Chair for Network Architectures and Services
Department of Informatics
TU München – Prof. Carle

Network Security

Chapter 9

Attacks and Attack Detection

(Prevention,
Detection and Response)

Attacks and Attack Detection

- Have you ever been attacked (in the IT security sense)?
- What kind of attacks do you know?

Part I: Attack Prevention

- Part 0: Attacks
- Part I: Attack Prevention
- Part II: Attack Detection
- Part III: Response Mechanisms

Attacks by Impact

- Disruptive:
  - The goal is to fully deny the victim‘s service to its clients
- Degrading:
  - A portion of the victim’s resources (e.g. 30%) are occupied by the attackers.
  - Can remain undetected for a signification time period
  - Customers experience slow response times or now service during high load periods. → Customers go to an other Service Provider.
- Leakage of data
  - Confidential data, passwords, password files, keys, …
- Control
  - Being able to command a machine (may not interfere with normal operation)

System Vulnerabilities

- Origin of attacks:
  - Remote attacks: attacker breaks into a machine connected to same network, usually through flaw in software
  - Local attacks: malicious user gains additional privileges on a machine (usually administrative)
- Attacking techniques against a system:
  - *Buffer overflow:*
    - Intentional manipulation of program state by causing an area of memory to be written beyond its allocated limits
  - *Race condition:*
    - Exploiting non-atomic execution of a series of commands by inserting actions that were “unforeseen” by the programmer
  - *Exploiting trust in program input / environment:*
    - It is often possible to maliciously craft input / environment variables to have deleterious side effects
    - Programmers are often unaware of this

Scans and Port Scans

- Scans
  - A scan is an active attack to obtain information about a network and its systems. The attacker contacts machines and requests information in a systematic way and analyzes the result.
  - Port Scan: scan is to see which ports are open on a machine

- Can leak info about
  - Network Topology
  - Operating System
  - Applications and Application Versions
  - …

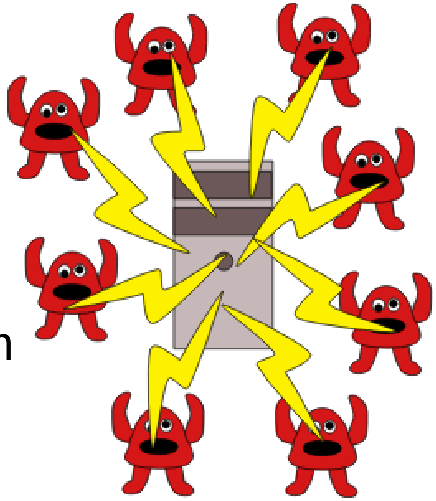
- Used to
  - Use information for subsequent attacks

Denial of Service attacks

- What is Denial of Service?
  - *Denial of Service (DoS) attacks aim at denying or degrading legitimate users' access to a service or network resource, or at bringing down the servers offering such services*

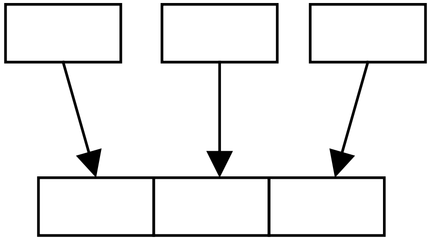
Denial of Service Attacking Techniques

- *Resource destruction* (disabling services):
  - Hacking into systems
  - Making use of implementation weaknesses as buffer overflow
  - Deviation from proper protocol execution
- *Resource depletion* by causing:
  - Storage of (useless) state information
  - High traffic load (requires high overall bandwidth from attacker)
  - Expensive computations ("expensive cryptography"!)
  - Resource reservations that are never used (e.g. bandwidth)
- Origin of malicious traffic:
  - Genuineness of source addresses: either genuine or forged
  - Number of sources:
    - single source, or
    - multiple sources *(Distributed DoS, DDoS)*

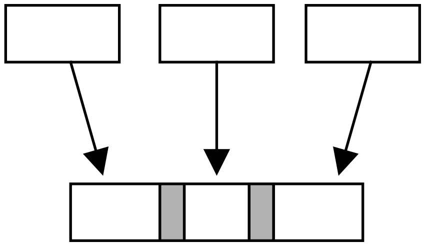


Examples: Resource Destruction (ancient)

- Ping-of-Death:
  - Maximum size of TCP/IP packet is 65536 bytes
  - Oversized packet may crash, freeze, reboot system
- Teardrop:
  - Fragmented packets are reassembled using the Offset field.
  - Overlapping Offset fields might cause system to crash.
- Take-Home Message:
  - Only a few packets can be sufficient to bring down a system.



