CIS 4930: Secure IoT

Prof. Kaushal Kafle

Lecture 17

Class Notes

- **Project phase 3: Security analysis of IoT apps (Android)**
    - Project proposal is due today!
- We'll have an online class on 12/12.

Denial-of-Service (DoS)

Denial-of-Service (DoS)

- Intentional prevention of access to valued resource
  - CPU, memory, disk (system resources)
  - DNS, print queues (services)
  - Web server, database, media server (applications)
- **This is an attack on availability**
- Launching DoS attacks is easy
- Preventing DoS attacks is very hard

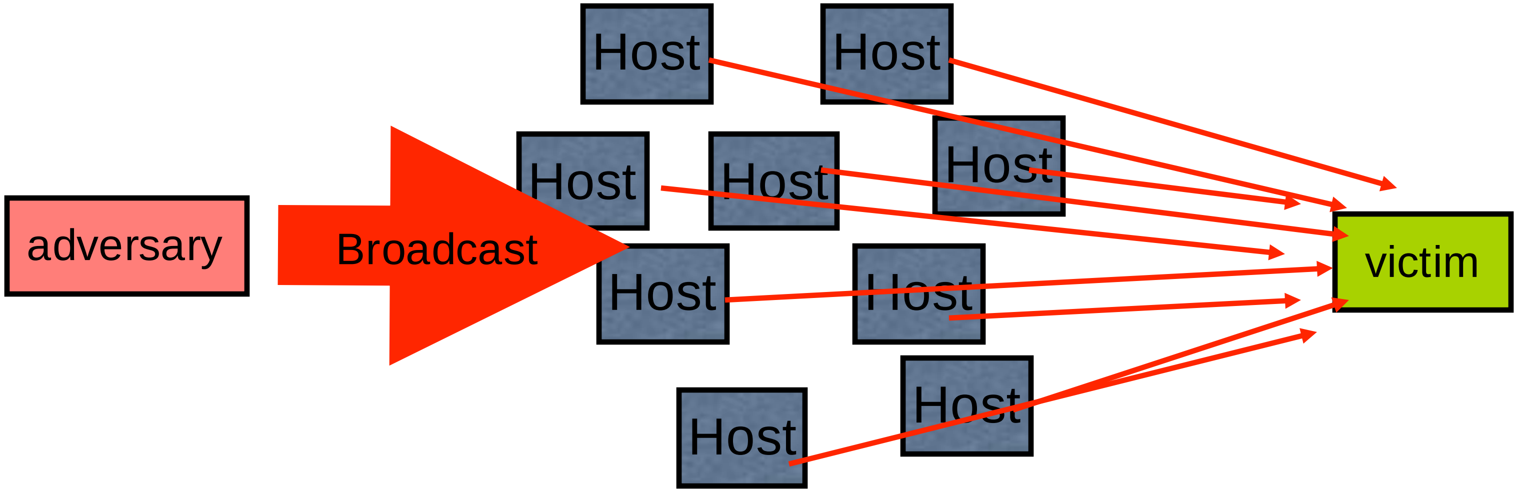
Canonical DoS - Request Flood

- Overwhelm some resource with requests
- e.g., web-server, phone system
- Most effective when processing request is expensive

Smurf Attacks

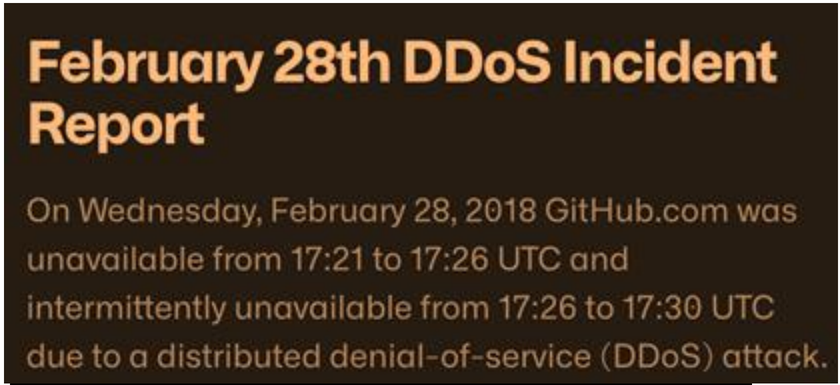
Example: SMURF Attacks

- Simple DoS attack:
  - Send a large number PING packets to a network's broadcast IP addresses (e.g., 192.168.27.254)
  - Set the source packet IP address to be your victim
  - All hosts will reflexively respond to the ping at your victim
  - … and it will be crushed under the load.
  - This is an **amplification attack** and a **reflection attack**



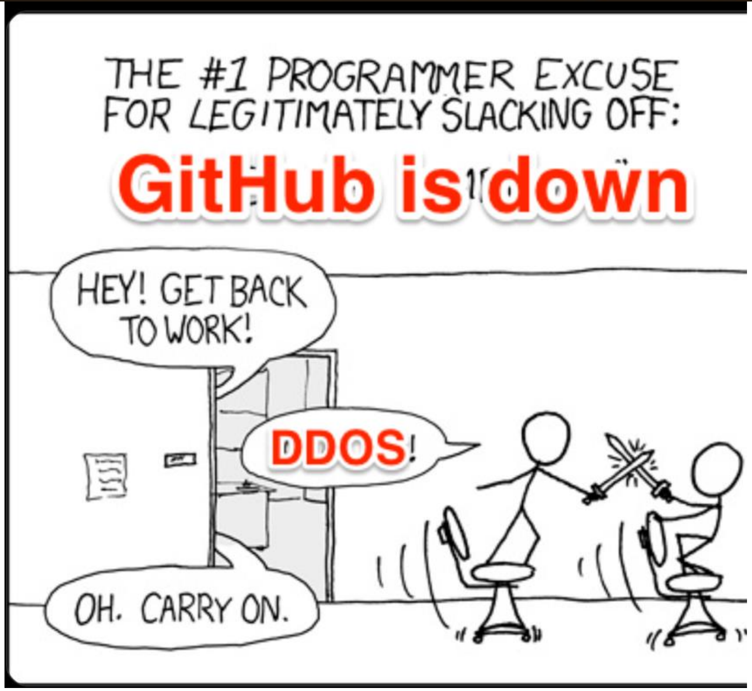
Distributed Denial-of-service (DDoS)

- DDoS: Network oriented attacks aimed at preventing access to network, host or service
  - Saturate the target's network with traffic
  - Consume all network resources (e.g., SYN flooding)
  - Overload a service with requests
    - Use "expensive" requests (e.g., "sign this data")
  - Can be extremely costly
- Result: service/host/network is unavailable
- Criminals sometimes use DDoS for racketeering (e.g., Mirai)
- Note: IP addresses of perpetrators are often hidden (spoofed)



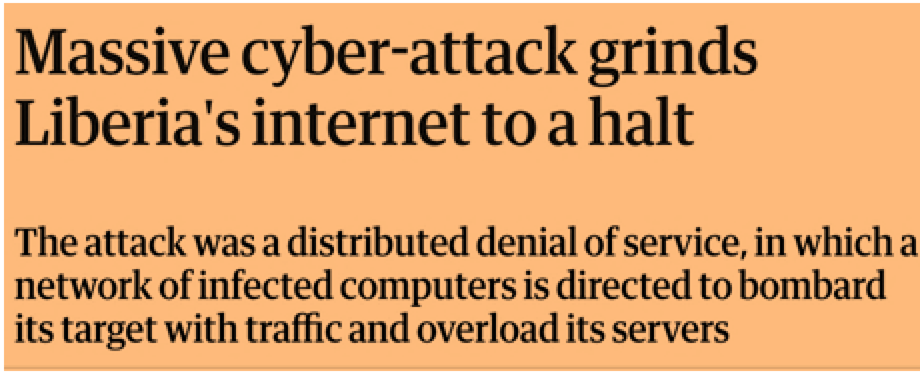
February 28th DDoS Incident Report

On Wednesday, February 28, 2018 GitHub.com was unavailable from 17:21 to 17:26 UTC and intermittently unavailable from 17:26 to 17:30 UTC due to a distributed denial-of-service (DDoS) attack.



