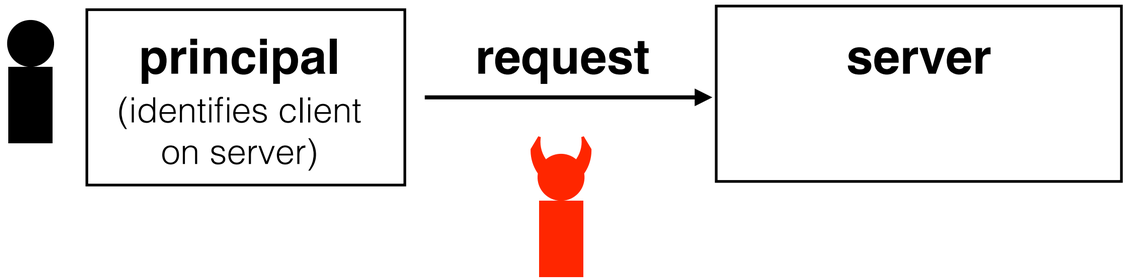
# 6.1800 Spring 2024

## Lecture #25: Network-based attacks

preventing access — *denying service* — to online resources

**Katrina LaCurts | lacurts@mit.edu | 6.1800 2024**

# we've been dealing with adversaries on the network for two lectures




**adversary's goal:** observe or tamper with packets

**today, our adversaries are still on the network, but they have new goals**

the primary method they'll use to achieve this goal is a **DDoS attack**, made more effective with a **botnet**

# 'Denial of service condition' disrupted US energy company operations

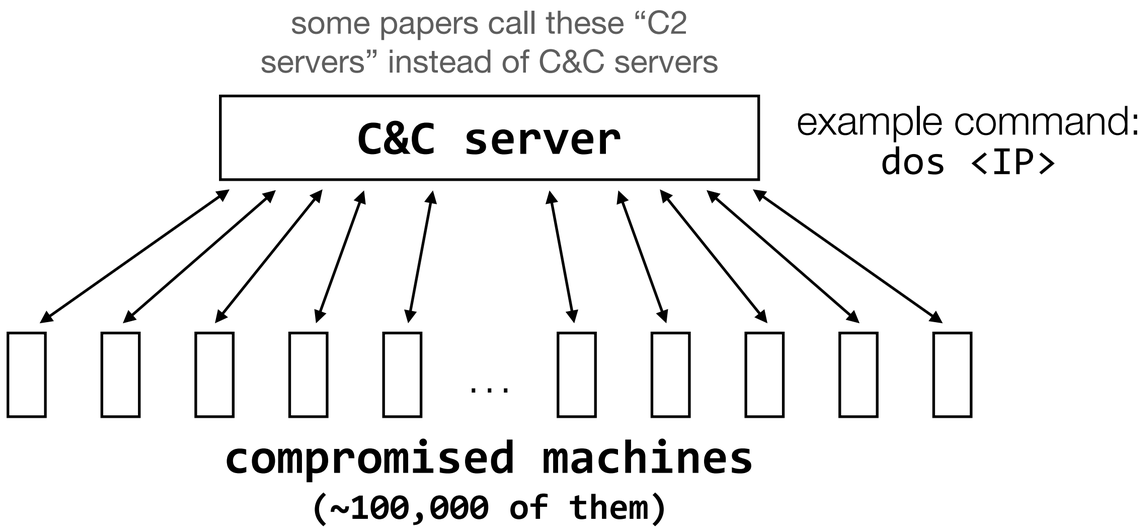
**Zack Whittaker** @zackwhittaker / 4 days ago

Comment

**policy:** maintain **availability** of the service

**threat model:** adversary controls a **botnet**, and is aiming to prevent access to a legitimate service via **DDoS attacks**

**botnets:** large collections of compromised machines controlled by an adversary




these machines can become compromised in a variety of ways. the mirai botnet, for example, works by attempting to log in to many machines using common username/password combinations. this has been effective for IoT devices that often have a common default password.