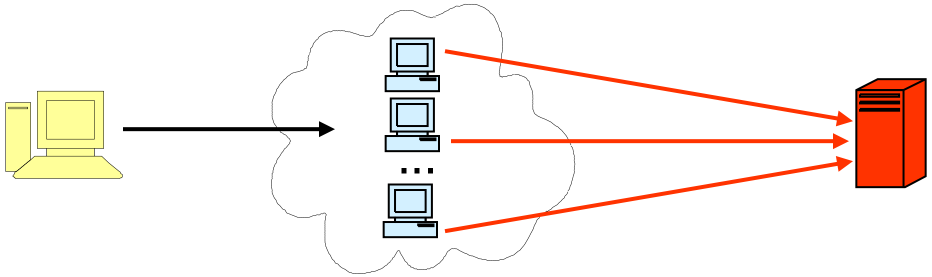
Normal Behavior



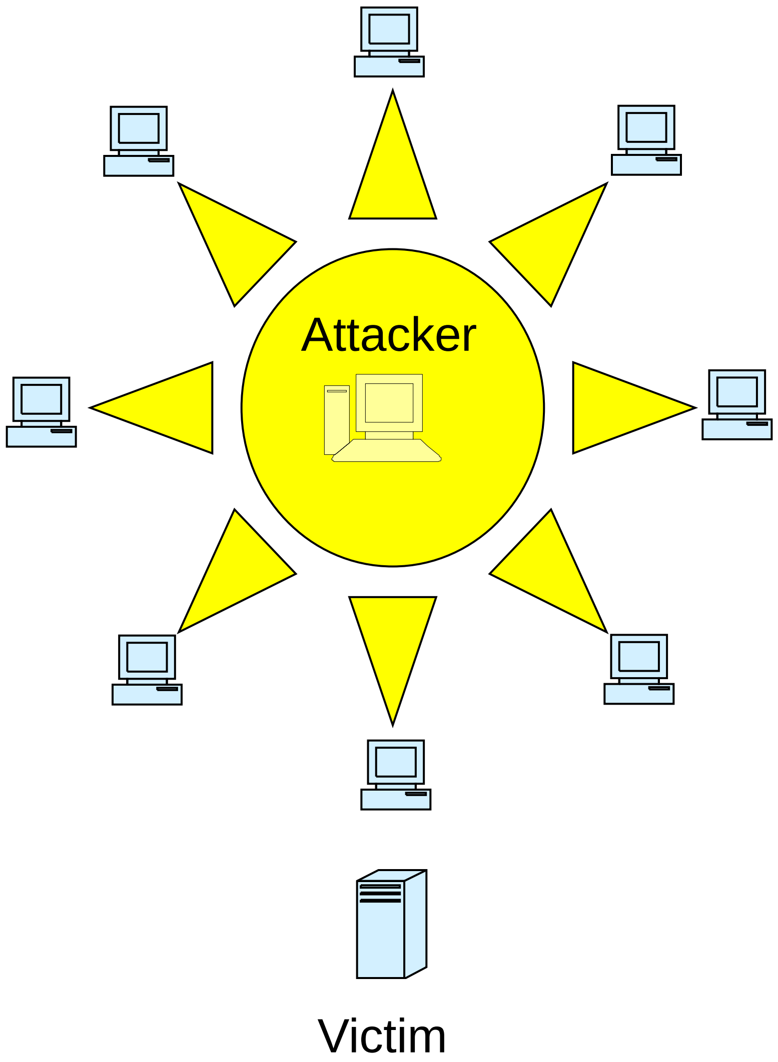
Teardrop Attack

Resource Depletion Example 1: Abusing ICMP

- Two main reasons make ICMP particular interesting for attackers:
  - It may be addressed to broadcast addresses
  - Routers respond to it
- The *Smurf* attack - ICMP echo request to broadcast:
  - An attacker sends an ICMP echo request to a broadcast address with the source addressed forged to refer to the victim
    - local broadcast: 255.255.255.255;
    - directed broadcast: (191.128.0.0/24) 191.128.0.255
  - Routers (often) allow ICMP echo requests to broadcast addresses
  - All devices in the addressed network respond to the packet
  - The victim is flooded with replies to the echo request
  - With this technique, the network being abused as an (unaware) attack amplifier is also called a *reflector network:*

