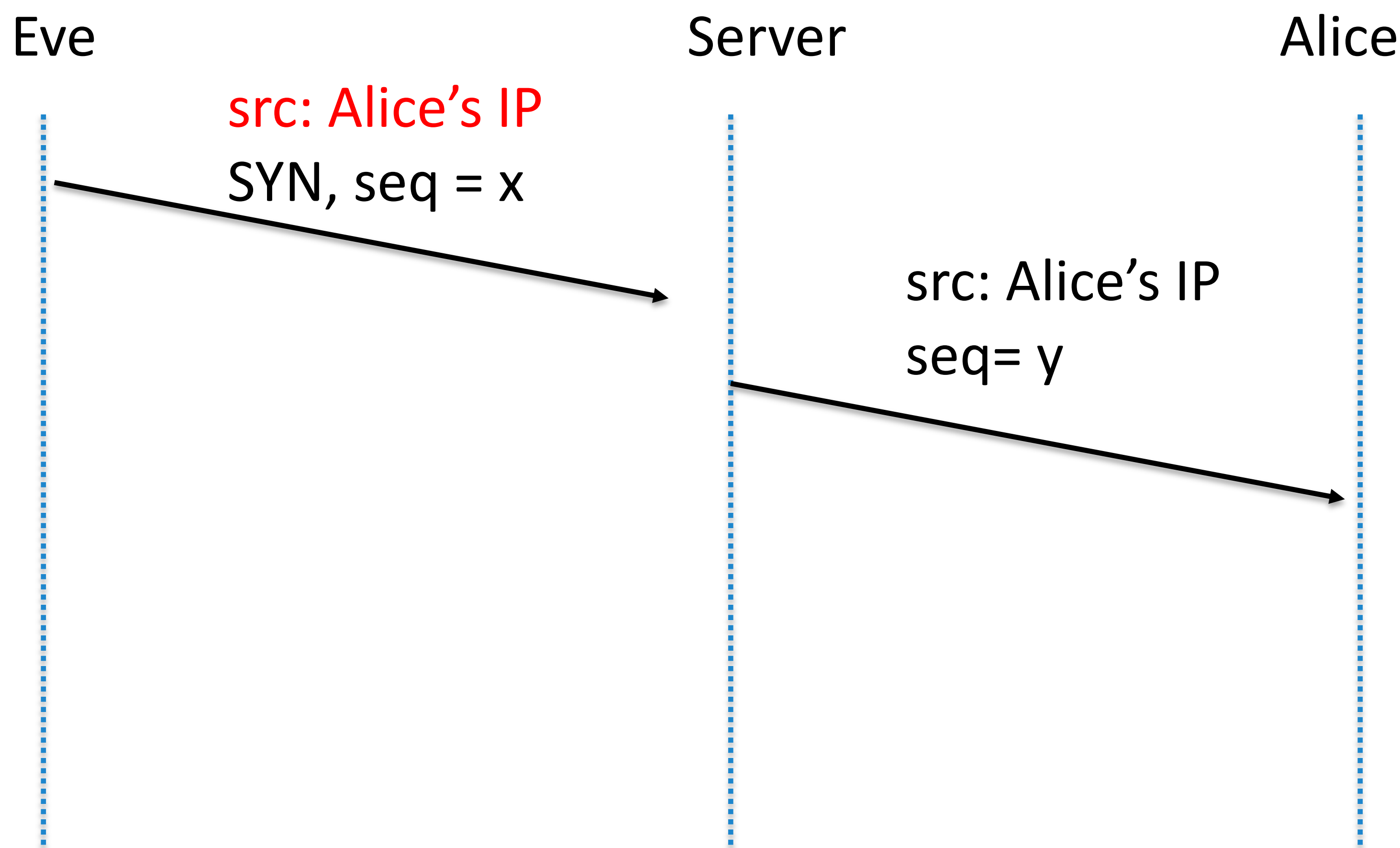
Blind spoofing



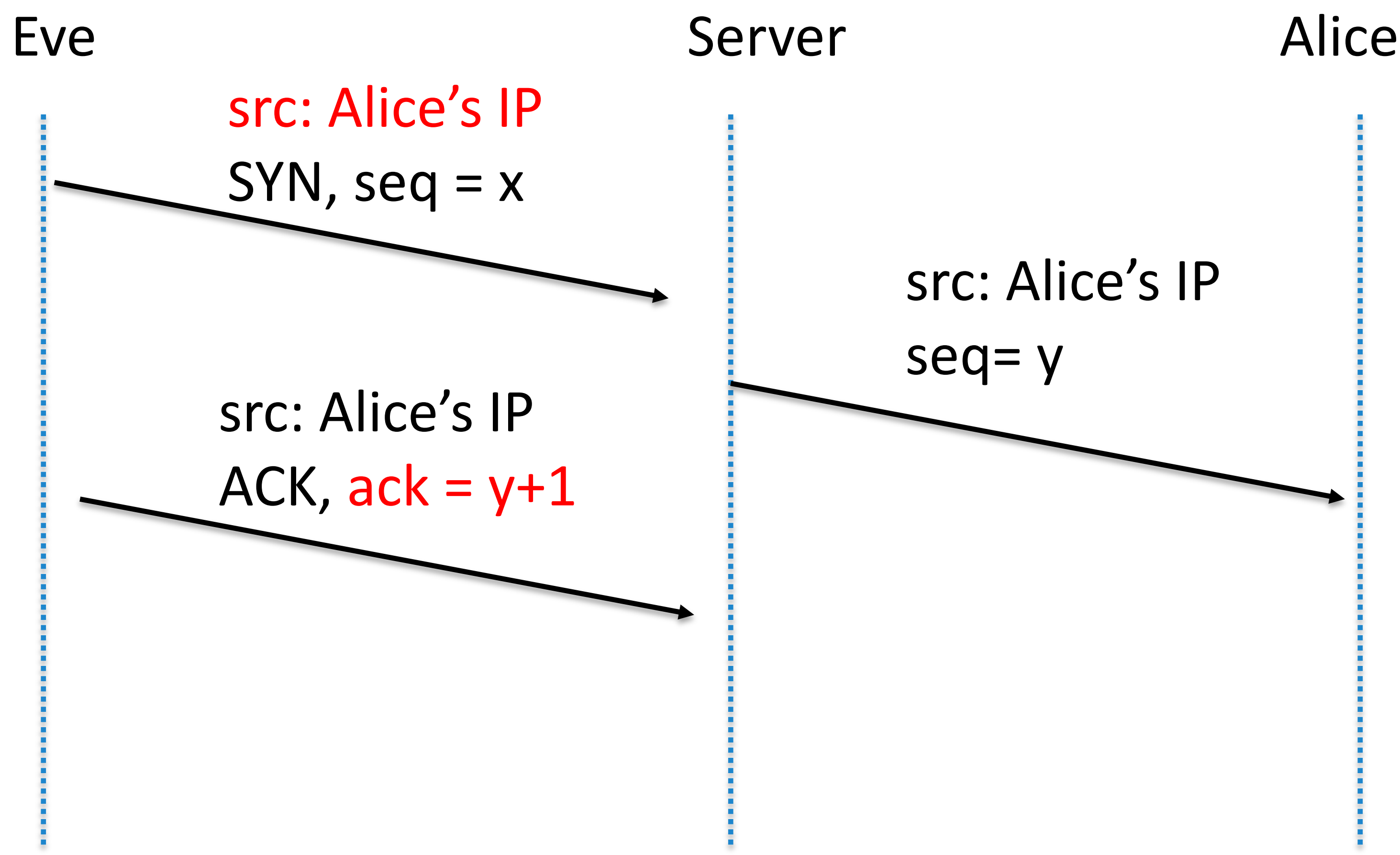
Blind spoofing



