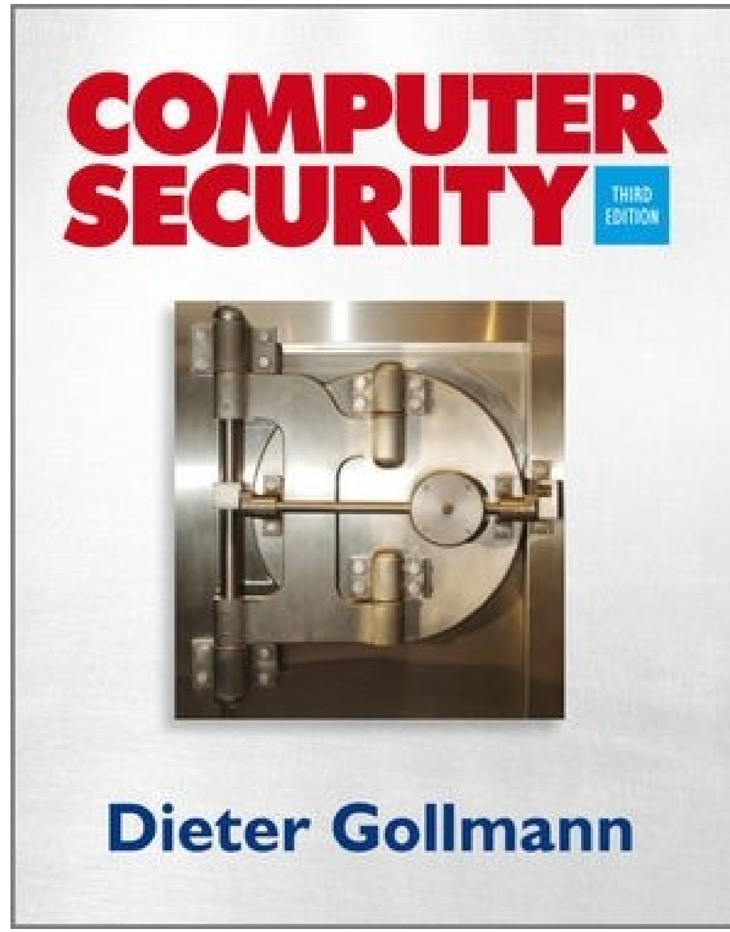
Computer Security 3e

Dieter Gollmann



Chapter 17: Network Security

Agenda

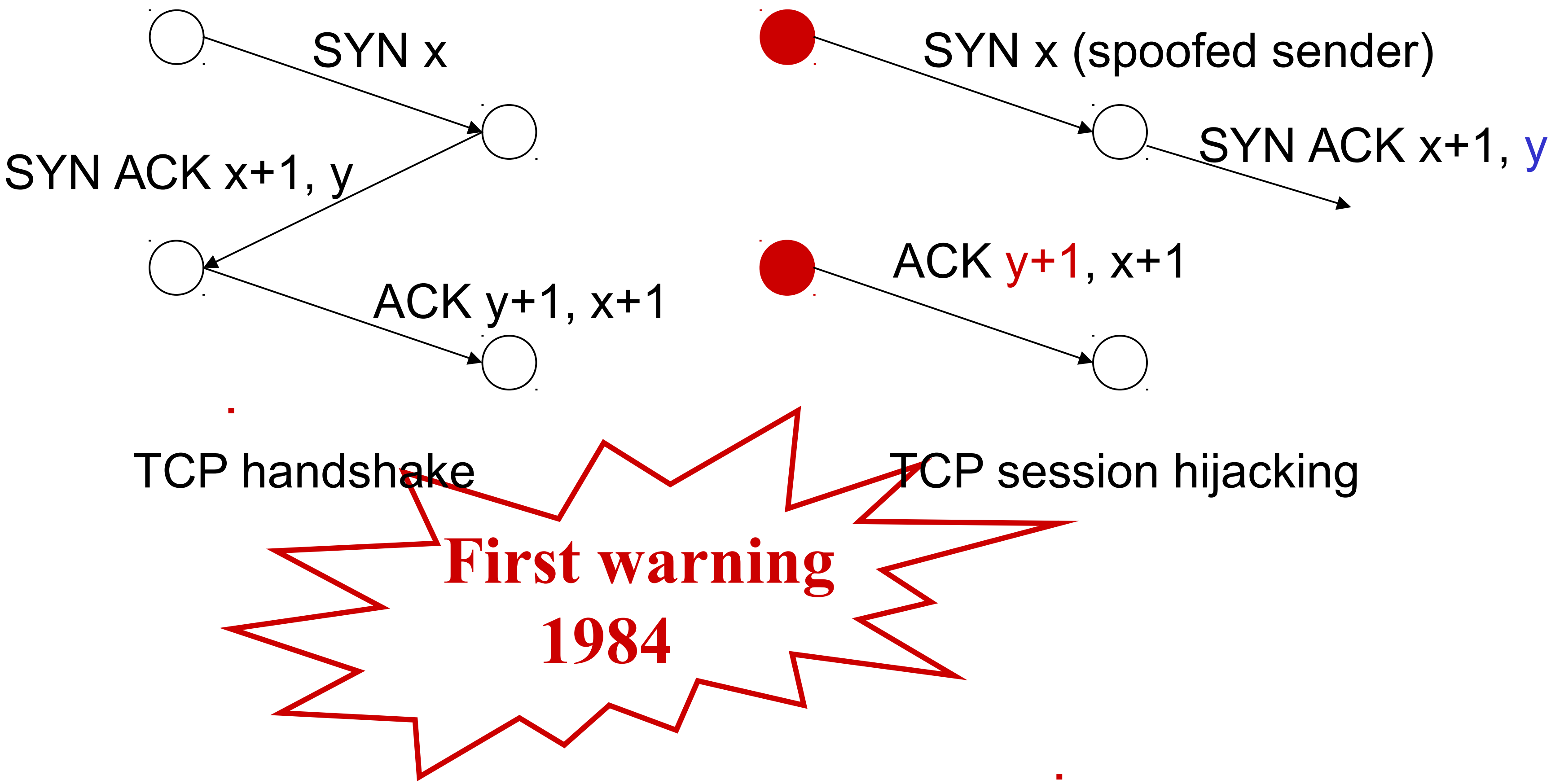
- Net adversary
- TCP attacks
- DNS attacks
- Firewalls
- Intrusion detection
- Honeypots