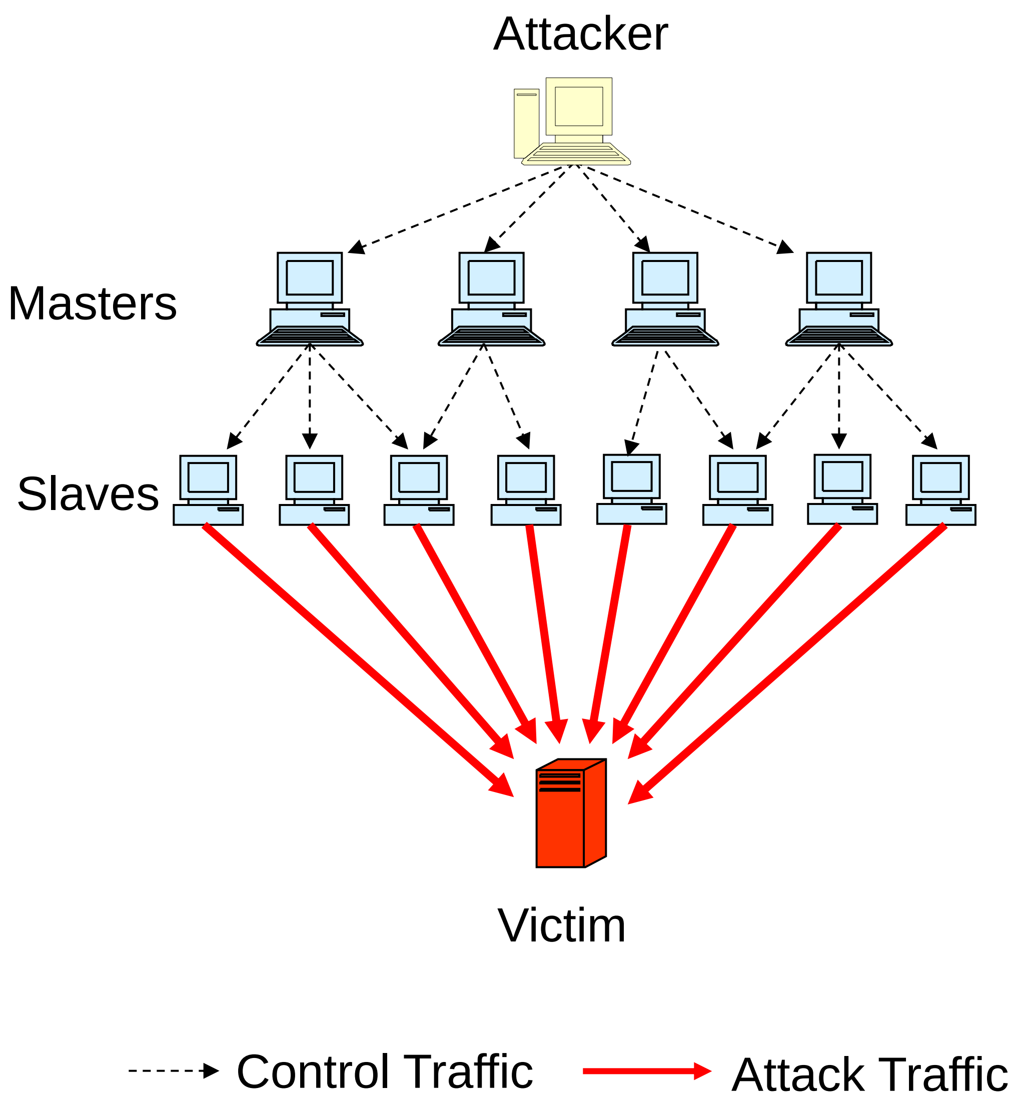


Resource Depletion with Distributed DoS (1)



- Category *Overwhelming the victim with traffic*
- Attacker intrudes multiple systems by exploiting known flaws
- Attacker installs DoS-software:
  - „Root Kits“ are used to hide the existence of this software
- DoS-software is used for:
  - Exchange of control commands
  - Launching an attack
  - Coordinating the attack

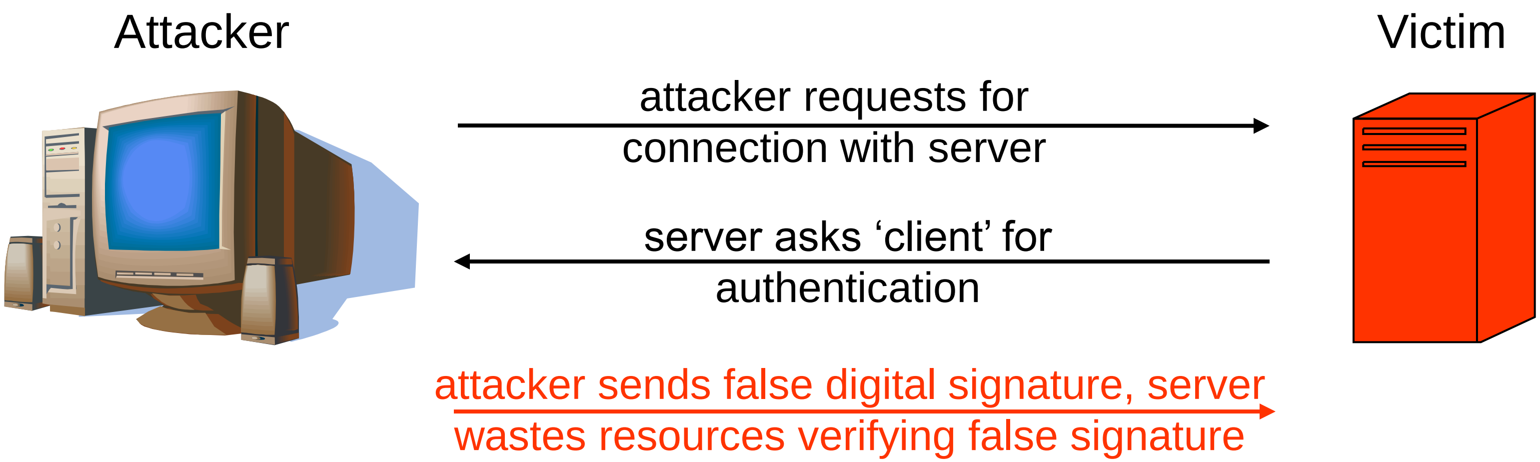
Resource Depletion with Distributed DoS (2)



