Network Security on the Web

COMP 30650: Networks and Internet Systems

Dr. Gavin McArdle
Email: gavin.mcardle@ucd.ie
Office: A1.09 Computer Science

RECAP

Confidentiality

- Encryption
  - Symmetric
  - Public key

Authentication & Integrity

  - MAC
  - Digital Signatures

TODAY’S PLAN

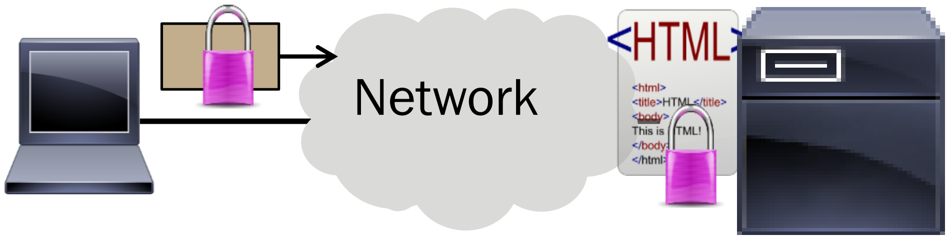
Applying Security

- Security in the Web
- PKI
- DNS Spoofing
  - Security Extensions
- Firewalls
- Distributed Denial of Service/Access
- IP Spoofing

WEB SECURITY - HTTPS

Securing the web

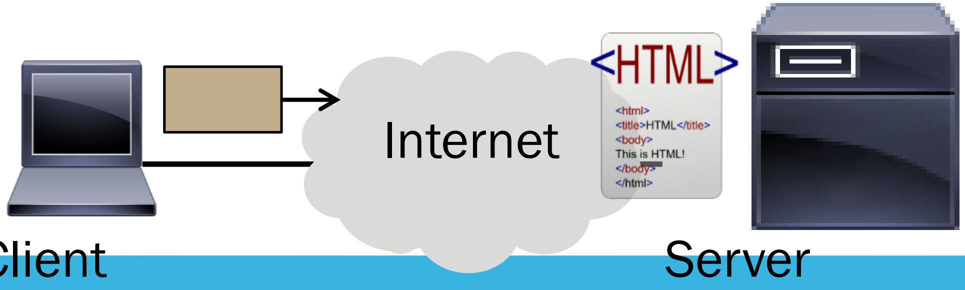
- Focus on SSL/TLS for HTTPS
- **Secure Sockets Layer/Transport Layer Security**
- Based on certificates



GOAL AND THREAT MODEL

Much can go wrong on the web!

- Clients encounter malicious content
- Web servers are target of break-ins
- Fake content/servers trick users
- Data sent over network is stolen …

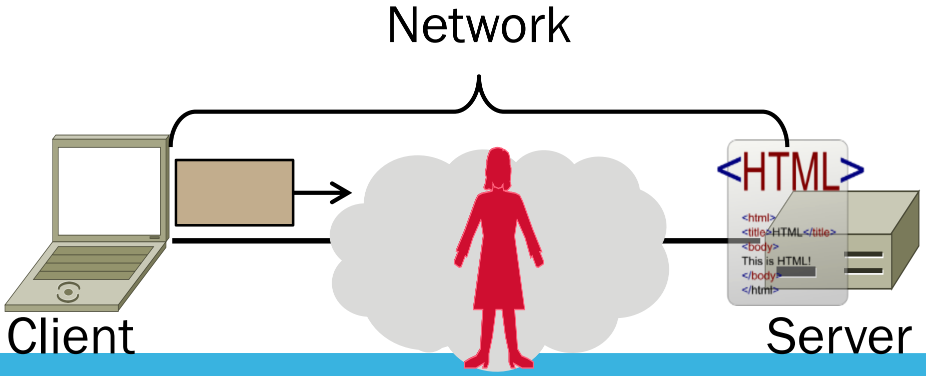


GOAL AND THREAT MODEL

Goal of HTTPS is to secure HTTP

We focus on network threats:

1. Eavesdropping client/server traffic
2. Tampering with client/server traffic
3. Impersonating web servers



HTTPS CONTEXT

HTTPS (HTTP Secure) is an add-on

- Means HTTP over SSL/TLS
- SSL (Secure Sockets Layer) precedes TLS (Transport Layer Security)

Motivated by secure web commerce

- Slow adoption, now widespread use
- Can be used by any app, not just HTTP

SSL/TLS OPERATION

Protocol provides:

1. Verification of identity of server (and optionally client)
2. Message exchange between the two with confidentiality, integrity, authenticity and freshness

Consists of authentication phase (handshake that sets up encryption) followed by data transfer phase

SSL/TLS AUTHENTICATION

Must allow clients to securely connect to servers not used before

- Client must authenticate server
- Server typically doesn't identify client

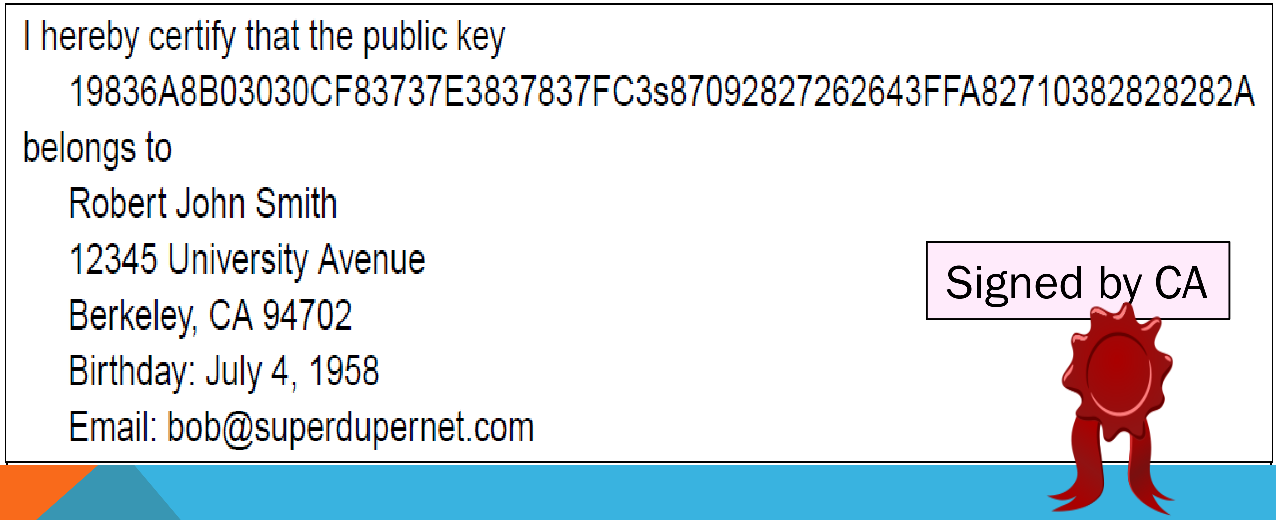
Uses public key authentication

- But how does client get server's key?
- With certificates »

CERTIFICATES

A certificate binds public key to an identity, e.g., domain

- Distributes public keys when signed by a party you trust
- Commonly in a format called X.509