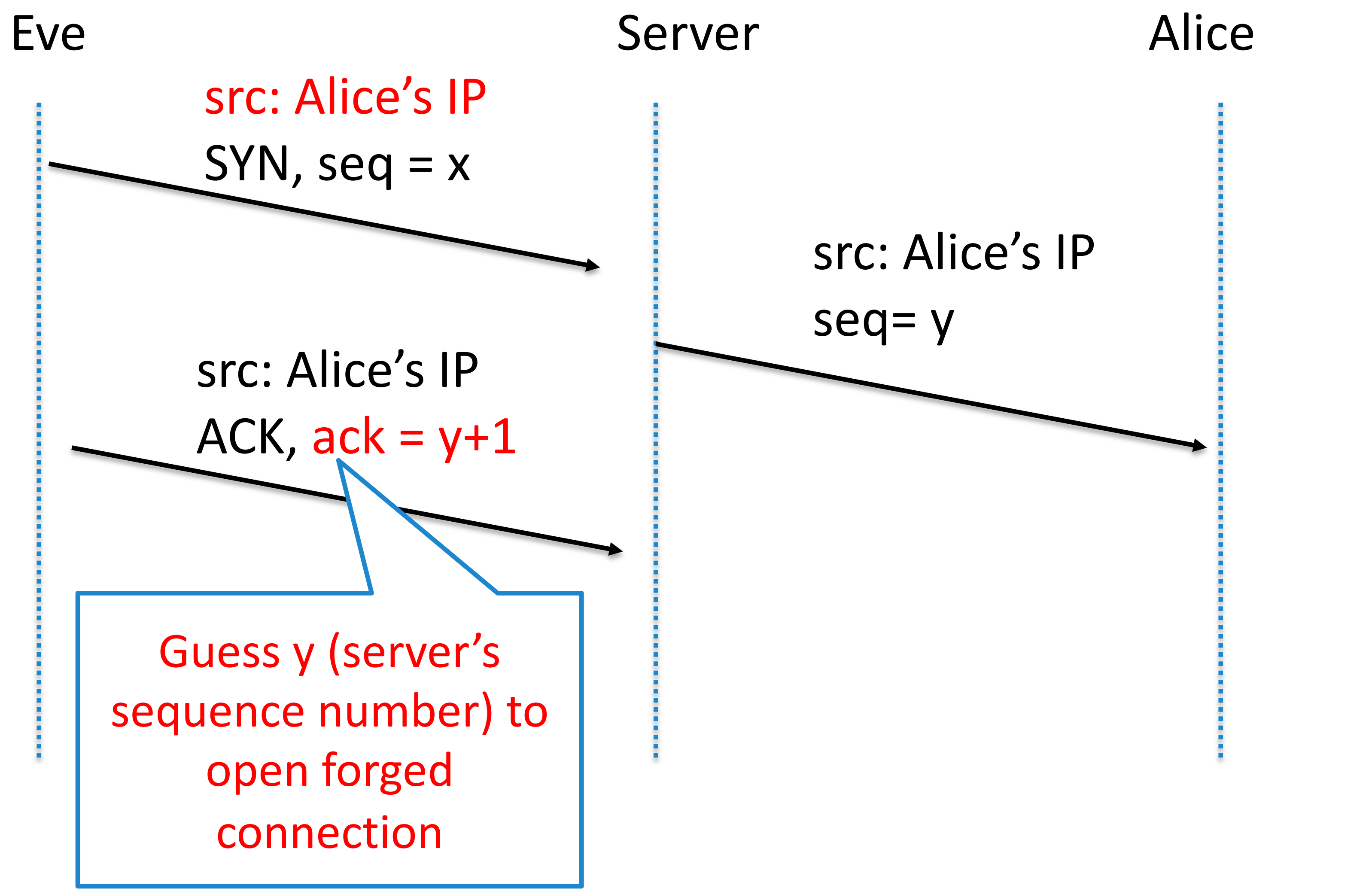
Blind spoofing



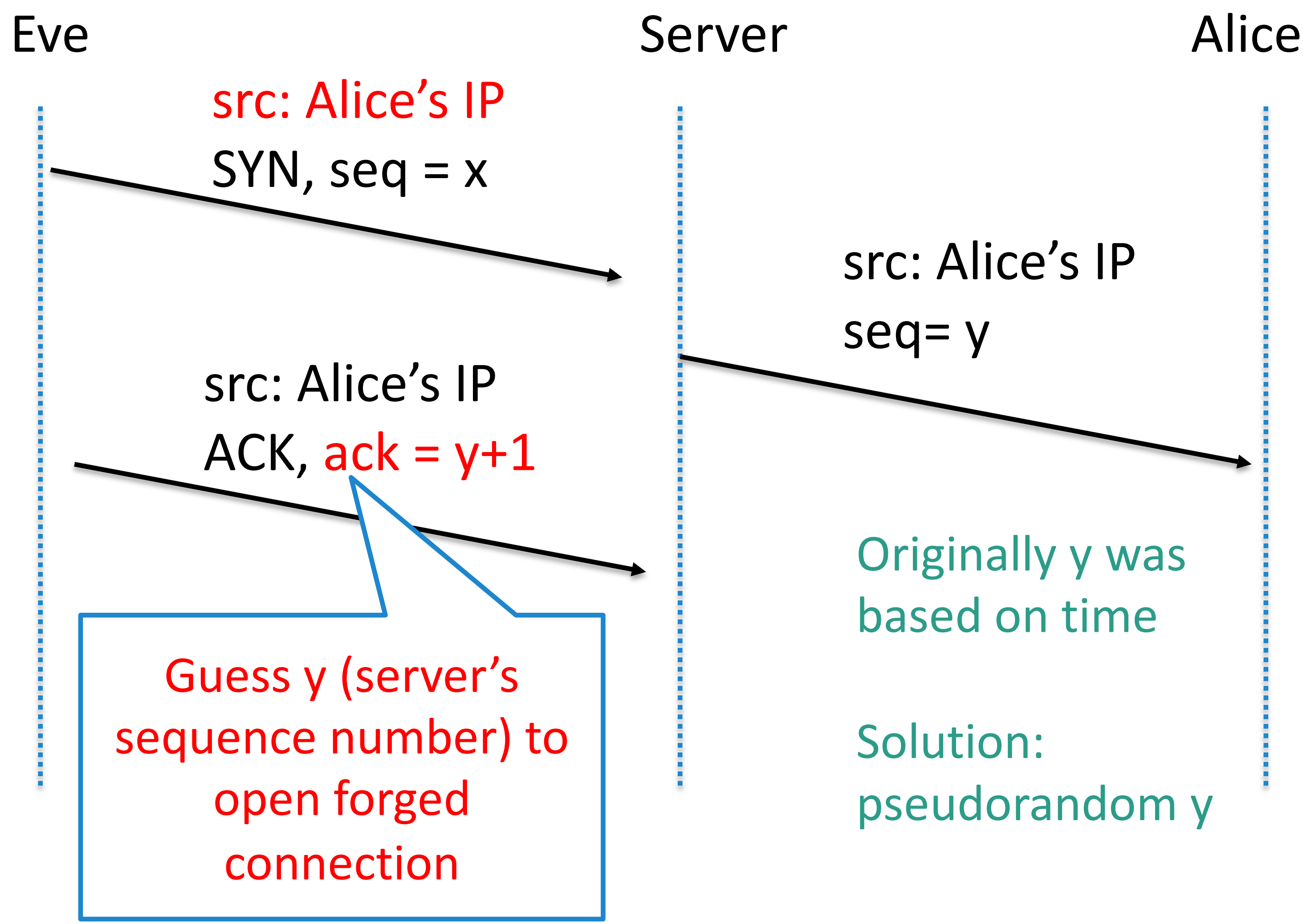
Blind spoofing



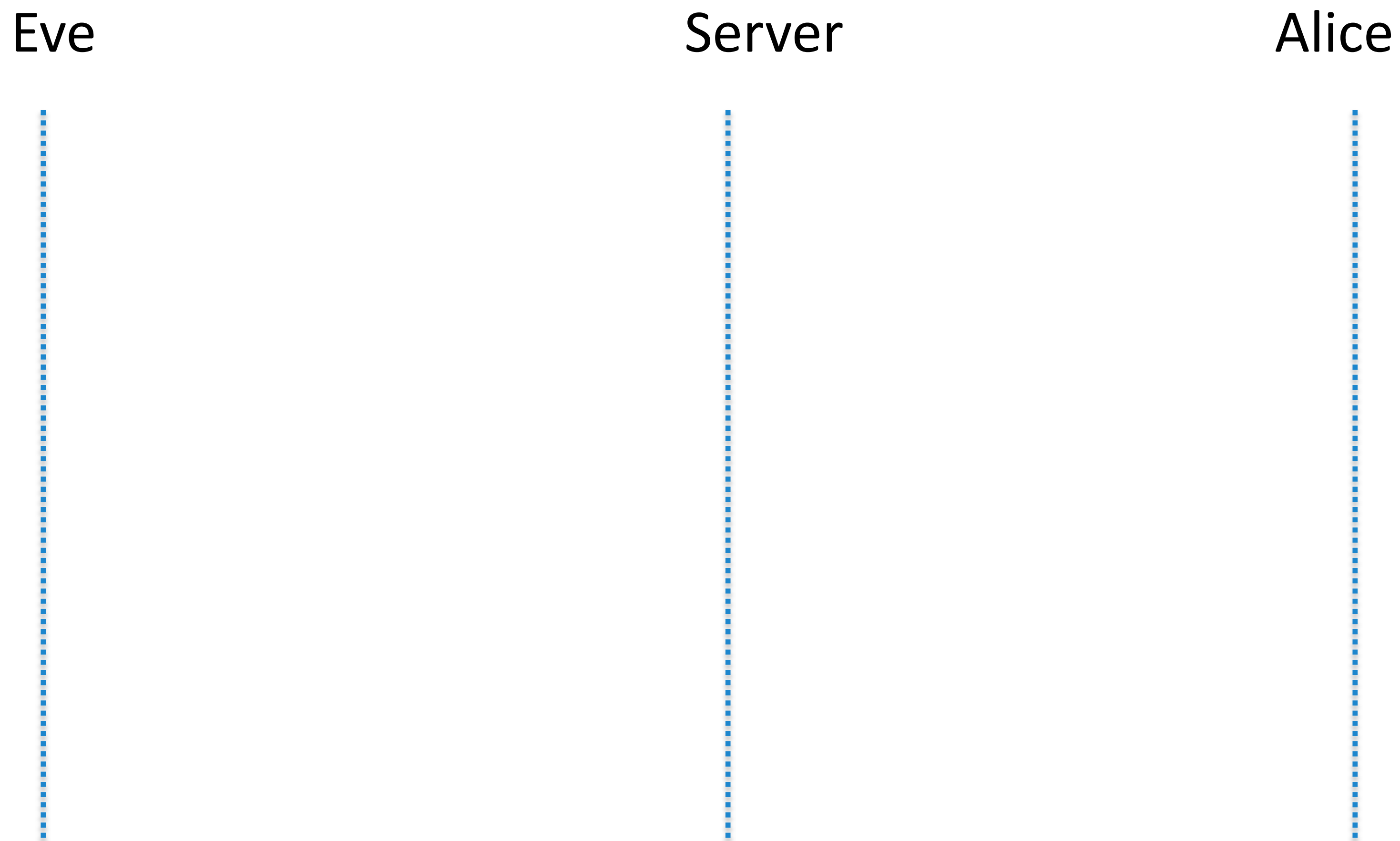