(D)DoS Techniques 101
(Don't do these.)

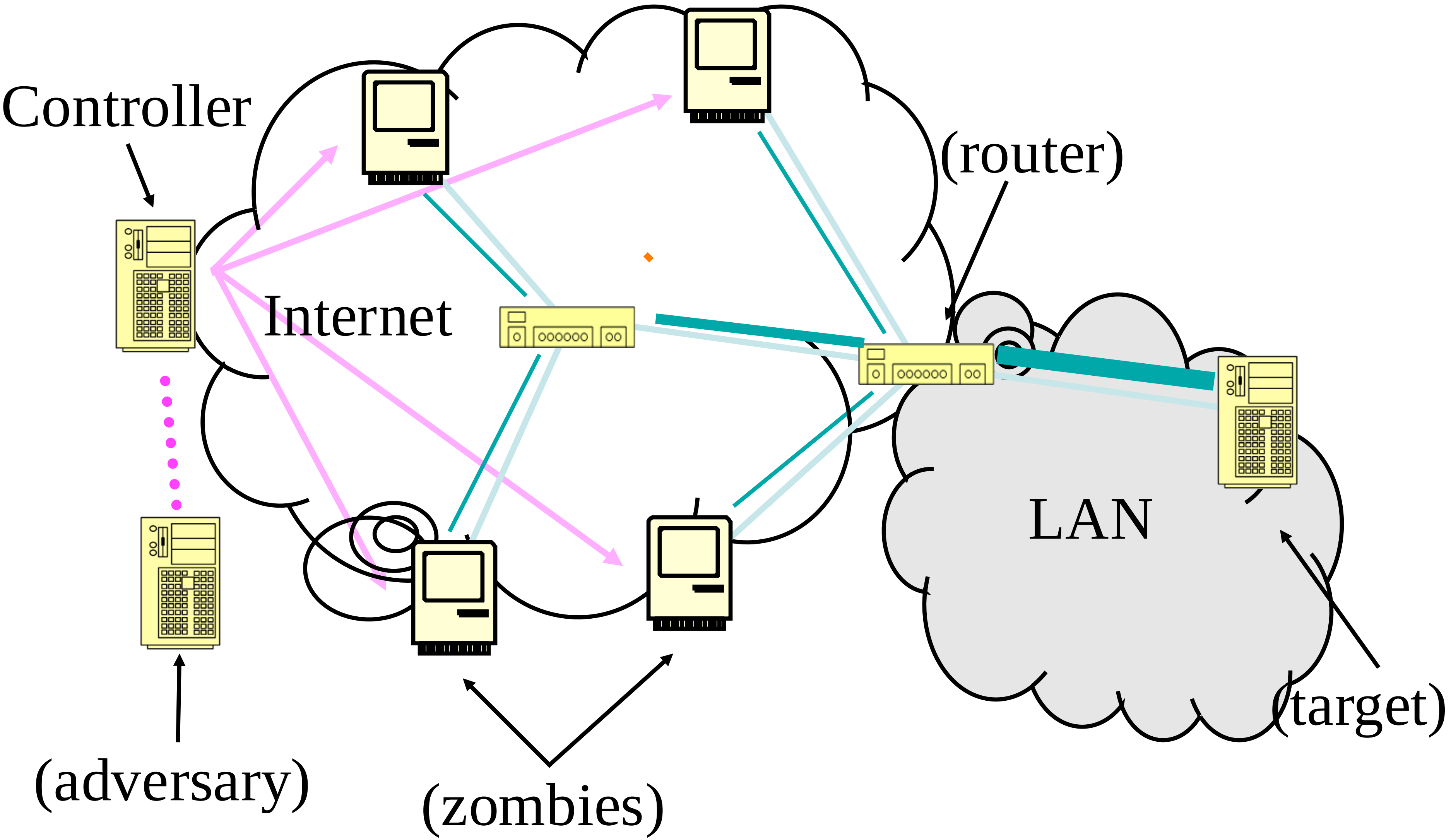
- Send a stream of legitimate requests
- Send a few malformed packets
  - causing failures or expensive error handling
  - low-rate packet dropping (TCP congestion control)
  - "ping of death"
- Abuse legitimate access
  - Compromise service/host
  - Use its legitimate access rights to consume the rights for domain (e.g., local network)



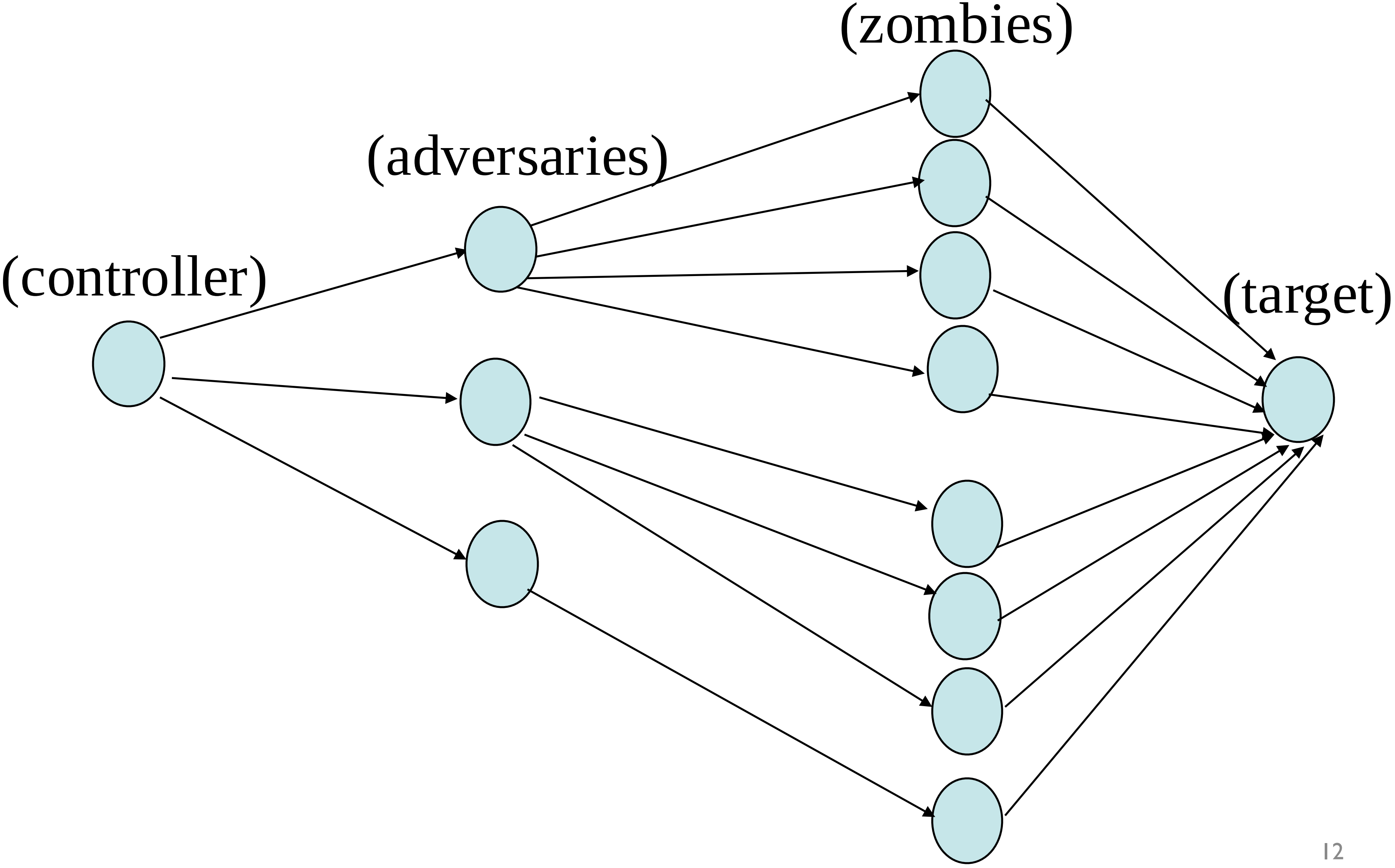
Massive cyber-attack grinds Liberia's internet to a halt