- The attacker classifies the compromised systems in:
  - Master systems
  - Slave systems
- Master systems:
  - Receive command data from attacker
  - Control the slaves
- Slave systems:
  - Launch the proper attack against the victim
- During the attack there is no traffic from the attacker

Resource Depletion with CPU Exhaustion

- Category *CPU exhaustion by causing expensive computations:*
  - Here: attacking with bogus authentication attempts



  - The attacker usually either needs to receive or guess some values of the second message, that have to be included in the third message for the attack to be successful
  - Also, the attacker, must trick the victim *repeatedly* to perform the expensive computation in order to cause significant damage

➡ Be aware of DoS-Risks when introducing security functions into protocols!!!

Part I: Attack Prevention

- Part 0: Attacks
- Part I: Attack Prevention
- Part II: Attack Detection
- Part III: Response Mechanisms

Attack Prevention

- *Prevention:*
  - All measures taken in order to avert that an attacker succeeds in realizing a threat
  - Examples:
    - Cryptographic measures: encryption, computation of modification detection codes, running authentication protocols, etc.
    - Firewall techniques: packet filtering, service proxying, etc.
  - Preventive measures are by definition taken *before an attack takes place*

➡ Attention: it is generally impossible to prevent every potential attack!

Prevention: Defense Techniques Against DoS Attacks (1)

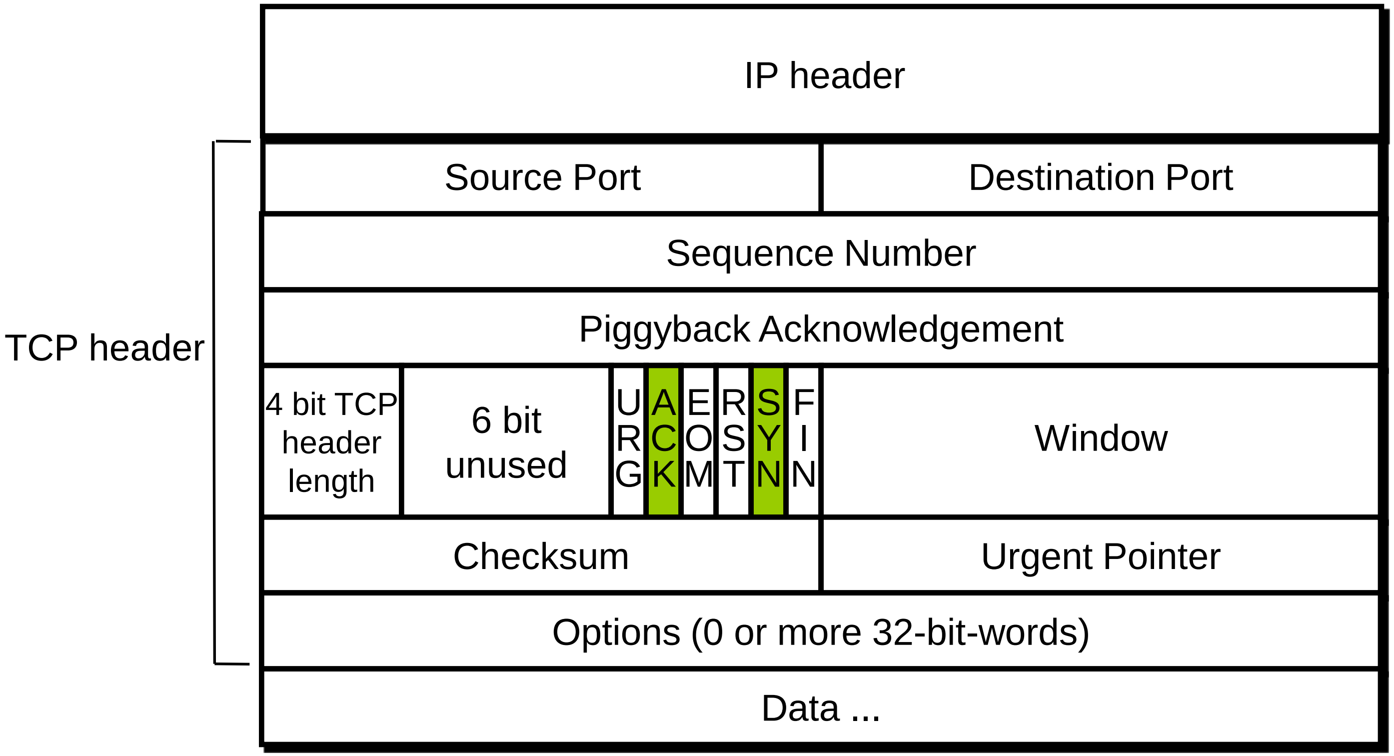
- Defenses against disabling services:
  - Hacking defenses:
    - Good system administration
    - Firewalls, logging & intrusion detection systems
  - Implementation weakness defenses:
    - Code reviews, stress testing, etc.
  - Protocol deviation defenses:
    - Fault tolerant protocol design
    - Error logging & intrusion detection systems
    - “DoS-aware protocol design”:
      - Be aware of possible DoS attacks when reassembling packets
      - Do not perform expensive operations, reserve memory, etc., before authentication