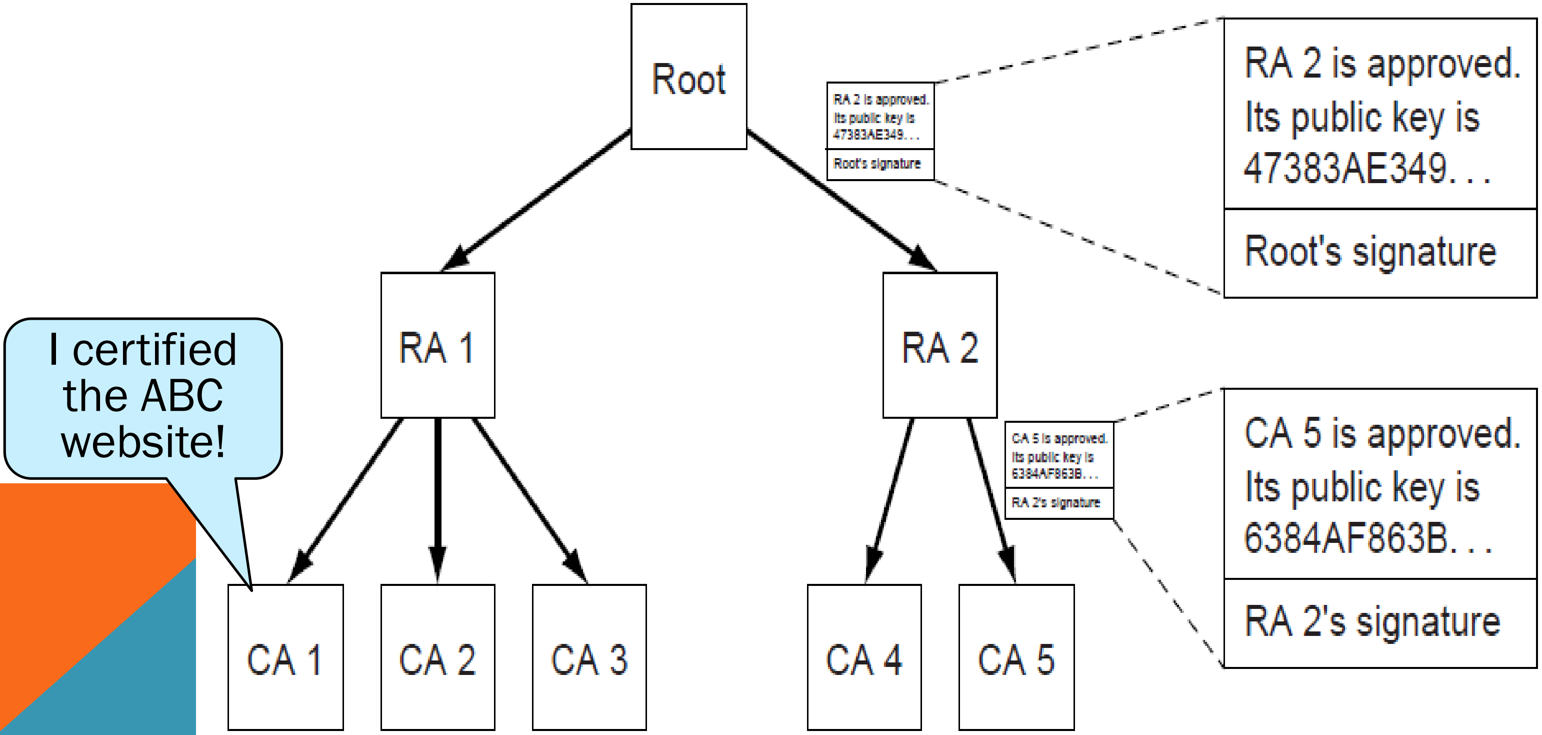
I hereby certify that the public key
19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A
belongs to
Robert John Smith
12345 University Avenue
Berkeley, CA 94702
Birthday: July 4, 1958
Email: bob@superdupernet.com

Signed by CA

PKI (PUBLIC KEY INFRASTRUCTURE)

Adds hierarchy to certificates to let many parties issue

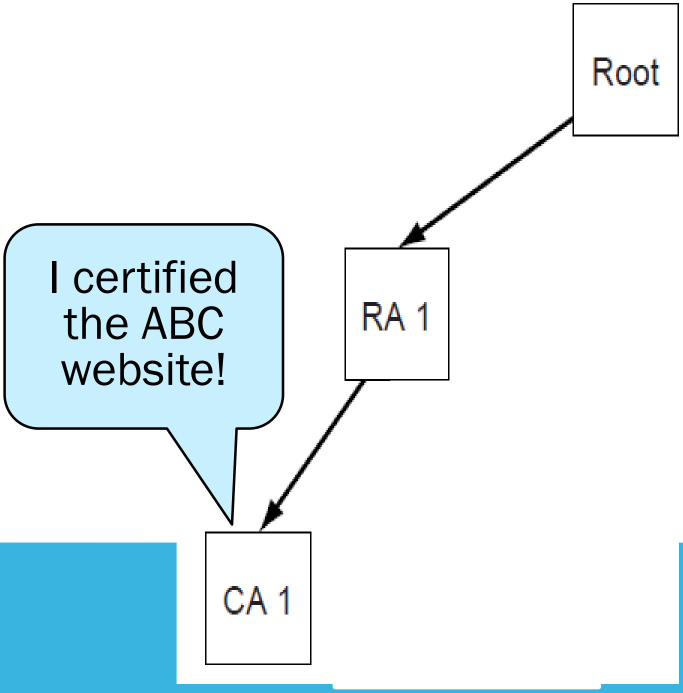
- Issuing parties are called CAs (Certificate Authorities)


Root
RA 2 is approved. Its public key is 47383AE349. . .
Root's signature
RA 2 is approved. Its public key is 47383AE349. . .
Root's signature
RA 1
RA 2
I certified the ABC website!
CA 5 is approved. Its public key is 6384AF863B. . .
RA 2's signature
CA 5 is approved. Its public key is 6384AF863B. . .
RA 2's signature
CA 1
CA 2
CA 3
CA 4
CA 5

PKI

Need public key of PKI root and trust in servers on path to verify a public key of website ABC