The attack was a distributed denial of service, in which a network of infected computers is directed to bombard its target with traffic and overload its servers

The canonical DDoS attack



Adversary Network



Why DDoS?

- Motivations:
  - An axe to grind
  - Curiosity (script kiddies)
  - Blackmail / racketeering
  - Information warfare
  - Distraction

Q: An easy fix?

- How do you solve distributed denial of service?

Simple DDoS Mitigation

- **Ingress/Egress Filtering**: Inspect and filter incoming/outgoing network packets
  - Helps spoofed sources, not much else
- **Ingress filtering:**
  - Verify if the source IP address of incoming traffic is valid
  - E.g., if a packet from over the internet arrives with an internal IP (192.168.x.x, 10.x.x.x), the packet can be dropped
  - Prevents the device from becoming a target
- **Egress filtering**:
  - Verify if traffic going out of the network has a source IP that is not in network
  - E.g., a network has IP range 131.247.x.x/x, drop all outgoing packets with IP outside of this range
  - Prevents the device from becoming an amplifier
- **Challenges**: Overhead, Misconfiguration, Scalability in dynamic environment (e.g. increasing no. of devices, diverse config requirements, cloud servers on VMs hosted dynamically)

Pushback

- Initially, detect the DDoS and flag the sources/types/links of DDoS traffic
- **Pushback** on upstream routers
  - Contact upstream routers using PB protocol
  - Indicate some filtering rules (based on observed flows)
- Repeat as necessary towards sources
- Focus is on stopping malicious traffic closer to the source of that traffic rather than at the victim's end.
- Works well in wonderful magic land where it rains chocolate and doughnuts (/s)

Traceback

- With small probability (e.g., 1/20,000), routers include identity of previous hop with packet data
- For large flows, targets can reconstruct path to source
- Statistics say that the path will be exposed
- Focus is on identification of true origin of spoofed packets to help with filtering rules.

DDoS Reality

- None of the “protocol oriented” solutions have really seen any adoption
  - too many untrusting, ill-informed, mutually suspicious parties must play together
- Real Solution (or reality)
  - Large ISPs police their ingress/egress points very carefully
  - Watch for DDoS attacks and filter appropriately
  - Develop products that coordinate view from many vantage points in the network to identify upswings in traffic

Botnets

Botnets

- A **botnet** is a network of software robots (bots) run on **zombie machines** which are controlled by **command and control** networks
  - **IRCbots** - command and control over IRC (one of the first avenues for botnets)
  - **Bot master** - owner/controller of network

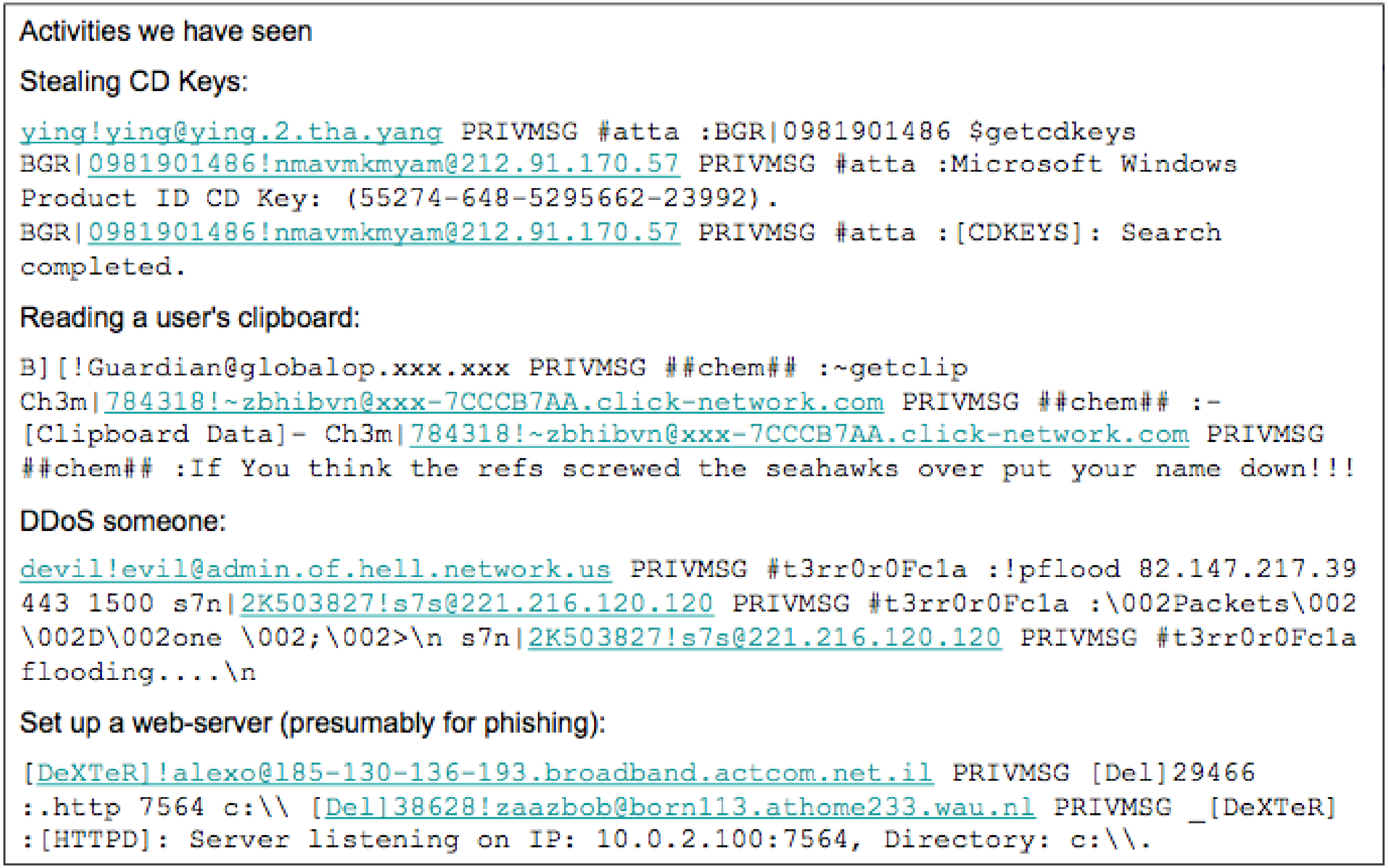
What are botnets being used for?