Net Adversary

- A botnet consists of bots (drones), i.e. programs installed on the machines of unwitting Internet users and receiving commands from a bot controller.
- Botnet attacks do not target communications links; you do not face an adversary in charge of the entire Internet, but you can no longer assume that the end points of links are safe harbours.
- Net adversary: malicious network node able to
  - read messages directly addressed to it,
  - spoof arbitrary sender addresses,
  - try to guess fields sent in unseen messages.

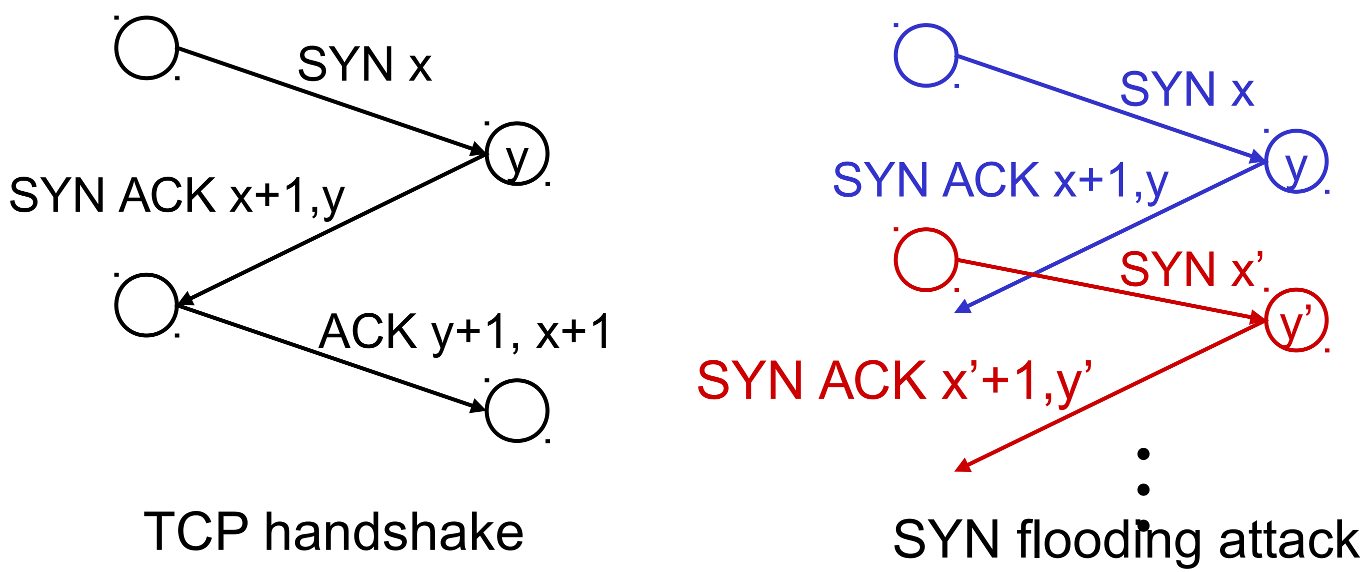
TCP Session Hijacking

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



TCP SYN Flooding Attacks

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



Domain Name System (DNS)

- Essential infrastructure for the Internet.
- Maps host names to IP addresses (and vice versa).
- Originally designed for a friendly environment; hence only basic authentication mechanisms.
- Historic note: DNS created in the 1980s (e.g., RFC 819, August 1982); strong political obstacles to globally deployable cryptographic protection.
- Some serious attacks reported in recent years.
- We will look at those attacks and at available countermeasures.