Blind spoofing



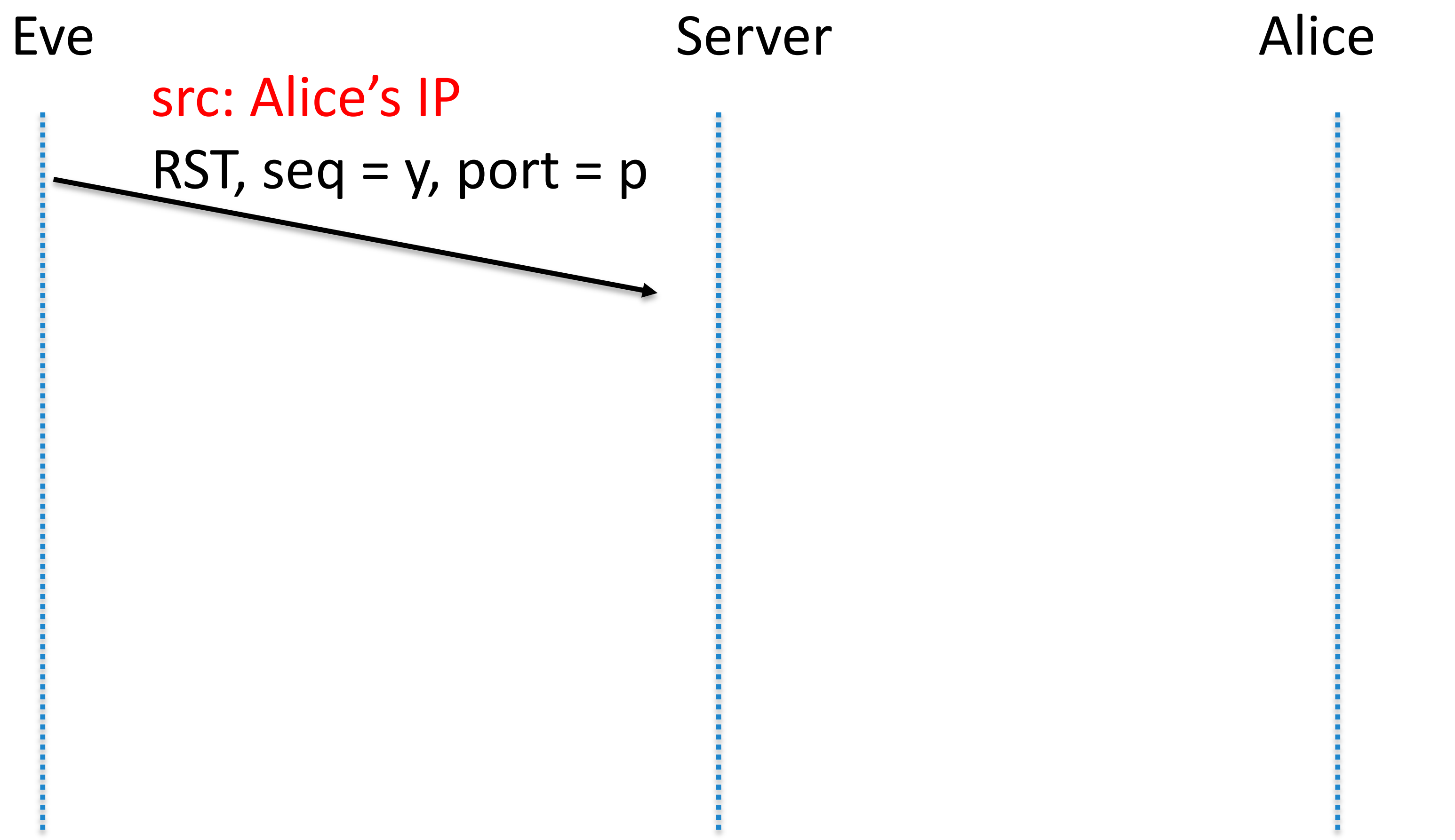
Blind spoofing



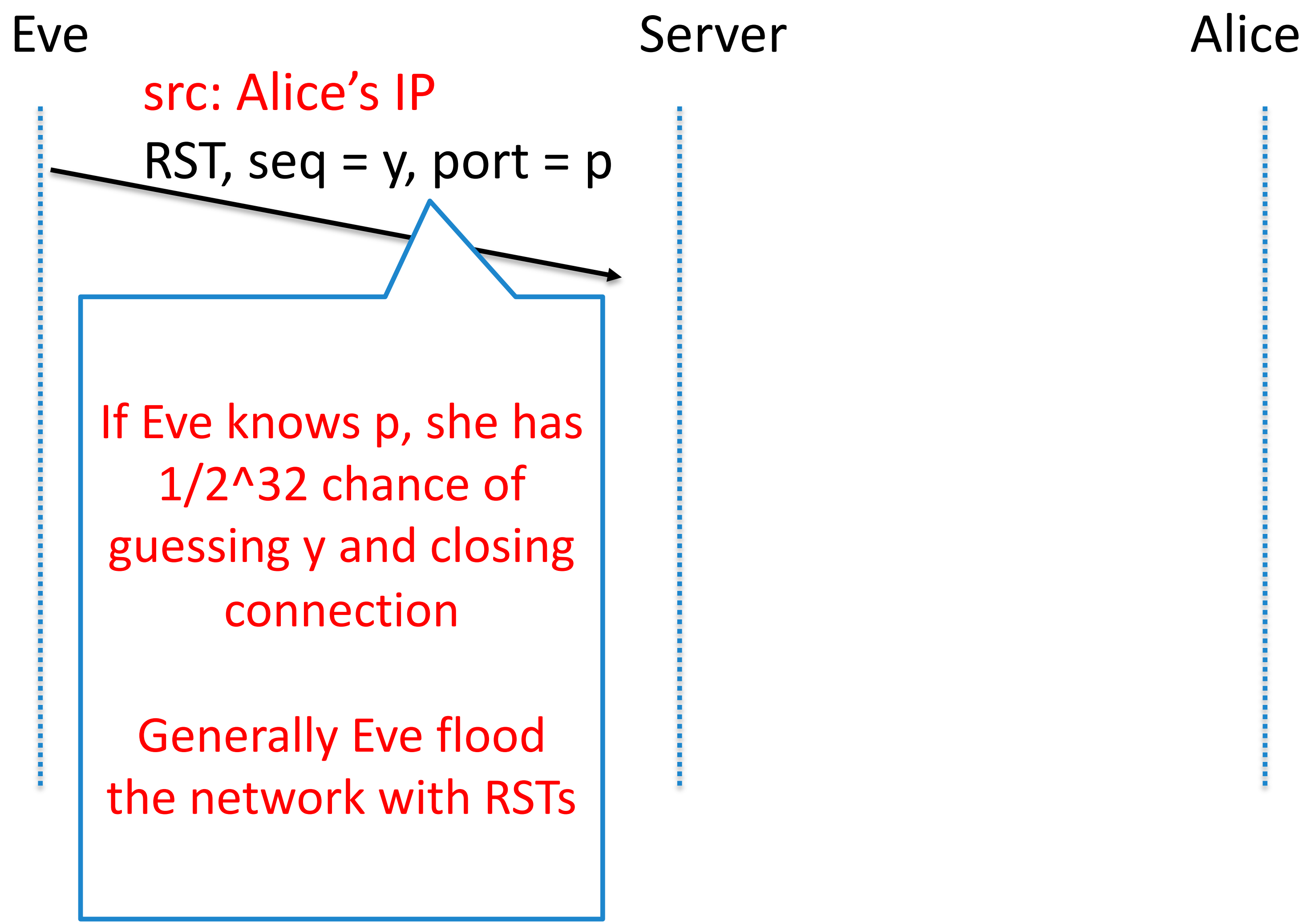
