ECE 455: CYBERSECURITY

Lecture #8

Daniel Gitzel

Announcement

- **Continue work on final project.**
  - We’ll have a check-in next class.
  - You should know if your proposal is viable by this week.
- **Class on Tuesday next week.**
  - Paper and quiz as usual.

NETWORK SECURITY

Introduction

- Net adversary
- TCP attacks
- DNS attacks
- Firewalls
- Intrusion detection
- Honeypots

Secure End-to-End Channels

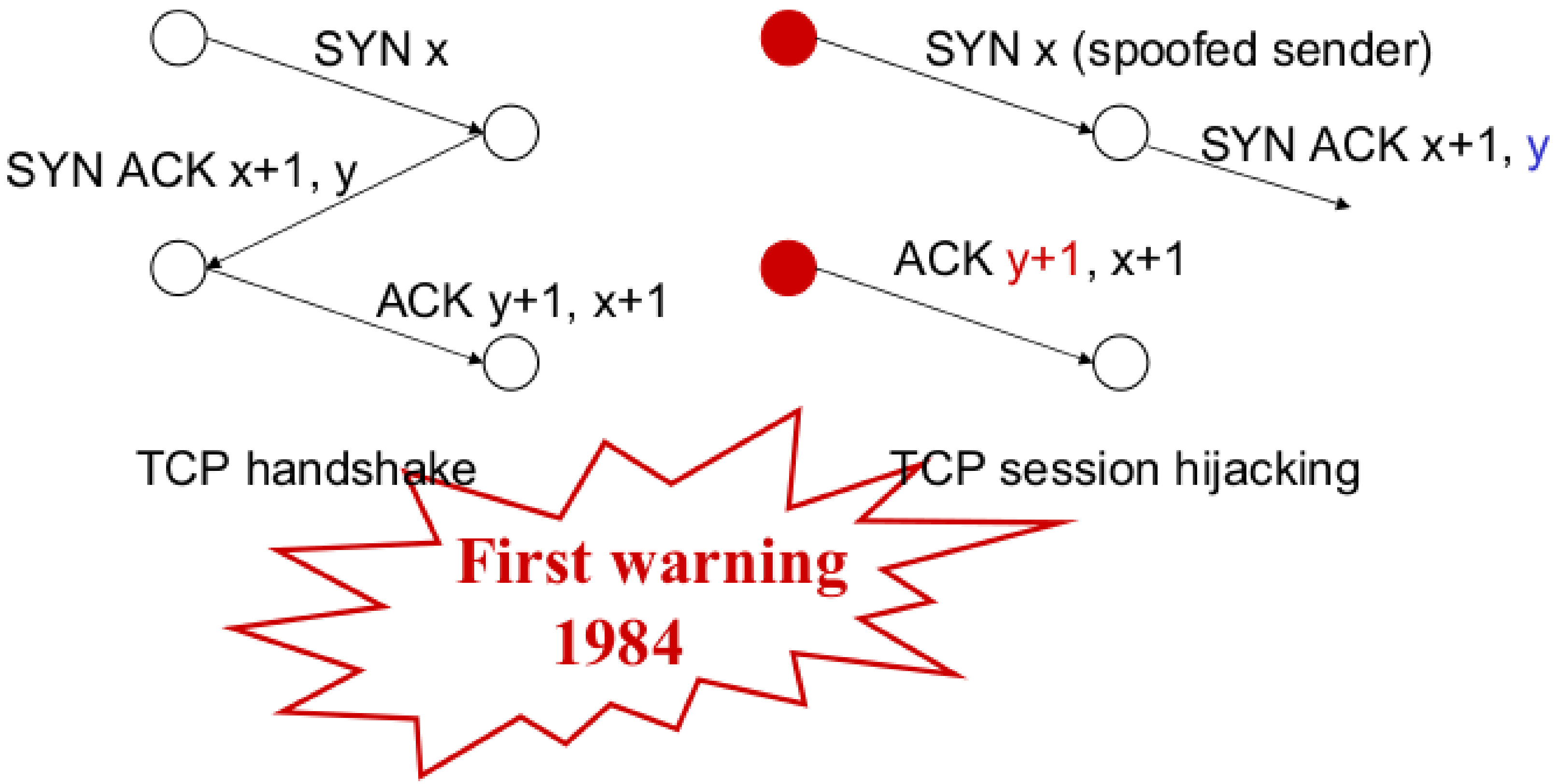
- **End-to-end = protect channel from originating client to intended server, between endpoints**
  - no need to trust intermediaries
- **Dealing with threats:**
  - Eavesdropping?
    - Encryption (including session keys)
  - Manipulation (injection, MITM)?
    - Integrity (use of a MAC); replay protection
  - Impersonation? (someone pretending as you)
    - Signatures
  - Availability?

Net Adversary

- **A botnet consists of bots, programs running on the machines of unwitting Internet users and receiving commands from a bot controller.**
- **Net adversary threat model**
- **A malicious network node able to:**
  - read messages directly addressed to it,
  - spoof arbitrary sender addresses,
  - try to guess fields sent in unseen messages.

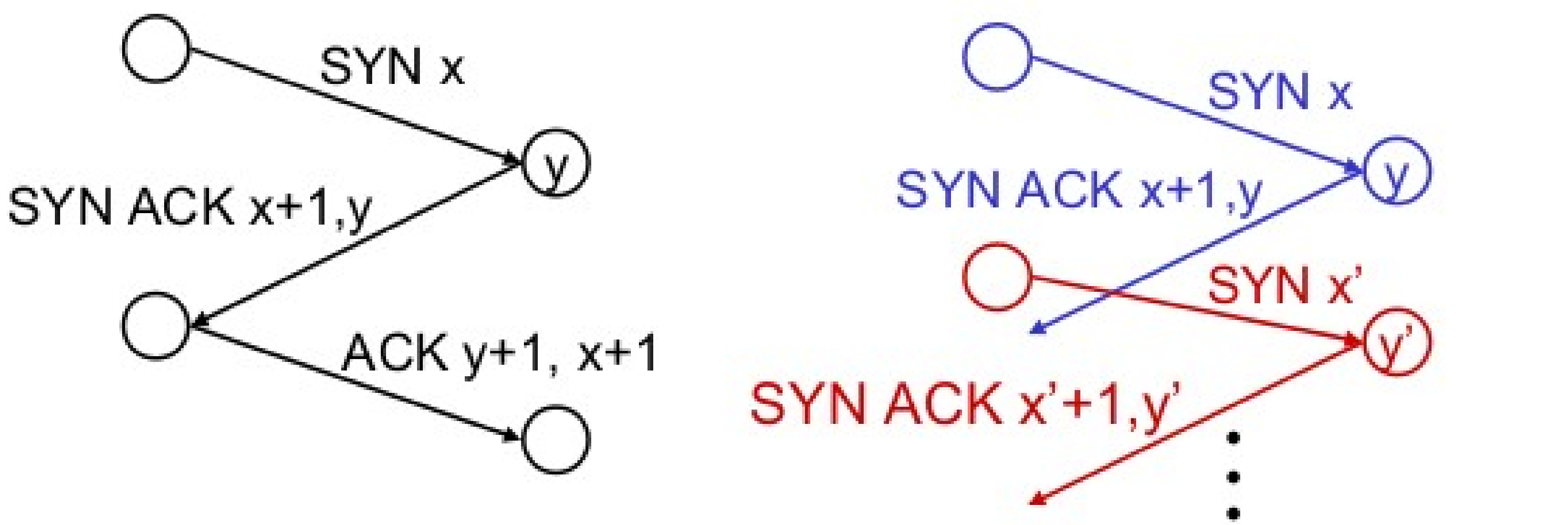
Classic TCP Session Hijacking

- Predict challenge to send messages that appear to come from a trusted host.



Classic TCP SYN Flooding Attack

- Exhaust responder's resources by creating half-open TCP connection requests.



TCP handshake

SYN flooding attack

Intro to DNS

Overview

- Why DNS is privacy and security problem.
- How privacy and security issues can be mitigated.
- How to secure your DNS queries.

What is DNS?