**policy:** maintain **availability** of the service

**threat model:** adversary controls a **botnet**, and is aiming to prevent access to a legitimate service via **DDoS attacks**

## network intrusion detection systems:

attempt to detect network attacks so that users can then prevent them (detection is the first step to prevention)

botnets are sophisticated, so we can't rely on just blocking "bad" IP addresses

**signature-based** NIDS match traffic against known signatures

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 7597
(msg:"MALWARE-BACKDOOR QAZ Worm Client Login
access"; flow:to_server,established;
content:"qazwsx.hsq"; metadata:ruleset community;
reference:mcafee,98775; classtype:misc-activity;
sid:108; rev:11;)
```

an example of a signature

**policy:** maintain **availability** of the service

**threat model:** adversary controls a **botnet**, and is aiming to prevent access to a legitimate service via **DDoS attacks**

## network intrusion detection systems:

attempt to detect network attacks so that users can then prevent them (detection is the first step to prevention)

botnets are sophisticated, so we can't rely on just blocking "bad" IP addresses

**signature-based** NIDS match traffic against known signatures

**anomaly-based** NIDS match traffic against a model of "normal" traffic

```
for each packet:
  search packet for "root"
```

**problem:** string might be split across packets

**policy:** maintain **availability** of the service

**threat model:** adversary controls a **botnet**, and is aiming to prevent access to a legitimate service via **DDoS attacks**

## network intrusion detection systems:

attempt to detect network attacks so that users can then prevent them (detection is the first step to prevention)

botnets are sophisticated, so we can't rely on just blocking "bad" IP addresses

**signature-based** NIDS match traffic against known signatures

**anomaly-based** NIDS match traffic against a model of "normal" traffic

```
stream = []
for each packet:
  add packet data to stream
  search stream for "root"
```

**problem:** packets might arrive out of order

**policy:** maintain **availability** of the service

**threat model:** adversary controls a **botnet**, and is aiming to prevent access to a legitimate service via **DDoS attacks**

## network intrusion detection systems:

attempt to detect network attacks so that users can then prevent them (detection is the first step to prevention)

botnets are sophisticated, so we can't rely on just blocking "bad" IP addresses

**signature-based** NIDS match traffic against known signatures

**anomaly-based** NIDS match traffic against a model of "normal" traffic