Prevention: Defense Techniques Against DoS Attacks (2)

- Defenses against resource depletion:
  - Generally:
    - Rate Control (ensures availability of other functions on same system) i.e. a potential reason to implement QoS mechanisms
    - Accounting & Billing (“if it is for free, why not use it excessively?”)
    - Identification and punishment of attackers
  - Authentication of clients plays an important role for the above measures
  - Memory exhaustion: stateless protocol operation
- Concerning origin of malicious traffic:
  - Defenses against single source attacks:
    - Disabling of address ranges (helps if addresses are valid)
  - Defenses against forged source addresses:
    - **Ingress Filtering at ISPs** (incoming packets from outside of ISP with IP source address from ISP blocked)
    - **Egress Filtering** (block outgoing packets with source address from other network)
    - “Verify” source of traffic (e.g. with exchange of “cookies”)
  - Widely distributed DoS: ???

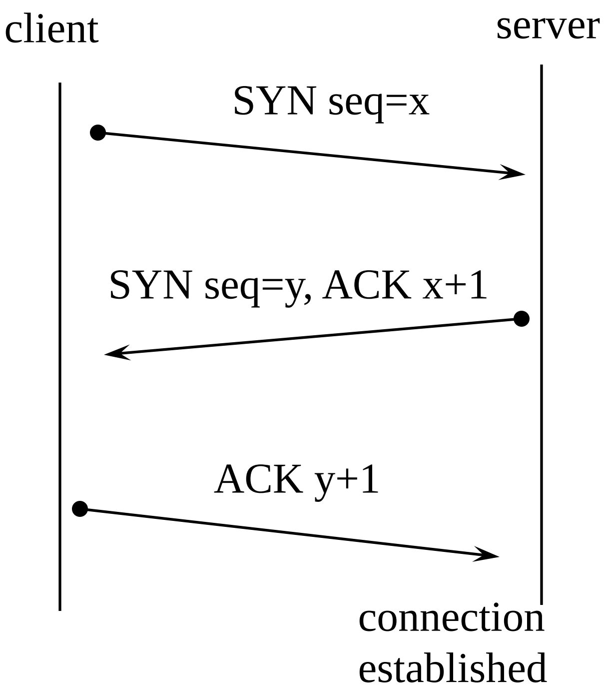
Example: TCP SYN Flood Attack (1)

- The TCP protocol Header:

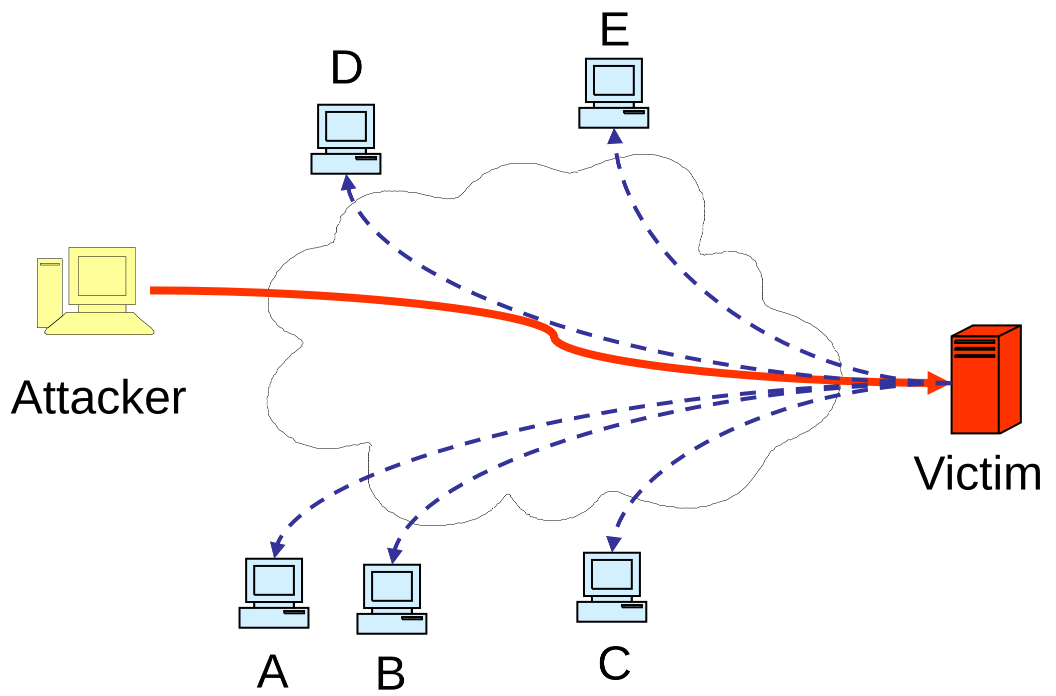


Example: TCP SYN Flood Attack (2)

- TCP 3-Way Handshake:
  - The client sends a 'TCP SYN' message
    - seq number = *x (*chosen by the client)
    - ACK flag = 0
    - SYN flag = 1
  - The server sends a 'TCP SYN ACK'
    - seq number = *y* (chosen by the server)
    - ack number = *x + 1*
    - ACK flag = 1
    - SYN flag = 1
  - The client sends a 'CONNECT ACK'
    - seq number = *x + 1*
    - ack number = *y + 1*
    - ACK flag = 1
    - SYN flag = 0
  - The handshake ensures that both sides are ready to transmit data.



Example: TCP SYN Flood Attack (3)



Connection Table
A
B
C
D
E
...

TCP SYN packets with forged source addresses (“SYN Flood”)

TCP SYN ACK packet to assumed initiator (“Backscatter”)

No response comes back. ⇒ Too many half-opened connections.

⇒ The backlog queue (connection table) fills up.

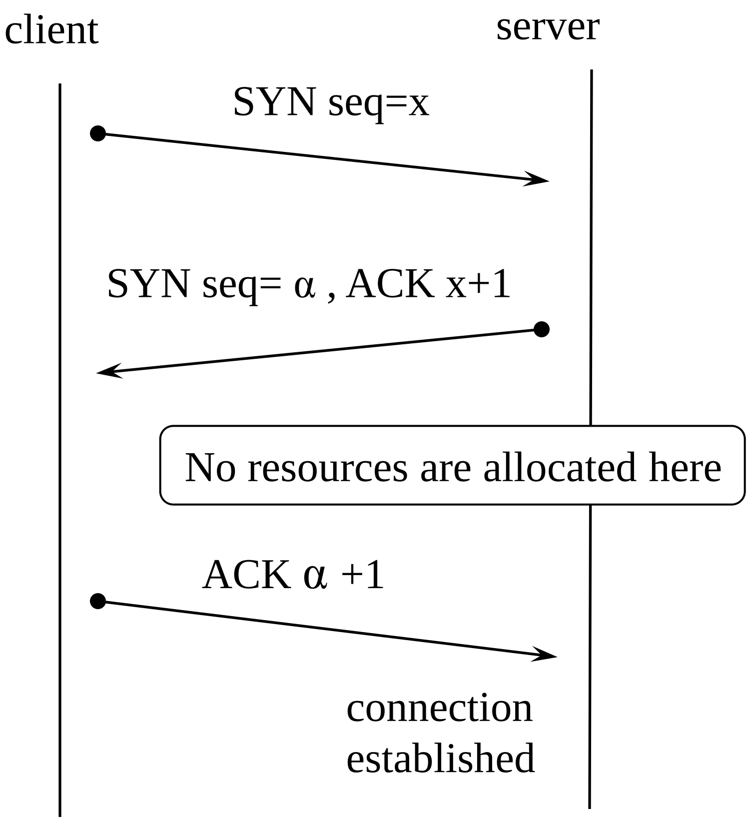
⇒ Legitimate users can not establish a TCP connection with the server.

