Lecture #25: Network-based attacks

preventing access — *denying service* — to online resources
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

**principal**
(identifies client on server)

**request**

**server**

**adversary’s goal:** prevent legitimate access to an internet resource
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

adversary’s goal: prevent legitimate access to an internet resource
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**
threat model: adversary controls a botnet, and is aiming to prevent access to a legitimate service via DDoS attacks
**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*
**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
**botnets:** large collections of compromised machines controlled by an adversary

the Mirai paper calls these “C2 servers” instead of C&C servers

**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**
**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

The Mirai paper calls these “C2 servers” instead of C&C servers.

**C&C server**

Example command:
```
dos <IP>
```

**compromised machines**

(≈100,000 of them)
botnets: large collections of compromised machines controlled by an adversary

the Mirai paper calls these “C2 servers” instead of C&C servers

example command: dos <IP>


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
**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*
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
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
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:"qazwxs.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

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**
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”
**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:**

- **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

**botnets** are sophisticated, so we can’t rely on just blocking “bad” IP addresses

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

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

**network intrusion detection systems:**

- attempt to detect network attacks so that users can then prevent them (detection is the first step to prevention)
- **signature-based** NIDS match traffic against known signatures
- **anomaly-based** NIDS match traffic against a model of “normal” traffic

**problem:** string might be split across packets
network intrusion detection systems:

- **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**

---

```python
stream = []
for each packet:
    add packet data to stream
    search stream for “root”
```

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

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:

- signature-based NIDS match traffic against known signatures
- anomaly-based NIDS match traffic against a model of “normal” traffic

Threat model: Adversary controls a botnet, and is aiming to prevent access to a legitimate service via DDoS attacks.

Policy: Maintain availability of the service.

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 than 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.

Botnets are sophisticated, so we can’t rely on just blocking “bad” IP addresses.

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

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

policy: maintain availability of the service
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**

**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

[Diagram showing network traffic flow from adversary to receiver with TTL=23 and seq=1]
**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**

**Network Intrusion Detection Systems (NIDS):** attempt to detect network attacks so that users can then prevent them (detection is the first step to prevention)

- **Signature-based** NIDS match traffic against known signatures
- **Anomaly-based** NIDS match traffic against a model of “normal” traffic

**Botnets** are sophisticated, so we can’t rely on just blocking “bad” IP addresses

---

Received by **NIDS**, not by **receiver**, because of **TTL**
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

---

[Diagram showing network flows]

- **adversary** sends a packet with TTL=23 and seq=1, which is detected by NIDS (15 hops).
- The packet is then sent to the receiver (5 hops).
- The receiver receives the packet with TTL=17 and seq=1, which NIDS does not detect (15 hops).
- The packet is marked as received by NIDS, not by receiver, because of TTL limitations.

Received by NIDS, not by receiver, because of TTL.
**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**

**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**

**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.

---

**victim's webserver**
**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

---

```
GET largeFile.zip
DO bigQuery
```

victim’s webserver
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

this attack is similar to the HTTP flood attack in the Mirai paper
**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

TCP handshake

SYN
**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

![TCP handshake diagram]
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

this attack is similar to the SYN flood attack in the Mirai paper
**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.

This attack is similar to the SYN flood attack in the Mirai paper.
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

this attack is similar to the SYN flood attack in the Mirai paper
**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.

---

This attack is similar to the SYN flood attack in the Mirai paper.
**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

this attack is similar to the SYN flood attack in the Mirai paper
**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

---

this attack is similar to the SYN flood attack in the Mirai paper
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

this attack is similar to the SYN flood attack in the Mirai paper
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

this attack is similar to the SYN flood attack in the Mirai paper
**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
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
threat model: adversary controls a botnet, and is aiming to prevent access to a legitimate service via DDoS attacks

policy: maintain availability of the service

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
**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

**Diagram:**

The diagram shows a sequence of packets (seq=1 to seq=7) exchanged between two hosts. The packets are acknowledged (ack=1 to ack=3) to simulate a typical network interaction. According to the diagram, the 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

*DNS nameservers* (preferably DNSSEC-enabled)

**victim's IP:**

1.2.3.4

this is a DNS amplification attack; it is *not* the “DNS flood” attack mentioned in the Mirai paper
**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

---

**DNS request:** src=1.2.3.4

**DNS nameservers**

(preferably DNSSEC-enabled)

**victim’s IP:**

1.2.3.4

this is a DNS amplification attack; it is *not* the “DNS flood” attack mentioned in the Mirai paper
**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

---

Diagram:

```
DNS request: src=1.2.3.4

DNS nameservers
(preferably DNSSEC-enabled)

DNS response: dst=1.2.3.4

victim’s IP:
1.2.3.4
```

this is a DNS amplification attack; it is **not** the “DNS flood” attack mentioned in the Mirai paper
**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

this is a DNS amplification attack; it is not the “DNS flood” attack mentioned in the Mirai paper
**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.

---

**DNS request:** `src=1.2.3.4`

**DNS response:** `dst=1.2.3.4`

**DNS nameservers** (preferably DNSSEC-enabled)

---

Victim's IP: `1.2.3.4`

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

---

This is a DNS amplification attack; it is *not* the “DNS flood” attack mentioned in the Mirai paper.
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.

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.