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.
- Protocols such as BIND, MSDNS, PowerDNS, DJBDNS, for resolving host names to IP addresses.
- We will explain issues at a general, simplified level.

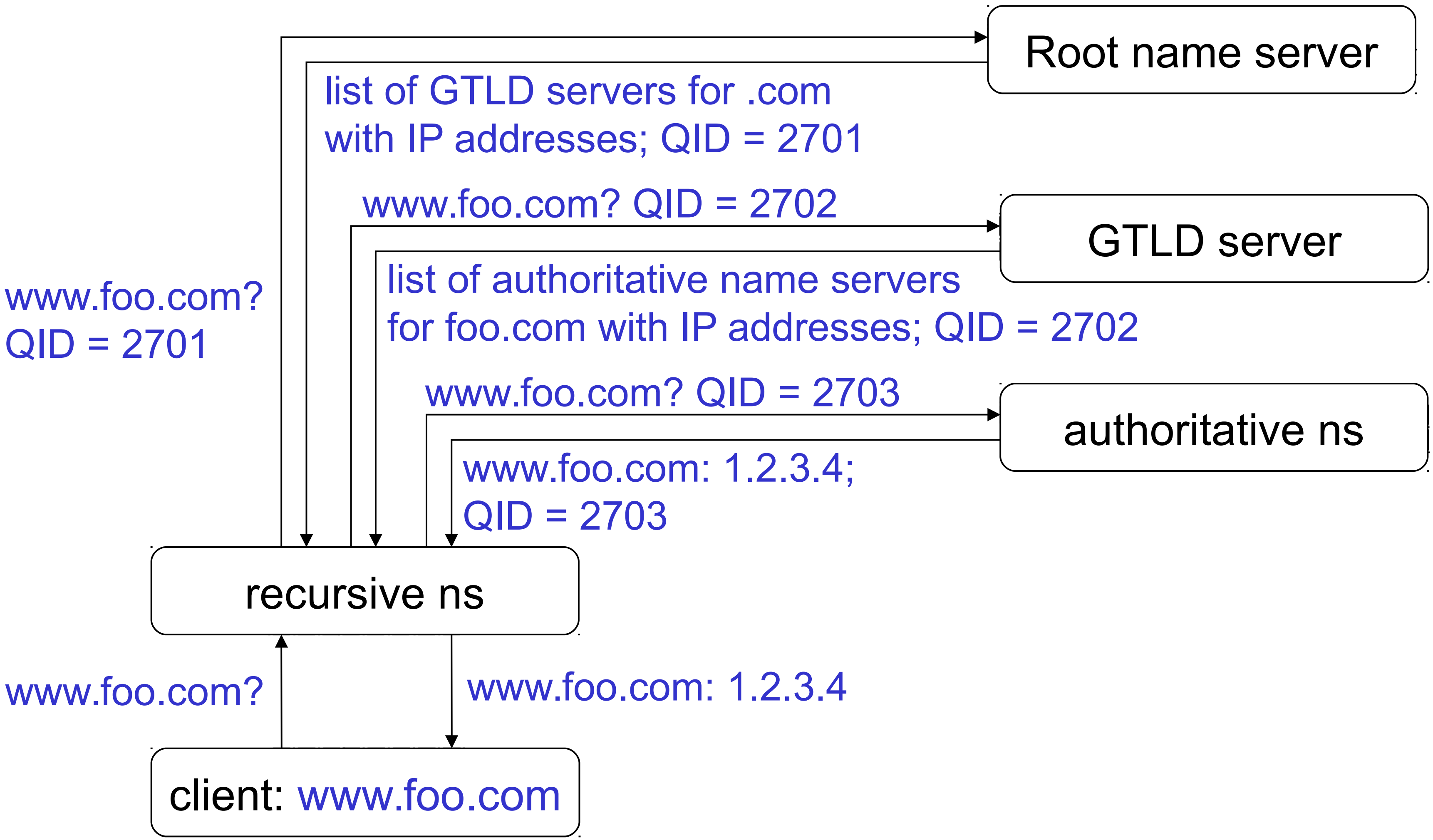
DNS Infrastructure

- 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, .cn, …
  - 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 name servers in their TLD.
- Recursive name servers pass client requests on to other name servers and cache answers received.

IP Address Lookup – Simplified

- Client sends request to its local recursive name server asking to resolve a host name (target).
- Recursive name server refers request to one of the root servers.
- Root server returns list of GTLD servers for the target's TLD; also sends glue records that give the IP addresses of those servers.
- Recursive name server refers request to one of the GTLD servers.
- GTLD server returns list of authoritative name server for the target's domain, together with their IP addresses (glue records).
- Local recursive name server refers the request to one of the authoritative name servers.
- Authoritative mail server provides authoritative answer with IP address to local name server.
- Local recursive name server sends answer to client.

Name resolution



Cache & Time-to-live

- Simplified description left out an important aspect.
- Performance optimisation: when name server receives an answer, it stores answer in its cache.
- When receiving a request, name server first checks whether answer is already in its cache; if this is the case, the cached answer is given.
- 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?
- Long TTL = high security, low TTL = low security?

