Chapter 6

Network Security

Intruders

- Significant issue for networked systems is hostile or unwanted access
- Either via network or local
- Can identify classes of intruders:
  - masquerader: pretend to be an “acceptable” user
  - misfeasor: authentic user performing unauthorized actions
  - clandestine user: secretly accessing the network/performing actions
- Varying levels of competence

Intruders

- Clearly a growing publicized problem
  - from “Wily Hacker” in 1986/87
  - to clearly escalating CERT stats
- Range
  - benign: explore, still costs resources
  - serious: access/modify data, disrupt system
- Led to the development of CERTs
  - **C**omputer **E**mergency **R**esponse **T**eam
- Intruder techniques and behavior patterns constantly shifting, have common features

Examples of intrusion

- Remote user (even root) compromise
- Web server defacement
- Guessing / cracking passwords
- Copying viewing sensitive data / databases
- Capturing internal network traffic
- Using an unsecured modem / debug port to access network
- Impersonating a user to reset password
- Using an unattended workstation
- Encrypting data and requesting ransom
- Damaging / destroying data or user accounts
- ...

Hackers

- **Motivated by curiosity**, sometimes thrill of access and status
  - hacking community a strong meritocracy
  - status is determined by level of competence
- Benign intruders might be tolerable
  - do consume resources and may slow performance
  - can’t know in advance whether benign or malign
- IDS / IPS / VPNs can help counter
- Awareness led to establishment of CERTs
  - collect / disseminate vulnerability info / responses
- Current consensus on best way to deal with friendly hackers:
  - **Vulnerability Rewards Programs** (VRPs) that pay a bounty for newly discovered vulnerabilities
  - run by manufacturer or third parties
  - often coupled with agreements for *coordinated disclosure*

Hacker behavior example

1. Select target using IP lookup tools (`nmap`, Shodan)
2. Map network for accessible services (`nmap`, Shodan)
3. Identify potentially vulnerable services (OpenVAS, Metasploit)
4. Brute force (guess) passwords
5. Elevate privileges (Metasploit)
6. Install remote administration tool (Metasploit)
7. Wait for admin to log on and capture password
8. Use password to access remainder of network

Good collection of free tools: https://www.kali.org/

Criminal enterprise

- Organized groups of hackers now a threat
  - corporation / government / loosely affiliated gangs, sometimes supported by countries (and therefore often well-funded)
  - typically young
  - sources from many different countries
- Criminal hackers usually have specific targets
  - many possible targets (financial and identity theft, sabotage, false information campaigns, etc.)
  - motivated either financially or politically
- Once penetrated act quickly and get out
  - exception: “Advanced Persistent Threats” with the goal of staying undetected over long time (often years)
- IDS / IPS help but less effective
- Sensitive data needs strong protection → proper key management

Criminal enterprise behavior

1. Act quickly and precisely to make their activities harder to detect
2. Exploit perimeter via vulnerable ports
3. Use Trojan horses (hidden software) to leave back doors for reentry
   Note: Professional groups often **build their own tools**, antivirus scanners therefore may not have seen the patterns before.
4. Use sniffers to capture passwords
5. Do not stick around until noticed
6. Make few or no mistakes

Insider attacks

- **Among most difficult to detect and prevent**
- Employees have access and (sometimes extensive) systems knowledge
- May be motivated by revenge / entitlement
  - when employment terminated
  - taking customer data when moving to competitor
  - can also be politically motivated (planted spies)
- IDS / IPS may help but also need:
  - least privilege, monitor logs, strong authentication, termination process to block access, and mirror data

Insider behavior example

1. Create network accounts for themselves and their friends
2. Access accounts and applications they wouldn't normally use for their daily jobs
3. Conduct furtive instant-messaging chats
4. Perform large downloads and file copying
5. Access the network during off hours
6. Insert backdoors into code or systems configuration
7. Sign modified code with organization keys

But: many of these could also have legitimate reasons → distinguishing between real insider attack and exceptional usage patterns is **hard**!

Intrusion techniques

- Aim to gain access and/or increase privileges on a system
- Often use system / software vulnerabilities
- Primary goal often is to acquire passwords / access tokens / keys
  - to then exercise access rights of owner
- Basic attack methodology
  1) target acquisition and information gathering
  2) initial access
  3) privilege escalation
  4) covering tracks

Password guessing

- One of the most common attacks
- Attacker knows a login (from email/web page etc.)
- Then attempts to guess password for it
  - defaults, short passwords, common word searches
  - user info (variations on names, birthday, phone, common words/interests)
  - exhaustively searching all possible passwords
- Check by login or against stolen password file
- Success depends on password chosen by user
- Surveys show many users choose poorly

Mitigation: unique, high-entropy passwords (password manager)

Password capture

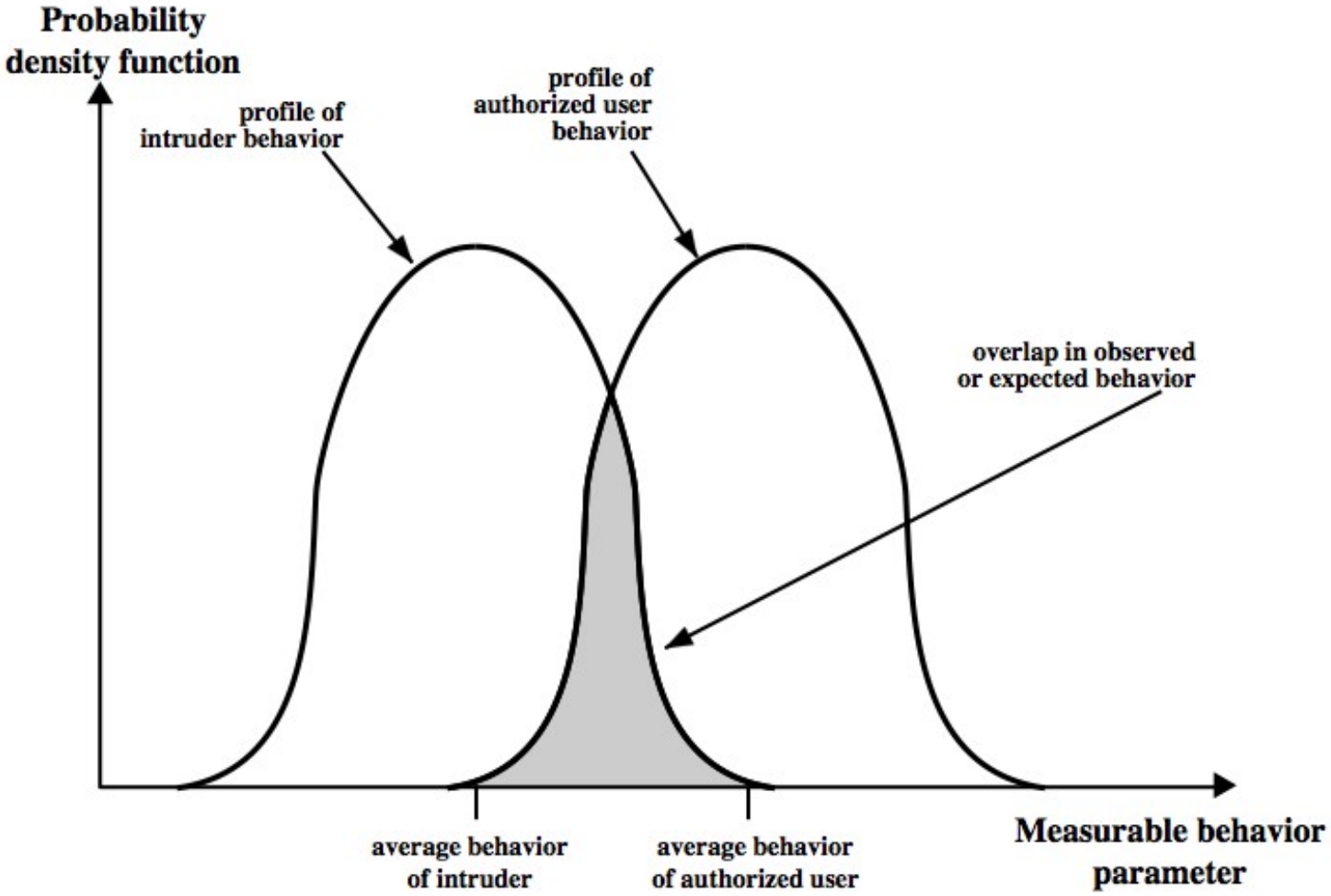
- Another attack involves **password capture**
  - watching over shoulder as password is entered
  - using a trojan horse program to collect
  - monitoring an insecure network login
    - eg. telnet, FTP, web, email
  - extracting recorded info after successful login (web history/cache, etc.)
  - faking login pages of legitimate web pages / apps → **phishing**
- Using valid login/password can impersonate user
- Users need to be educated to use suitable precautions/countermeasures

Mitigation: second factor authentication (FIDO2/WebAuthn)

Intrusion detection

- Inevitably will have security failures
- Need also to detect intrusions to
  - block if detected quickly
  - act as deterrent
  - collect information to improve security
- Assume intruder will behave differently to a legitimate user
  - but will have imperfect distinction between legitimate and malicious
  - problem: how do we describe/learn/… what a legitimate user does, which also changes over time?