Ready for Business?
Help your customers find you.
KINCAID & SON CONSTRUCTION CO INC
Kornerstone Konstruction
Krueger Contracting
Kuk Construction LLC
Kymco LLC
L & C Services Inc
L H Construction Inc.
LPI Inc
Lee's Custom Design
Liberty Way Contractors
Little Bear Construction
LITTLE SUSITNA CONSTRUCTION COMPANY
Loken Construction
LONGSTAFF ENGINEERS & CONSTRUCTION
Lynx Construction Inc
M B Contracting Co Inc
MCF Construction
NANUQ INC
NEESER CONSTRUCTION INC
Nielson Construction Inc
No-Sea-M Seamless Gutters
Nordic Constructors
North Country Services
North Pole Contractors Inc
Northern Petroleum Testing & Services Inc
Northland Alaska Inc
Northland Construction Enterprises
Northridge Development Corporation
Northrim Siding Inc
Northwest Homes Inc
Nugget Construction Inc
Nugget Construction & Rigging
Nuna Contractors Inc
Olgoonik Corporation
Origin D B Inc
Osage Construction Company
OSBORNE CONSTRUCTION CO

History

- Use of names in place of a host's IP address dates back to the ARPANET era.
- Paul Mockapetris proposed a distributed and dynamic DNS database in 1983.
- In November 1987, IETF published the DNS specifications in RFC 1034 and RFC 1035, essentially DNS as it exists today.

Domain Name System (DNS)

- **Essential infrastructure for the Internet**
  - Critical-path for just about everything we do
  - Maps host names to IP addresses (and vice versa)
- **Design only scales if we can minimize lookup traffic**
  - Lots of caching!
  - Pre-fetching additional answers
- **Originally designed for a friendly environment; only basic authentication mechanisms**
- **Directly interacting w/ DNS: dig program on Unix**
  - Allows querying of DNS system
  - Dumps each field in DNS responses

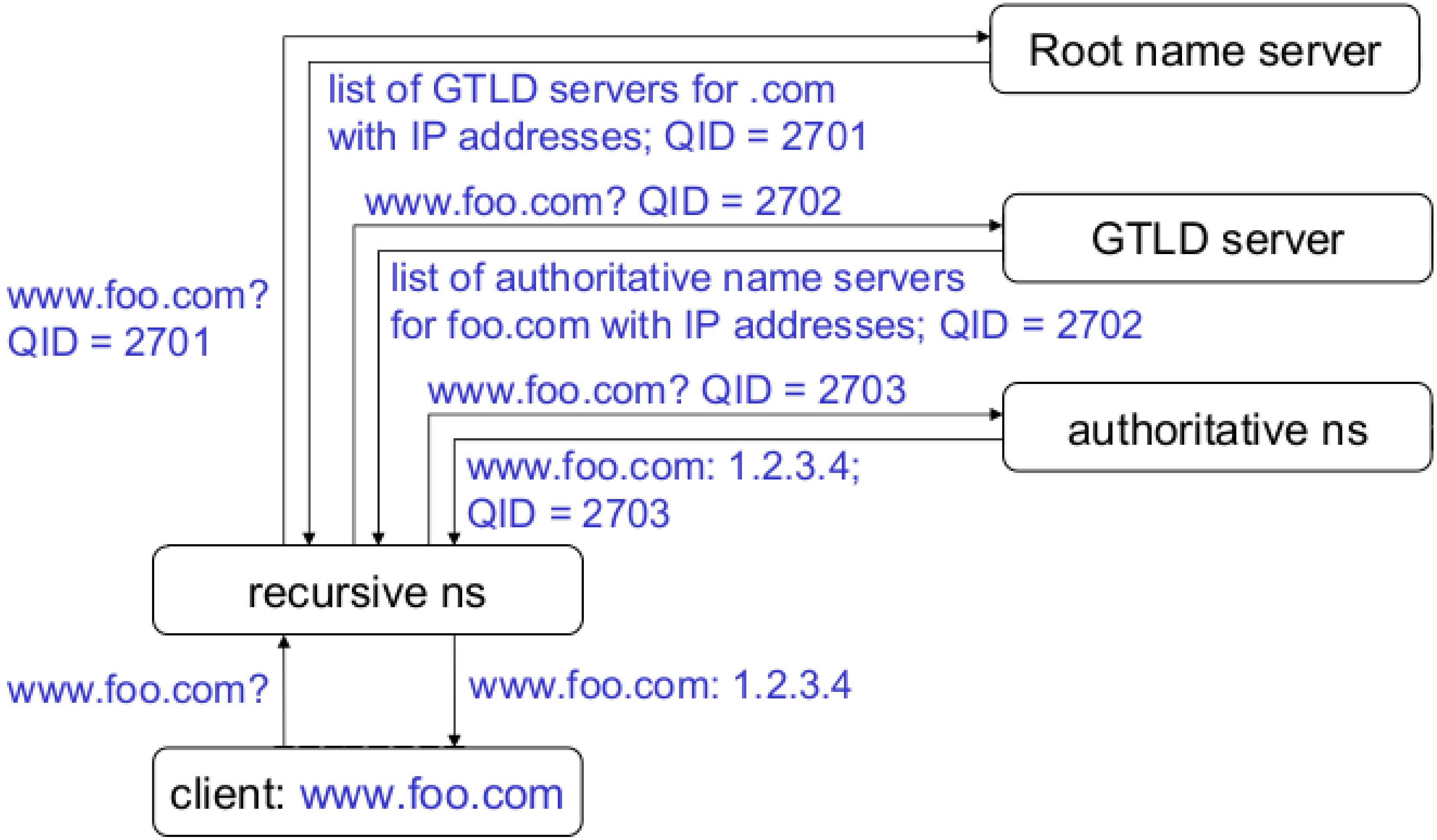
Domain Name System (DNS)

- **Distributed directory service for domain names (host names) used for:**
  - look up IP address for host name, host name for IP address.
  - anti-spam: Sender Policy Framework uses DNS records.
  - basis for same origin policies applied by web browsers.
- **Various types of resource records.**
- **Host names and IP addresses collected in zones managed by authoritative name servers.**

DNS Infrastructure

- **13 root servers; all name servers configured with the IP addresses of these root servers.**
- **Global Top Level Domain (GTLD) servers for top level domains: .com, .net, .org, etc.**
  - There can be more than one GTLD server per TLD.
  - Root servers know about GTLD servers.
- **Authoritative name servers provide mapping between host names and IP addresses for their zone.**
- **GTLD servers know authoritative servers in their TLD**
- **Recursive name servers pass client requests on to other name servers and cache answers received.**