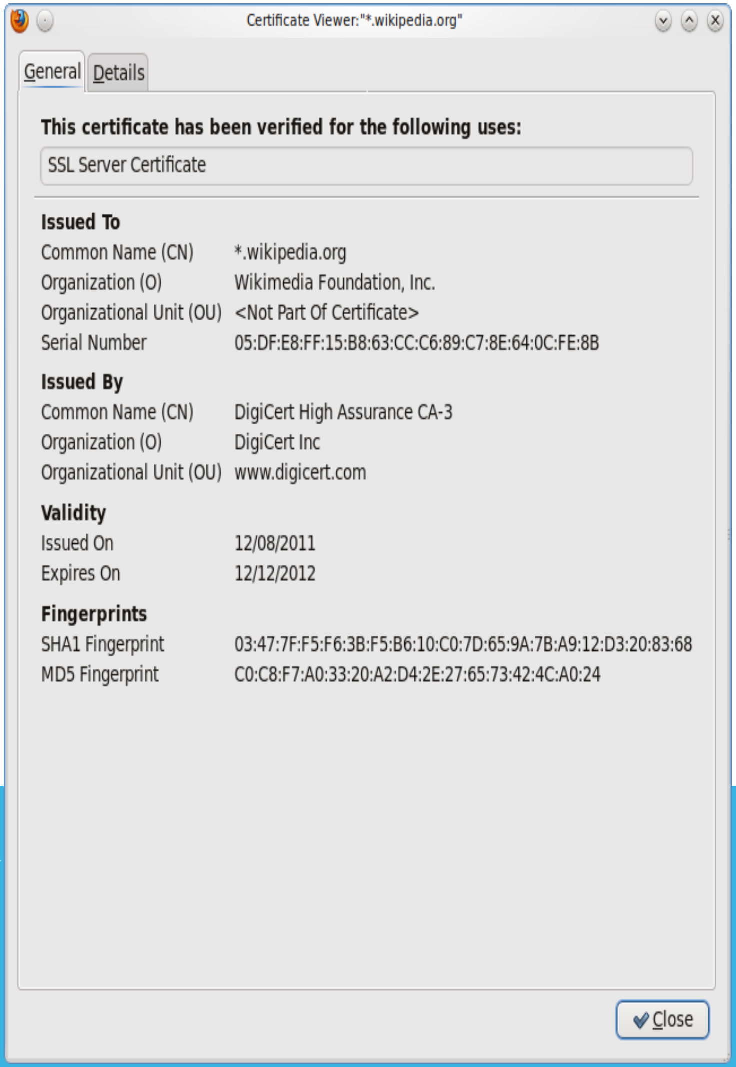
- Browser has Root’s public key
- {RA1’s key is X} signed Root
- {CA1’s key is Y} signed RA1
- {ABC’s key Z} signed CA1



PKI

Browser/OS has public keys of the trusted roots of PKI

- >100 root certificates!
- That’s a problem …
- Inspect your web browser



This certificate has been verified for the following uses:

SSL Server Certificate

Issued To

Common Name (CN)	*.wikipedia.org
Organization (O)	Wikimedia Foundation, Inc.
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	05:DF:E8:FF:15:B8:63:CC:C6:89:C7:8E:64:0C:FE:8B

Issued By

Common Name (CN)	DigiCert High Assurance CA-3
Organization (O)	DigiCert Inc
Organizational Unit (OU)	www.digicert.com

Validity

Issued On	12/08/2011
Expires On	12/12/2012

Fingerprints

SHA1 Fingerprint	03:47:7F:F5:F6:3B:F5:B6:10:C0:7D:65:9A:7B:A9:12:D3:20:83:68
MD5 Fingerprint	C0:C8:F7:A0:33:20:A2:D4:2E:27:65:73:42:4C:A0:24

Certificate for wikipedia.org issued by DigiCert

PKI

Real-world complication:

- Public keys may be compromised
- Certificates must then be revoked

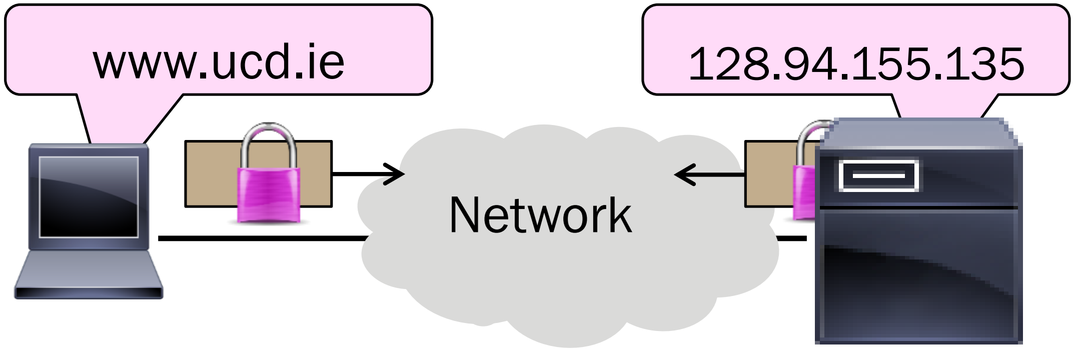
PKI includes a CRL (Certificate Revocation List)

- Browsers use to weed out bad keys

DNS SECURITY

Securing Internet naming

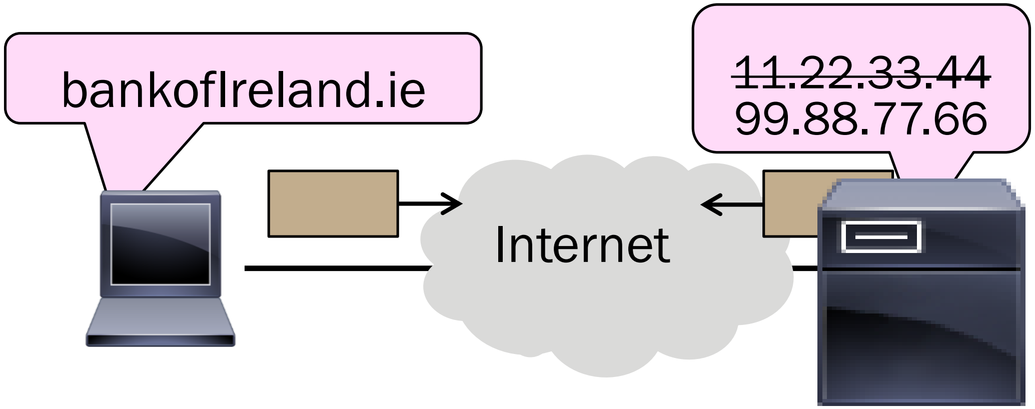
- DNS security extensions (DNSSEC)



GOAL AND THREAT MODEL

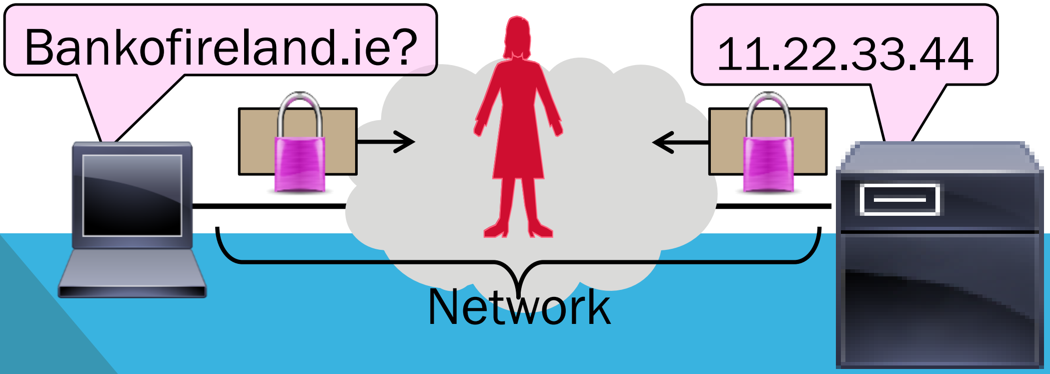
Naming is a crucial Internet service

- Binds host name to IP address
- Wrong binding can be disastrous ...



GOAL AND THREAT MODEL

- Goal is to secure the DNS so that the returned binding is correct
  - Integrity/authenticity vs confidentiality
- Attacker can intercept/tamper with messages on the network



DNS SPOOFING

How can a network attacker corrupt the DNS?