Light-weight Authentication

- Messages on Internet cannot be intercepted; 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 its answer; applies also to answer from authoritative name server.
- 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?

- If query is not passed by mistake to the attacker, her chance of generate faking a answer is 2^{-16}.
- If
  - root servers entries at the local name server are correct,
  - routing tables in the root servers are correct,
  - routing tables in the GTLD servers are correct,
  - cache entries at recursive name server are correct,

  the attacker will not see original query ID.
- Security relies on the assumption that routing from local recursive name server to authoritative name server is correct.
- Attack method: guess QID to subvert cache entries.

Compromising Authentication

- If routing to and from root servers and GTLD servers cannot be compromised, the attacker can only try to improve her chances of guessing a query ID.
- Some (earlier) versions of BIND used a counter to generate the QID (as on slide 5!).
- Cache poisoning attack:
  1. Ask recursive name server to resolve host name in attacker's domain.
  2. Request to attacker's name server contains current QID.
  3. Ask recursive name server to resolve host name you want to take over; send answer that includes next QID and maps host name to your chosen IP address.
  4. If your answer arrives before the authoritative answer, your value will be cached; the correct answer is dropped.

Predictable Challenges

- Lesson: If you want to perform authentication without cryptography, do not use predictable challenges.
- More ways of improving the attack's chances:
  - To account for other queries to the recursive name server concurrent to the attack, send answers with QIDs from a small window.
  - To increase the chance that fake answer arrives before authoritative answer, slow down authoritative name server with a DoS attack.
  - To prevent that a new query for the host name restores the correct binding, set a long time to live.

Bailiwick Checking

- Performance optimization: name servers send additional resource records to recursive name server, just in case they might come useful.
- Might save round trips during future name resolution.
- Works fine if all name servers are well behaved.
- Do not trust your inputs: malicious name server might provide resource records for other domains, e.g. with IP addresses of its choice.
- Bailiwick checking: additional resource records not coming from the queried domain, i.e. records "out of bailiwick", not accepted by recursive name server.