piracy

mining

attacks

hosting



Activities we have seen

Stealing CD Keys:

```
ying!ying@ying.2.tha.yang PRIVMSG #atta :BGR|0981901486 $getcdkeys
BGR|0981901486!nmavmkmyam@212.91.170.57 PRIVMSG #atta :Microsoft Windows
Product ID CD Key: (55274-648-5295662-23992).
BGR|0981901486!nmavmkmyam@212.91.170.57 PRIVMSG #atta :[CDKEYS]: Search
completed.
```

Reading a user's clipboard:

```
B][!Guardian@globalop.xxx.xxx PRIVMSG ##chem## :~getclip
Ch3m|784318!~zbhibvn@xxx-7CCCB7AA.click-network.com PRIVMSG ##chem## :-
[Clipboard Data]- Ch3m|784318!~zbhibvn@xxx-7CCCB7AA.click-network.com PRIVMSG
##chem## :If You think the refs screwed the seahawks over put your name down!!!
```

DDoS someone:

```
devil!evil@admin.of.hell.network.us PRIVMSG #t3rr0r0Fc1a :!pflood 82.147.217.39
443 1500 s7n|2K503827!s7s@221.216.120.120 PRIVMSG #t3rr0r0Fc1a :\002Packets\002
\002D\002one \002;\002>\n s7n|2K503827!s7s@221.216.120.120 PRIVMSG #t3rr0r0Fc1a
flooding....\n
```

Set up a web-server (presumably for phishing):

```
[DeXTeR]!alexo@185-130-136-193.broadband.actcom.net.il PRIVMSG [Del]29466
:.http 7564 c:\\ [Del]38628!zaazbob@born113.athome233.wau.nl PRIVMSG _[DeXTeR]
:[HTTPD]: Server listening on IP: 10.0.2.100:7564, Directory: c:\\.
```

IRC

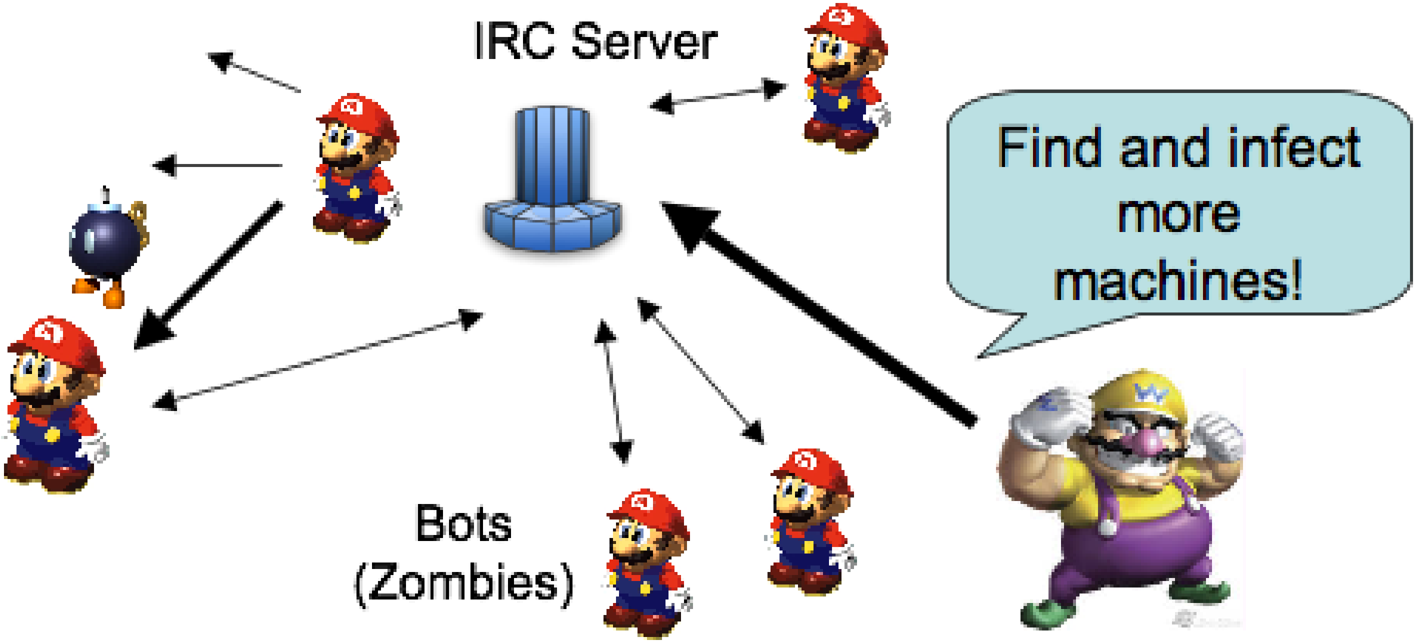
- Internet Relay Chat
  - before AOL chat rooms

AOL Instant Messenger is no more, but the memories will last forever

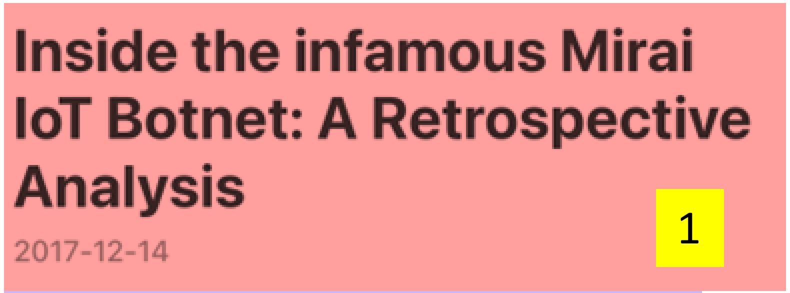
- Supports one-to-many or many-to-many chat
- Supports many **channels** (sometimes password protected)
- Client/server architecture

IRC botnets

- Botmaster creates an IRC server.
- Infected bots are instructed to connect to this IRC server.
- Bots remain idle and wait for Botmaster’s instructions.
- Botmaster sends the command to attack specific victim.