```
stream = []
for each packet:
  get sequence number
  add to stream in the correct order
  search stream for "root"
```

**problem:** this is a bit more difficult that it looks on the slide, and requires keeping a lot of state

it's certainly not impossible; after all, your computer reconstructs TCP byte streams all the time

**problem 2:** it doesn't even work

**policy:** maintain **availability** of the service

**threat model:** adversary controls a **botnet**, and is aiming to prevent access to a legitimate service via **DDoS attacks**

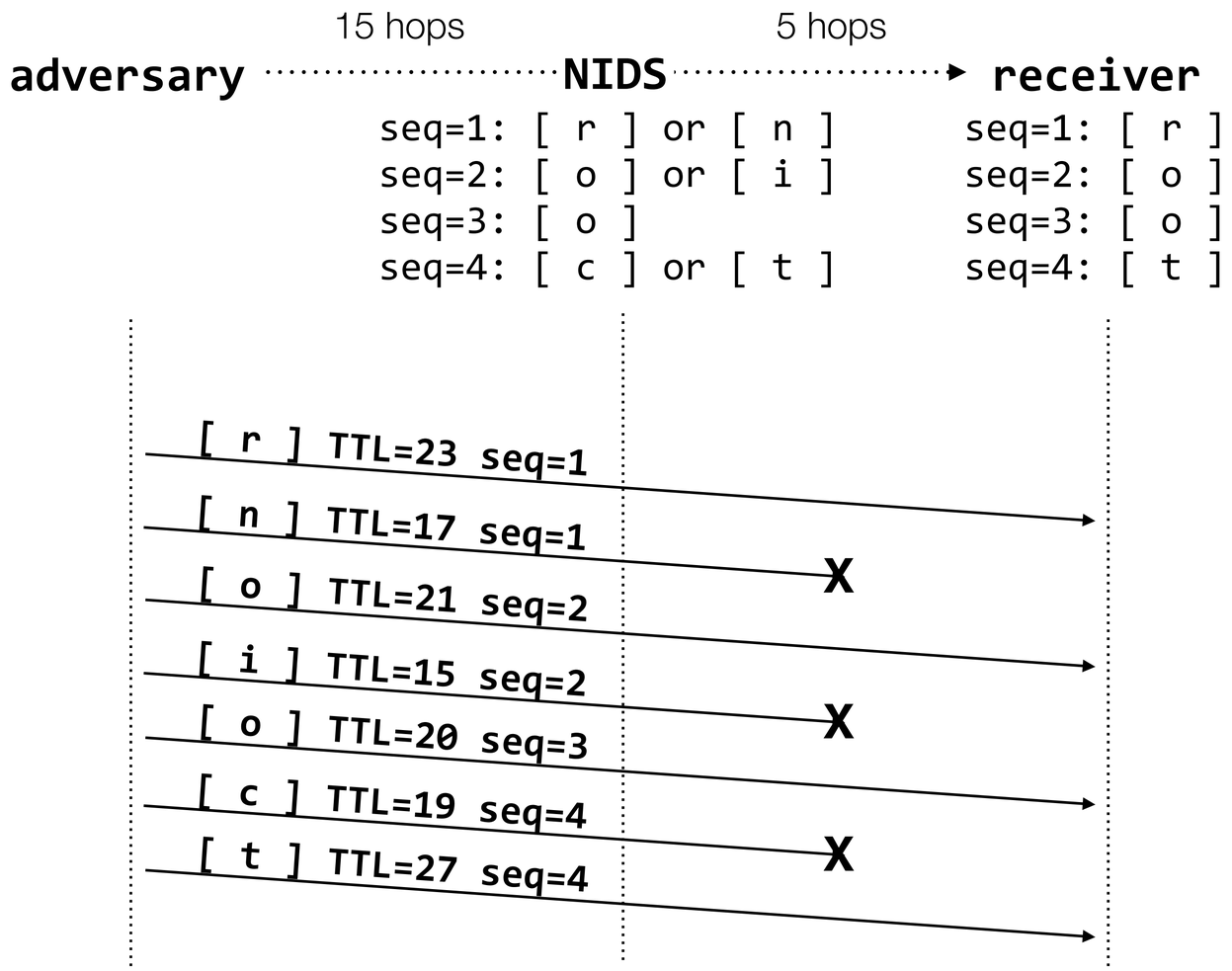