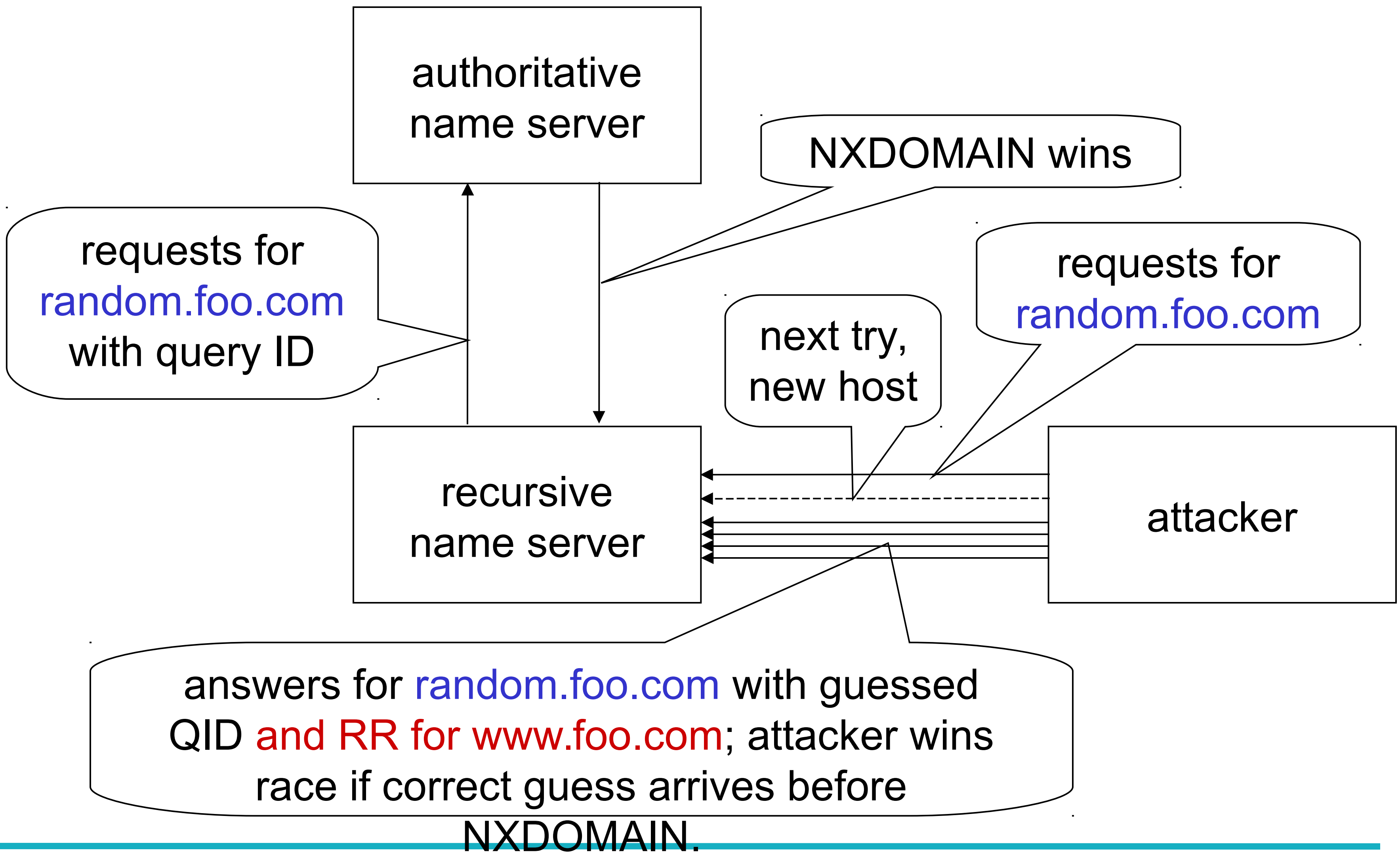
DNS Attack – Next Try

- Attacker in a race with authoritative name server.
- If authoritative answer comes first, the attacker's next attempt has to wait until TTL expires.
- Attacker does not ask for www.foo.com but for a host random.foo.com that is not in recursive name server's cache; triggers a new name resolution request.
  - Defeats TTL as a measure to slow down attacker;
  - TTL not intended as a security mechanism!
- Authoritative name server for foo.com unlikely to have entry for random.foo.com.
- NXDOMAIN answer indicating that host doesn't exist.

Dan Kaminsky's Attack (2008)

- Attacker sends requests for random.foo.com to recursive 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's answer wins race with NXDOMAIN, entry www.foo.com 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



Severity of Attack

- Very serious attack: attacker becomes name server for domains of her choice.
- Attack increases chance of guessing a QID correctly by trying many random host names.
- Reportedly success within 10 seconds.
- Many ways for triggering name resolution at recursive name server.
- Alternative attack strategy: send many faked name server redirects for www.foo.com with guessed QID (version in Kaminsky's black hat talk).

Countermeasures

- Increase search space for attacker: run queries on random ports.
  - Attacker now must guess QID & port number.
- Restrict access to local recursive name server: split name server (split-split name server).
- Trust levels for resource records: access control to prevent unauthorized overwriting of authoritative data.
- DNSSec: cryptographic authentication using digital signatures; give up on QID as a security feature.
- Name server does not reply to malformed queries??
  - Actually helps the attacker.

DNS Rebinding Attacks

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; binds this name to two IP addresses, to its own and to the target's address.
- Client downloads applet from attacker.org; script connects to target; permitted by same origin policy.
- Defence: 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.
- Defence: Don't trust the DNS server on time-to-live; pin host name to original IP address;
  - J. Roskind: Attacks against the Netscape browser. in RSA Conference, April 2001.
  - Duration of pinning is browser dependent.

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.
- General security issue: error handling procedures written without proper consideration of their security implications.

DNS Rebinding Attack

- Next round – browser plug-ins, e.g. Flash.
- Plug-ins may do their own pinning.
- Dangerous constellation:
  - 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.
- Defence (centralize controls): one pinning database for all plug-ins
  - E.g., let plug-ins use the browser's pins.
  - Feasibility depends on browser and plug-in.

DNS Rebinding Attack

- More sophisticated authorisation system: Client browser refers to policy obtained from DNS server when deciding on connection requests.
- Defence: don’t ask DNS server for the policy but the system with the IP address a DNS name is being resolved to.
    - Related to reverse DNS lookup.
    - Similar to defences against bombing attacks in network security.

Firewalls

Introduction

- Cryptographic mechanisms protect data in transit (confidentiality, integrity).
- Authentication protocols verify the source of data.
- We may also control which traffic is allowed to enter our system (ingress filtering) or to leave our system (egress filtering).
- Access control decisions based on information like addresses, port numbers, ...

Firewall

- Firewall: a network security device controlling traffic flow between two parts of a network.
- Often installed between an entire organisation's network and the Internet.
- Can also be installed in an intranet to protect individual departments.
- All traffic has to go through the firewall for protection to be effective.
  - Dial-in lines, 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

Packet Filter

- Inspect headers of IP packets, also TCP and UDP port numbers.
- Rules specify which packets are allowed through the firewall, and which are dropped.
  - Actions: bypass, drop, protect (IPsec channel).
- 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 (eg: FTP, IRC, H323).

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 realise 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 – Eexamples

- Permissive policies: Allow all traffic, but block ...
  - Irc
  - telnet
  - snmp
  - …
- Restrictive policies: block all traffic, but allow ...
  - http
  - Pop3
  - Smtp
  - ssh
  - …

Rule Order

- A firewall policy is a collection of rules.
- Packets can contain several headers (→ IPsec).
- When setting a policy, you have to know in which order rules (and headers) are evaluated.
- Two main options for ordering rules:
  - Apply first matching entry in the list of rules.
  - Apply the entry with the best match for the packet.