Intrusion detection



Approaches to intrusion detection

- Statistical anomaly detection
  - attempts to define normal/expected behavior
  - profile based – **learning** “normal” behavior from data
  - threshold to distinguish classification
  - detect anomalies as significant deviations from profile
- Rule-based detection
  - attempts to **define** proper behavior
  - penetration identification based on definition of improper behavior
  - rules are written by domain experts
  - can use allow (white) or block/warn (black/gray) lists

Audit records

- Fundamental tool for intrusion detection
- Native audit records
  - part of all common multi-user OS
  - already present for use
  - may not have info wanted in desired form
- Detection-specific audit records
  - created specifically to collect wanted info
  - at cost of additional overhead on system

Base-rate fallacy

- Practically an intrusion detection system needs to detect a substantial percentage of intrusions with few false alarms
  - if too few intrusions detected → false sense of security
  - if too many false alarms → ignored / waste time
- This is very hard to do
- Existing systems seem not to have a good record

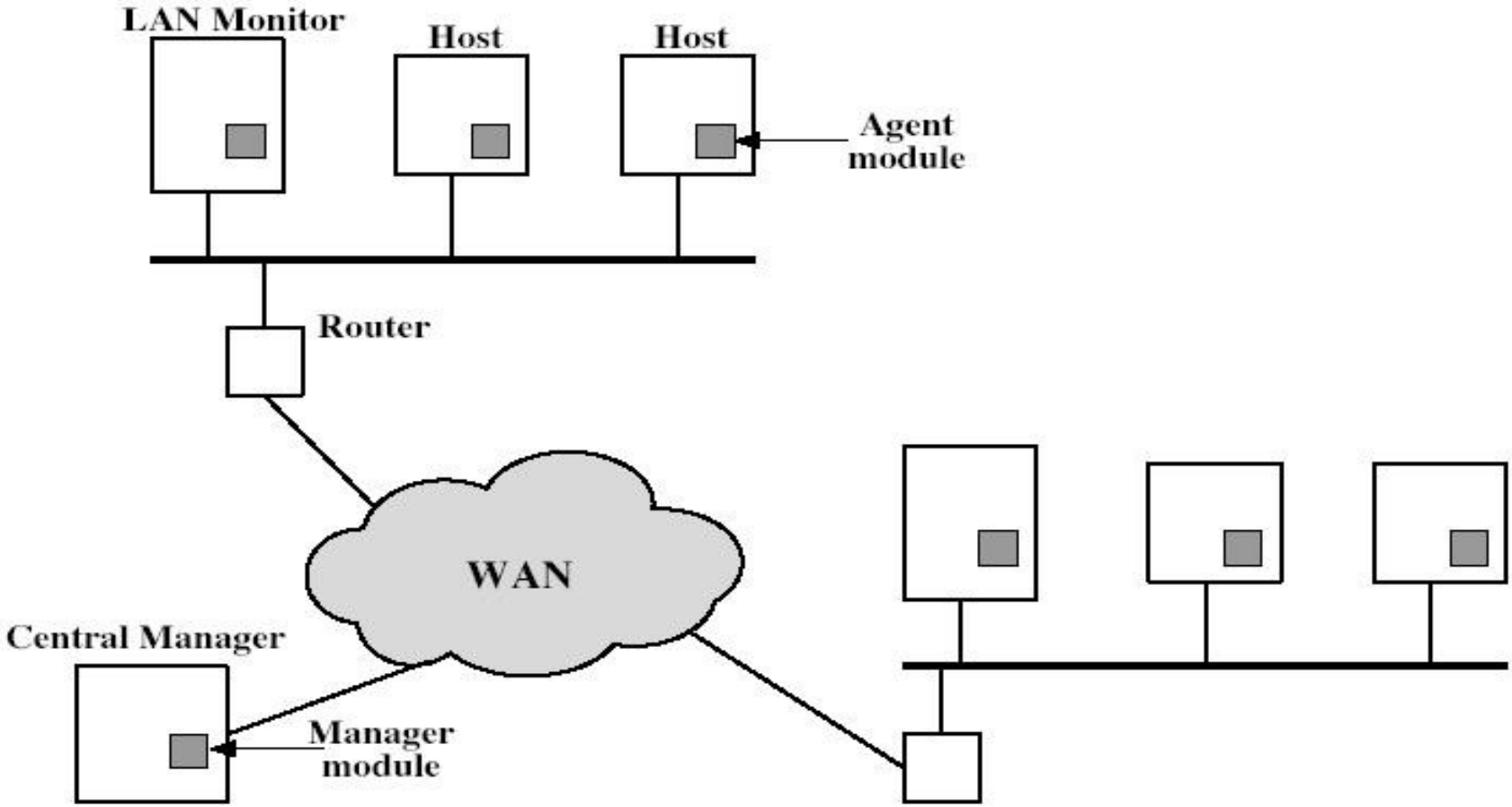
Base-rate fallacy

- Assume we have a “terrorist detector”, which is **99.9% correct**.
  - every terrorist is detected without failure (this is hard, but pretend)
  - 1 in 1000 innocents is mistakenly labeled as terrorist (99.9%)
  - also assume 1 in 100.000 persons is actually a terrorist
- We now let all Austrians pass in front of the detector. How likely is it that an alarm from the detector actually marks a terrorist?
  - 8 Million Austrians → 80 terrorists → all detected
  - 8 Million Austrians → 8000 false alarms
  - 80 of 8080 are actually terrorists → 0,99% of all alarms are real, and
  - **99% of all alarms are false positives**
    - Anyone detected as terrorist is almost guaranteed innocent!
- Intrusion detection questions:
  - How many connections/packets/... per day?
  - How good is your detector?
  - What if the detector accuracy is symmetric, i.e. some attacks are not recognized?

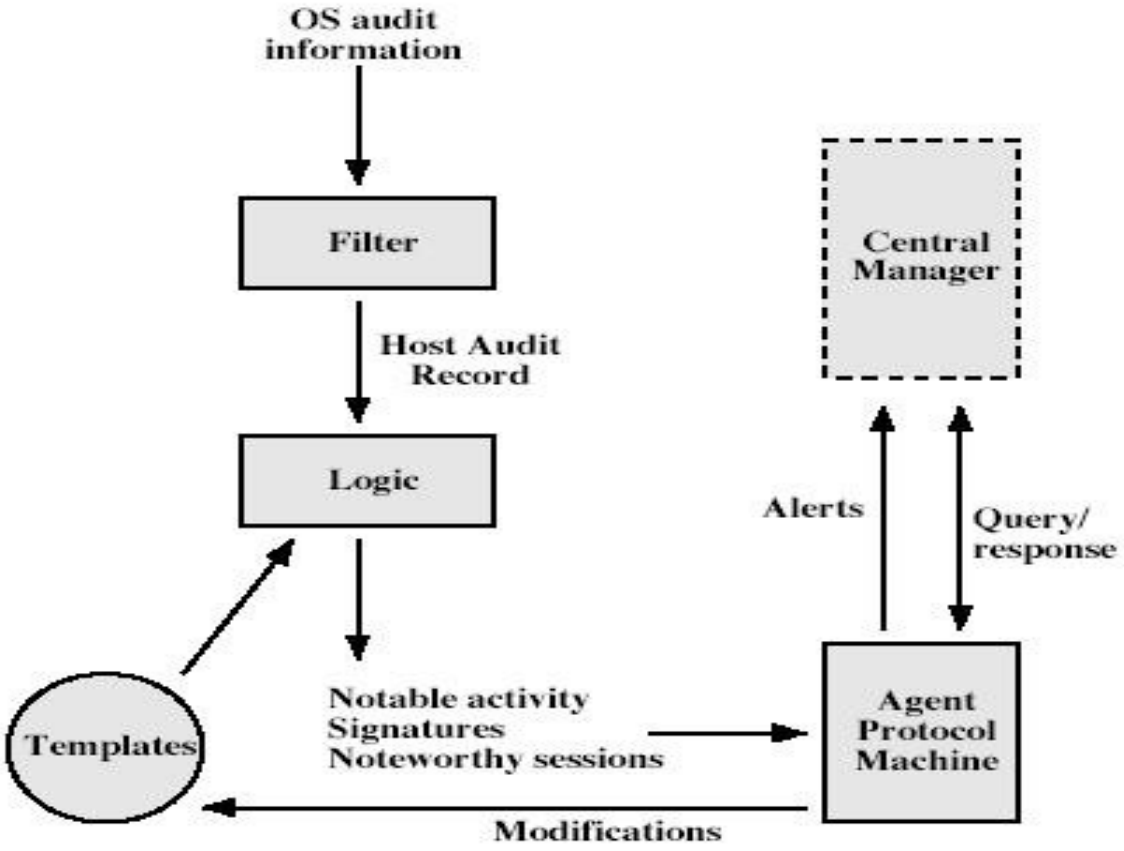
Distributed intrusion detection

- Traditional focus is on single systems
- But typically have networked systems
  → use (distributed) **Network Intrusion Detection Systems (NIDS)**
- More effective defense has these working together to detect intrusions
- Issues
  - dealing with varying audit record formats
  - integrity and confidentiality of networked data
  - centralized or decentralized architecture