## network intrusion detection systems:

attempt to detect network attacks so that users can then prevent them (detection is the first step to prevention)

botnets are sophisticated, so we can't rely on just blocking "bad" IP addresses

**signature-based** NIDS match traffic against known signatures

**anomaly-based** NIDS match traffic against a model of "normal" traffic

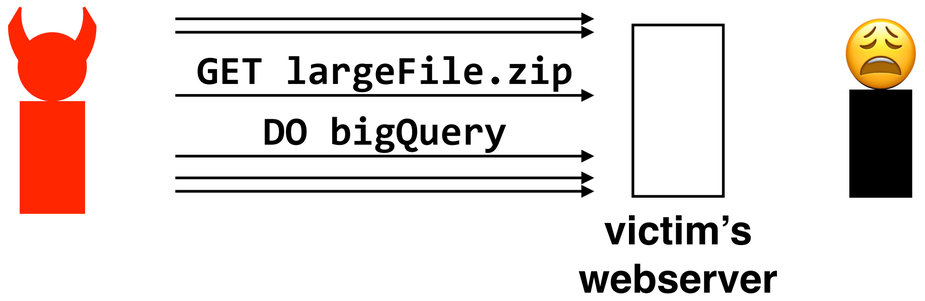


**policy:** maintain **availability** of the service

**threat model:** adversary controls a **botnet**, and is aiming to prevent access to a legitimate service via **DDoS attacks**

**additional challenge:**

some DDoS attacks mimic legitimate traffic, and/or attempt to exhaust resources on the server itself