IP Address Lookup – Review

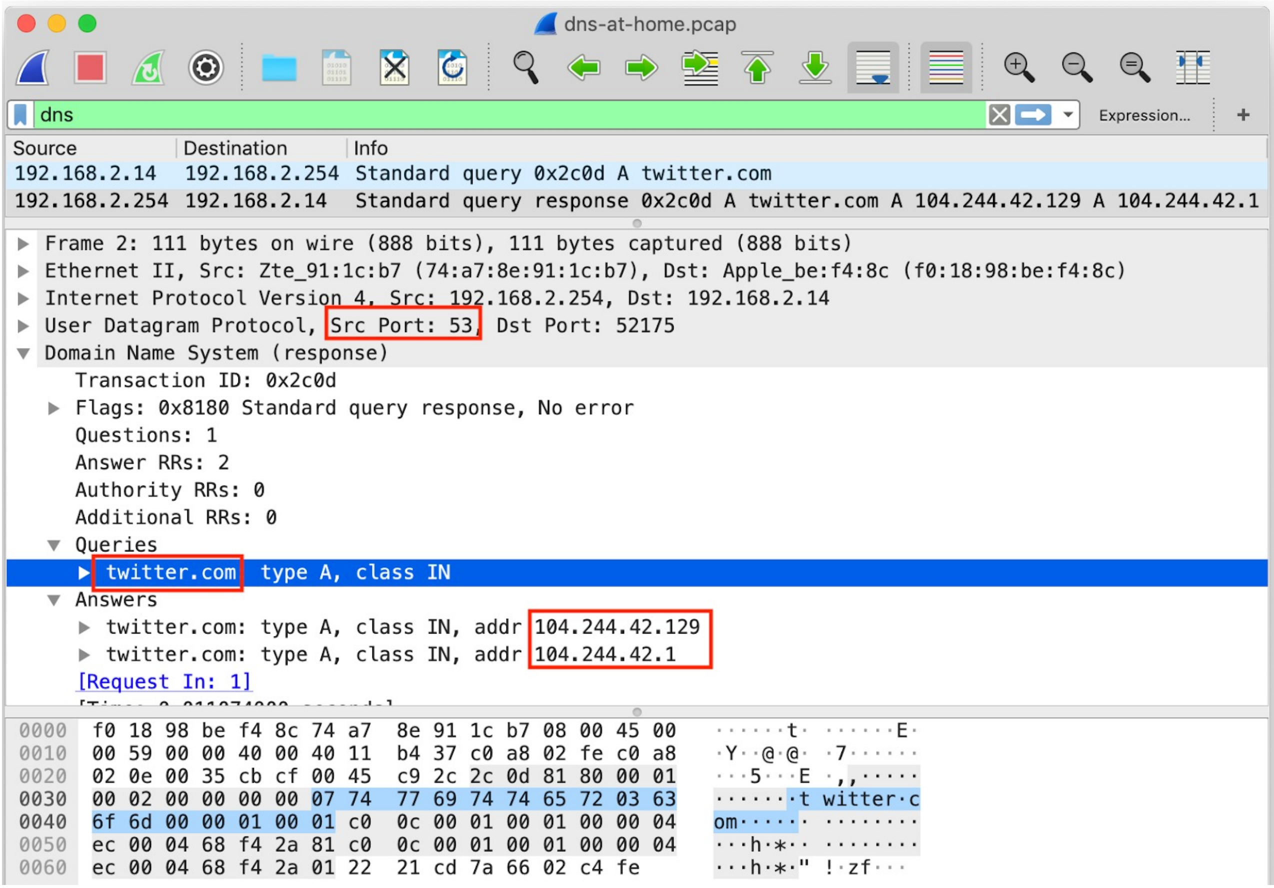


Cache & Time-to-live

- **Performance optimization**: stores map in cache
- Name server first checks its cache
- Answer remains in cache until it expires; time-to-live (TTL) of answer is set by sender.
- **Design question**: reasons for setting TTL by sender, reasons for setting TTL by receiver?
- Does Long TTL = high security, low TTL = low security?

DNS and Privacy

Unencrypted DNS



Why is DNS a privacy concern?

- Most ISPs log DNS queries.
- Mass surveillance[1].
- Fingerprinting and re-identification of individuals.

1. https://www.europarl.europa.eu/RegData/etudes/STUD/2015/527409/EPRS_STU%282015%29527409_REV1_EN.pdf

Why is DNS a privacy concern?

- MORECOWBELL[2] and QUANTUMDNS [3]

- Some ISPs embed user information (e.g. a user id or MAC address).

2. http://goodtimesweb.org/surveillance/2015/MORECOWBELL-Analysis-Grothoff-etal.pdf
3. https://www.wired.com/2014/03/quantum/

Why is unencrypted DNS a privacy concern?

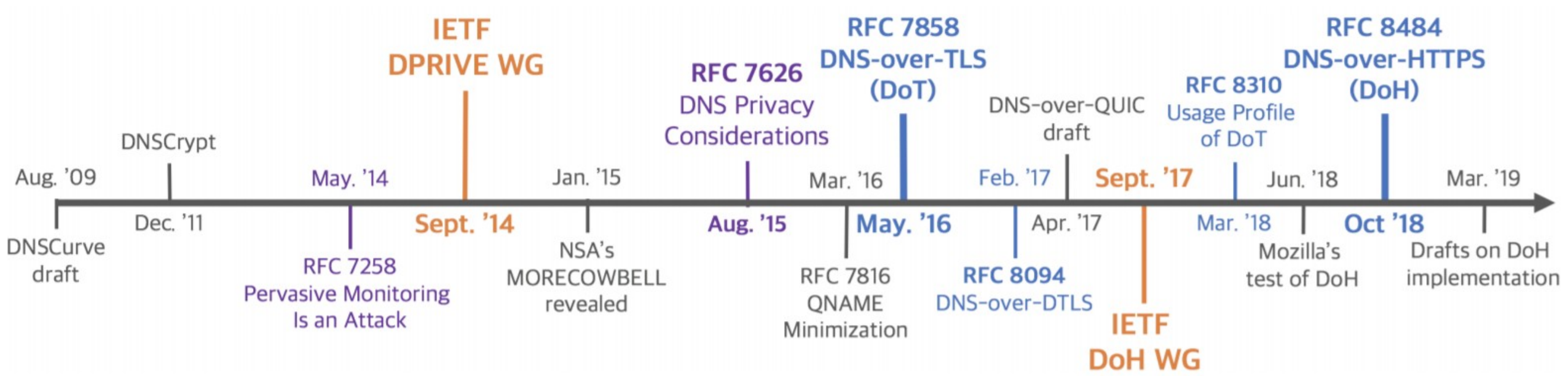
- DNS queries are sent in clear text.
- Most ISPs are Hijacking DNS Traffic and doing Ad/DNS Redirect.
- A number of consumer ISPs use or used DNS hijacking for their own purposes.

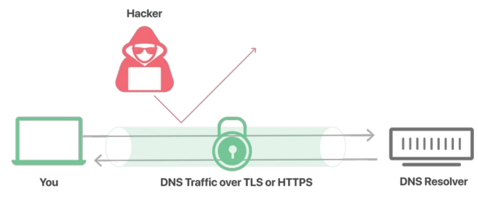
https://krebsonsecurity.com/2019/02/a-deep-dive-on-the-recent-widespread-dns-hijacking-attacks/

What about DNSSEC?

- In 1993, the IETF started a public discussion around how DNS could be made more trustworthy. Eventually, (DNSSEC) formally published in 2005.
- Origin authentication and data integrity.
- It doesn't encrypt communications.
- Most home routers/devices do not support DNSSEC.
- DNSSEC has nothing to do with last-mile DNS security.

DNS Encryption Protocols





DNS Encryption Protocols

Protocol	Released	Internet Standard	Prevalent
DNSCurve	2010		
DNSCrypt	2011		
DNS-over-TLS (DoT)	2016		
DNS-over-DTLS	2017		
DNS-over-QUIC	2017		
DNS-over-HTTPS (DoH)	2018		

DNS-over-TLS (DoT)

- DNS queries and answers via the Transport Layer Security (TLS) protocol.

- The goal of this protocol is to increase user privacy and security.

- DoT clients authenticate to the DNS server using Simple Public Key Infrastructure (SPKI).

How does DNS-over-TLS (DoT) Work?

- Client establishes a TCP connection to the DNS server over port 853.
- Server presents its certificate and the client checks it against the stored hash.
- Client and server do a TLS handshake, passing keys and starting an encrypted session.
- From there on, the data within the encrypted session follows the same rules as DNS over TCP.
- TLS encryption takes a little bit of a toll on its performance. However, the secure TLS connection remains open and is reused for future DNS queries.

DNS over HTTPS (DoH)

- DNS resolution via the HTTPS protocol.

- The goal of DoH is to increase user privacy and security.

- DoH is essentially HTTPS.

DNS over HTTPS (DoH)

- Requests are sent as an HTTP POST or GET method with queries in DNS message format.

- No certificate management.

- Enables web applications to access DNS through existing browser APIs.

DoT/DoH Native support

Devices/OS/Applications	**DNS-over-TLS (DoT)**	**DNS over HTTPS (DoH)**
Android Phones (version10+)		
Apple iPhone (iOS 14)		
Windows 10 macOS Catalina Linux		
Firefox Chrome Edge		

DoH/DoT Public Resolvers

Features	Google DNS	Quad9 DNS	Cloudflare DNS
DNS-over-TLS (DoT)			
DNS-over-HTTPS (DoH)			
Unfiltered DNS			
Block Malware (optional)			
Adult Content (optional)			

Disadvantages of DoH/DoT

- Latency
  - Encryption, Handshake, Socket Management add overhead
- A Single Point of Failure
  - Web Apps rely on browser implementations
    - Few browsers exist, most use Chrome or Safari

DNS Attacks

Light-weight Authentication

- **Threat model:**
  - Attacker can only read messages forwarded to her
  - Anybody can pretend to be an authoritative name server for any zone
- **How does a recursive name server know that it has received a reply from an authoritative name server?**
  - Recursive name server includes a 16-bit query ID (QID) in its requests.
  - Responding name server copies QID into reply
  - Recursive name server caches first answer for a given QID and host name; then discards this QID.
  - Drops answers that do not match an active QID.

Authentication – Security?