RST hijacking



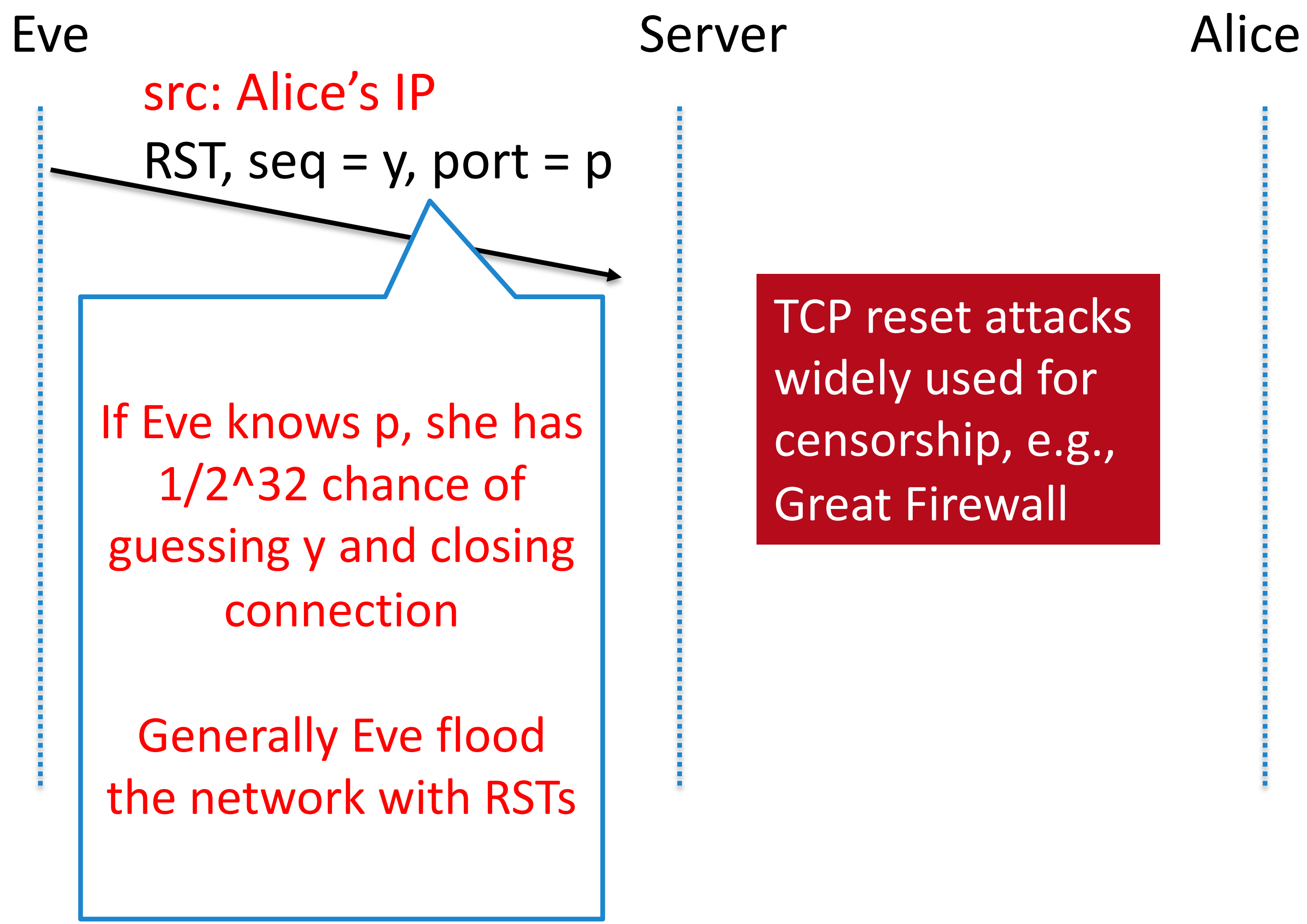
RST hijacking



RST hijacking



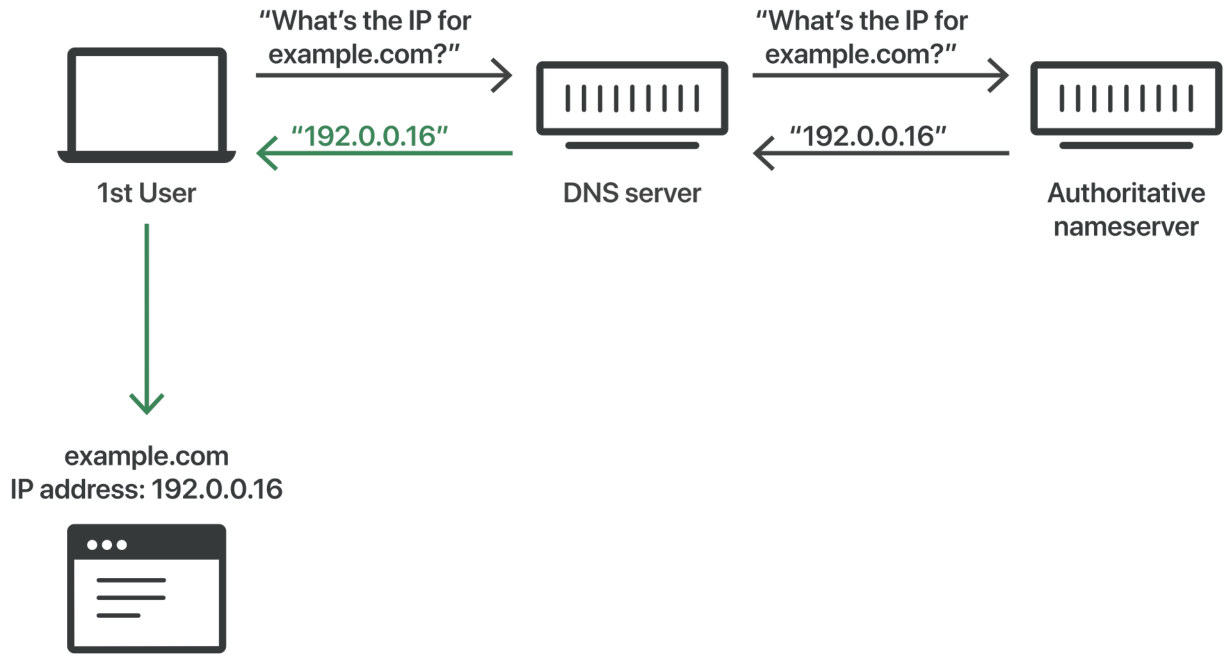