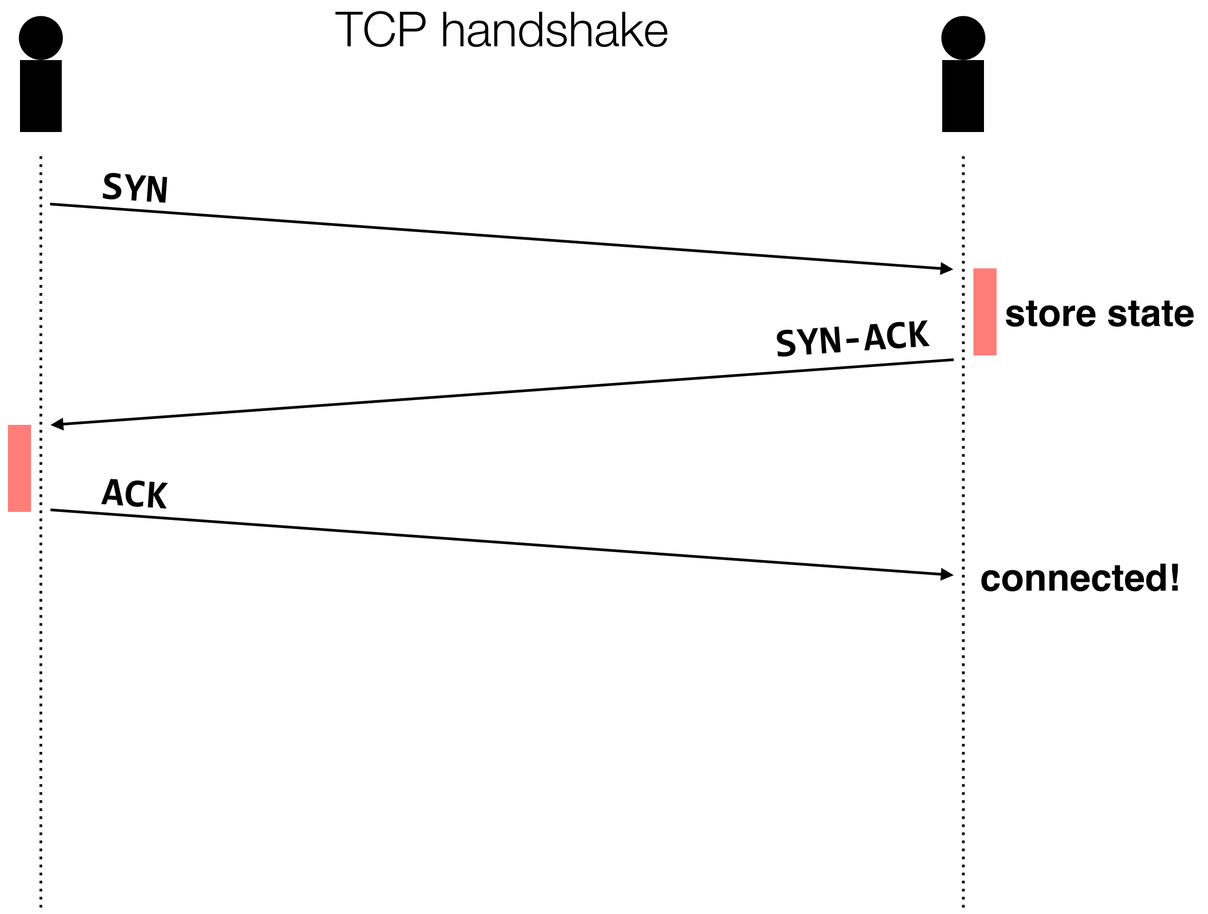


**policy:** maintain **availability** of the service

**threat model:** adversary controls a **botnet**, and is aiming to prevent access to a legitimate service via **DDoS attacks**

**additional challenge:**

some DDoS attacks mimic legitimate traffic, and/or attempt to exhaust resources on the server itself

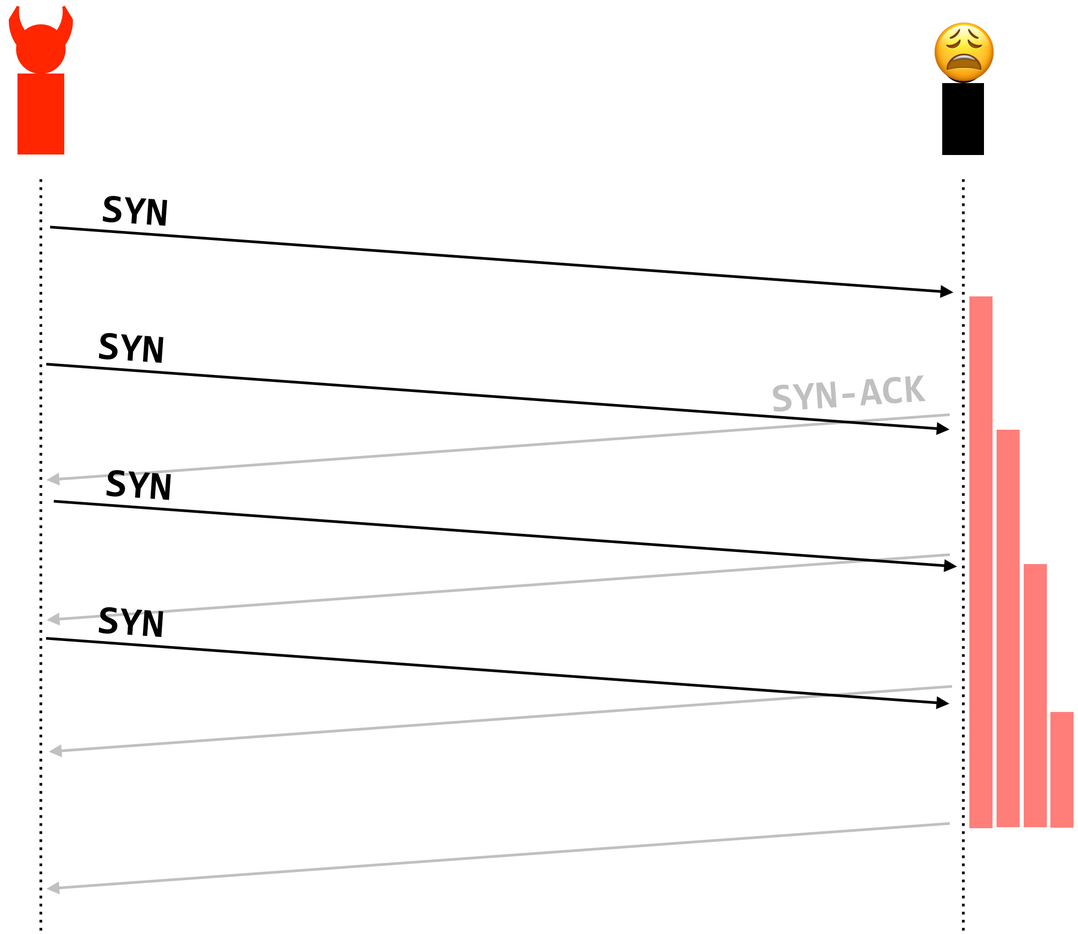


**policy:** maintain **availability** of the service

**threat model:** adversary controls a **botnet**, and is aiming to prevent access to a legitimate service via **DDoS attacks**

**additional challenge:**

some DDoS attacks mimic legitimate traffic, and/or attempt to exhaust resources on the server itself