- **Attack method**
  - Guess QID to subvert cache entries.
- **If query is not passed by mistake to the attacker her chance of generate faking a answer is 2^-16**
- **Security relies on correct routing from local name to authoritative name server.**

DNS Cache Poisioning

- Ask recursive name server to resolve host name in attacker's domain.
- Request to attacker's name server contains current QID.
- Attackers asks recursive name server to resolve victim host name
- Attacker sends answer that includes next QID and maps victim host name to chosen IP address
- If attacker's answer arrives first; the correct answer is dropped and cache is poisoned

Predictable Challenges

- **Do not use predictable challenges (e.g. QID)**
- **Attacker can improve chances:**
  - Send answers with QIDs from a small window.
  - Slow down authoritative name server with a DoS attack.
  - Prevent that a new query from restoring the correct binding, set a long time to live.

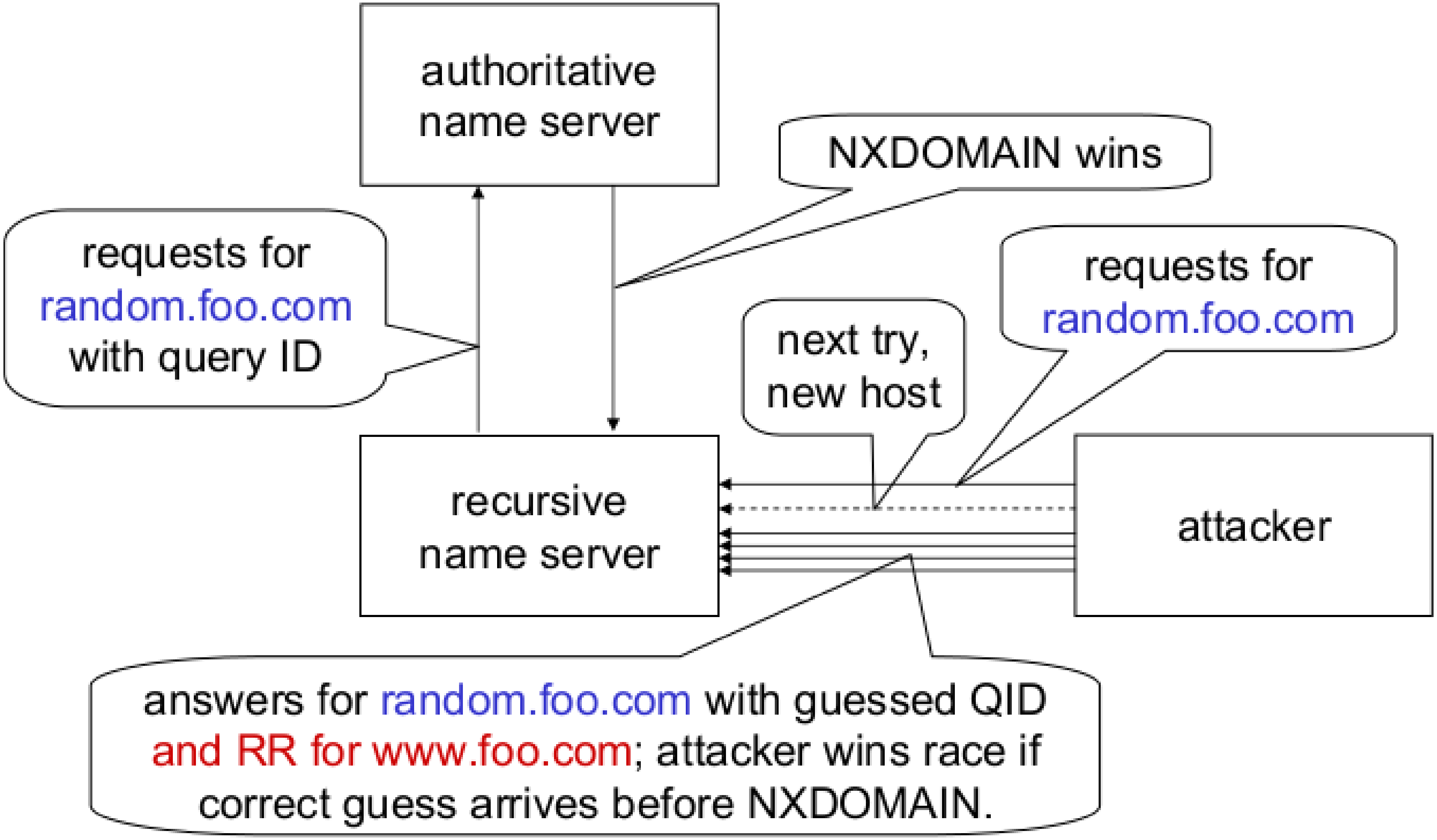
Bailiwick Checking

- **Bailiwick: an area of jurisdiction**
- **Optimize perf:**
  - Name servers send additional resource records
  - Might save round trips
  - Assumes benign servers
  - Malicious name server sends records for other domains
- **Bailiwick checking rejects records outside of the queried domain (i.e. out of jurisdiction)**

Dan Kaminsky’s Attack (2008)

- Attacker requests random.foo.com from name server
- Recursive name server refers request to authoritative name server for foo.com
- Attacker sends answers for random.foo.com with guessed QIDs and additional resource record for www.foo.com (in bailiwick)
- If guessed QID is correct and attacker wins race with NXDOMAIN, poison entry is cached with a TTL set by attacker
- Recursive name server will now direct all queries for www.foo.com to attacker’s IP address

Dan Kaminsky’s Attack



Countermeasures

- **Run queries on random ports**
  - Attacker now must guess QID & port number
- **Restrict access to local recursive name server: split name server**
- **Access control for records prevent unauthorized overwriting**
- **DNSSec: authentication using digital signatures**
- **Server does not reply to malformed queries??**

Split-split Name Server

- **Split Network Topology**
  - Local users who want to connect to the outside world
  - Remote users who want to connect to local hosts
- **Split Name Servers**
  - Recursive name server for internal queries to resolve (external) host names
  - Non-recursive authoritative name server for zone to resolve external queries for host names in zone
- **DNS server facing external users does not cache resource records so there is no cache to poison**
- **No defense against local attackers**

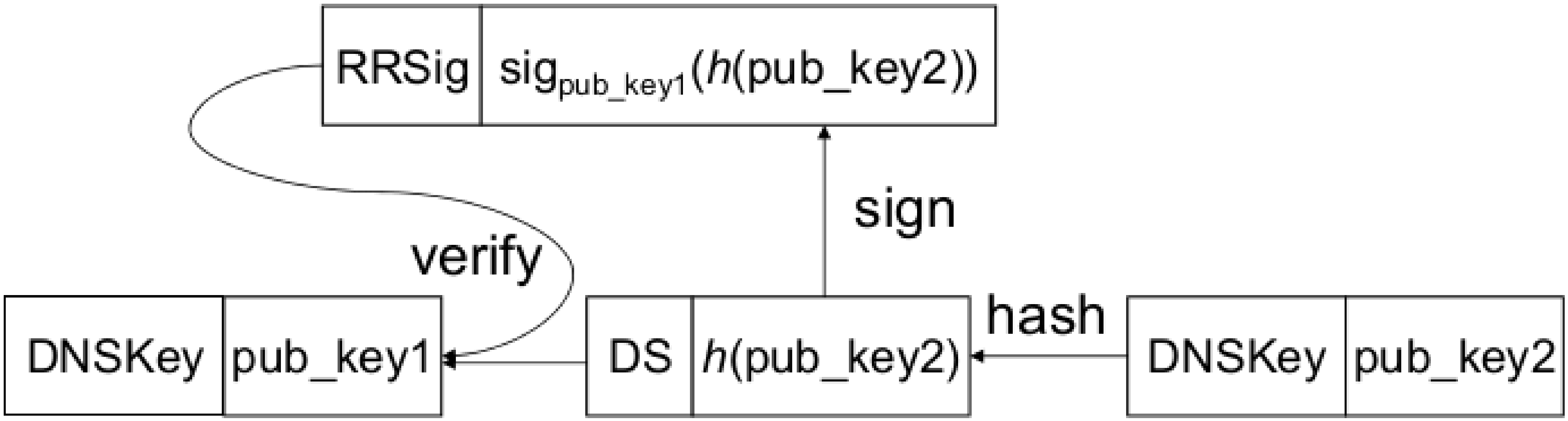
DNSSec

- **DNS Security Extensions, protect resource records with digital signatures**
- **Several new resource record types introduced:**
  - RRSIG resource records contain digital signatures of other resource records.
  - DNSKEY resource records contain the public keys of zones.
  - DS (Delegation Signer) resource records contain hashes of DNSKEY research records.