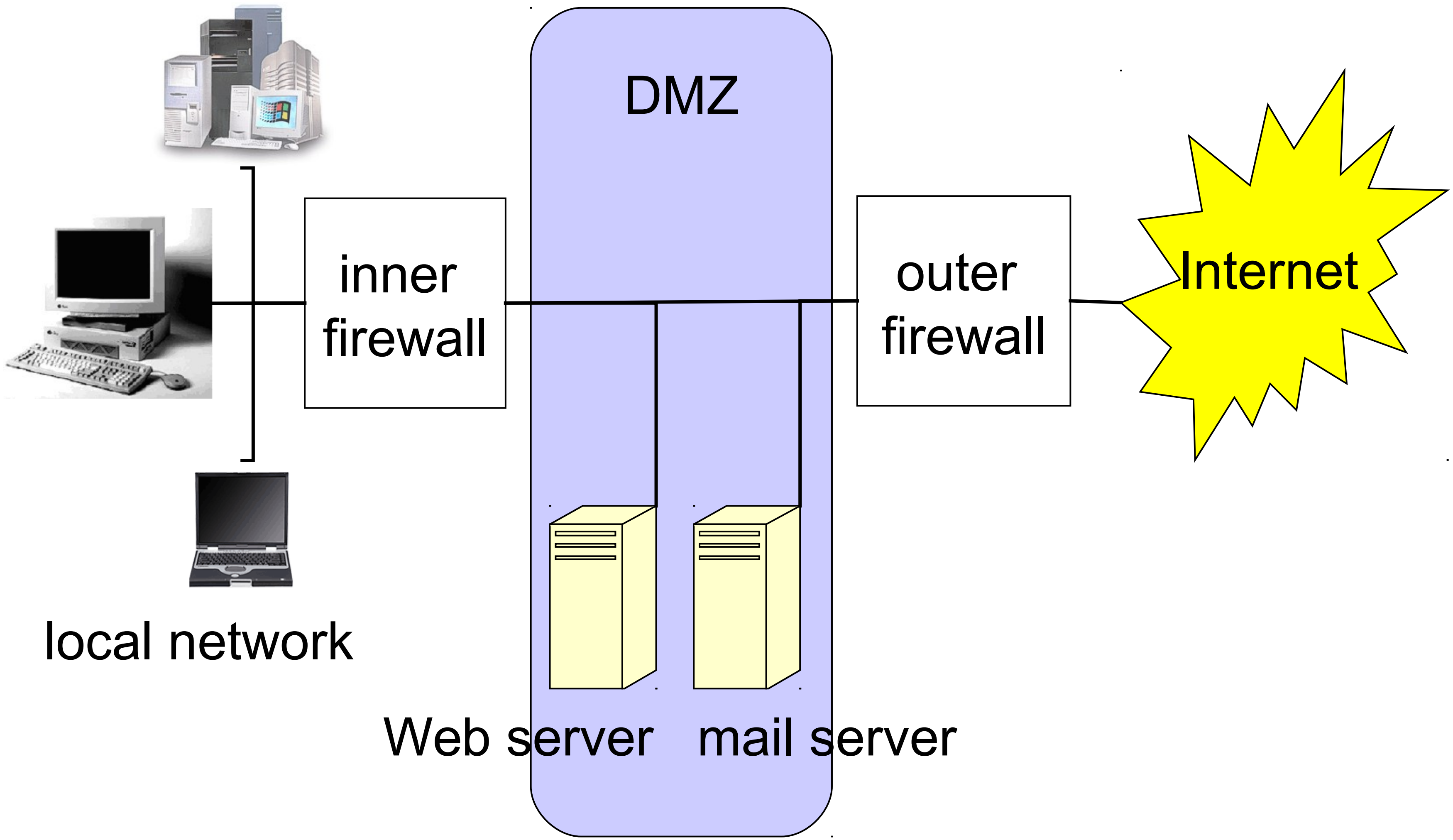
Typical Firewall Ruleset

- 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 to put, for example, 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: “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 tunnelling: sending data for one protocol through another protocol circumvents the firewall.
  - As more and more administrators block almost all ports but have to leave port 80 open, more and more protocols are tunnelled through http to get through the firewall.
- Encrypted traffic cannot be examined and filtered.

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.

Comment

- 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).
- We will now look at ways of detecting network attacks.

Intrusion Detection Systems

- Passive supervision of network, analogue to intruder alarms.
  - Creates more work for personnel.
  - Provides security personnel with volumes of reports that can be presented to management …
- Two approaches to Intrusion Detection:
  - Knowledge-based IDS – Misuse detection
  - Behaviour-based IDS – Anomaly detection
- Network based and host based IDS.
- Given the (near) real-time nature of IDS alerts, an IDS can also be used as response tool.

Knowledge-based IDS

- Knowledge-based IDS looks for patterns of network traffic or activity in log files that indicate suspicious behaviour, using information such as:
  - known vulnerabilities of particular OS and applications;
  - known attacks on systems;
  - given security policy.
- 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.