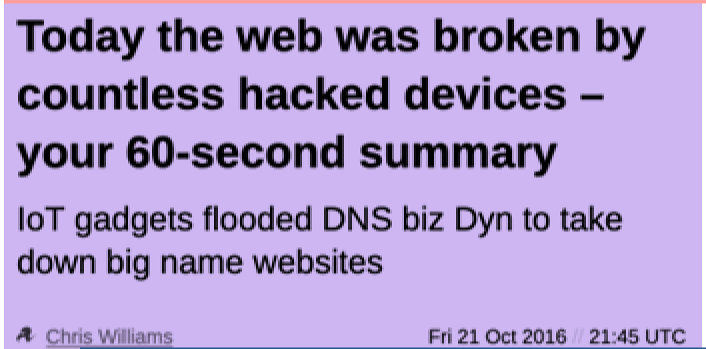


Mirai



Inside the infamous Mirai IoT Botnet: A Retrospective Analysis [1]

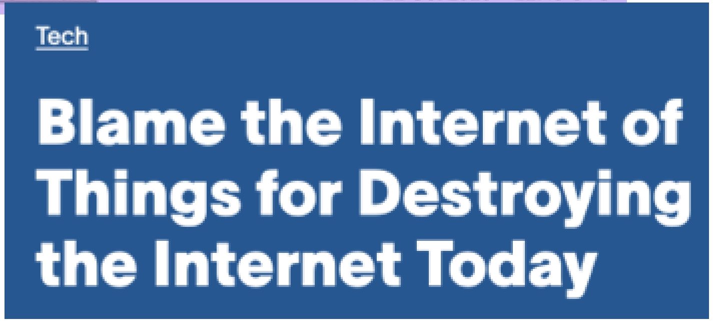
2017-12-14



Today the web was broken by countless hacked devices – your 60-second summary

IoT gadgets flooded DNS biz Dyn to take down big name websites

Chris Williams Fri 21 Oct 2016 // 21:45 UTC



Tech

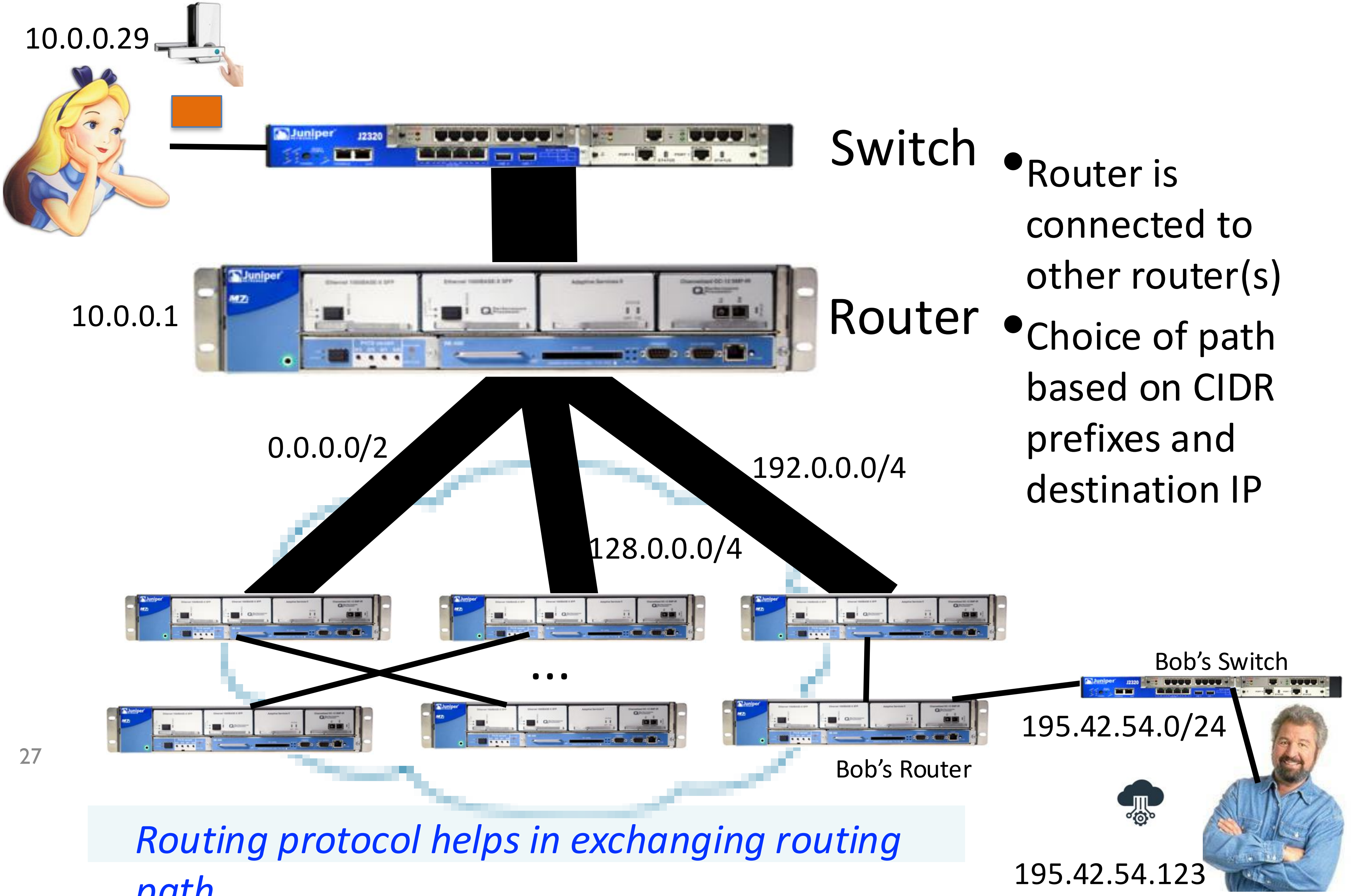
Blame the Internet of Things for Destroying the Internet Today

- Works like a combination of a worm and a botnet
- Self-propagating
- Infects vulnerable IoT devices
- Infected IoT devices are turned into zombies
- C&C servers issue commands to the devices on which victim to target

[1] https://blog.cloudflare.com/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/

Routing

Routing outside of the local subnet



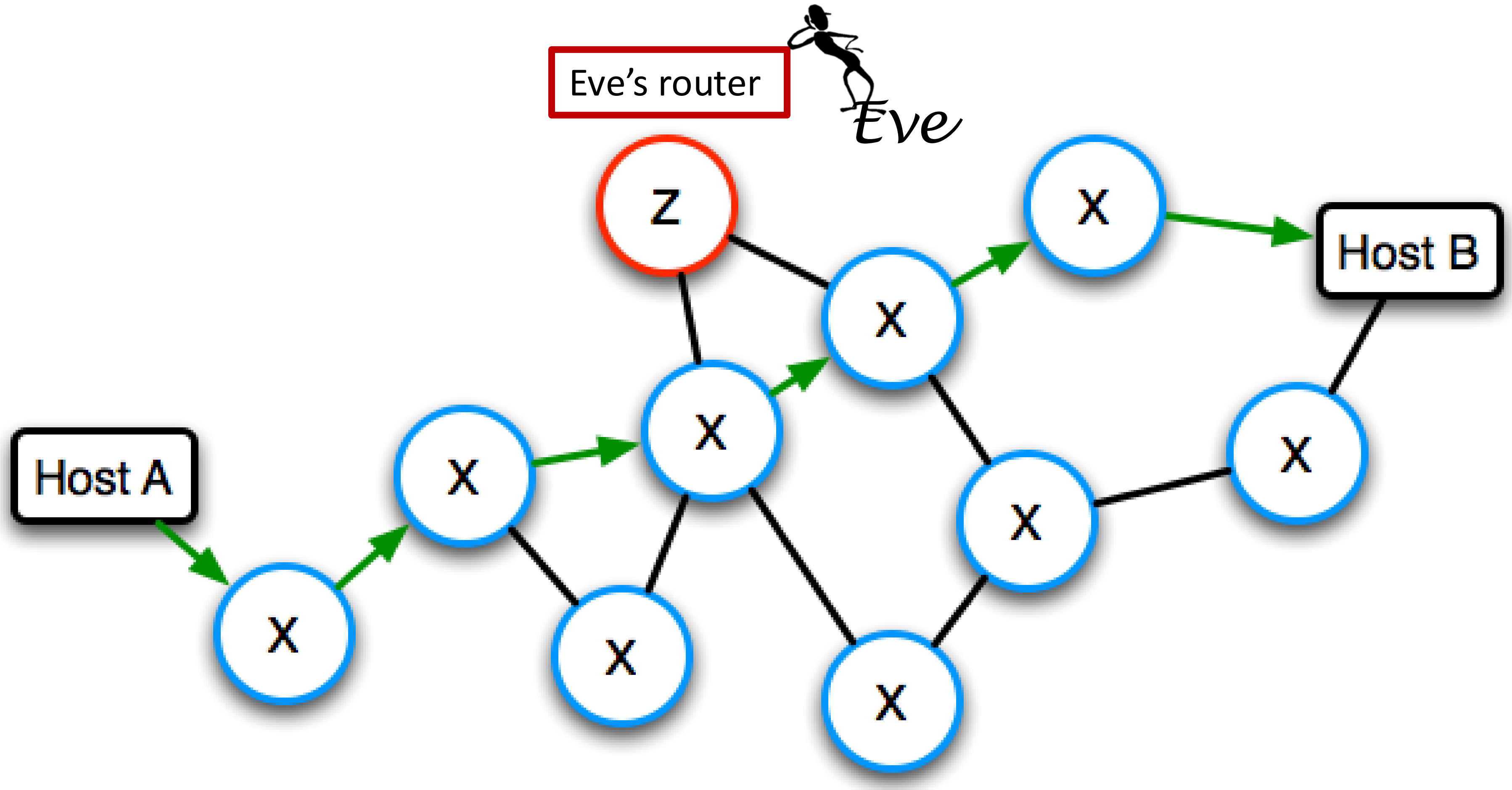
- Router is connected to other router(s)
- Choice of path based on CIDR prefixes and destination IP