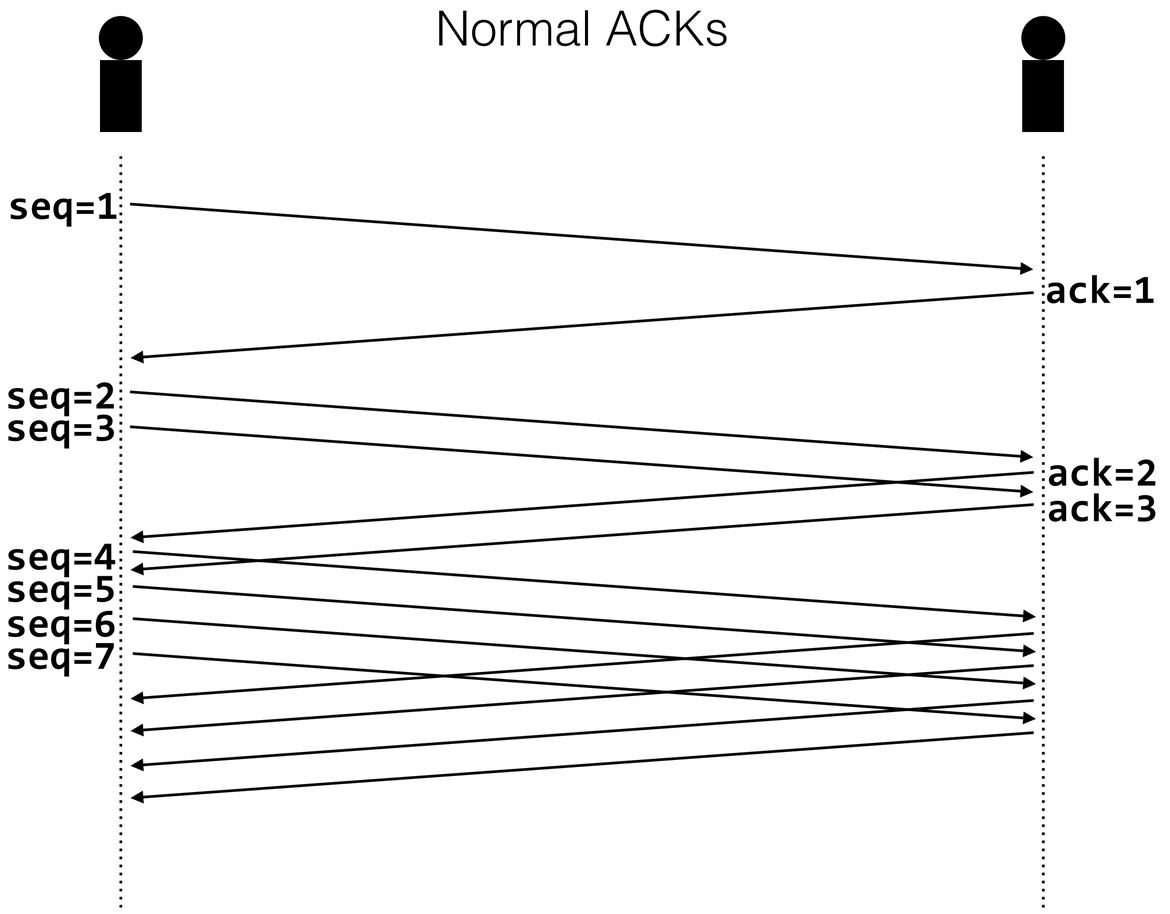


**policy:** maintain **availability** of the service

**threat model:** adversary controls a **botnet**, and is aiming to prevent access to a legitimate service via **DDoS attacks**

**additional challenge:**

some DDoS attacks mimic legitimate traffic, and/or attempt to exhaust resources on the server itself

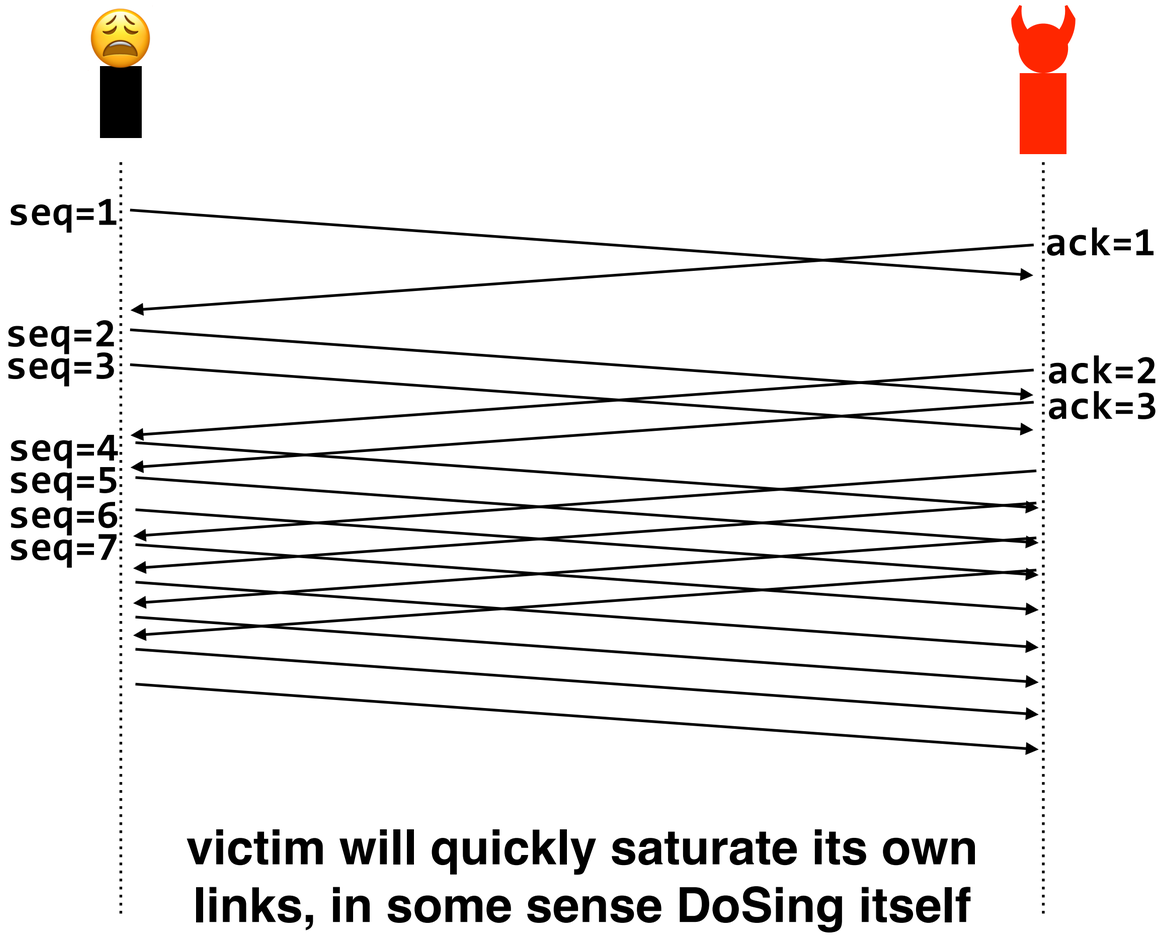


**policy:** maintain **availability** of the service

**threat model:** adversary controls a **botnet**, and is aiming to prevent access to a legitimate service via **DDoS attacks**

**additional challenge:**

some DDoS attacks mimic legitimate traffic, and/or attempt to exhaust resources on the server itself