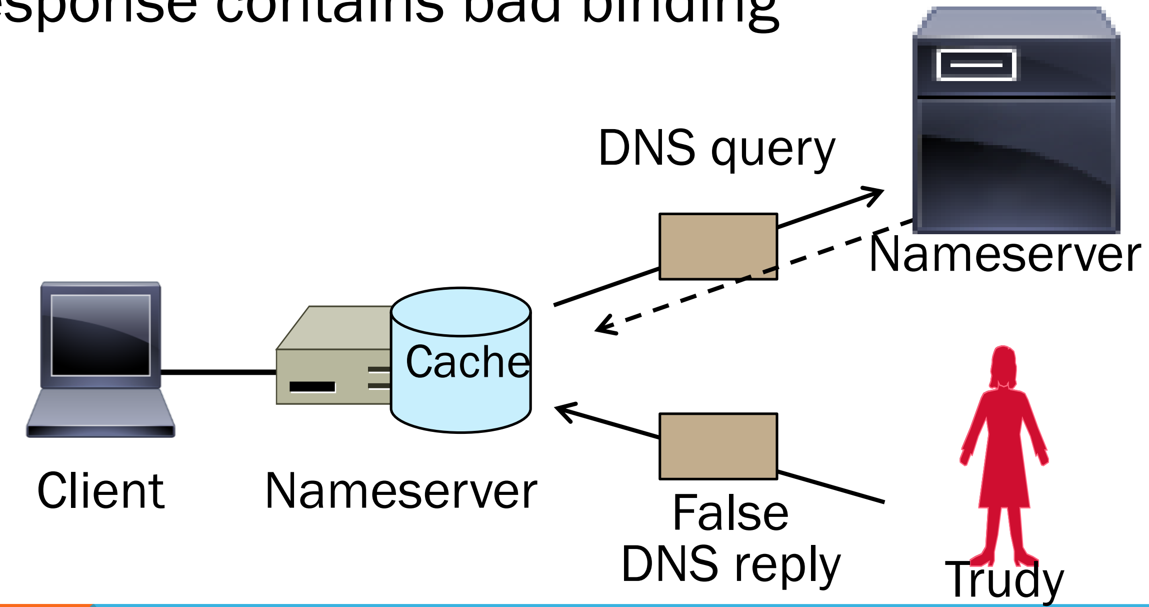
Trudy can trick a nameserver into caching the wrong binding

- By using the DNS protocol itself
- This is called DNS spoofing

DNS SPOOFING

Trudy returns a fake DNS response that appears to be true

- Fake response contains bad binding



DNS SPOOFING

Lots of questions!

1. How does Trudy know when the DNS query is sent and what it is for?
2. How can Trudy supply a fake DNS reply that appears to be real?
3. What happens when the real DNS reply shows up?

There are solutions to each issue …

DNS SPOOFING

Lots of questions!

1. How does Trudy know when the DNS query is sent and what it is for?
   - Trudy can make the query herself!
   - Nameserver works for many clients
   - Trudy is just another client
2. How can Trudy supply a fake DNS reply that appears to be real?
3. What happens when the real DNS reply shows up?

There are solutions to each issue ...

DNS SPOOFING

Lots of questions!

1. How does Trudy know when the DNS query is sent and what it is for?
2. How can Trudy supply a fake DNS reply that appears to be real?
   - A bit more difficult. DNS checks:
     - Reply is from authoritative nameserver (e.g., .com)
     - Reply ID that matches the request
     - Reply is for outstanding query
3. What happens when the real DNS reply shows up?

There are solutions to each issue …

DNS SPOOFING

Lots of questions!

1. How does Trudy know when the DNS query is sent and what it is for?
2. How can Trudy supply a fake DNS reply that appears to be real?
   - Put IP of authoritative nameserver as the source IP address
   - ID is 16 bits (64K). Send many guesses! (Or if a counter, simple to predict.)
   - Send reply right after query
3. What happens when the real DNS reply shows up?

There are solutions to each issue …

DNS SPOOFING

Lots of questions!

1. How does Trudy know when the DNS query is sent and what it is for?
2. How can Trudy supply a fake DNS reply that appears to be real?
3. What happens when the real DNS reply shows up?
   - There is no outstanding query after fake reply is accepted
   - So real reply will be discarded

DNSSEC (DNS SECURITY EXTENSIONS)

As well as the usual A, NS records to map a domain name to IP address, DNSSEC extends DNS with new record types

- RRSIG for digital signatures of records
- DNSKEY for public key validation
- DS for public keys for delegation

Clients query DNS as usual, then validate replies to check that content is authentic

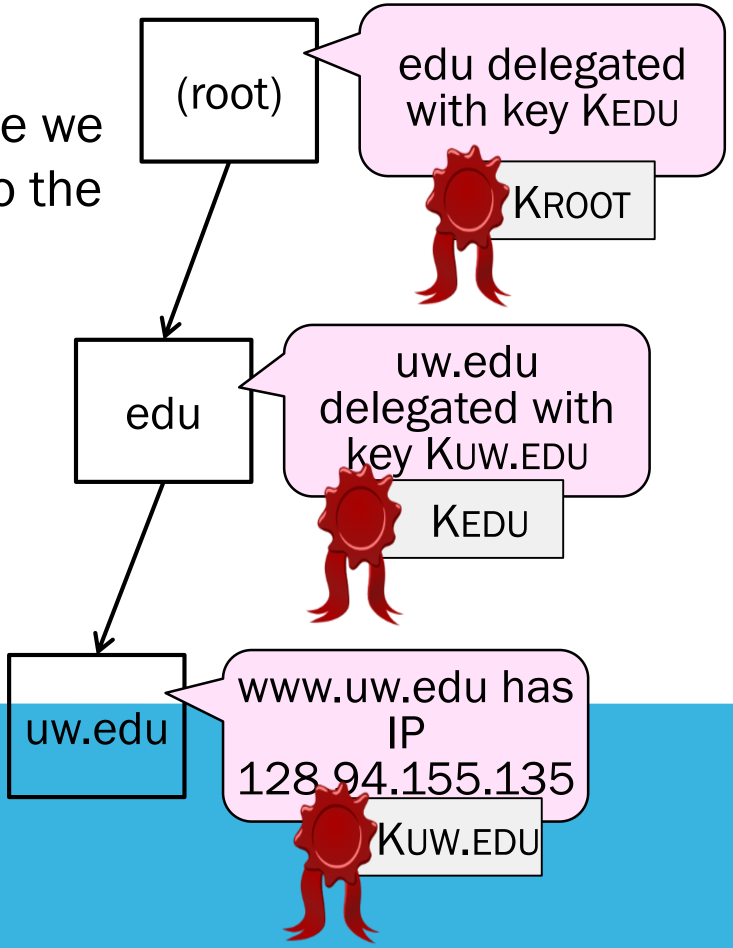
DNSSEC – VALIDATING REPLIES

Client queries www.uw.edu as usual

- Replies include signatures/keys
- Same as we have seen. We need to make sure we can trust the keys so we have to validate up to the root node which is trusted

Client validates answer:

1. KROOT is a trust anchor
2. Use KROOT to check KEDU
3. Use KEDU to check KUW.EDU
4. Use KUW.EDU to check IP



(root)

edu delegated with key KEDU

KROOT

edu

uw.edu delegated with key KUW.EDU

KEDU

uw.edu

www.uw.edu has IP 128.94.155.135

KUW.EDU

FIREWALLS

Firewalls

- Protecting hosts by restricting network connectivity

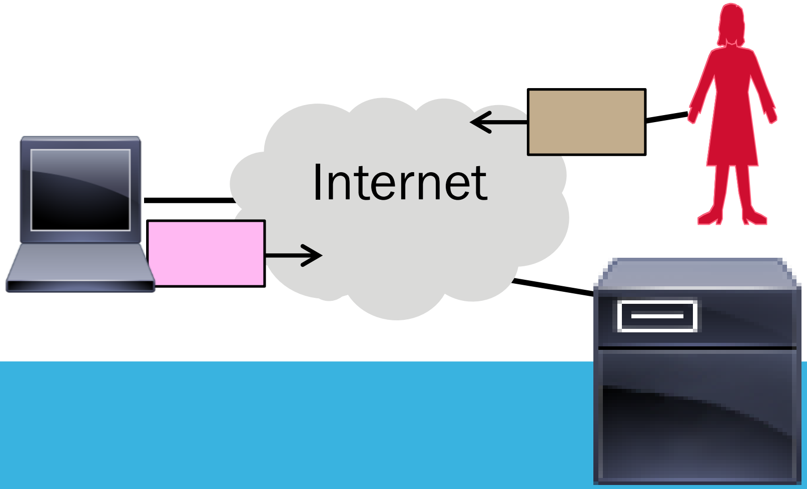
MOTIVATION

The best part of IP connectivity

- You can send to any other host

The worst part of IP connectivity

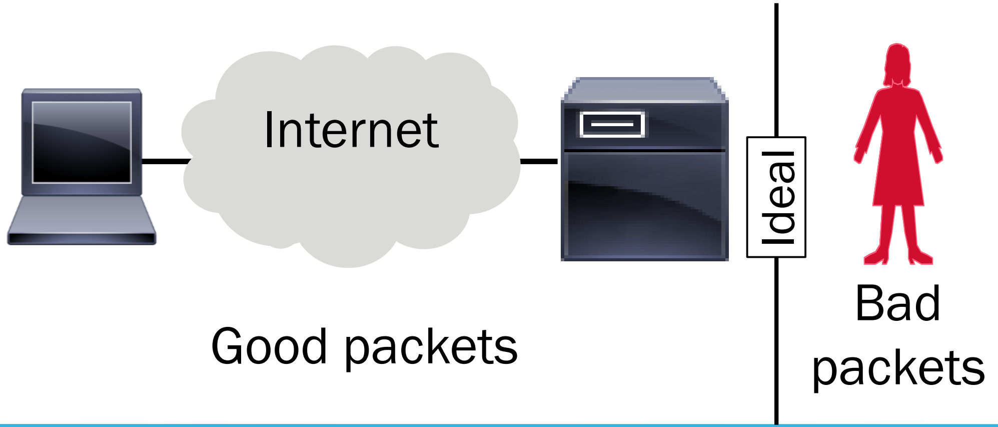
- Any host can send packets to you!
- There’s nasty stuff out there …



GOAL AND THREAT MODEL

Goal of firewall is to implement a boundary to restrict IP connectivity:

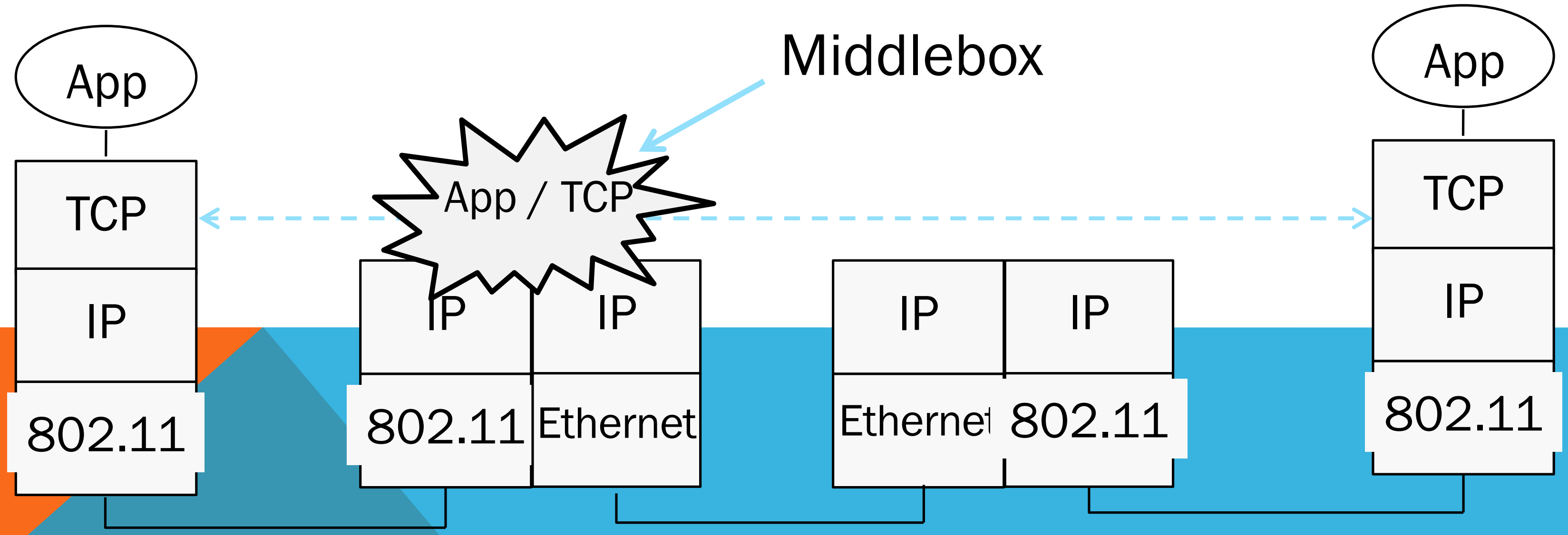
- You can talk to hosts as intended
- Trudy can't talk to you over network