DNSSec

- Build chain by alternating DNSKEY and DS records.
- Key in DNSKEY record verifies the signature on the next DS record
- Hash in the DS record links to next DNSKEY record, and so on.
- Verification in the resolver has to find a trust anchor for the chain (root verification key).

DNSSec – Chain



DNS Rebinding

- **Same origin policy**
  - Script in a web page can only connect back to the server it was downloaded from.
- **To make a connection, the client's browser needs the IP address of the server.**
- **Authoritative DNS server resolves 'abstract' DNS names in its domain to 'concrete' IP addresses.**
- **The client's browser 'trusts' the DNS server when enforcing the same origin policy.**
- **Trust is Bad for Security!**

DNS Rebinding Attack

- **“Abuse trust”: Attacker creates attacker.org domain and name server**
- **Evil names server binds attacker.org IP and then switches to victim IP address**
- **Client downloads script from attacker.org; script connects to target; permitted by same origin policy.**
- **Defense: Same origin policy with IP address.**
  - D. Dean, E.W. Felten, D.S. Wallach: Java security: from HotJava to Netscape and beyond, 1996 IEEE Symposium on Security & Privacy.

DNS Rebinding Attack

- Client visits attacker.org; attacker’s DNS server resolves this name to attacker’s IP address with short time-to-live.
- Attack script waits before connecting to attacker.org.
- Binding at browser has expired; new request for IP address of attacker.org, now bound to target address.
- Defense: Don’t trust the DNS server on time-to-live; pin host name to original IP address

DNS Rebinding Attack

- Attacker shuts down its web server after the page has been loaded.
- Malicious script sends delayed request to attacker.org.
- Browser's connection attempt fails and pin is dropped.
- Browser performs a new DNS lookup and is now given the target's IP address.
- Error handling procedures has security implications!

DNS Rebinding Attack

- **Next round – browser plug-ins**
- **Plug-ins may do their own pinning.**
- **Dangerous combinations:**
  - Communication path between plug-ins.
  - Each plug-in has its own pinning database.
- **Attacker may use the client's browser as a proxy to attack the target.**
  - DDOS, send spam, etc.

FIREWALLS

Introduction

- **Cryptographic mechanisms protect data in transit**
- **Authentication protocols verify the source of data.**
- **Control which traffic is allowed to enter or leave our system**
- **Access control decisions based on information like addresses, port numbers, protocol, etc.**

Firewall

- **Firewall: a network security device controlling traffic flow between two parts of a network.**
- **Often installed between an organization's network and the Internet**
- **All traffic has to go through the firewall for protection to be effective.**
  - Wireless LANs, USB devices!?

Purpose

- Firewalls control network traffic to and from the protected network.
- Can allow or block access to services (both internal and external).
- Can enforce authentication before allowing access to services.
- Can monitor traffic in/out of network.

Types of Firewalls

- Packet filter
- Stateful packet filter
- Circuit-level proxy
- Application-level proxy

Packet Filter

- Inspect headers of IP packets, TCP and UDP ports
- Rules specify which packets are allowed through the firewall, and which are dropped.
- Actions: bypass, drop, protect
- Rules may specify source / destination IP addresses, and source / destination TCP / UDP port numbers.
- Rules for traffic in both directions.
- Certain common protocols are difficult to support securely (e.g. FTP).

Example

- **TCP/IP packet filtering router.**
  - Router which can throw packets away.
- **Examines TCP/IP headers of every packet going through the Firewall, in either direction.**
- **Packets can be allowed or blocked based on:**
  - IP source & destination addresses
  - TCP / UDP source & destination ports
- **Implementation on router for high throughput.**

Stateful Packet Filter

- **Packet filter that understands requests and replies**
  - e.g. for TCP: SYN, SYN-ACK, ACK
- **Rules need only specify packets in one direction**
  - from client to server – the direction of the first packet in a connection
- **Replies and further packets in the connection are automatically processed.**
- **Supports wider range of protocols than simple packet filter (FTP, IRC).**

Stateful Packet Filter & FTP

- **Client sends ftp-request to server**
- **Firewall stores connection state**
  - FTP-Server Address
  - state of connection (SYN, ACK, ...)
- **If correct FTP-server tries to establish data connection, packets are not blocked.**

Circuit-level proxy

- Similar to a packet filter, except that packets are not routed.
- Incoming TCP/IP packets accepted by proxy.
- Rules determine which connections will be allowed and which blocked.
- Allowed connections generate new connection from firewall to server.
- Similar specification of rules as packet filter.

Application-level Proxy

- **Layer-7 proxy server.**
  - “Client and server in one box”.
- **For every supported application protocol.**
  - SMTP, POP3, HTTP, SSH, FTP, NNTP...
  - Packets received and processed by server.
  - New packets generated by client.
- **MITM?**

Application-level Proxy

- **Complete server & client implementation in one box for every protocol the firewall should handle.**
  - Client connects to firewall.
  - Firewall validates request.
  - Firewall connects to server.
- **Response comes back through firewall and is also processed through client/server.**
- **Large amount of processing per connection.**
- **Can enforce application-specific policies.**

Firewall Policies

- **Permissive: allow by default, block some.**
  - Easy to make mistakes.
  - If you forget something you should block, it’s allowed, and you might not realize for a while.
  - If somebody finds find a protocol is allowed, they might not tell you ....
- **Restrictive: block by default, allow some.**
  - Much more secure.
  - If you forget something, someone will complain and you can allow the protocol.

Firewall Policies – Examples

- **Permissive policies: Allow all traffic, but block …**
  - IRC
  - telnet
- **Restrictive policies: block all traffic, but allow …**
  - http
  - POP3
  - SMTP
  - ssh