RST hijacking



Attacks on DNS

DNS poisoning

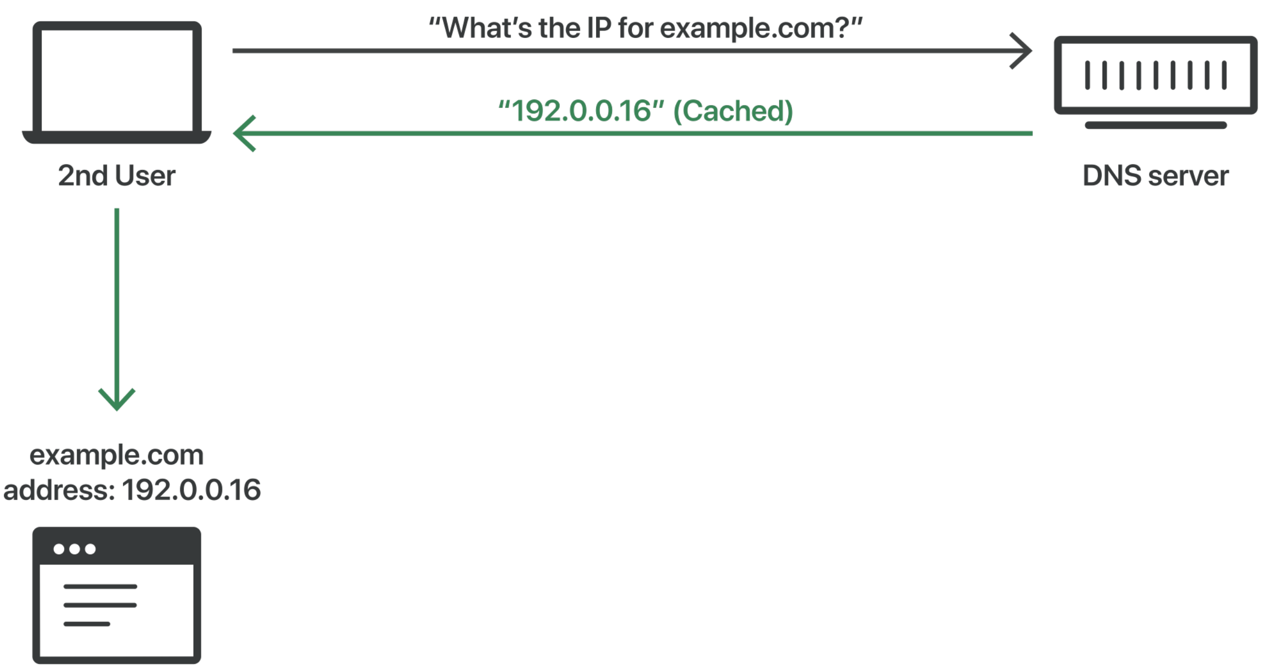
- How DNS works: non cached version



Src: https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

DNS poisoning

- How DNS works: cached version (used in practice)