Example: TCP SYN Flood Protection

- Anti-spoofing features
  - E.g. Ingress / Egress filtering
- Load Balancing and replication of resources:
  - The attack will pass unnoticed.
  - With a sufficient number of attackers the server can still be saturated.
- TCP stack tweaking
  - Increase backlog size
    - limited by the kernel memory of the server (each entry ~600 Bytes)
  - Decrease waiting time for the third packet of the TCP handshake
    - helps but has drawback that slower clients cannot connect
- SYN cookies (see subsequently)

Example: SYN Flood Protection with TCP SYN cookies (1)

- SYN cookies are a particular choice of the initial *seq number* by the server.
- The server generates **the initial sequence number α** such as:
  - $\alpha = h(K, S_{SYN})$
  - **K**: a secret key, changed over time
  - S_{SYN}: source addr of the SYN packet
  - **h** is a cryptographic hash function.
- At arrival of the ACK message, the server calculates α again.
- Then, it verifies if the *ack number* is correct.
- If yes, it assumes that the client has sent a SYN message recently and it is considered as normal behavior.



Example: SYN Flood Protection with TCP SYN cookies (2)

- Advantages:
  - The server does not need to allocate resources after the first SYN packet.
  - The client does not need to be aware that the server is using SYN cookies.

    ⇒ **SYN cookies don't requires changes in the specification of the TCP protocol**.
- Disadvantages:
  - Calculating α is CPU power consuming.

    ⇒ Moved the vulnerability from memory overload to CPU overload.
  - TCP options can not be negotiated (e.g. large window option)

    ⇒ Use only when an attack is assumed.
  - Is vulnerable to cryptoanalysis: even if h is a secure function the sequence numbers generated by the server may be predicted after receiving/ hijacking a sufficient number of cookies.

    ⇒ **The secret code need to be changed regularly, e.g. by including a timestamp.**
- N.B. SYN cookies are integrated in the Linux Kernel with MD5 as hash function.
  - top 5 bits: t mod 32, where t is a 32-bit time counter that increases every 64 seconds;
  - next 3 bits: an encoding of an MSS selected by the server in response to the client's MSS;
  - bottom 24 bits: a server-selected secret function of the client IP address and port number, the server IP address and port number, and t.

- Introduction
- Host IDS vs. Network IDS
- Knowledge-based Detection
- Anomaly Detection

Introduction

- Prevention is not sufficient in practice

- What can be attained with intrusion detection?
  - Detection of attacks and attackers
  - Detection of system misuse (includes misuse by legitimate users)
  - Limitation of damage (if response mechanisms exist)
  - Gain of experience in order to improve preventive measures
  - Deterrence of potential attackers

Introduction (2)

- *Intrusion*
  - Definition 1
    - "An Intrusion is unauthorized access to and/or activity in an information system."
  - Definition 2 (more general)
    - "…Any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource." [HLM91]
- As seen in Definition 2, the term "Intrusion" is often used in the literature to characterize any kind of attacks.
- *Intrusion Detection*
  - All measures taken to recognize an attack *while or after it occurred*
  - Examples:
    - Recording and analysis of audit trails
    - On-the-fly traffic monitoring and intrusion detection.

Attack Detection: Classification

- Classification by the scope of the detection:
  - Host-based Intrusion Detection Systems (HIDS)
  - Network- based Intrusion Detection Systems (NIDS)

- Classification by detection strategy:
  - Knowledge-based detection
  - Anomaly detection
  - Hybrid attack detection

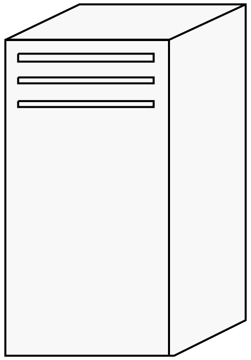
Part II: Attack Detection

- Introduction
- Host IDS vs. Network IDS
- Knowledge-based Detection
- Anomaly Detection

Host Intrusion Detection Systems (HIDS)

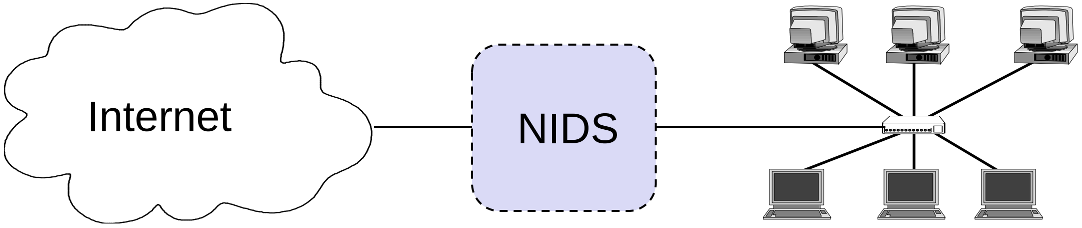
- Use information available on a system, e.g. OS-Logs, application-logs, timestamps
- Can easily detect attacks by insiders, as modification of files, illegal access to files, installation of Trojans or root kits
- Drawbacks:
  - Has to be installed on every system.
  - The attack packets can not be detected before they reach the victim

    ⇒ Host-based IDS are helpless against bandwidth saturation attacks.