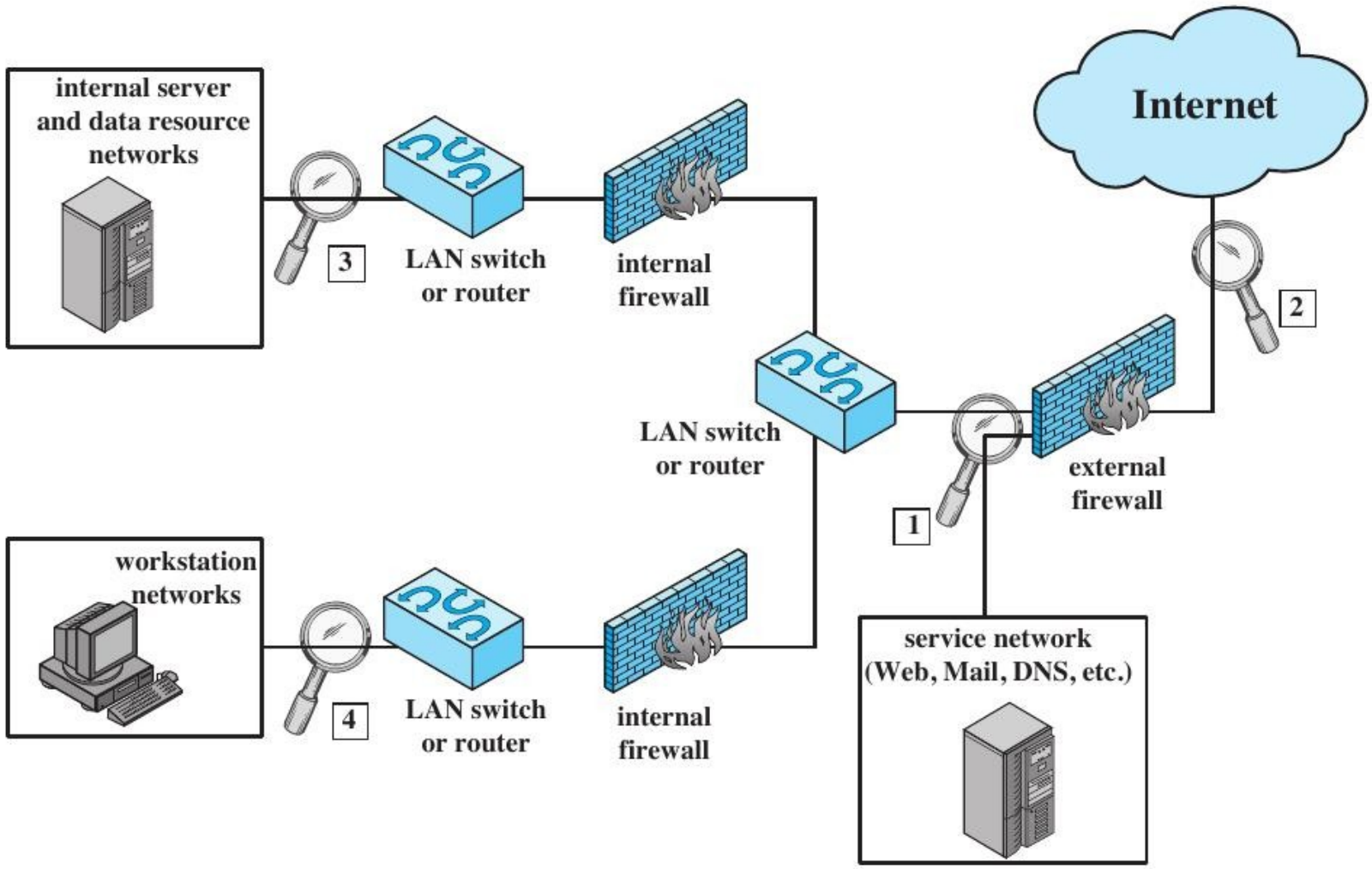
Distributed intrusion detection: architecture



Distributed intrusion detection: agent implementation



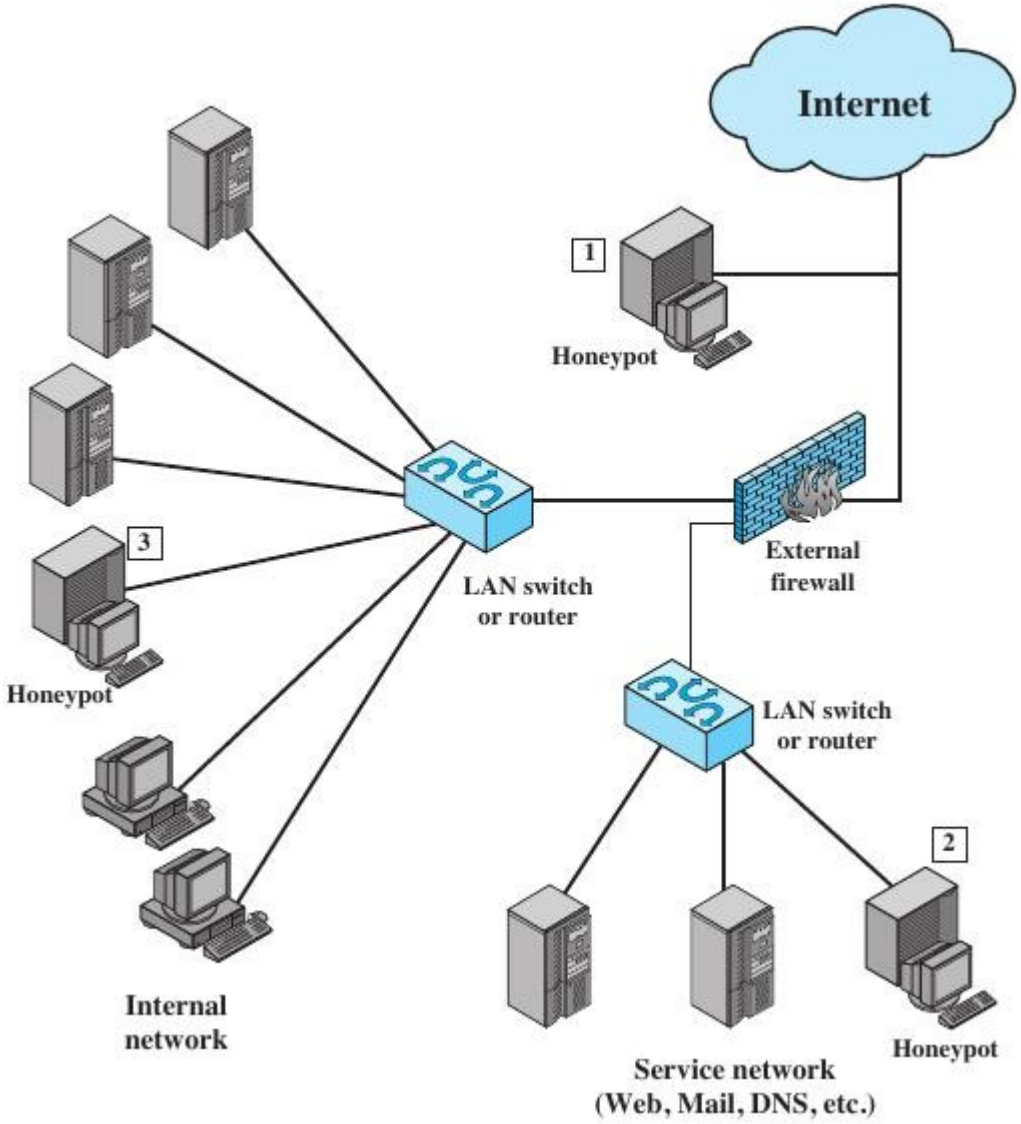
Example of distributed NIDS sensor deployment



Honeypots

- Decoy systems to lure attackers
  - away from accessing critical systems
  - to collect information of their activities
  - to encourage attacker to stay on system so administrator can respond
- Are filled with fabricated information
- Instrumented to collect detailed information on attackers activities
- Single or multiple networked systems
- Cf. IETF Intrusion Detection WG standards

Example of honeypot deployment



Firewalls

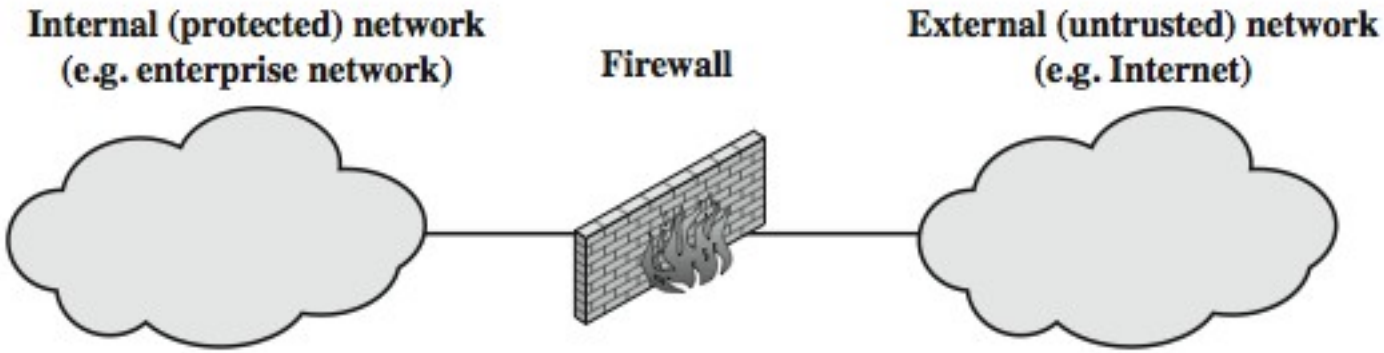
- Most networks in current use are connected to the Internet in one way or the other (often necessary e.g. for OS/virus/IDS signature updates even on isolated networks)
- Has persistent security concerns
  - can't easily secure every system in organization individually
- Typically use a **Firewall**
- To provide **perimeter defence**
- As part of comprehensive security strategy

  Note: with mobile devices roaming in different networks, there no longer is a perimeter → central firewalls no longer work

What is a firewall?

- A **choke point** of control and monitoring
- Interconnects networks with differing trust
- Imposes restrictions on network services
  - only authorized traffic is allowed
- Auditing and controlling access
  - can implement alarms for abnormal behavior
- Provide NAT and usage monitoring
- Implement VPNs using WireGuard, IPsec, OpenVPN, etc.
- Must be hardened against penetration to the system itself

What is a firewall?