Routing protocol helps in exchanging routing path

Routing Security

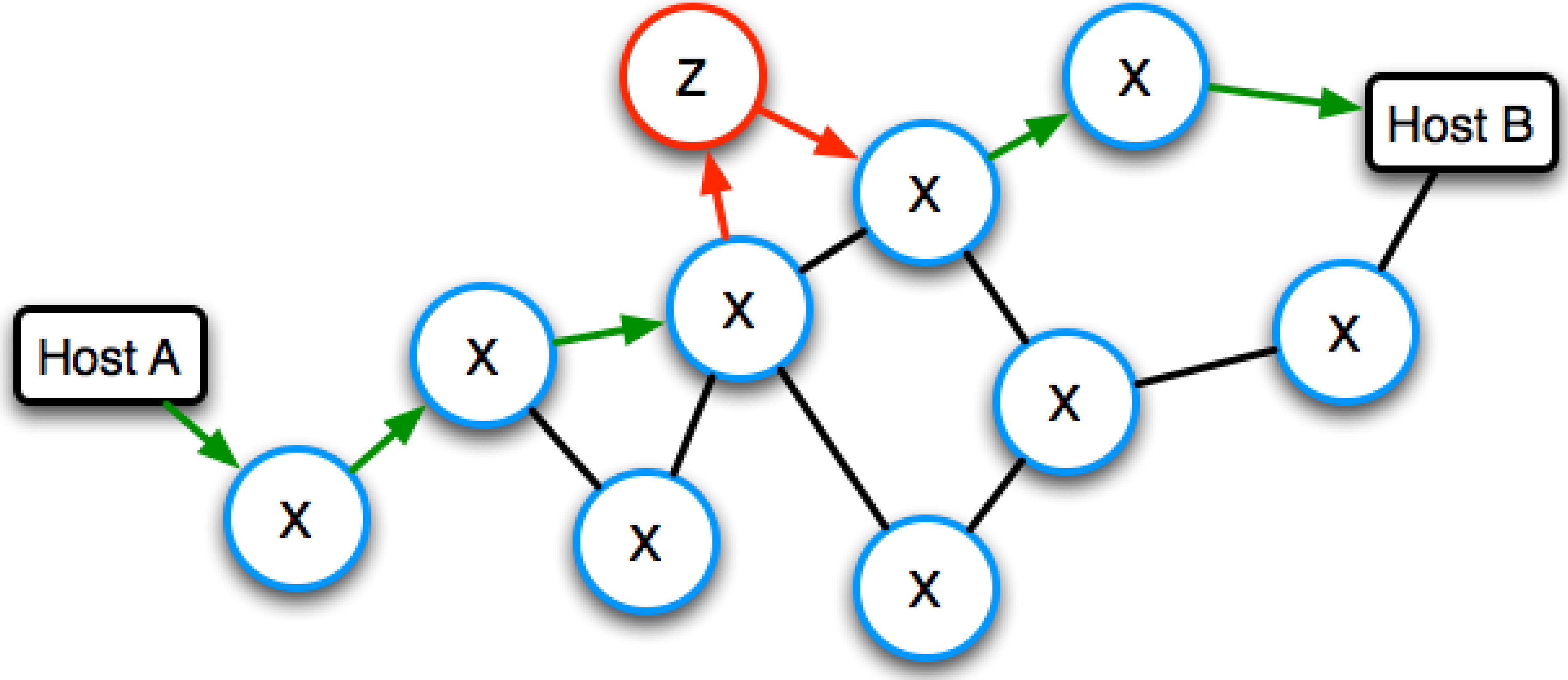
- Bad guys/gals/Internet-enabled-toasters/vacuum-cleaners *can* play games with routing protocols.
- *But why…?*
- Implications for diverted traffic:
  - Enemy can see the traffic.
  - Enemy can easily modify the traffic.
  - Enemy can drop the traffic.
- *Routing security in a nutshell*: Cryptography can mitigate effects, but not stop them.

Routing



Routers exchange path information.