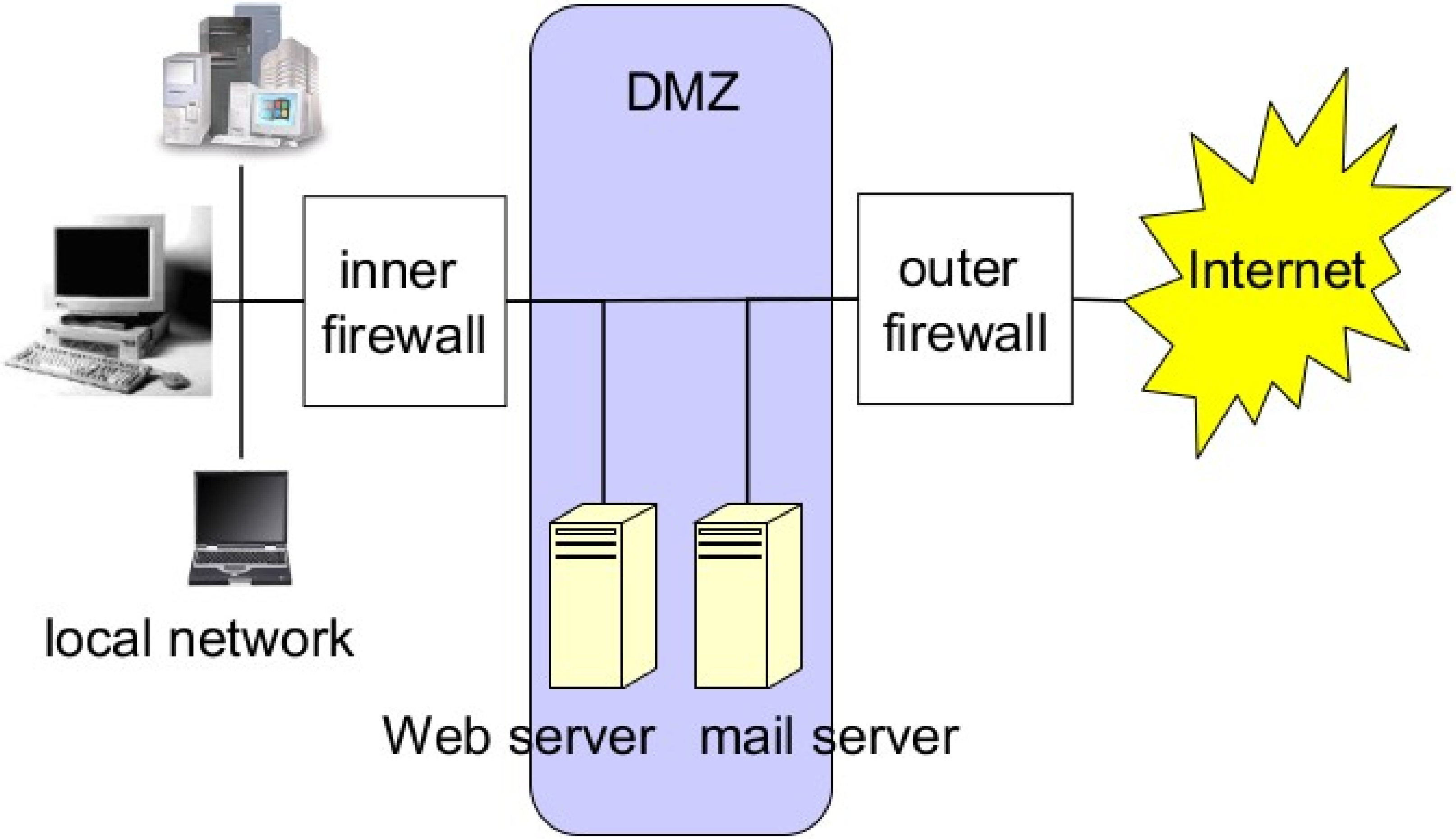
Typical Firewall Ruleset

- **Allow from internal network to Internet:**
  - HTTP, FTP, HTTPS, SSH, DNS
- **Allow reply packets**
- **Allow from anywhere to Mail server:**
  - TCP port 25 (SMTP) only
- **Allow from Mail server to Internet:**
  - SMTP, DNS
- **Allow from inside to Mail server:**
  - SMTP, POP3
- **Block everything else**

Firewall Location

- **Firewall can only filter traffic which goes through it.**
- **Where should we put a mail server?**
  - Requires external access to receive mail from the Internet.
    - Should be on the inside of the firewall
  - Requires internal access to receive mail from the internal network.
    - Should be on the outside of the firewall
- **Solution: “a perimeter network” (aka DMZ).**

DMZ


DMZ
inner firewall
outer firewall
Internet
local network
Web server
mail server

Firewalls – Limitations

- **Firewalls do not protect against insider threats.**
- **Blocking services may create inconveniences for users.**
- **Network diagnostics may be harder.**
- **Some protocols are hard to support.**
- **Protocol tunneling: sending data for one protocol through another protocol circumvents the firewall.**
  - More and more protocols are tunneled through http to get through the firewall
- **Encrypted traffic cannot be examined and filtered**
  - Some solutions can! HTTPS proxy

INTRUSION DETECTION SYSTEMS

Reminder: Security Strategies

- **Prevention:** take measures that prevent your assets from being damaged.
- **Detection:** take measures so that you can detect when, how, and by whom an asset has been damaged.
- **Reaction:** take measures so that you can recover your assets or to recover from a damage to your assets.

Security Strategies

- Cryptographic mechanisms and protocols are fielded to prevent attacks.
- Perimeter security devices (e.g. firewalls) mainly prevent attacks by outsiders.
- Although it would be nice to prevent all attacks, in reality this is rarely possible.
- New types of attacks occur: denial-of-service (where crypto may make the problem worse).
- How to we detect network attacks?

Vulnerability Assessment

- **Examines the “security state” of a network:**
  - Open ports
  - Software packages running (which version, patched?)
  - Network topology
  - Returns prioritized lists of vulnerabilities
- **Only as good as the knowledge base used**
  - Have to be updated to handle new threats
- **Vulnerability Assessment Methods**
  - Software solutions (ISS Scanner, Stat, Nessus etc.)
  - Audit Services (manual Penetration tests etc)
  - Web based commercial (Qualys, Security Point etc)
  - All have draw-backs and cannot detect all possible vulnerabilities

Intrusion Detection Systems (IDS)

- **Passive supervision of network (like an intruder alarm)**
  - Creates more work for personnel.
  - Provides security personnel with volumes of reports that can be presented to management (can be overwhelming or ignored)
- **Approaches to Intrusion Detection:**
  - Knowledge-based IDS – Misuse detection
  - Behavior-based IDS – Anomaly detection
- **IDS can also be used as response tool.**

Knowledge-based IDS

- Looks for suspicious patterns of network traffic or log files (heuristics):
  - Known vulnerabilities of particular OS and applications
  - Known attacks on systems
- Example “signatures” might include:
  - Number of recent failed login attempts on a sensitive host;
  - Bit patterns in an IP packet indicating a buffer overrun attack;
  - Certain types of TCP SYN packets indicating a SYN flood DoS attack.
- Also known as misuse detection IDS
  - More useful against insider threats

Knowledge-based IDS

- Only as good as database of attack signatures:
  - New vulnerabilities constantly being discovered and exploited
  - Vendors need to research latest attacks and customers need to install updates
  - Effective database difficult to build: large number of vulnerabilities and exploitation methods
  - Large databases makes IDS slow to use
- All commercial IDS look for attack signatures.

Behaviour-based IDS

- Wouldn’t it be nice to be able to detect new attacks?
- Anomaly detection uses statistical techniques to detect attacks
- First establish base-line behavior: what is “normal” for this system?
- Then gather new statistical data and measure deviation from base-line
- If a threshold is exceeded, issue an alarm

Behaviour-based IDS

- Example: monitor number of failed login attempts at a sensitive host over a period;
  - If a burst of failures occurs, an attack may be under way;
  - Or maybe the admin just forgot his password?
- False positives (false alarm): attack flagged when none is taking place.
- False negatives: attack missed because it fell within the bounds of normal behavior.
- Same issue as biometrics (separating two different distributions)

[1] Richard Bejtlich: Interpreting Network Traffic: A Network Intrusion Detector’s Look at Suspicious Events

Anomaly Detection

- IDS does not need to know about security vulnerabilities in a particular system:
  - base-line defines normality;
  - IDS does not need to know details of the construction of a buffer overflow packet.
- Anomalies are not necessarily attacks; normal and forbidden behavior may overlap:
  - Legitimate users may deviate from baseline, causing false positives (e.g. user goes on holiday, works late in the office, forgets password, or starts to use new application).
  - If base-line is adjusted dynamically and automatically, a patient attacker may be able to gradually shift the base-line over time so that his attack does not generate an alarm.
- There is no strong justification for calling anomaly detection “intrusion detection”.

IDS Architecture

- Distributed set of sensors – either located on hosts or on network – to gather data.
- Centralized console to manage sensor network, analyze data (→ data mining), report and react.
- Ideally:
  - Protected communications between sensors and console;
  - Protected storage for signature database/logs;
  - Secure console configuration;
  - Secured signature updates from vendor;
  - Otherwise, the IDS itself can be attacked and manipulated; IDS vulnerabilities have been exploited.