Firewall limitations

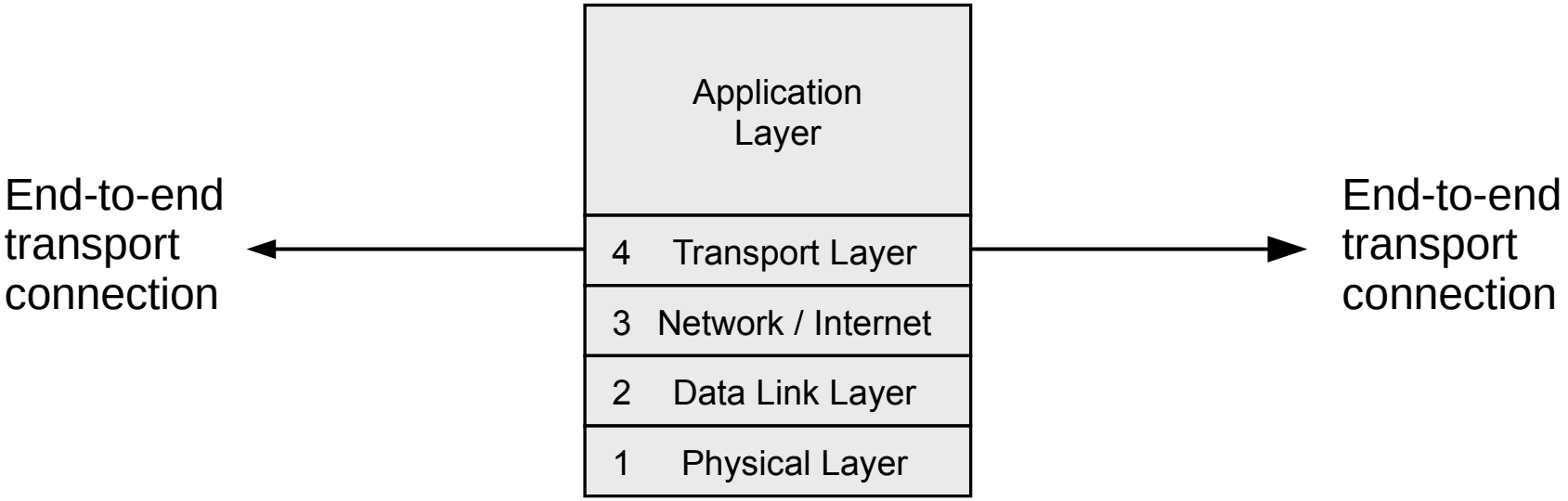
- Cannot protect from attacks bypassing it
  - e.g. sneaker net, utility modems / debug ports, trusted organisations, trusted services (e.g. TLS/SSH)
  - all mobile devices outside the trusted network
- Cannot protect against internal threats
  - e.g. disgruntled or colluding employees
- Cannot protect against access via WLAN
  - if improperly secured against external use
- Cannot protect against malware imported via laptop, PDA, storage infected outside

- Imperfect; but not using it is even worse!

Firewalls: Packet filters

- Simplest, fastest firewall component
- Foundation of any firewall system
- Examine each IP packet (no context) and permit or deny according to rules
- Hence restrict access to services (ports)
- Possible default policies
  - that not expressly permitted is prohibited → often used for incoming
  - that not expressly prohibited is permitted → often used for outgoing

Firewalls: Packet filters



Attacks on packet filters

- IP address spoofing
  - fake source address to be trusted
    - easy for UDP, hard for TCP
  - mitigation: add filters on router to block
- Source routing attacks
  - attacker sets a route other than default
  - mitigation: block source routed packets
- Tiny fragment attacks
  - split header info over several tiny packets
  - mitigation: either discard or reassemble before check

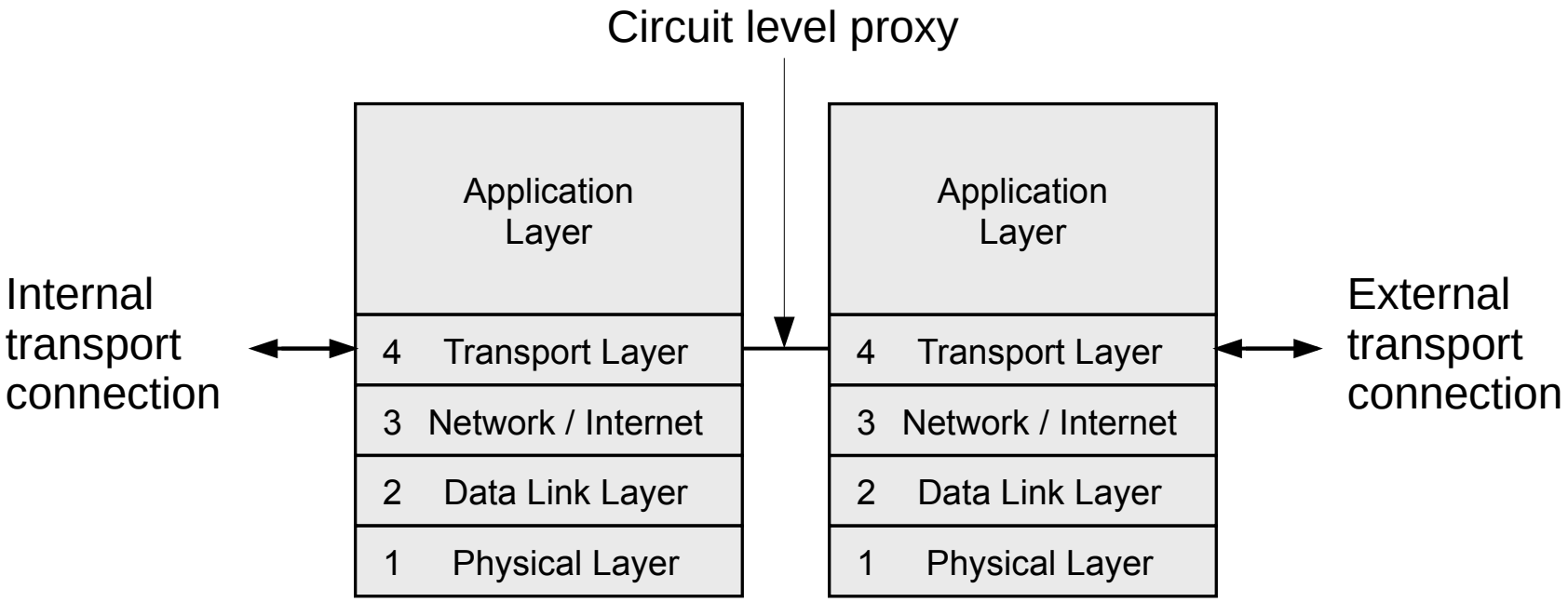
Firewalls:
Stateful packet filters

- Traditional packet filters do not examine higher layer context
  - i.e. matching return packets with outgoing flow
- Stateful packet filters address this need
- They examine each IP packet in context
  - keep track of client-server sessions
  - check each packet validly belongs to one
- Hence are better able to detect bogus packets out of context
- May even inspect limited application data

Firewalls: Circuit level gateway

- Relays two TCP connections
- Imposes security by limiting which such connections are allowed
- Once created usually relays traffic without examining contents
- Typically used when trusting internal users by allowing general outbound connections
- SOCKS protocol is commonly used for setting up circuits
  - Note: e.g., **Tor** acts as a SOCKS proxy

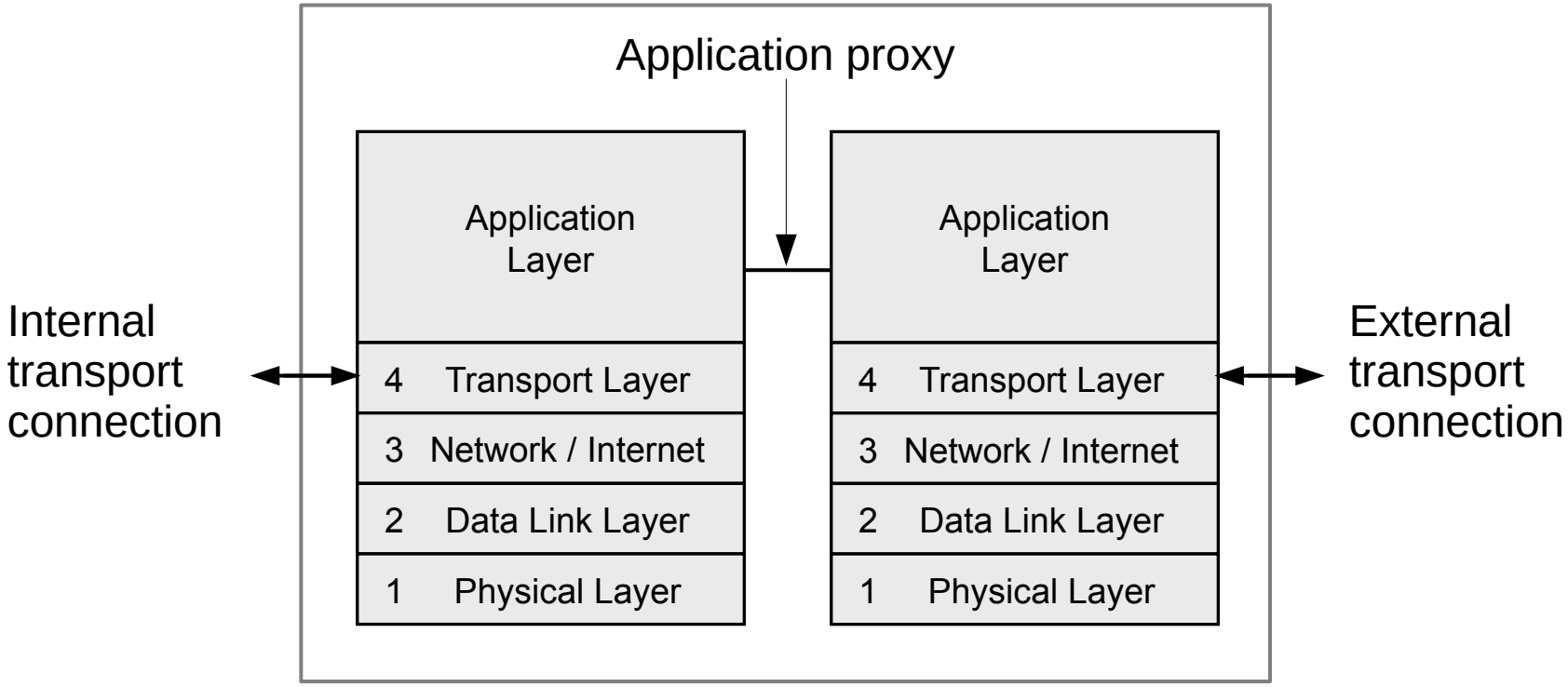
Firewalls:
Circuit level gateway



Firewalls: Application level gateway (proxy)