Knowledge-based IDS

- Only as good as database of attack signatures:
  - New vulnerabilities not in the database are constantly being discovered and exploited;
  - Vendors need to keep up to date with latest attacks and issue database updates; customers need to install these;
  - Large number of vulnerabilities and different exploitation methods, so effective database difficult to build;
  - Large database 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?
- Statistical anomaly detection uses statistical techniques to detect attacks.
- First establish base-line behaviour: 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.
- Also known as behaviour-based detection.

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.
  - See e.g. Richard Bejtlich: Interpreting Network Traffic: A Network Intrusion Detector's Look at Suspicious Events, Proceedings of the 12th Annual Computer Security Incidence Handling Conference, Chicago, 2000.
- False negatives: attack missed because it fell within the bounds of normal behaviour.
- This issue also applies to knowledge-based systems.

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 behaviour 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.
- Centralised console to manage sensor network, analyze data ($\rightarrow$ 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.

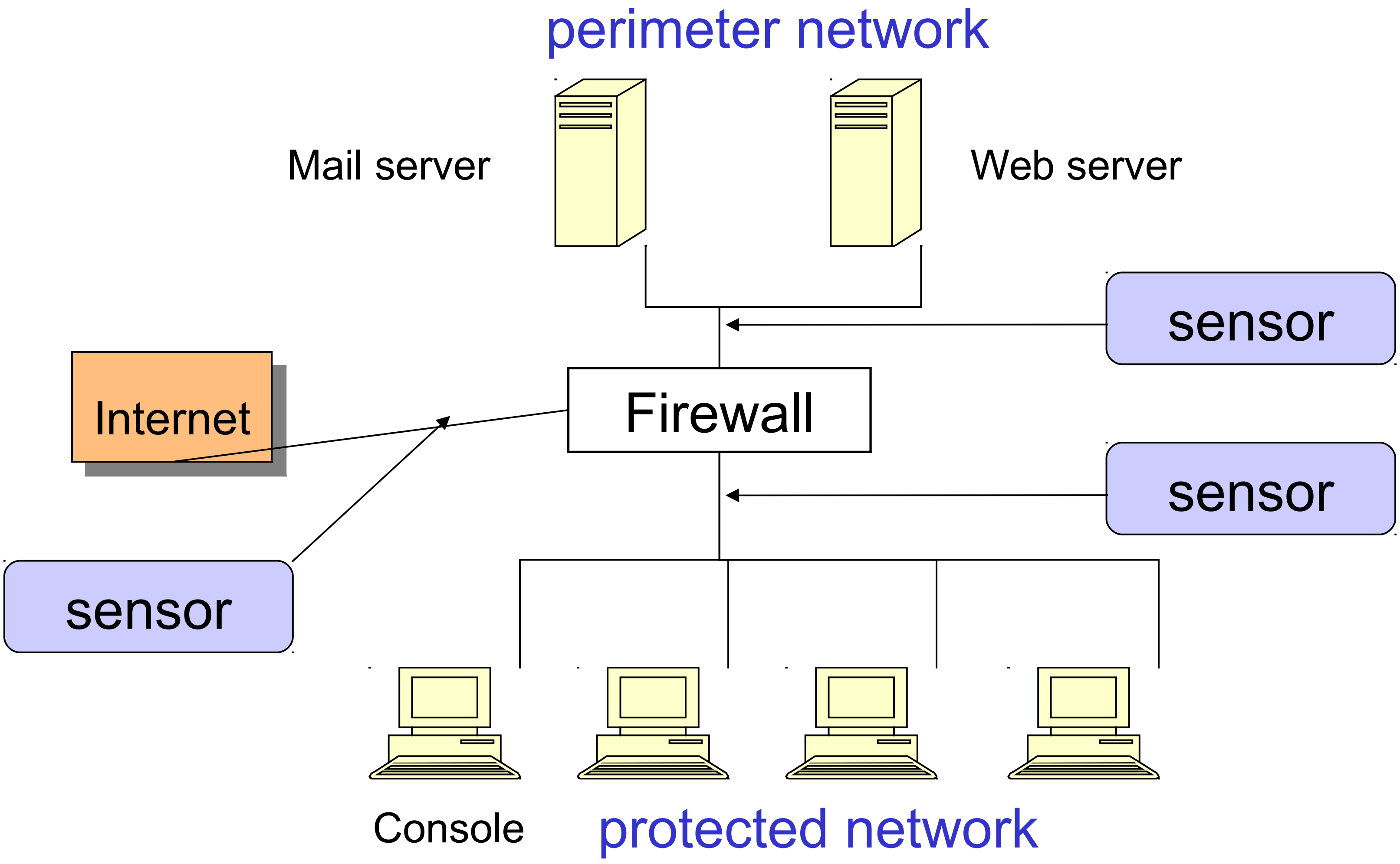
HIDS & NIDS

- 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.
- Monitors and analyzes all traffic in real-time.
- Attack recognition module uses three common techniques to recognize attack signatures:
  - Pattern, expression or bytecode matching;
  - Frequency or threshold crossing (e.g. detect port scanning activity);
  - Correlation of lesser events (in reality, not much of this in commercial systems).