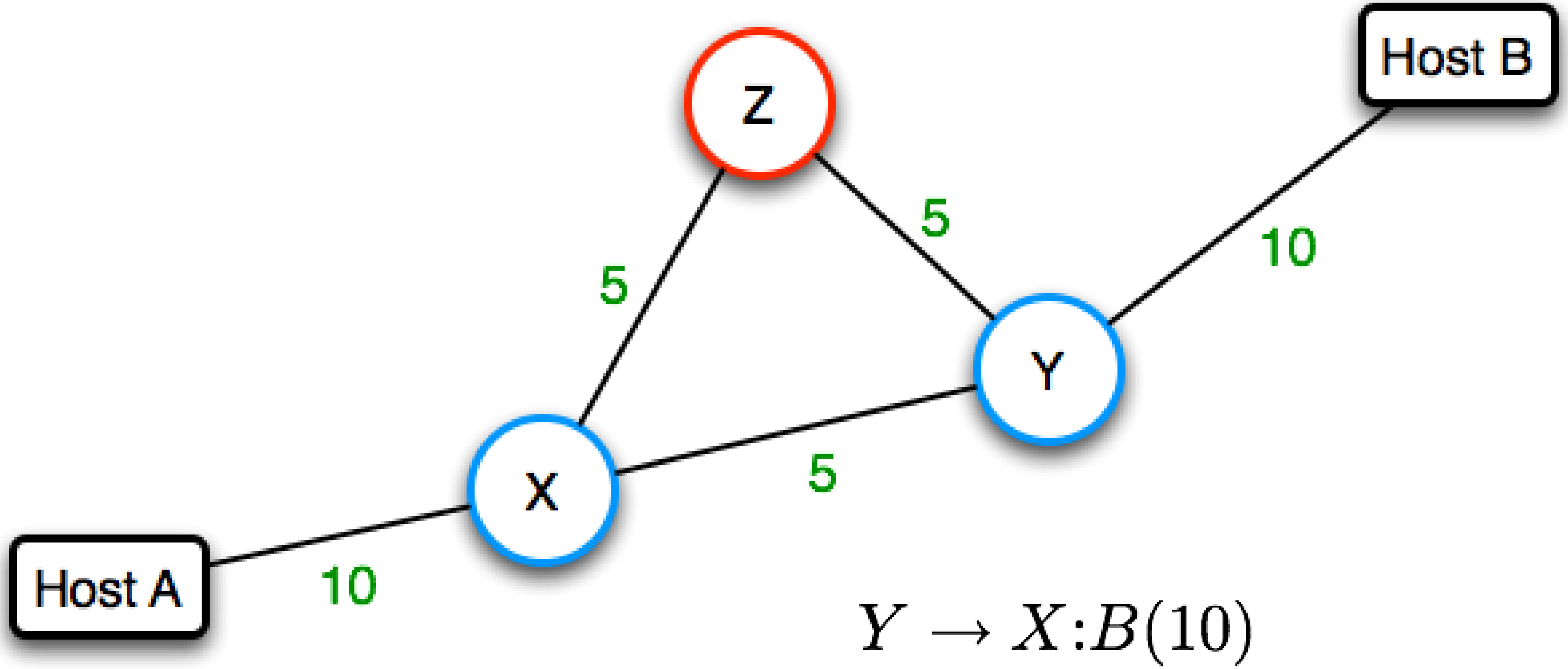
The Enemy's Goal



Routing Protocols

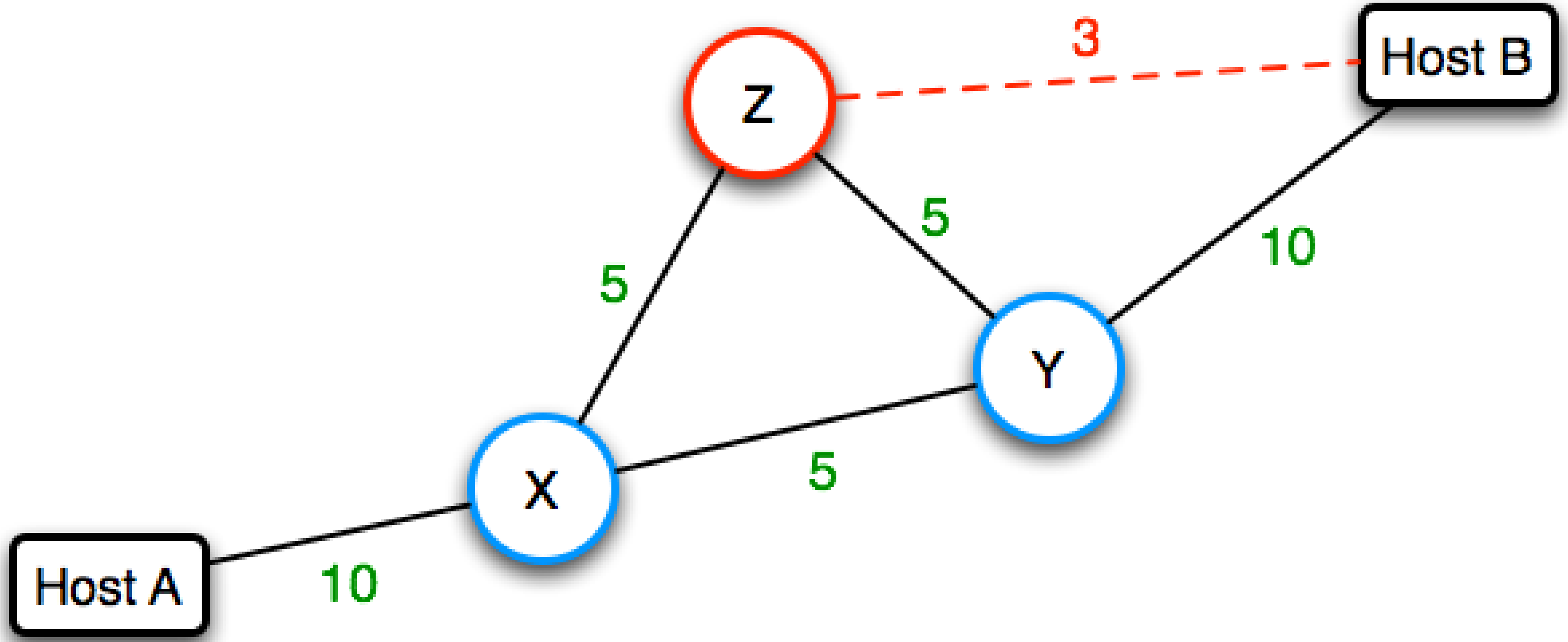
- Routers speak to each other
- They exchange topology and cost information
- Each router calculates the shortest path to each destination
- Routers forward packets along locally shortest path
- Attacker can lie to other routers
- *Examples of routing protocols (OSPF, BGP)*

Normal Behavior



$$
\begin{aligned}
&Y \rightarrow X{:}B(10)\\
&Y \rightarrow Z{:}B(10)\\
&Z \rightarrow X{:}Y(5), B(15)\\
&X \rightarrow A{:}Z(5), Y(5), B(15)
\end{aligned}
$$

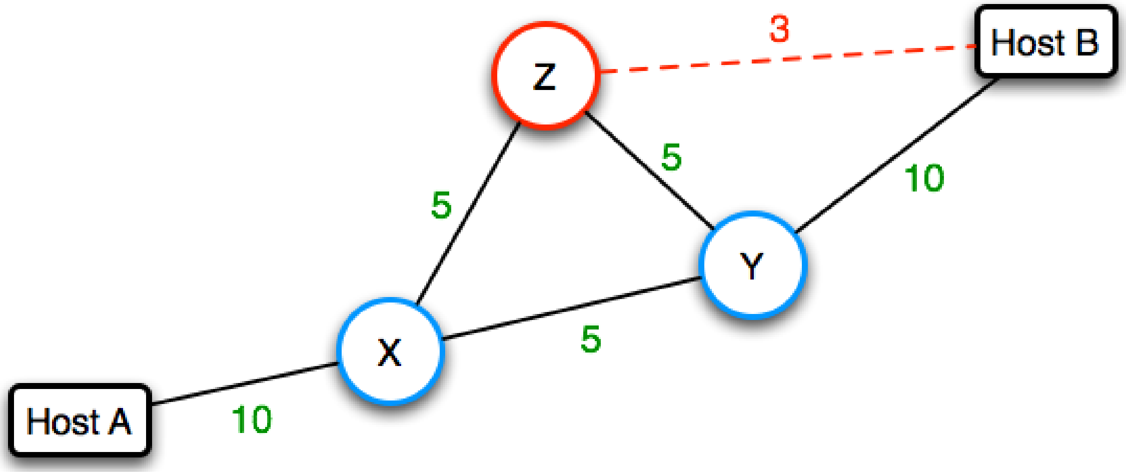
Malicious Behavior



$$Y \rightarrow X\!:\!B(10)$$
$$Y \rightarrow Z\!:\!B(10)$$
$$Z \rightarrow X\!:\!Y(5), B(3)$$
$$X \rightarrow A\!:\!Z(5), Y(5), B(8)$$