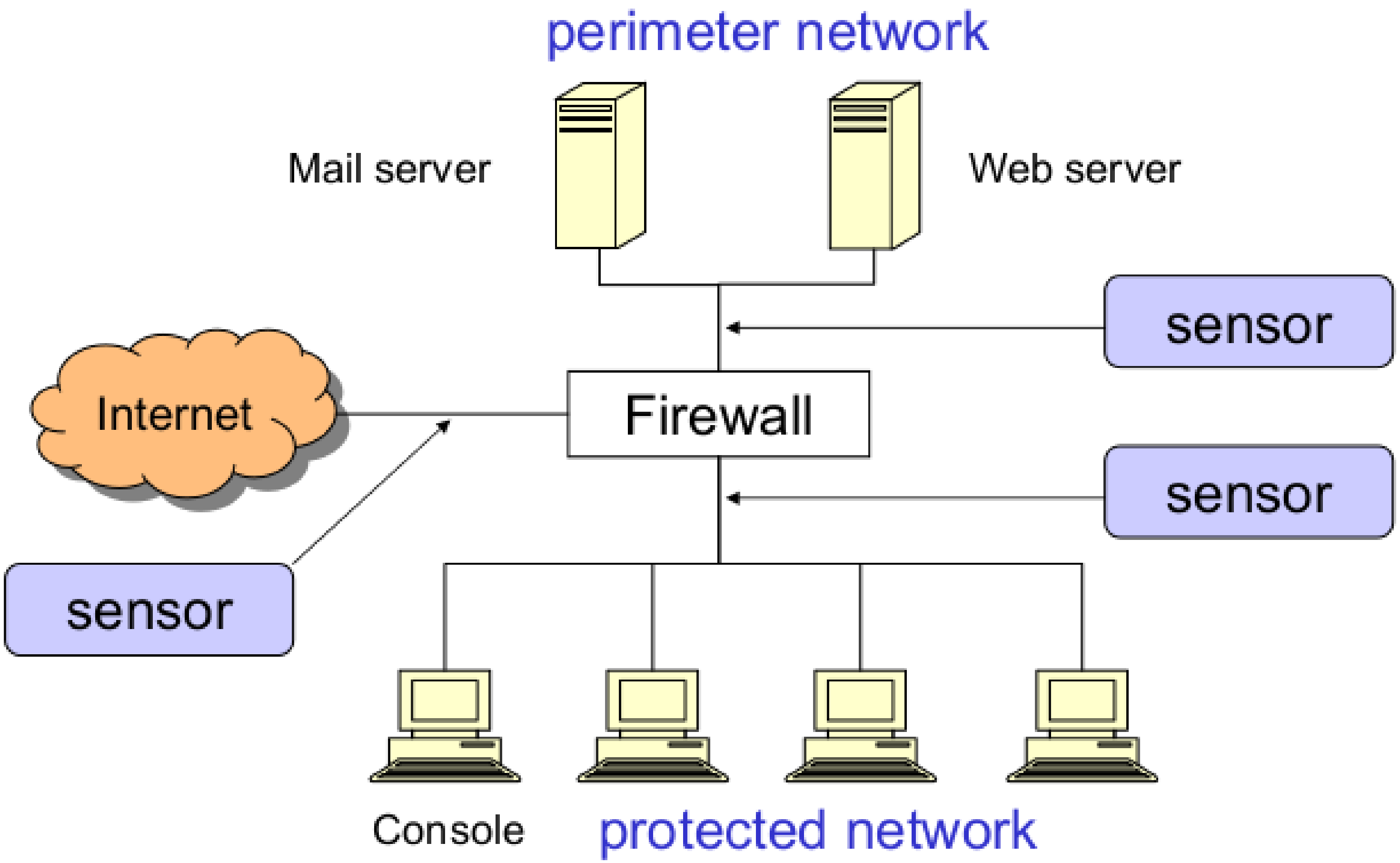
HIDS & NIDS

- **Network-based IDS** (NIDS) looks for attack signatures in network traffic.
- **Host-based IDS** (HIDS) looks for attack signatures in log files of hosts.
- Trend towards host-based IDSs.
- Attacks a NIDS can detect but a HIDS cannot:
  - SYN flood, Land, Smurf,Teardrop, BackOrifice,
- And vice-versa:
  - Trojan login script, walk up to unattended keyboard, encrypted traffic,
- For more reliable detection, combine both IDS types.

Network-based IDS

- Uses network packets as data source.
- Typically a network adapter running in “promiscuous mode”
  - This passes all traffic to the IDS’s instead of discarding frames with MAC address filtering
- Monitors and analyzes all traffic in real-time.
- Attack recognition module uses three common techniques to recognize attack signatures:
  - Pattern, expression or code matching;
  - Frequency or threshold crossing (e.g. detect port scanning activity);
  - Correlation of lesser events

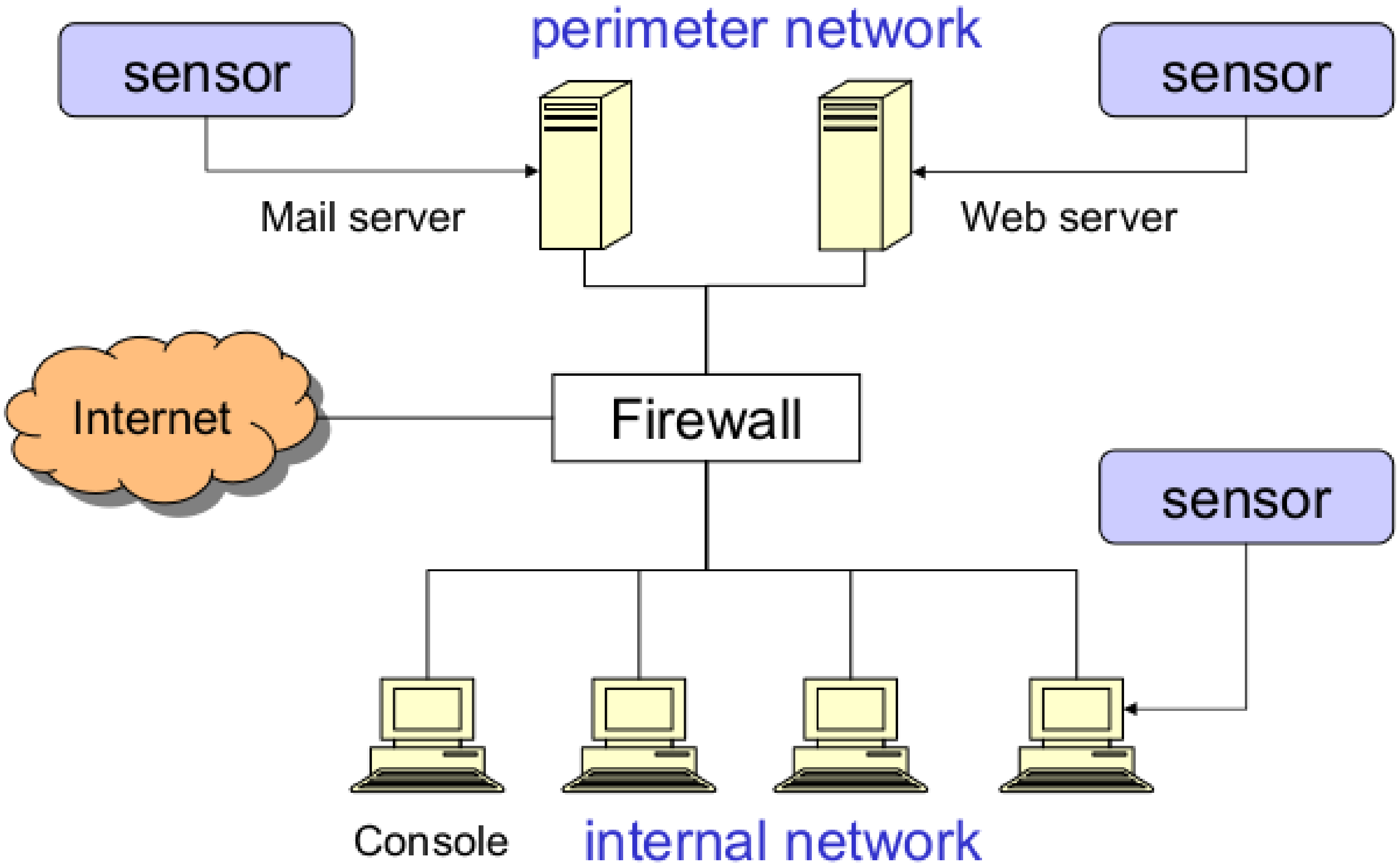
Placement of NIDS



Host-based IDS

- Typically monitors system, event, and security logs on Windows and syslog in Unix environments.
- Observe sequences of system calls to check whether a change from user to supervisor mode had been effected properly through a command like **su**.
- Verify checksums of key system files & executables at regular intervals for unexpected changes.
- Some products use regular expressions to refine attack signatures;
  - passwd program executed AND .rhosts file changed.
- Some products listen to port activity and alert when specific ports are accessed – limited NIDS capability.