- Have application specific gateway / proxy
  - like circuit-level gateway, but also knows and inspects the content
- Has full access to protocol
  - user requests service from proxy
  - proxy validates request as legal
  - then actions request and returns result to user
  - can log / audit traffic at application level
- Need separate proxies for each service
  - some services naturally support proxying
  - others are more (or very) problematic
    - e.g. proxying encrypted/signed connections

Firewalls: Application level gateway (proxy)



Bastion host

- Highly secure host system
- Runs circuit / application level gateways
- Or provides externally accessible services
- Potentially exposed to "hostile" elements
- Hence is secured to withstand this
  - hardened OS, essential services, extra authentication
  - proxies small, secure, independent, non-privileged
- May support 2 or more network connections
- May be trusted to enforce policy of trusted separation between these net connections

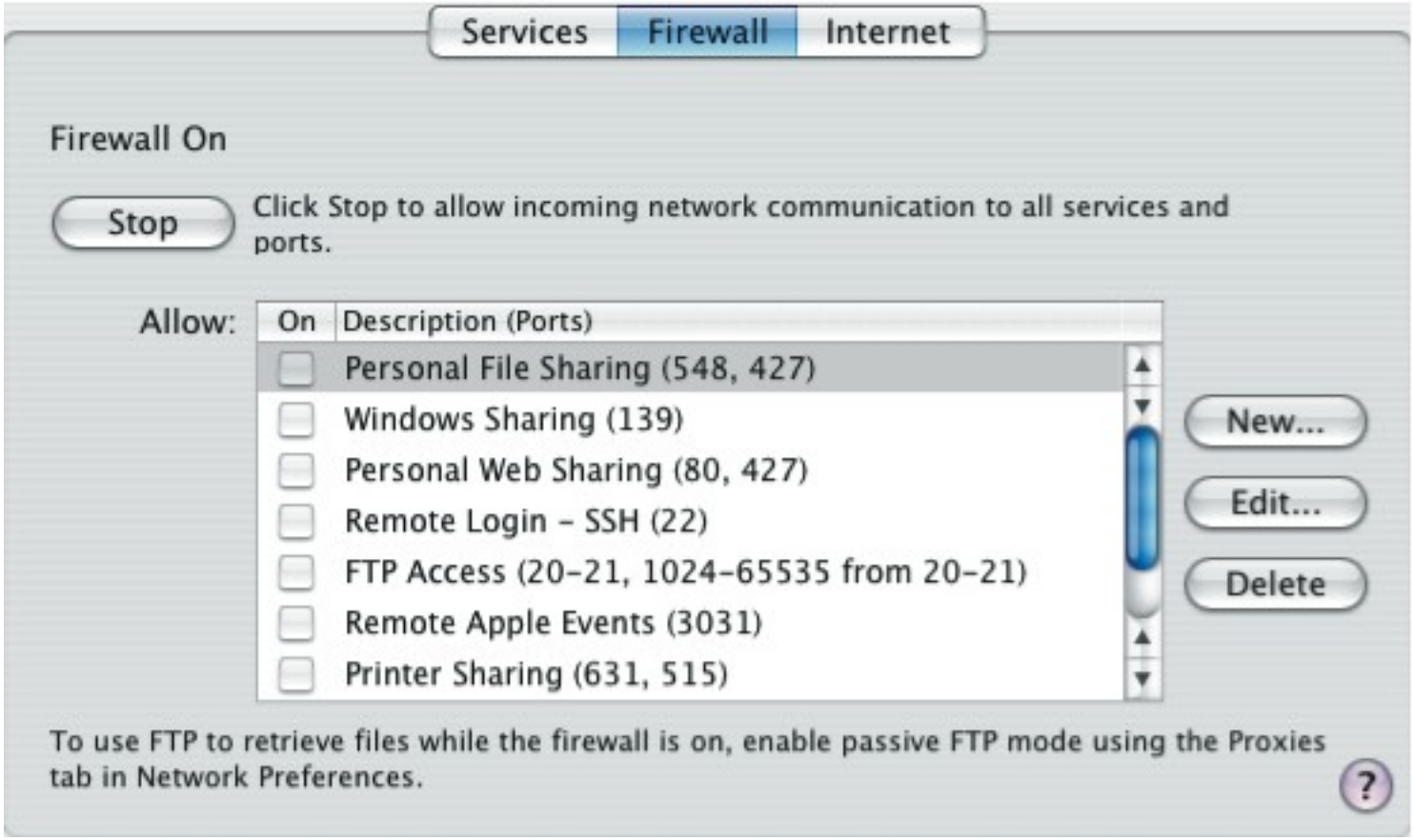
Host-based firewalls

- Software module used to secure individual host
  - available in many operating systems
  - or can be provided as an add-on package
- Often used on servers
- Advantages:
  - can tailor filtering rules to host environment
  - protection is provided independent of topology
  - provides an additional layer of protection

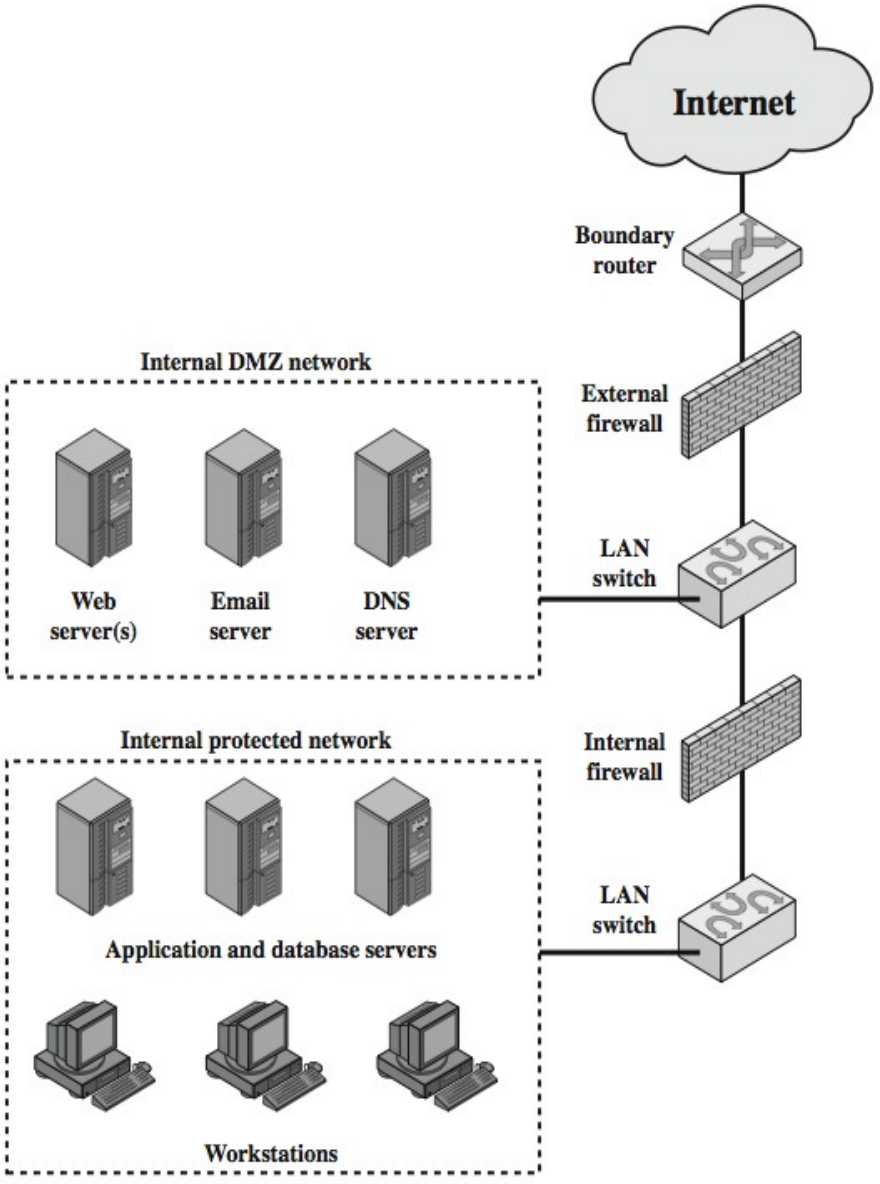
Personal firewalls

- Controls traffic between PC/workstation and Internet or enterprise network
- A software module on personal computer
- Or in home/office DSL/cable/ISP router
- Typically much less complex than other firewall types
- Primary role to deny unauthorized remote access to the computer
- And monitor outgoing activity for malware