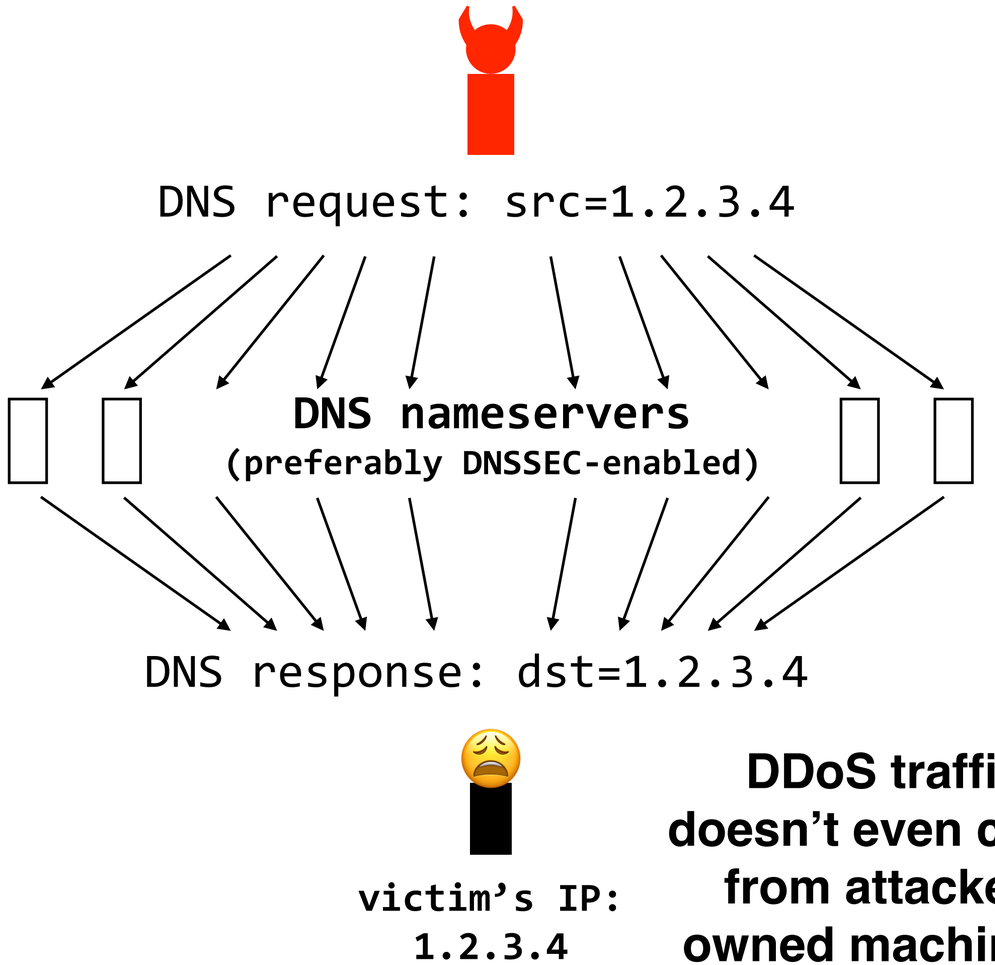

**victim will quickly saturate its own links, in some sense DoSing itself**

**policy:** maintain **availability** of the service

**threat model:** adversary controls a **botnet**, and is aiming to prevent access to a legitimate service via **DDoS attacks**

**additional challenge:**

some DDoS attacks mimic legitimate traffic, and/or attempt to exhaust resources on the server itself




**DDoS traffic doesn't even come from attacker-owned machines!**

this is known as a DNS amplification (or sometimes DNS reflection) attack

**DDoS attacks** prevent legitimate access to internet services. secure channels won't help us here, and **botnets** make DDoS attacks relatively easy to mount

DDoS attacks are difficult to prevent because they are sophisticated and can mimic legitimate traffic; **network-intrusion detection systems** help, but they're not perfect

robust, distributed systems are a good defense against DDoS attacks

network attacks are particularly devastating when they attack parts of the **network infrastructure** (e.g., DDoSing the DNS root zone, making fake BGP announcements)

these attacks are possible in part because the internet was not designed with them in mind