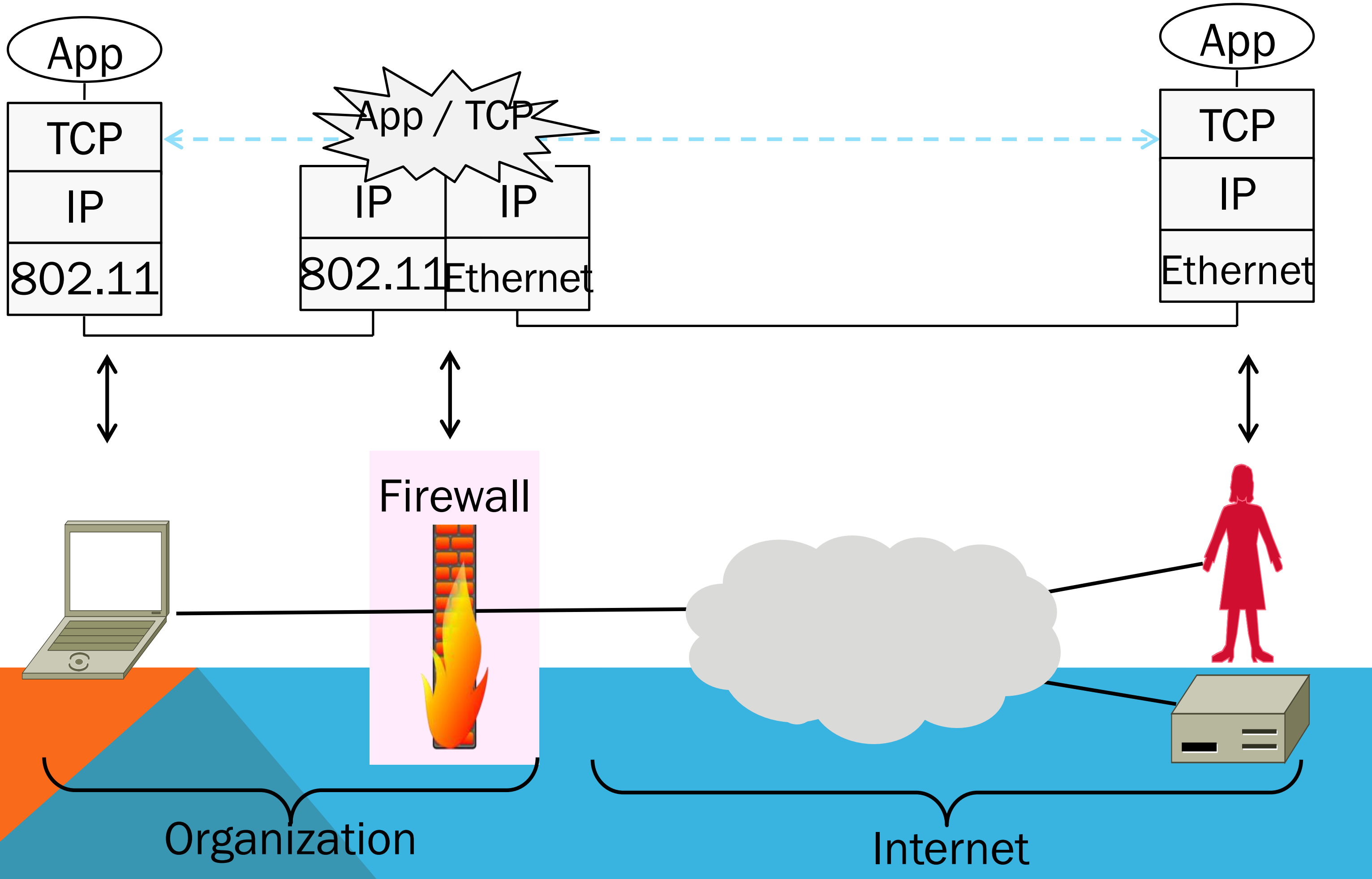
RECALL MIDDLEBOXES

Sit "inside the network" but perform "more than IP" processing on packets to add new functionality

- NAT box, Firewall / Intrusion Detection System



FIREWALL AS MIDDLEBOX



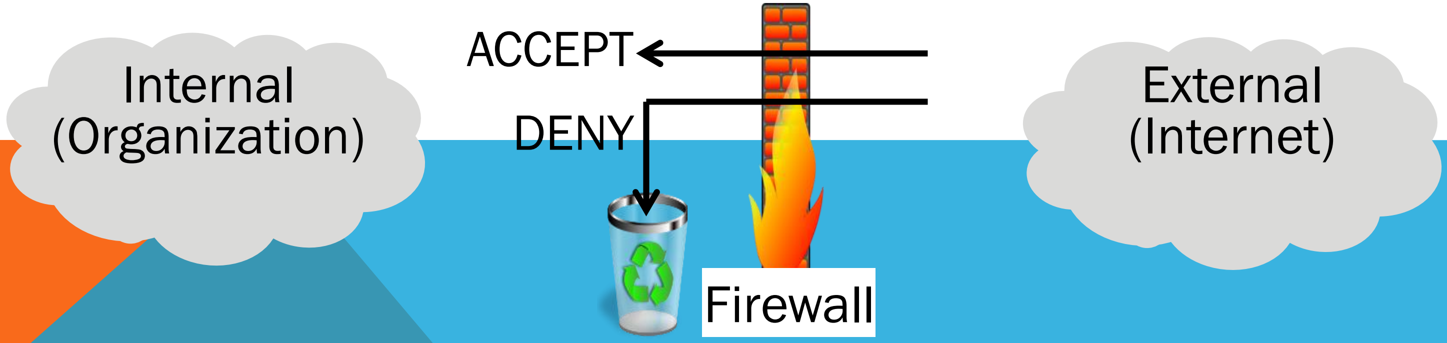
OPERATION

Firewall has two sides:

- Internal (organization) and external (Internet)

For each packet that tries to cross, decide whether to:

- ACCEPT = pass unaltered; or DENY = discard silently
- Decision is a local policy; firewall centralizes IT job



DESIGN

Key tension:

- How to translate desired policies into packet filtering rules

Policies are high-level statements

- Relate to usage of apps, content

Packet filtering is low-level

- Limited viewpoint in the network, e.g., no app messages, encryption

DESIGN

Stateless firewall

- Simplest kind of firewall
- Implements static packet filter rules
- Typically using TCP/UDP ports
- E.g., deny TCP port 23 (telnet)
- Can allow/disallow many types of services and destinations

DESIGN

Stateful firewall

- A step up from stateless
- Implements stateful packet filter rules that track packet exchanges
- NAT example: accept incoming TCP packets after internal host connects

DESIGN

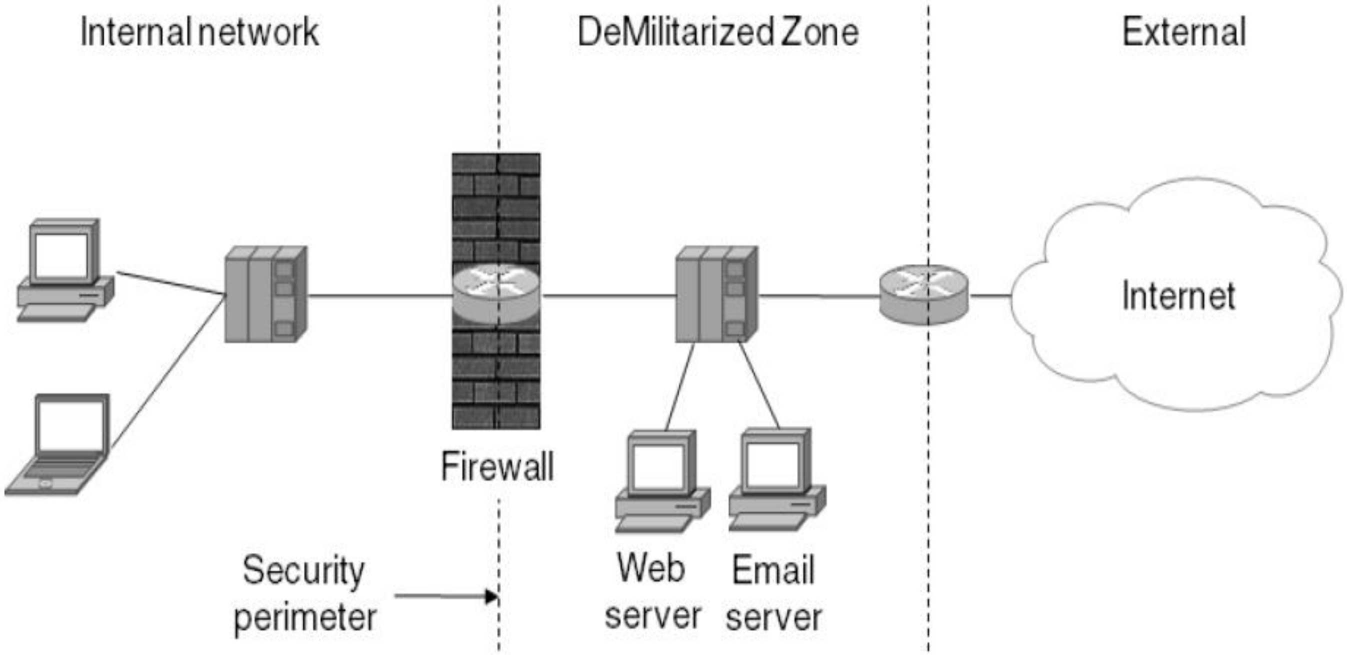
Application layer firewall:

- Another step up
- Implements rules based on app usage and content
- E.g., inspect content for viruses
- Tries to look beyond packets by emulating higher layers, e.g., by reassembling app messages

DEPLOYMENT

Firewall is placed around internal/external boundary

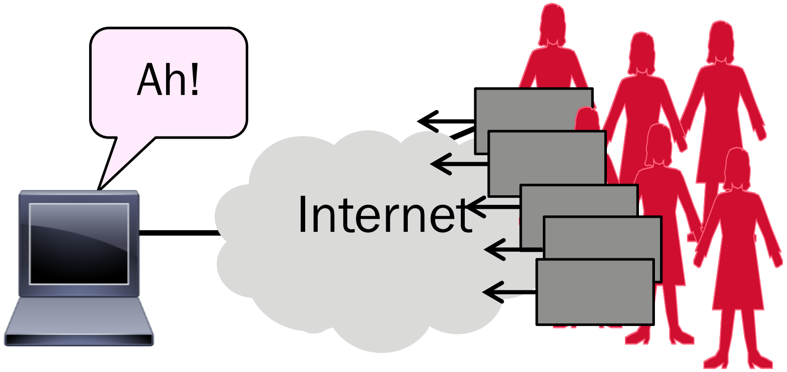
- Classic setup includes DMZ (DeMilitarized Zone) to put busy Internet hosts on the outside for better separation



DISTRIBUTED DENIAL-OF-SERVICE (DDOS)

Distributed Denial-of-Service (DDOS)

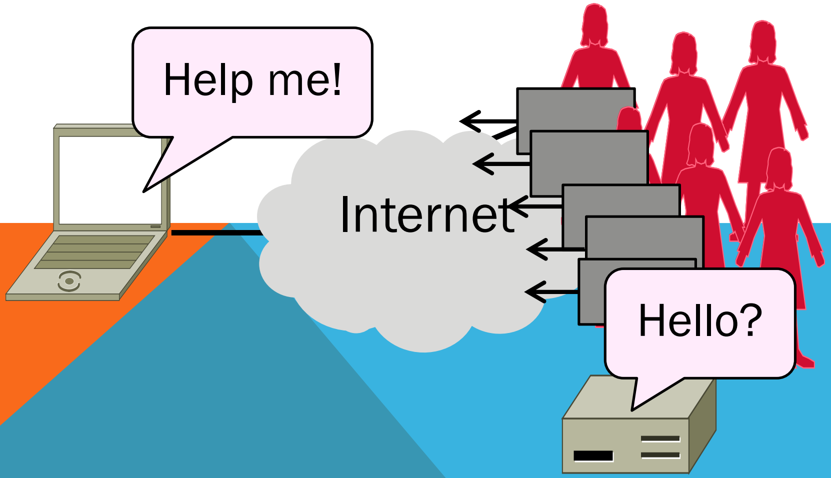
- An attack on network availability



MOTIVATION

Flooding a host with many packets can interfere with its IP connectivity

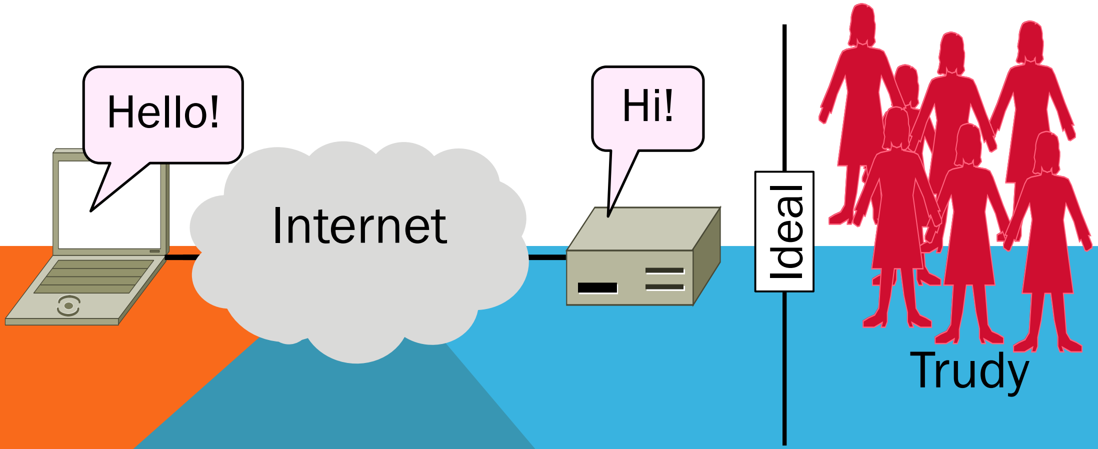
- Host may become unresponsive
- This is a form of denial-of-service



GOAL AND THREAT MODEL

Goal is for host to keep network connectivity for desired services

- Threat is Trudy may overwhelm host with undesired traffic



INTERNET REALITY

Distributed Denial-of-Service is a huge problem today!

There are no great solutions

- CDNs, network traffic filtering, and best practices all help

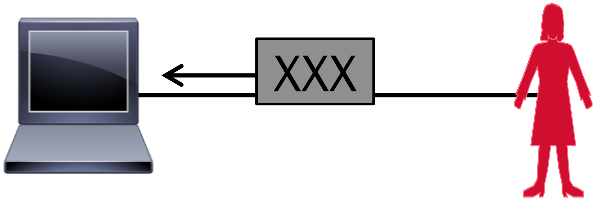
HOST DENIAL-OF-SERVICE

Strange packets can sap host resources!