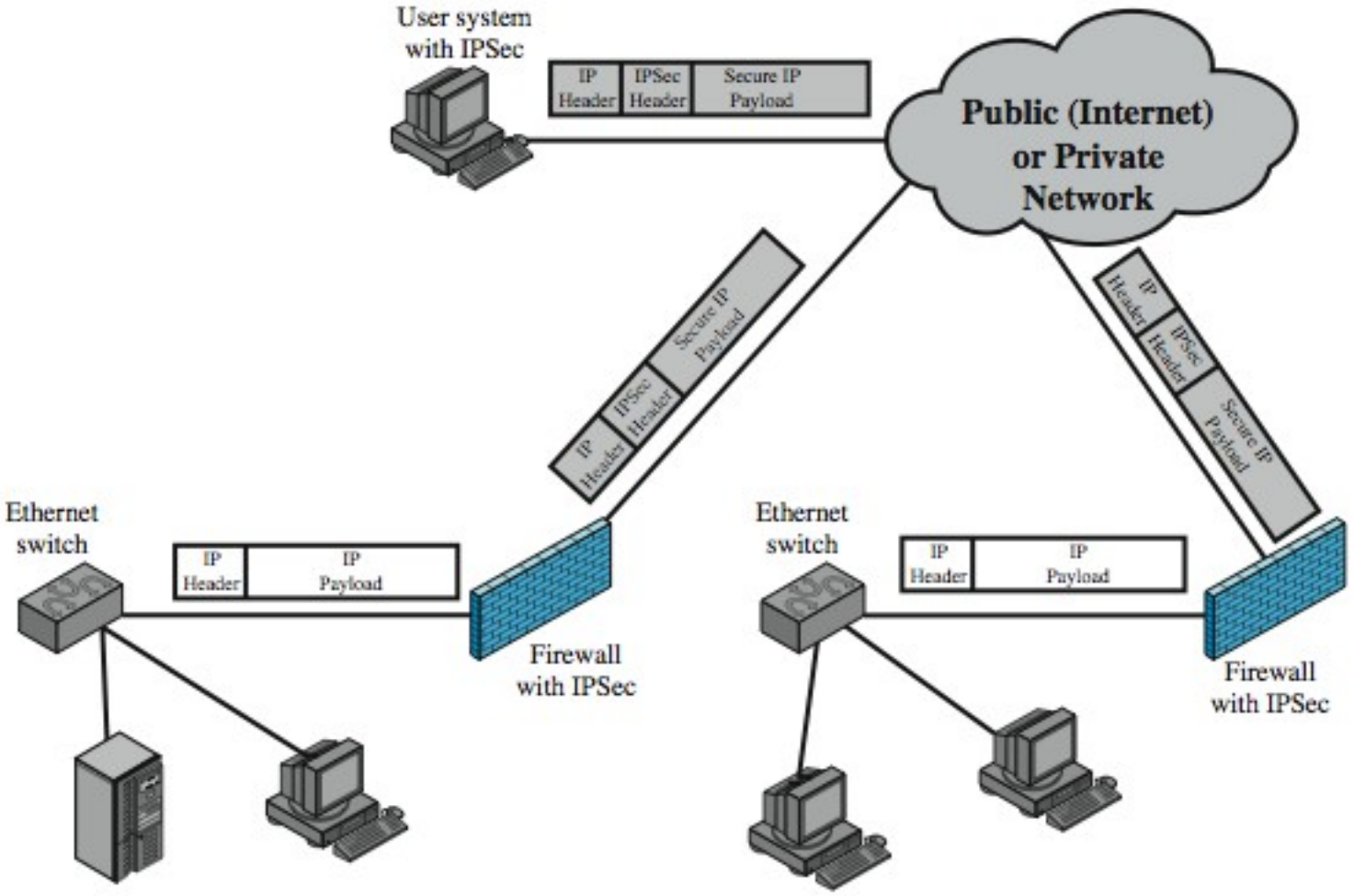
Personal firewalls



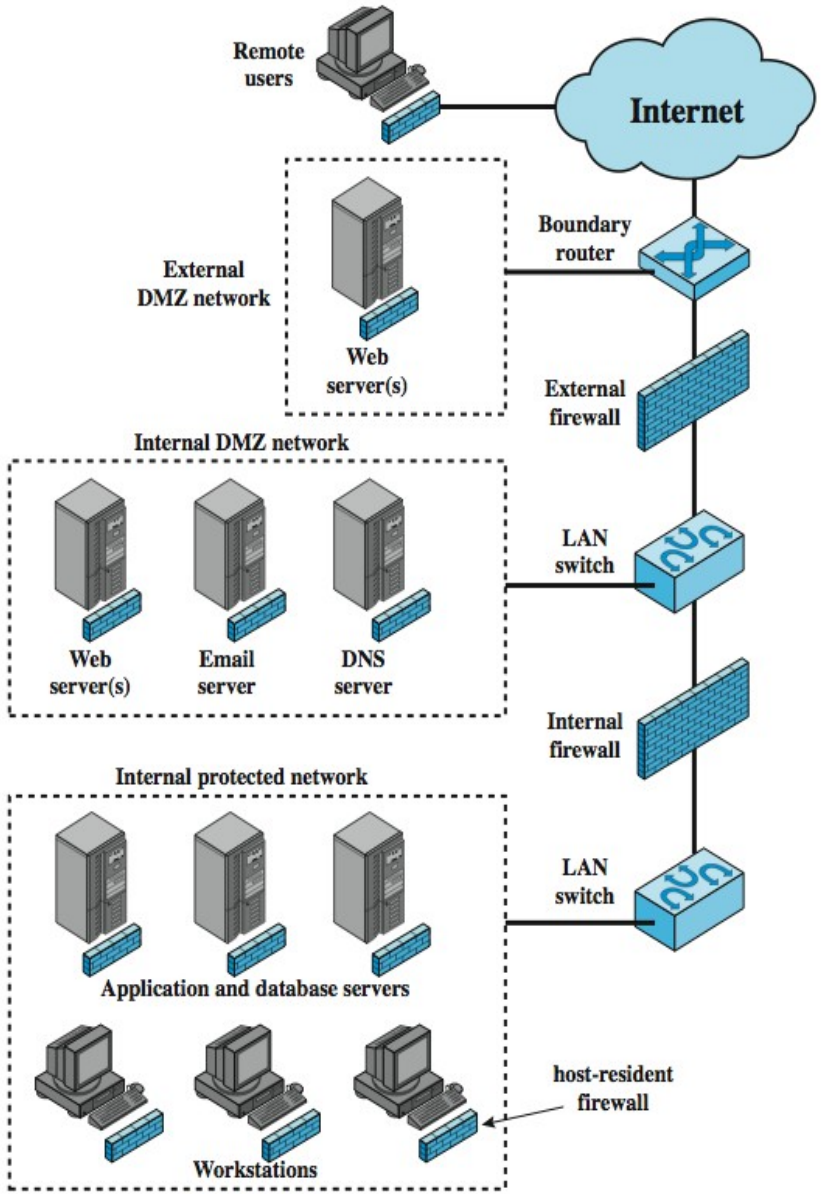
DMZ Networks



Virtual Private Networks (VPNs)



Distributed Firewalls



Intrusion Prevention Systems (IPS)

Host-based IPS (HIPS)

- Identifies attacks using both signature and anomaly detection techniques
  - signature: focus is on the specific content of application payloads in packets, looking for patterns that have been identified as malicious
  - anomaly: IPS is looking for behavior patterns that indicate malware
- Can be tailored to the specific platform
- Can also use a sandbox approach to monitor behavior

Network-based IPS (NIPS)

- Inline NIDS with the authority to discard packets and tear down TCP connections
- Uses signature and anomaly detection
- May provide flow data protection
  - monitoring full application flow content
- Can identify malicious packets using:
  - pattern matching
  - stateful matching
  - protocol anomaly
  - traffic anomaly
  - statistical anomaly

Firewalls vs. IDS/IPS

Firewall

- Tries to **prevent** „bad“ traffic
- Problem is classifying good vs. bad traffic in advance based on **static rules**
- Default policy is DROP-ALL with explicit accepts
- BUT: many protocols require so many different connections that firewall rule sets will often err on the accept side
- Therefore, even with stateful firewalls, new threats are hard to cover

Intrusion Detection/Prevention System (IDS/IPS)

- Idea is to **detect** „bad“ traffic and then act on it (log for IDS, block for IPS)
- Classification of good vs. bad traffic based on **static and heuristic matches**
- Advantage over firewalls: IDS/IPS can monitor more than one packet/session and then classify using more information about a connection
- Disadvantage: action (log/block) is often delayed, quick attacks within a few packets therefore not covered well