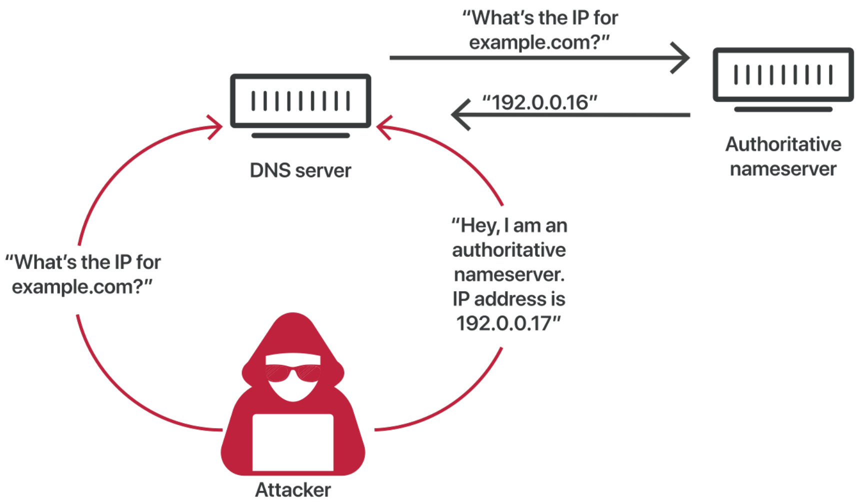


Src: https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

DNS poisoning

- Now the attack

Step 1

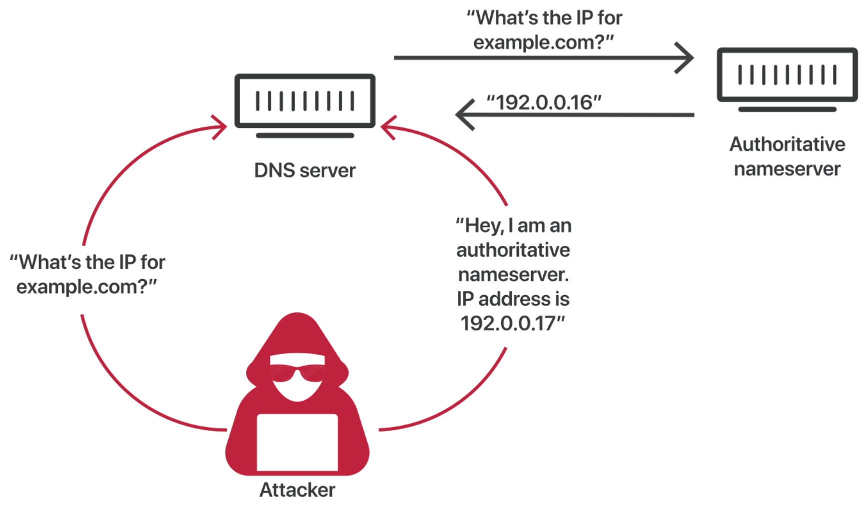


Src: https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

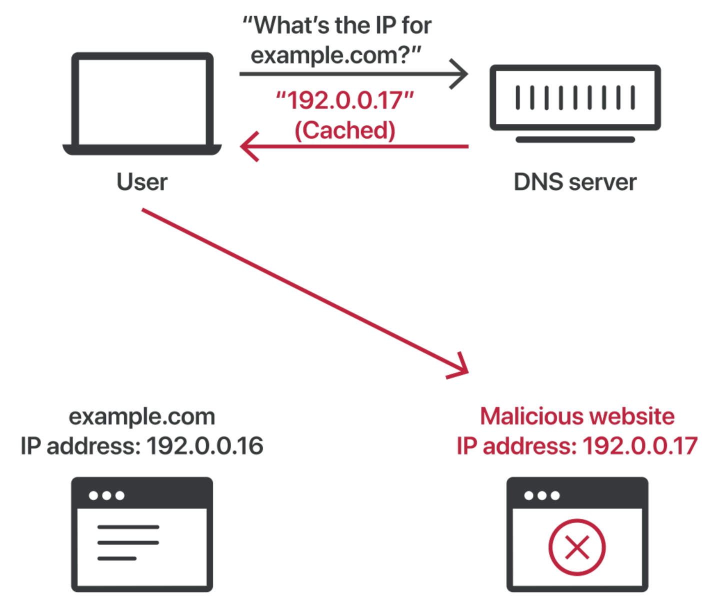
DNS poisoning

- Now the attack

Step 1



Step 2



Src: https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

DNS poisoning

- Why does it work?
  - DNS uses UDP – no authentication, no way of knowing which is true authoritative nameserver and which is not
- Challenges for the attacker?
  - Which DNS queries are not cached so the authoritative nameserver will be queried
  - Which authoritative nameserver the query will go to
  - Finally, the attacker has few milliseconds before response from the actual nameserver show up

DNS poisoning: Practical defense

- Include and randomize a 16 bit query ID (QID)
- Then the attacker has to
    - Race with the response from name server
    - Also has to guess the QID correctly – low chance

Kaminsky attack (2008)

- A devastating vulnerability that shook the security community
- own a whole domain, e.g., example.com
- First, ”glue records”

Kaminsky attack (2008)

- A devastating vulnerability that shook the security community
- pwn a whole domain, e.g., example.com
- First, ”glue records”

```
$ dig @ns1.example.com www.example.com
;; ANSWER SECTION:
www.example.com.     120      IN    A     192.168.1.10

;; AUTHORITY SECTION:
example.com.         86400    IN    NS    ns1.example.com.
example.com.         86400    IN    NS    ns2.example.com.

;; ADDITIONAL SECTION:
ns1.example.com.     604800   IN    A     192.168.2.20
ns2.example.com.     604800   IN    A     192.168.3.30
```

Kaminsky attack (2008)

- Now the exploit
- Setup
  - Attacker's javascript resides on Alice's machine
  - Asks for "doesnotexist.example.com"
  - Attacker also sets up a dns server which continuously send DNS responses like this:

```
$ dig doesnotexist.example.com
;; ANSWER SECTION:
doesnotexist.example.com.  120    IN  A    10.10.10.10

;; AUTHORITY SECTION:
example.com.             86400    IN  NS   www.example.com.

;; ADDITIONAL SECTION:
www.example.com.        604800    IN  A    10.10.10.20
```

Kaminsky attack (2008)

- Now the exploit
- Setup
  - Attacker's javascript resides on Alice's machine
  - Asks for "doesnotexist.example.com"
  - Attacker also sets up a dns server wl send DNS responses like this:

```
$ dig doesnotexist.example.com
;; ANSWER SECTION:
doesnotexist.example.com.   120    IN   A     1

;; AUTHORITY SECTION:
example.com.              86400    IN   NS    w

;; ADDITIONAL SECTION:
www.example.com.         604800    IN   A     10.10.10.20
```

Under attacker's control

Now domain traffic will go to the attacker

Kaminsky attack (2008)

- How does attacker know which queries would Alice ask?
  - The javascript is in attacker’s control
- How would attacker know the quid
  - It’s UDP – no authentication
  - First way: flood of responses with different QID
  - Second (better) way: Just send responses with QID X and let the javascript send queries 1000 times.
  - Now there is a 1000/2^16 chance (~1/65) chance that the whole domain will be owned

Kaminsky attack (2008)

- Immediate solution
  - As stupid as it might sound, they just increased the QID size to 32 bit (cannot be widely deployed, why?)

Kaminsky attack (2008)

- Immediate solution
  - As stupid as it might sound, they just increased the QID size to 32 bit (cannot be widely deployed, why?)
- Question
  - Another solution is to randomize the UDP port (used today)
  - Microsoft's updated DNS server is said to pre-allocate 2,500 UDP ports to use for these random queries
  - What is the increase in search space now?

Better solution: DNSSEC

- Crypto to the rescue
- DNS responses signed – now I know if the request is coming from authoritative name
- Higher levels vouch for lower levels
  - e.g., root vouches for .in, .in vouches for .ac, ...
- Root public key published
- Costly and slow adoption