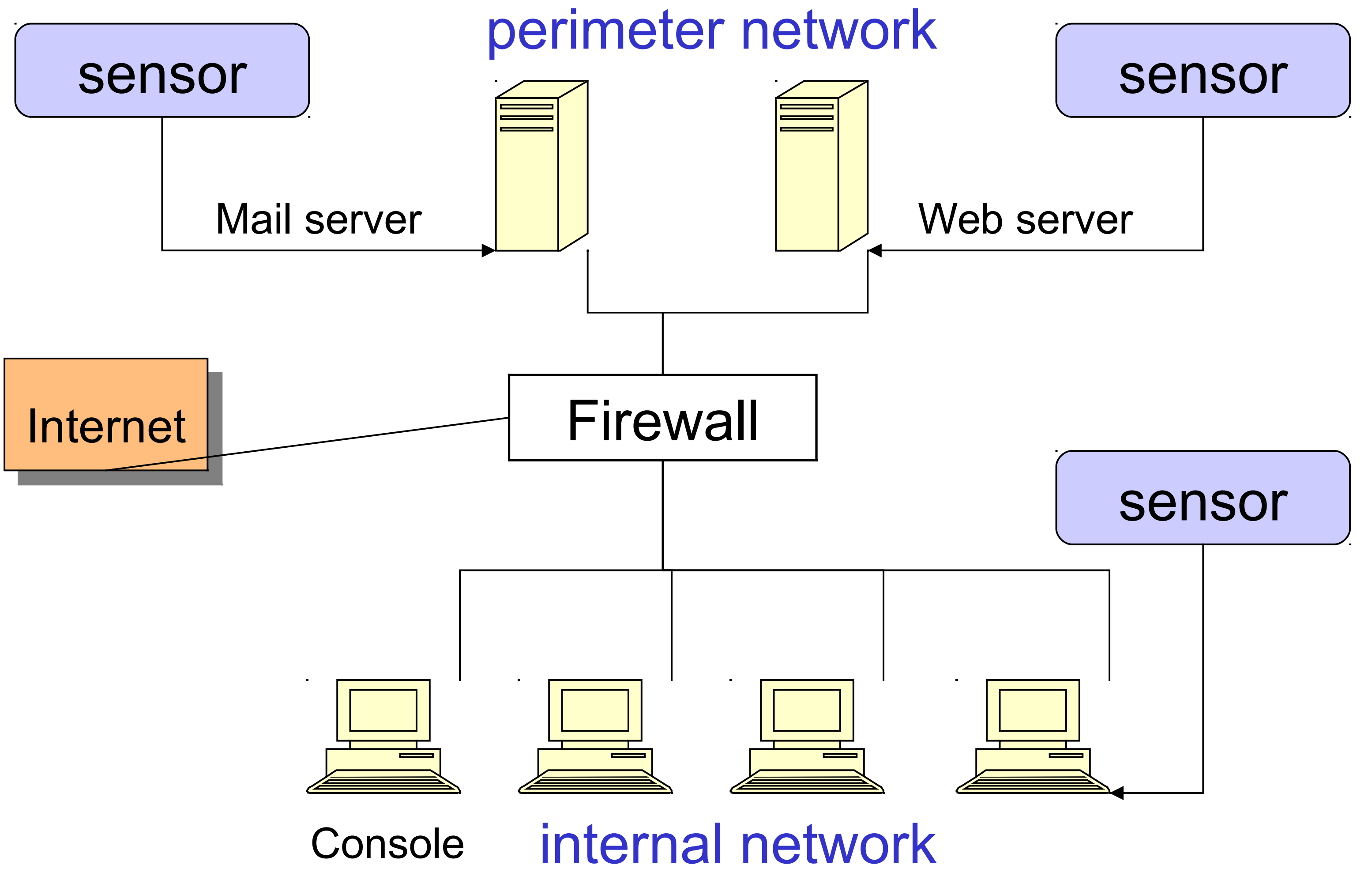
Placement of NIDS



Host-based IDS

- Typically monitors system, event, and security logs on Windows and syslog in Unix environments.
  - E.g., 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;
  - E.g., 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 – 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.
  - E.g., 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.
- 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”
  - Laurence Spitzner, “The value of honeypots”, SecurityFocus, October 2001
- Honeypot: technology to track, learn and gather evidence of hacker activities.

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 defence method.

Honeynet

- Network of honeypots.
- Supplemented by firewalls and intrusion detection systems – Honeywall.

- Advantages:
  - “More realistic” environment
  - Improved possibilities to collect data

Summary

- Apply prevention, detection and reaction in combination.
- IDS useful second line of defence (in addition to firewalls, cryptographic protocols, etc.).
- IDS deployment, customisation and management is generally not straightforward.
- Anomalies are not necessarily attacks.