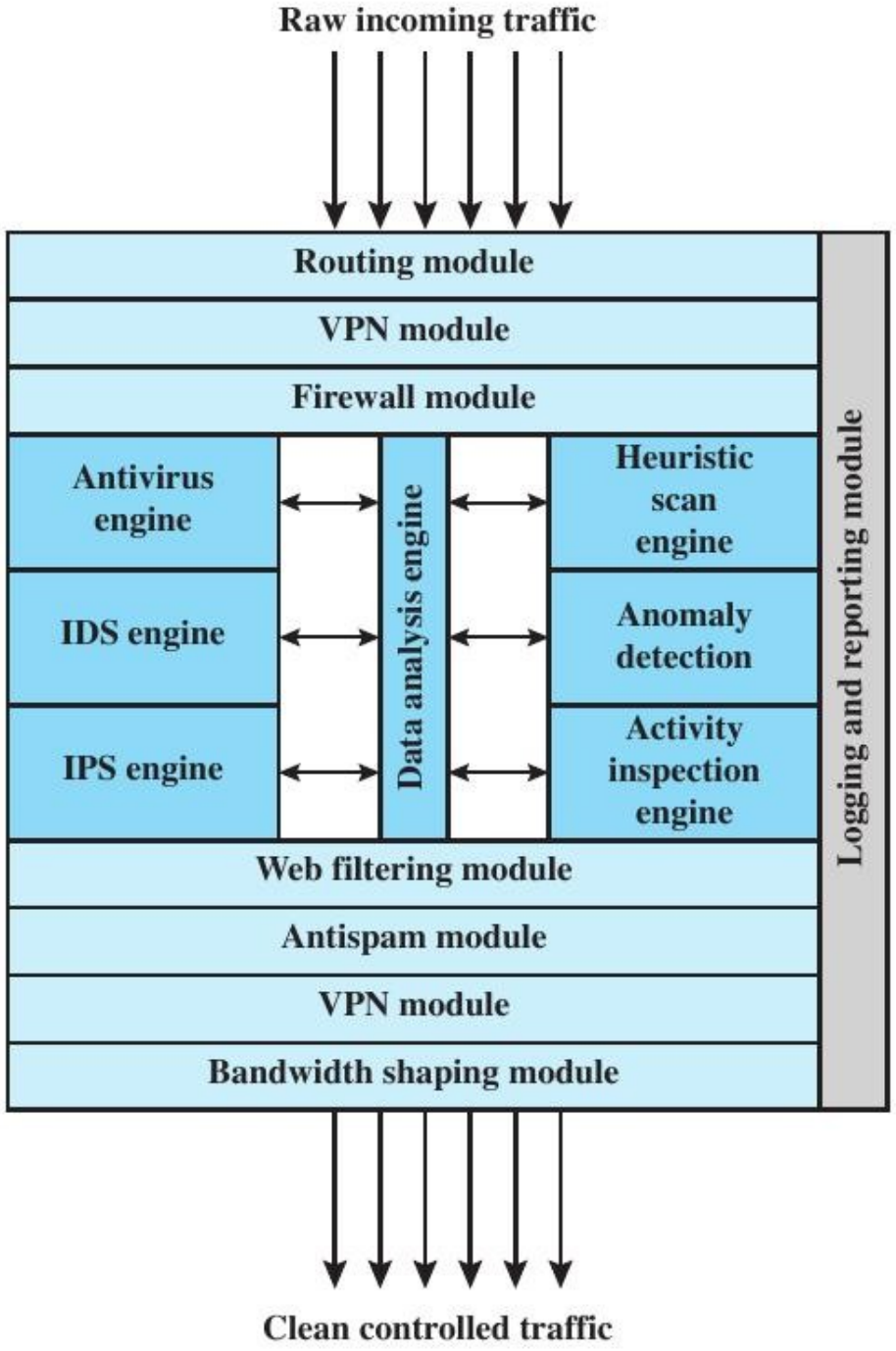
Firewalls vs. IDS/IPS

In practice, use both

- Firewalls for only allowing access to explicitly exported services and blocking everything else (rule set will still allow „bad“ traffic to pass in practice due to complexity issues)
- IDS for monitoring and reporting, especially concerning new attacks and uncommon network patterns
- IPS for protecting against dynamic attacks, e.g. denial-of-service (DoS)
- Note: IDS/IPS need signature updates like anti-virus software → typically requires maintenance contract with regular cost
- Note 2: IDS/IPS need to be distributed throughout the whole network, a single „choke point“ is not sufficient to reliably detect internal attacks

Unified Threat Management (UTM)

Combination of firewall, VPN gateway, IDS/IPS, virus scanning, etc.



Denial-of-Service (DoS)

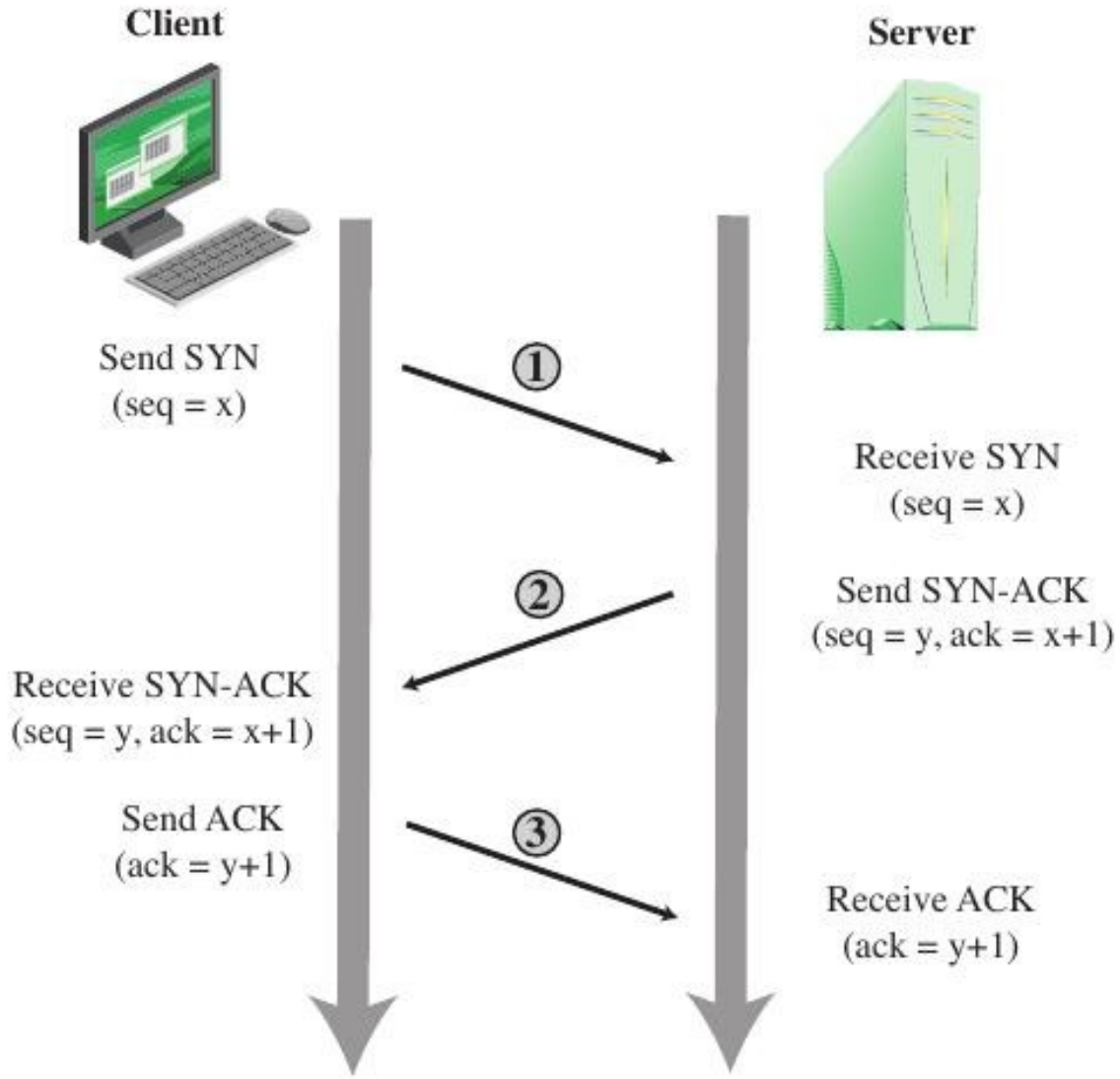
- DoS attacks try to make a service unavailable to others, are executed by unauthorized parties → direct violation of availability requirement
- NIST Computer Security Incident Handling Guide defines a DoS attack as

  "an action that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources such as central processing units (CPU), memory, bandwidth, and disk space."
- Can try to exhaust different resources
  - network bandwidth
  - system resources
  - application resources