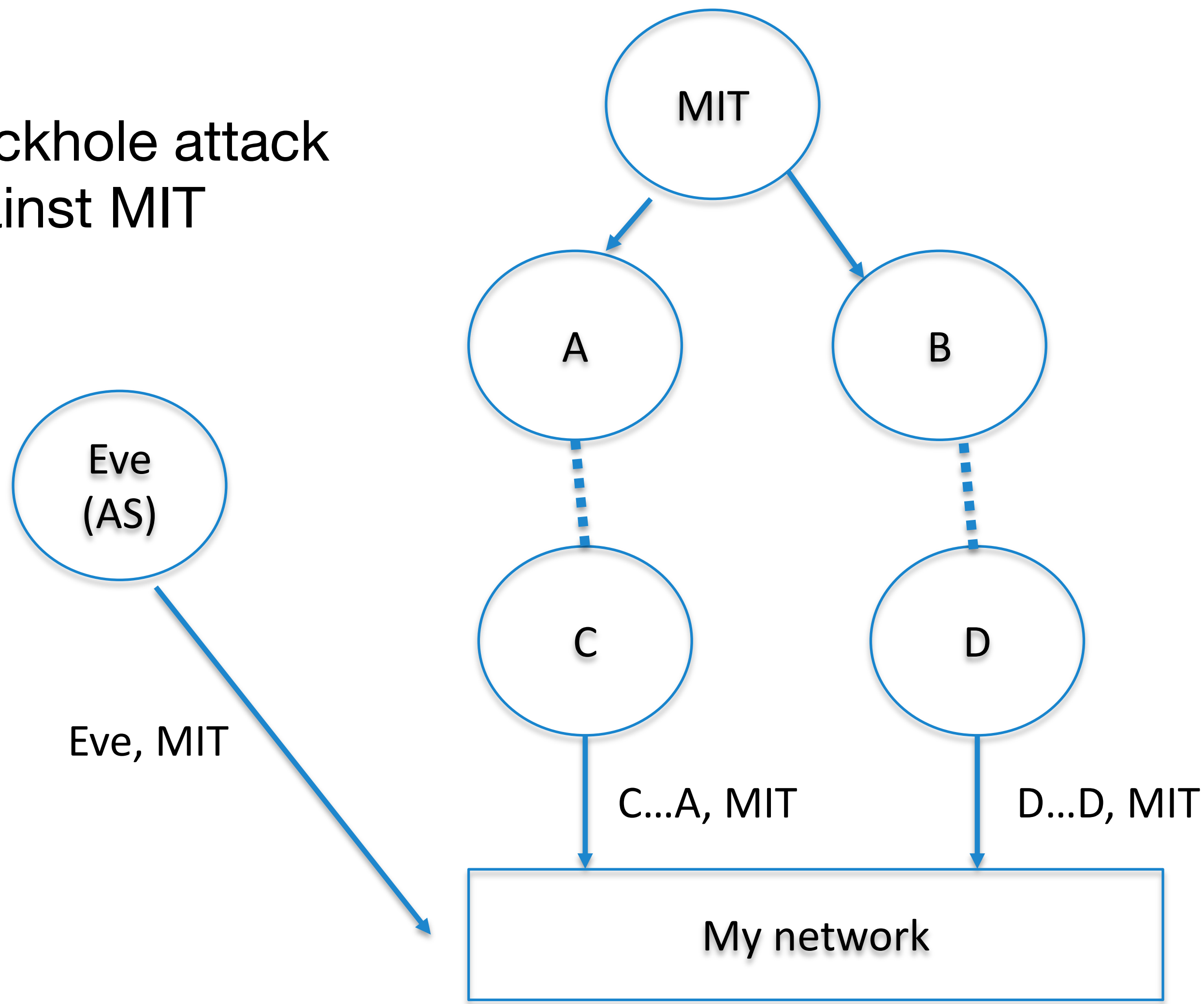
Attacks on BGP

Border gateway protocol (BGP)

- BGP is used to route between two AS's
- AS's advertised paths
- No verification: recipe for disaster

BGP hijacking

Blackhole attack against MIT



BGP hijacking in the real world

“in April 2018, a Russian provider announced a number of IP prefixes (groups of IP addresses) that actually belong to Route53 Amazon DNS servers. In short, the end result was that users attempting to log in to a cryptocurrency site were redirected to a fake version of the website controlled by hackers.”

https://www.cloudflare.com/learning/security/glossary/bgp-hijacking/

BGP hijacking in the real world

“In 2008, the Pakistani government-owned Pakistan Telecom attempted to censor Youtube within Pakistan by updating its BGP routes for the website. Seemingly on accident, the new routes were announced to Pakistan Telecom's upstream providers, and from there broadcast to the whole Internet. Suddenly, all web requests for Youtube were directed to Pakistan Telecom”

https://www.cloudflare.com/learning/security/glossary/bgp-hijacking/

S-BGP

- IP prefix announcements signed, Routes signed
  - previous hop authorizes next hop
- Higher levels vouch for lower levels
  - e.g., ICANN vouches for ARIN, ARIN vouches for AT&T, ...
- Root public key published

- Costly and slow adoption

TLS/SSL certificates

- The TL;DR
    - Use public key certificates issued by certificate authority(CA) to know if you are *really* talking to the domain you intend
    - The list of CAs are shipped with the browser
    - Question: What might be some problems?
    - One hint: can you really trust all CAs

Next: DoS attacks and mitigations

Outline

- Basics of computer networks
  - Or how computers talk to each other
- Basic network attacks
  - Attacking host-to-host data transmission protocols
  - Attacking network infrastructure
  - DDos, smurf attack, reflection attack
- Some mitigations

DoS Attacks

Denial of service (DoS)

- Prevent users from being able to access a specific computer, service, or piece of data
- In essence, an attack on availability
- Possible vectors:
  - Exploit bugs that lead to crashes
  - Exhaust the resources of a target
- Often very easy to perform...
- ... and fiendishly difficult to mitigate

DoS Attacker Goals & Threat Model

- Active attacker who may send arbitrary packets to anybody
- Goal is to reduce the availability of the victim

DoS Attacker model

- How much bandwidth is available to the attacker?
  - Can be increased by controlling more resources...
  - Or tricking others into participating in the attack

- What kind of packets do you send to victim?
  - Minimize effort and risk of detection for attacker...
  - While also maximizing damage to the victim

Standard DDos (Distributed Dos)

- What kind of packets do you send to the victim?
- Ideally, should be “connectionless”
  - Difficult to spoof TCP connections
- Should maximize the resources used by the victim

When would DoS attack work?

Effort (computation/memory resources)
of victim

>>

Effort (computation/memory resources)
of Attacker

TCP SYN flood

- TCP stack keeps track of connection state in data structures called Transmission Control Blocks (TCBs)
  - New TCB allocated by the kernel when a listen socket receives a SYN
  - TCB must persist for at least one RTO (Retransmission TimeOut)
- Attack: flood the victim with SYN packets
  - Exhaust available memory for TCBs, prevent legitimate clients from connecting
  - Crash the server OS by overflowing kernel memory
- Advantages for the attacker
  - No connection – each SYN can be spoofed, no need to hear responses
  - Asymmetry – attacker does not need to allocate TCBs

Exploiting asymmetry

- Example of a Distributed Denial of Service Attack (DDoS)
- Some DDoS is fueled by volunteers
  - E.g. Anonymous and Low Orbit Ion Canon (LOIC)
- Most DDoS is fueled by botnets