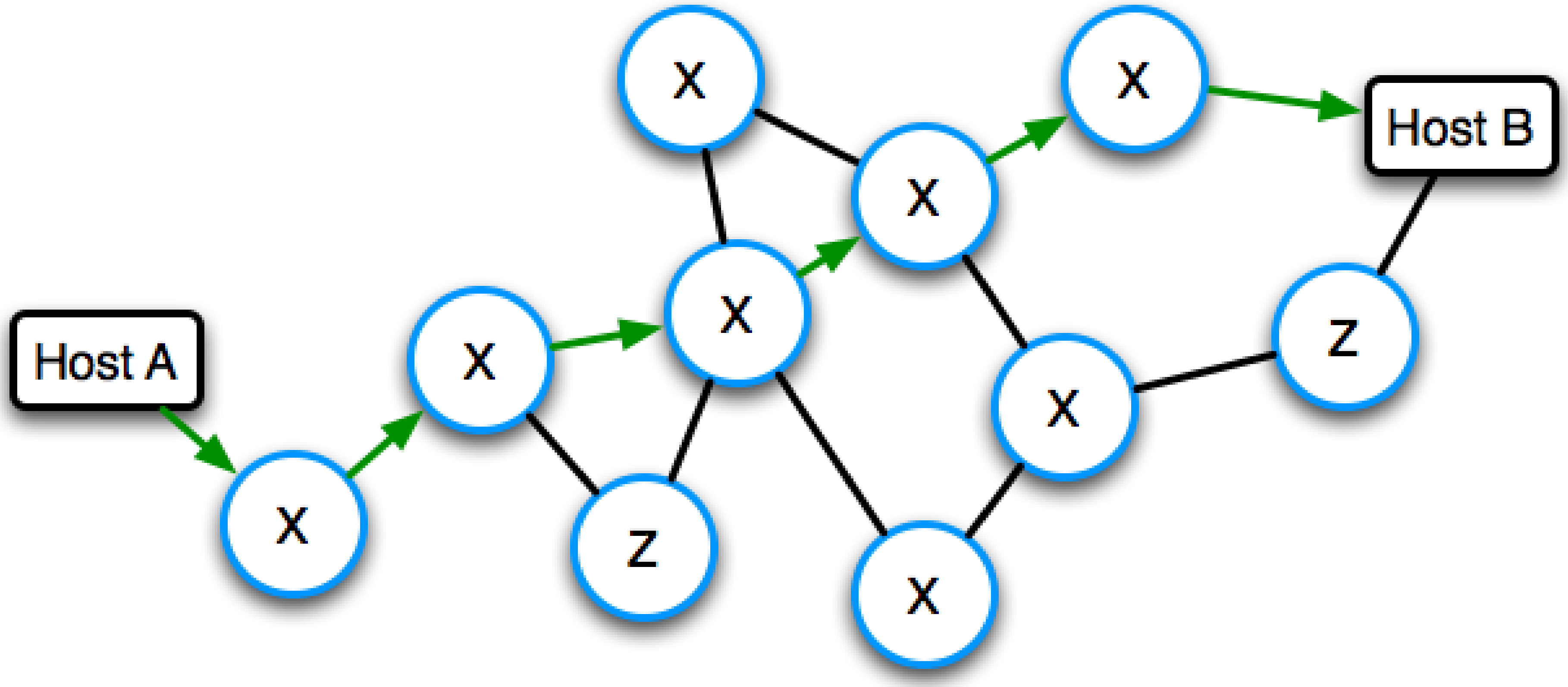
Why is this difficult?

- X (or Y) has no knowledge of Z's real connectivity.
- The problem isn't the link from X to Z:
  - The problem is the lack of integrity of the info being sent
  - Non-trivial complexity: Z might be deceived by some other neighbor Q

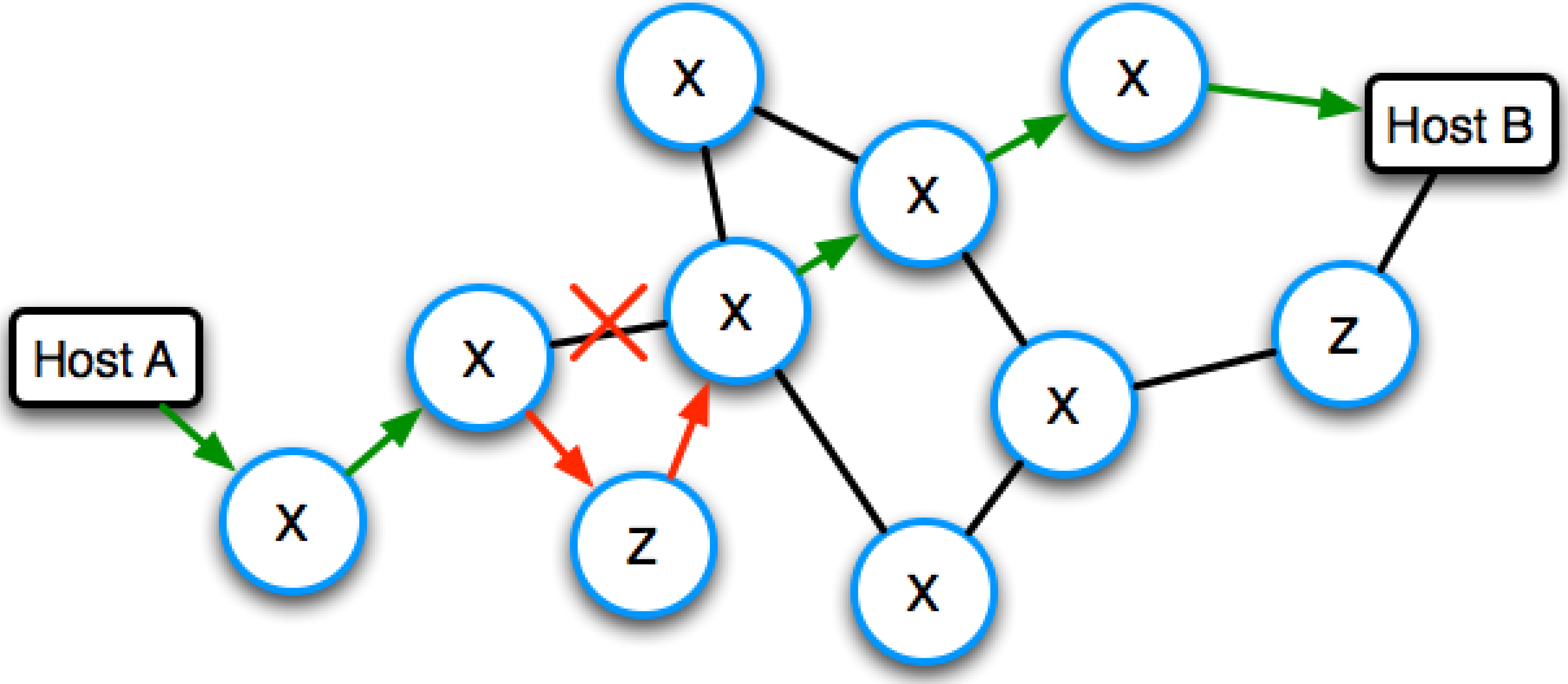


$$Y \rightarrow X : B(10)$$
$$Y \rightarrow Z : B(10)$$
$$Z \rightarrow X : Y(5), B(3)$$
$$X \rightarrow A : Z(5), Y(5), B(8)$$

Link Cutting



Link Cutting



- DoS a router
- Physically cut the path! (physical attacks not considered in our threat model)
- Forge routing message (e.g., intercept to say link no longer available)