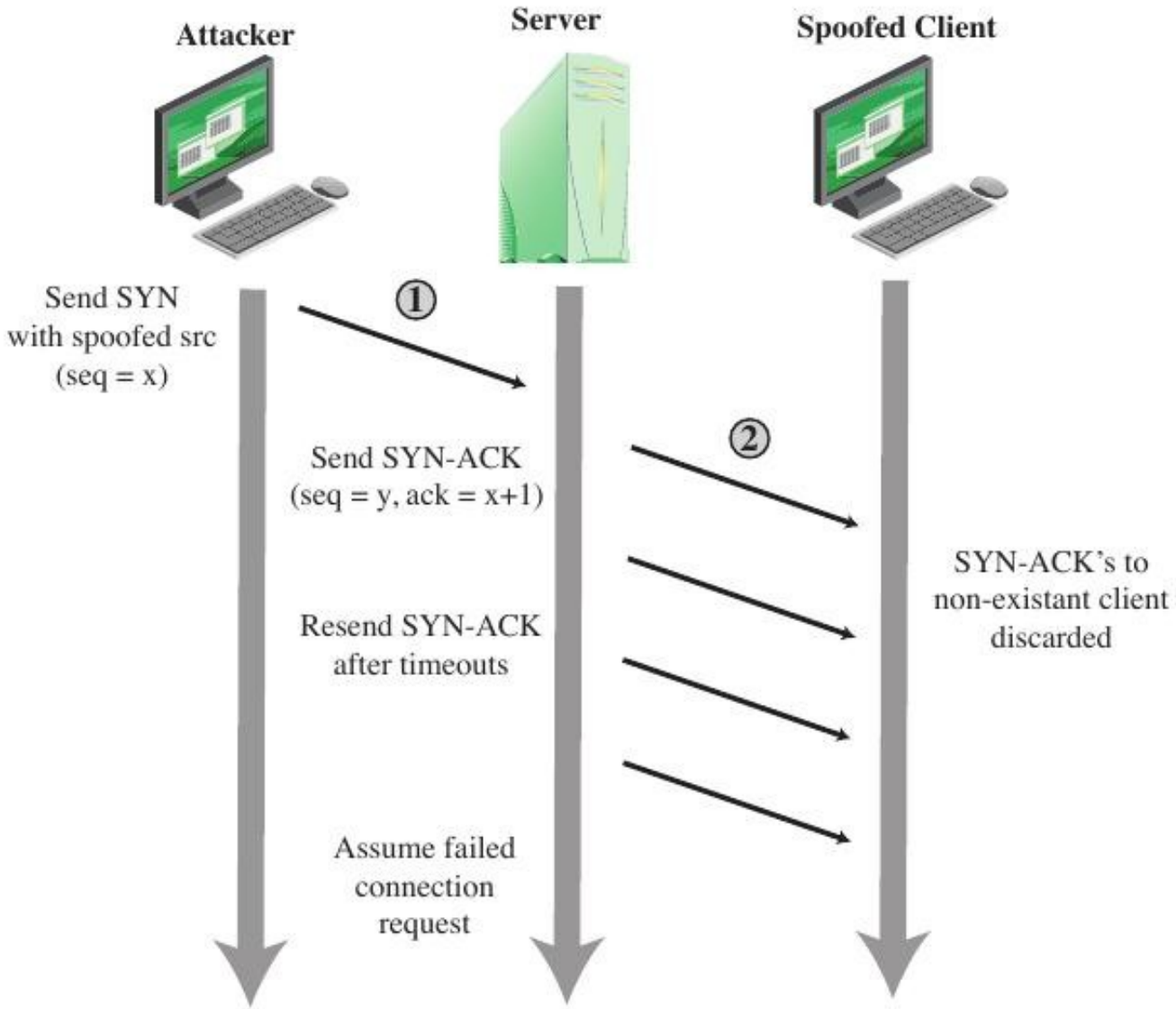
Examples for standard DoS techniques

- Simple ping flood (source has more network bandwidth than target)
- Source address spoofing (generates packets with source address faked to be that of the target and let other systems perform DoS with their replies)
- SYN spoofing
- Distributed DoS (DDoS)
- DDoS with reflectors (amplification)
- Application specific DoS (e.g. Slowloris for HTTP)
- Device specific DoS (e.g. overloading connection state tables causing dropped legitimate connections)

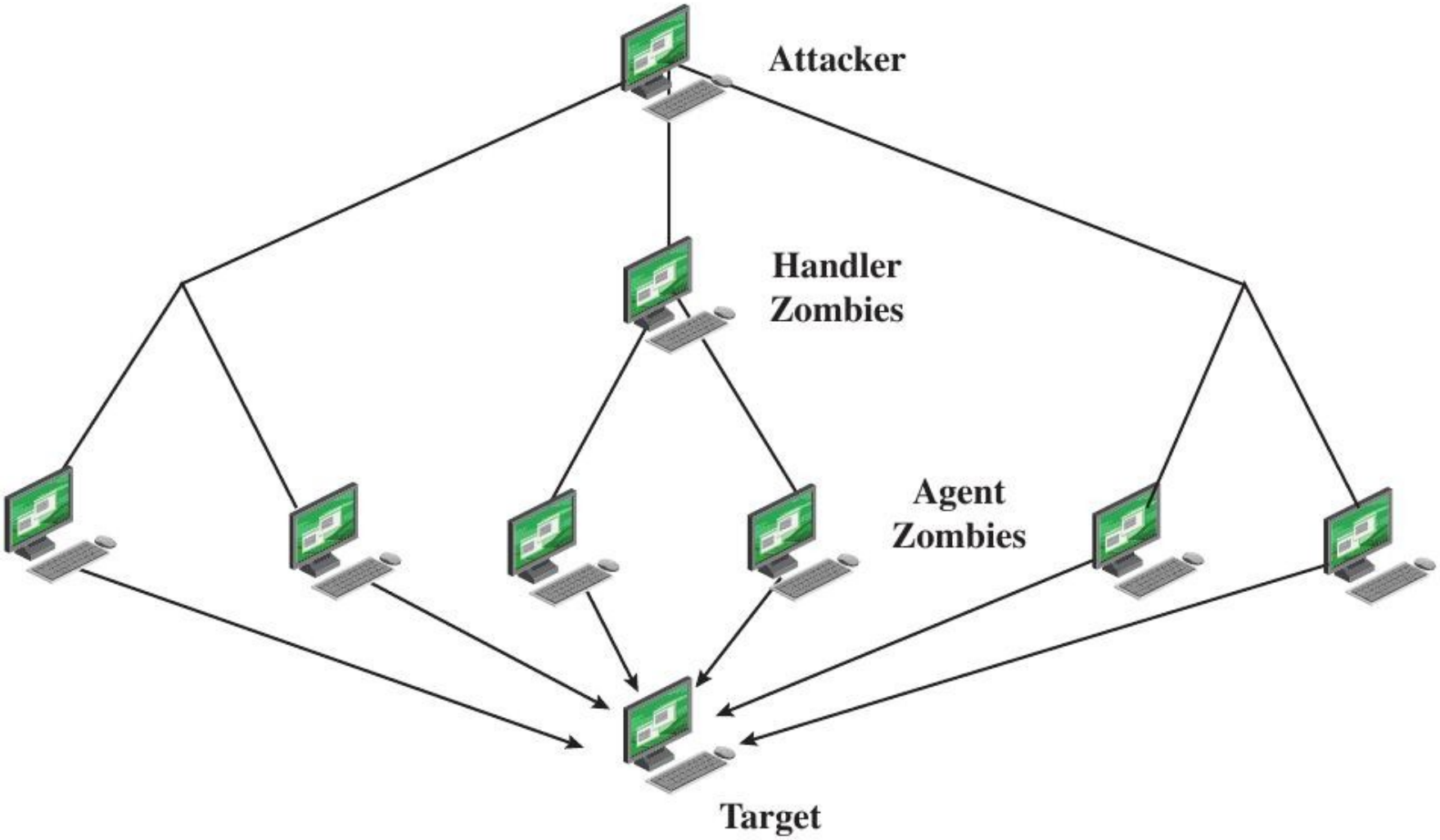
SYN spoofing: normal flow



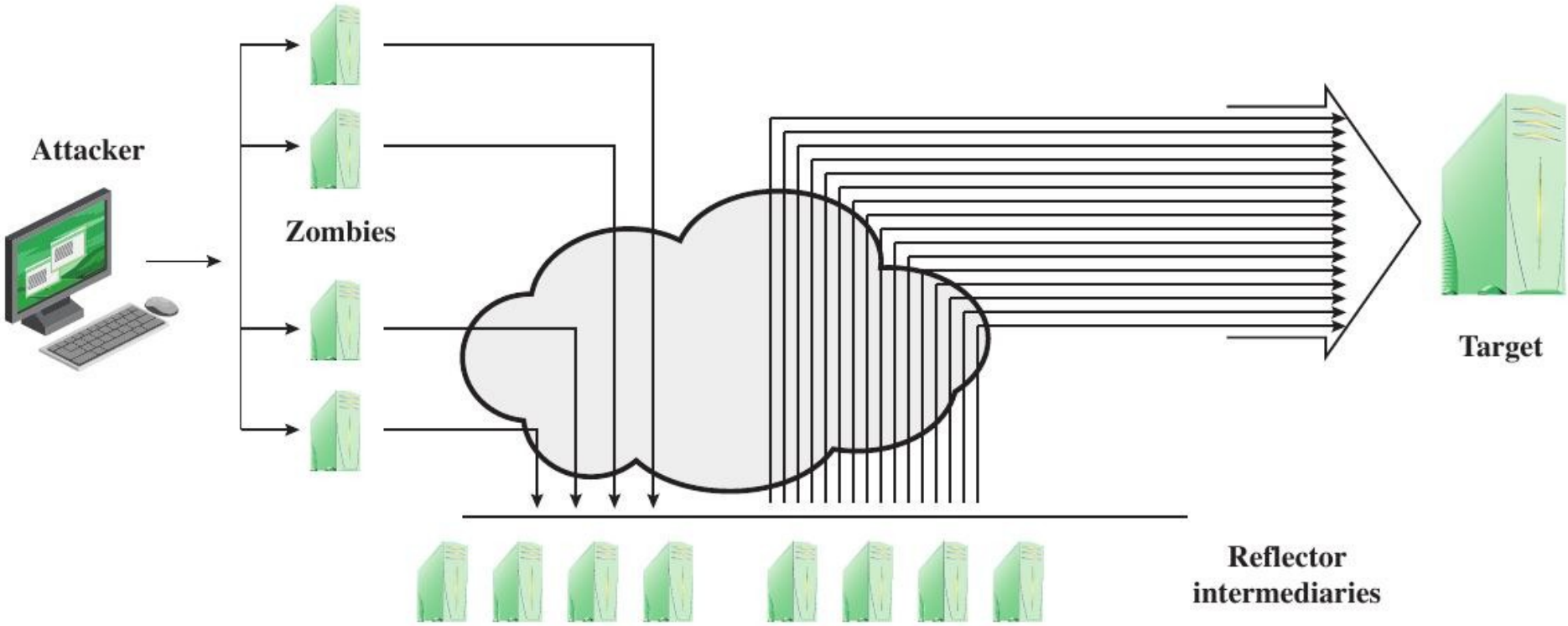
SYN spoofing: attack flow



DDoS attack architecture



DDoS attack with additional reflectors (amplification)



Countering DoS attacks

- Hard to counter DoS attacks on the receiving side
  - especially in DDoS case, there are always better network resources on the distributed Internet than the own connectivity
  - when upstream connection is overloaded, cannot even communicate to counter attack
- Therefore try to stop network DoS as close to the sender as possible
  - first step: own upstream Internet provider should block
  - second step: contact law enforcement (national and international) to block even closer to source → first need to locate source(s)
- Cloud-based: use CDN (Content Delivery Networks) – can identify & stop problems close to the source; only forward “good” traffic
- DoS on other resources (OS limits etc.) countered by same strategy
  - → block overload earlier (e.g. limit rate of incoming packets of this type on router/firewall before they hit the target system)