- “Ping of Death” malformed packet
- “SYN flood” sends many TCP connect requests and never follows up
- Few bad packets can overwhelm host

Patches exist for these vulnerabilities

- Read about “SYN cookies” for interest

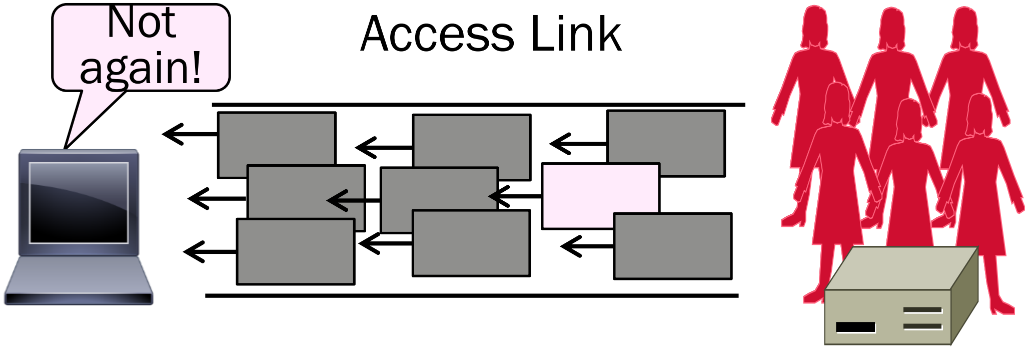


NETWORK DENIAL-OF-SERVICE

Network DOS needs many packets

- To saturate network links
- Causes high congestion/loss

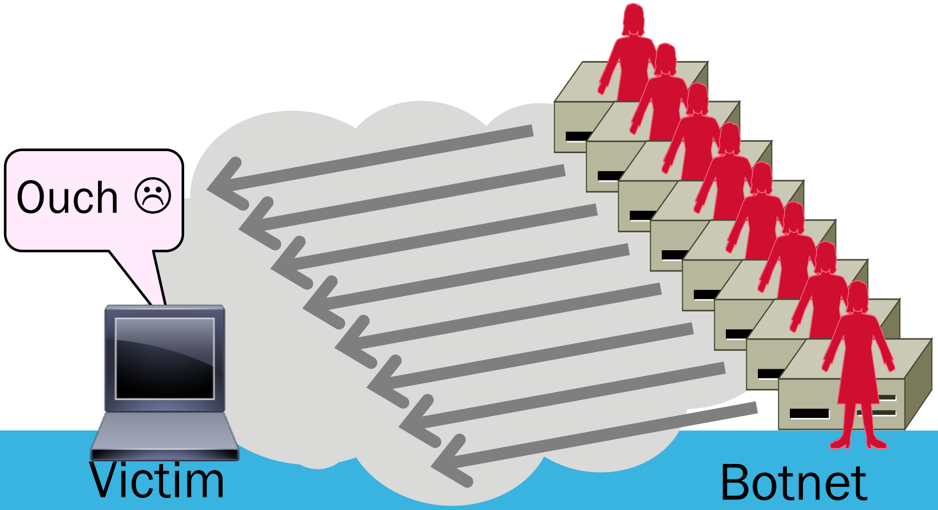
Helpful to have many attackers … or Distributed Denial-of-Service



DISTRIBUTED DENIAL-OF-SERVICE (DDOS)

Botnet provides many attackers in the form of compromised hosts

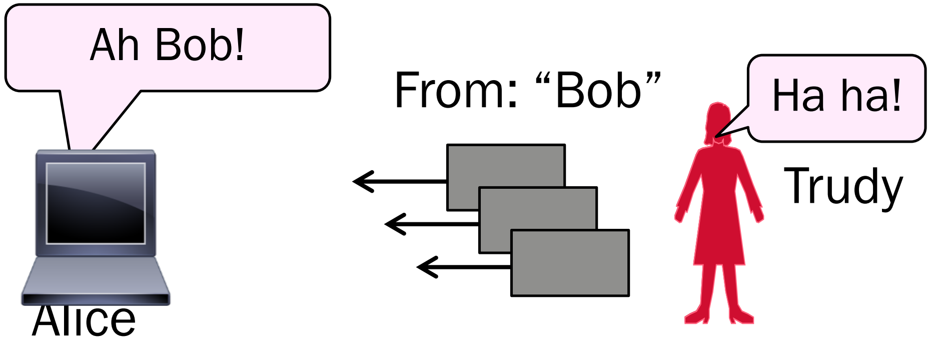
- Hosts send traffic flood to victim
- Network saturates near victim



COMPLICATION: SPOOFING

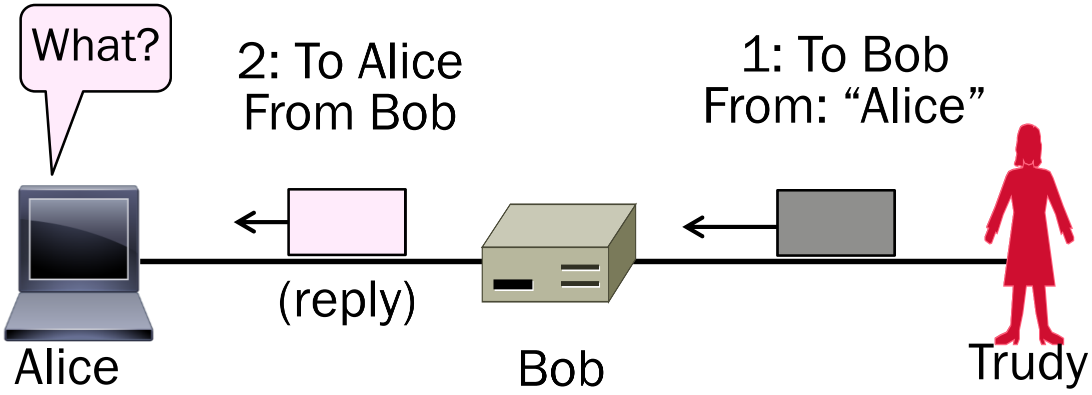
Attackers can falsify their IP address

- Put fake source address on packets
- Historically network doesn't check
- Hides location of the attackers
- Called IP address spoofing



SPOOFING

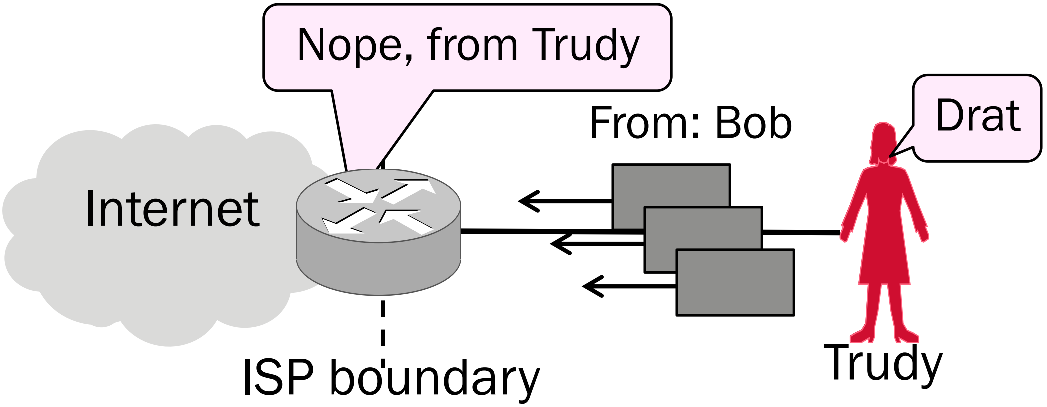
- Trudy can even trick Bob into really sending packets to Alice
- To do so, Trudy spoofs Alice to Bob



BEST PRACTICE: INGRESS FILTERING

Idea: Validate the IP source address of packets at ISP boundary

- Ingress filtering is a best practice, but deployment has been slow



FLOODING DEFENSES

1. **Increase network capacity around the server; harder to cause loss**
   - Use a CDN for high peak capacity
2. **Filter out attack traffic within the network (at routers)**
   - The earlier the filtering, the better
   - Ultimately what is needed, but ad hoc measures by ISPs today