it’s everyone’s favorite time of the year!

http://registrar.mit.edu/subjectevaluation
Colonial Pipeline hackers apologize, promise to ransom less controversial targets in future

‘We are apolitical, we do not participate in geopolitics’

By Mitchell Clark | May 10, 2021, 2:44pm EDT

how does an adversary — or anyone? — decide what systems are “controversial targets”?
Lecture #24: Network-based attacks
preventing access — *denying service* — to online resources
last week, we dealt with adversaries on the network

principal (identifies client on server) 

server

adversary's goal: observe or tamper with packets
this week, our adversaries are still on the network, but they have new goals

adversary’s goal: prevent legitimate access to an internet resource
this week, 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**.
this week, 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**
**botnets**: large collections of compromised machines controlled by an adversary

**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

---

compromised machines

(≈100,000 of them)
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

compromised machines (~100,000 of them)
**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  

- **C&C server**  
- **Compromised machines** (~100,000 of them)  

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

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)

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**.
**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:"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

```
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:**

- **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”

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)

**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

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

- for each packet:
  - search packet for “root”

for each packet:

- **problem:** string might be split across packets
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”
```
**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

---

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

**problem:** packets might arrive out of order
network intrusion detection systems:

- **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**

---

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

**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
**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

---

```python
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

**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

---

15 hops

adversary .................................. NIDS ..................................→ receiver

5 hops

---

Katrina LaCurts | lacurts@mit.edu | 6.033 2021
**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

**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

---

[Diagram of network traffic and attack]
**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

---

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)

- **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

**Diagram:**

```
adversary | NIDS | receiver
---------|------|---------
seq=1: [ r ] or [ n ]  seq=1: [ r ]
15 hops

[r] TTL=23 seq=1
[n] TTL=17 seq=1
```

received by NIDS, not by receiver, because of TTL
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

adversary-NIDS-receiver

seq=1: [ r ] or [ n ] seq=1: [ r ]
seq=2: [ o ] or [ i ] seq=2: [ o ]
seq=3: [ o ] seq=3: [ o ]

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

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
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
**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

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
**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

store state
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

TCP handshake:
- SYN
- SYN-ACK
- ACK
- connected!
- store state
**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

---

**Diagram:**
- SYN
- SYN-ACK

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

---

```
SYN
SYN
SYN
SYN
SYN-ACK
```

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

---

**Diagram:**

Normal ACKs

- seq=1
- seq=2
- seq=3

ack=1

---

Katrina LaCurts | lacurts@mit.edu | 6.033 2021
**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

---

**Normal ACKs**

- seq=1
- seq=2
- seq=3
- ack=1
- ack=2
- ack=3

---
**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

---

Normal ACKs:

```
seq=1
seq=2
seq=3
seq=4
seq=5
seq=6
seq=7
```

```
ack=1
ack=2
ack=3
```
**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

<table>
<thead>
<tr>
<th>seq=1</th>
<th>ack=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq=2</td>
<td>ack=2</td>
</tr>
<tr>
<td>seq=3</td>
<td>ack=3</td>
</tr>
</tbody>
</table>
**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

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`

---

**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

---

**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

---

![Diagram]

- **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).