Placement of HIDS



IDS Response Options

- Notify:
  - NIDS: alarm to console, email, SNMP trap, view active session
  - HIDS: alarm to console, email, SNMP trap
  - SNMP = Simple Network Management Protocol (traps allow for unsolicited messages to pass to modems, routers, switches, servers, etc.)
- Store:
  - NIDS: log summary, log network data
  - HIDS: log summary
- Action:
  - NIDS: kill connection (TCP reset), reconfigure firewall
  - HIDS: terminate user log in, disable user account, restore index.html

Dangers of Automated Response

- Attacker tricks IDS to respond, but response aimed at innocent target (say, by spoofing source IP address).
  - Similar to a reflection/amplification attack
- Users locked out of their accounts because of false positives.
- Repeated e-mail notification becomes a denial of service attack on sysadmin’s e-mail account;
- Repeated restoration of server data reduces website availability

IDS – Main Challenges

- Collecting and evaluating large amounts of data.
  - Combine events for more compact presentation.
- False positives, false negatives.
- Life intrusion detection systems generate lots of data.
  - DMZ with 60 hosts, monitored 7 days by NIDS with 244 signatures: 771,733 alerts created.
- Data mining applied for extracting useful information from such data collections.
- Context-aware systems filter out attacks that are irrelevant for the systems being monitored.
  - Ignore attacks on software or services you are not running.

Honeypots

- How to detect “**zero-day**” exploits? There is no attack signature yet.
  - Zero-day = brand new exploit or vulnerability
- How to “collect” new attacks for the knowledge base?
- Put systems online that mimic production systems but do not contain “real” data; anything observed on these systems is an attack.
- **Honeypot**: “a resource whose value is being attacked or compromised”
- Honeypot technologies track, learn and gather evidence of hacker activities

[2] Laurence Spitzner, “The value of honeypots”, SecurityFocus

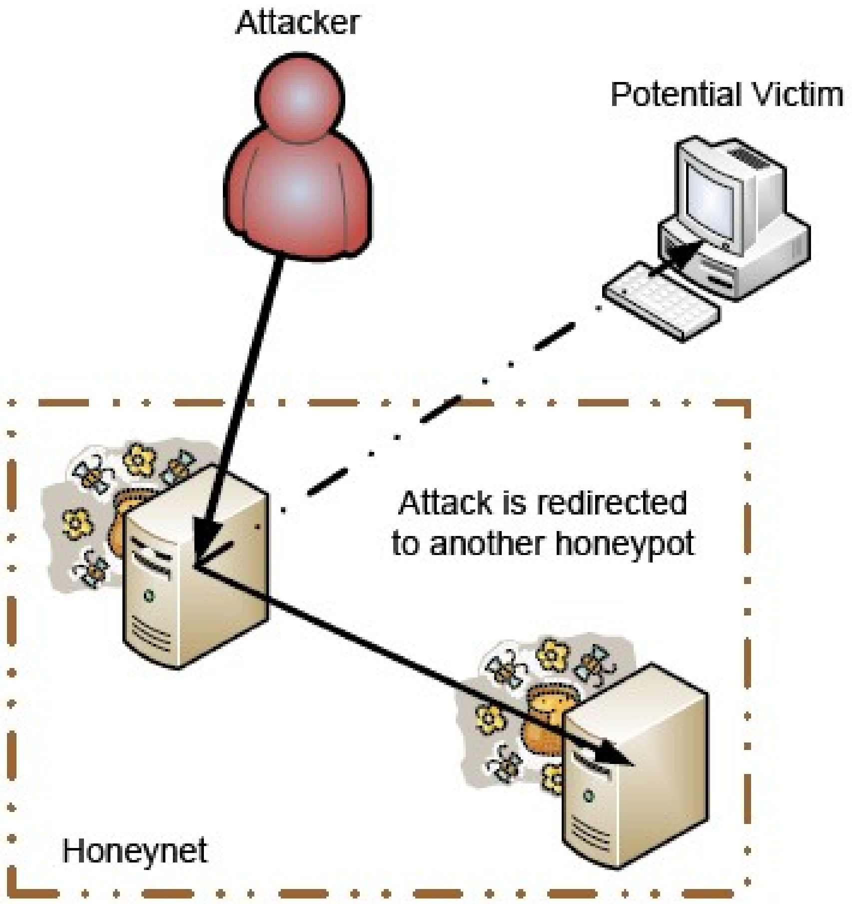
Honeypot Types

- Level of Involvement:
  - Low interaction: port listeners
  - Mid interaction: fake daemons
  - High interaction: real services
- Quality of information acquired increases with level of interaction.
- ‘Intelligent’ attackers will avoid obvious honeypots; tools for detecting honeypots exist.
- Risk that honeypot can be used as staging post in an attack increases with level of interaction.
- Pretending to be a honeypot has been proposed as a defense method!

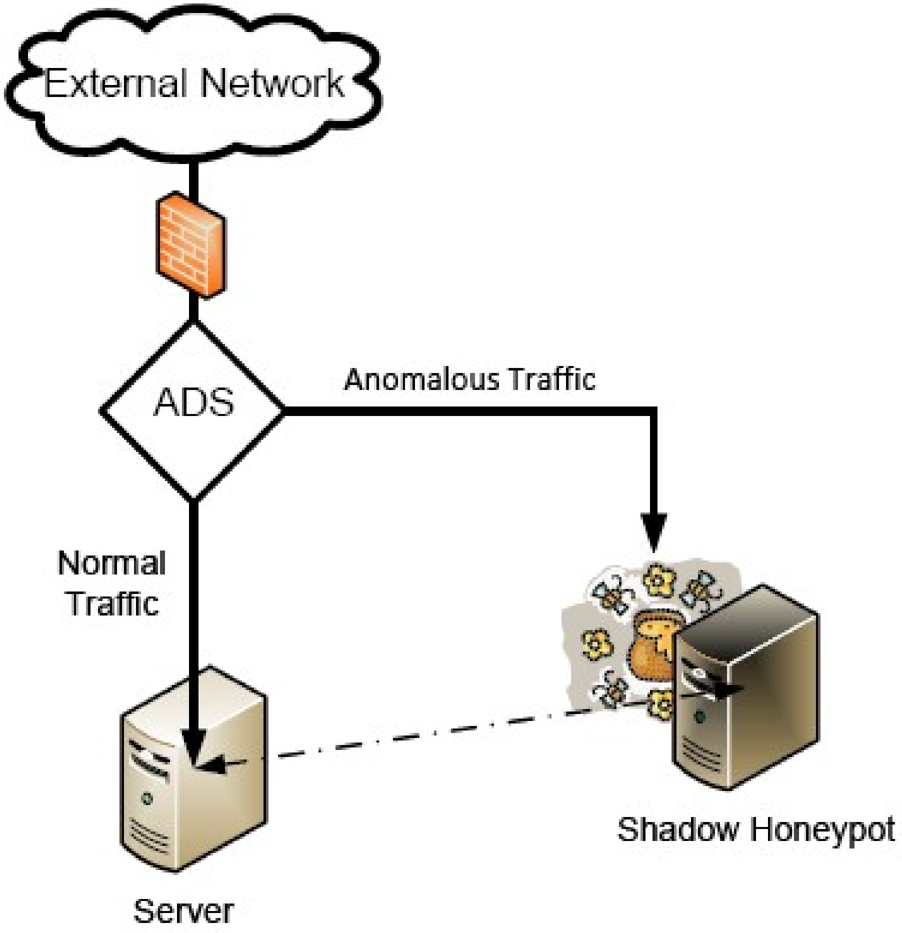
Honeypot Types

- **Honeynets: groups of honeypots**
  - Networked together to allow for more realism
  - Restrict outbound traffic (prevent using honeypot as an attack vector)
  - Redirect outbound traffic to another honeypot (trapping the attacker)
- **Shadow Honeypot: combine anomaly IDS and honeypot**
  - Honeypot mirrors server state
  - Redirect anomalous traffic to the honeypot (traffic segmentation)
  - Detected attacks reset the honeypot’s state (ready for next attack)

Honeynet


Attacker
Potential Victim
Attack is redirected
to another honeypot
Honeynet

Shadow Honeypot



Disadvantages of Honeypots

- Honeypots are not perfect, though:
  - Can be used by attacker to attack other systems [3]
  - Only monitor interactions made directly with the honeypot - the honeypot cannot detect attacks against other systems
  - Can potentially be detected by the attacker
  - Anomalies are not necessarily attacks

[3] honeynet Project. "Know Your Enemy: Honeynets." 24 March 2008.

Summary

- Apply prevention, detection and reaction in combination.
- IDS's are a useful second line of defense (in addition to firewalls, cryptographic protocols, etc.).
- IDS deployment, customization and management is generally not straightforward.
- IDS's are not fool-proof and require maintenance to remain effective
- Honeypots can provide information on new attacks by recording anomalous behavior