Network Intrusion Detection Systems (NIDS)

- Use information provided by the network, mainly packets sniffed from the network layer.
- Often used at the edges of the (sub-)networks (ingress/egress points)
- Can detect known attack signatures, port scans, invalid packets, attacks on application layer, DDoS, spoofing attacks
- Uses signature detection (stateful), protocol decoding, statistical anomaly analysis, heuristical analysis



Part II: Attack Detection

- Introduction
- Host IDS vs. Network IDS
- Knowledge-based Detection
- Anomaly Detection

Knowledge-based Attack Detection (1)

- Idea:
  - Store signatures of attacks in a database
  - Each communication is monitored and compared with database entries to discover occurrence of attacks.

Hand detected
➔ human

- The database is occasionally updated with new signatures.

- Advantage:
  - Known attacks can be reliably detected. Hardly “false positives” (see below for the definition of “false positives”)
  - Drawbacks:
    - Only known attacks can be detected.
    - Slight variations of known attacks are not detected.

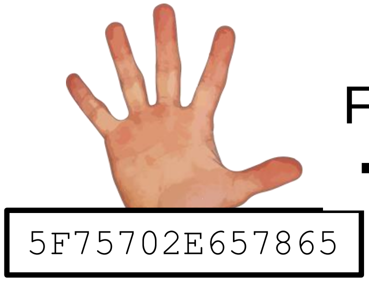
- Different appellations for “Knowledge-based” attack detection in the literature
  - “pattern-based” “signature-based” “misuse-based”.

Knowledge-based Attack Detection (2)

- Patterns can be specified at each protocol level
  - Network protocol (e.g. IP, ICMP)
  - Transport protocol (e.g. TCP, UDP)
  - Application protocol (e.g. HTTP, SMTP)

- Example of a rule in the IDS Snort (http://www.snort.org/)

```
alert tcp $HOME_NET any -> any 9996 \
 (msg:"Sasser ftp script to transfer up.exe"; \
content:"|5F75702E657865|"; depth:250; flags:A+; classtype: misc-
activity; \ sid:1000000; rev:3)
```



Fragment of Sasser located
➔ Sasser

Part II: Attack Detection

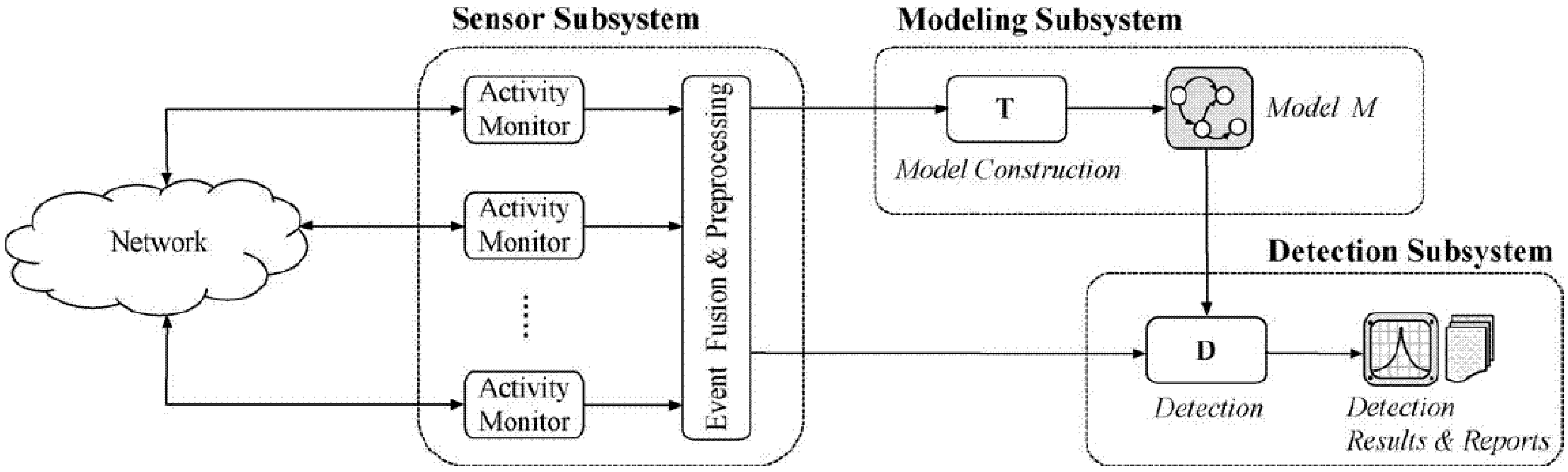
- Introduction
- Host IDS vs. Network IDS
- Knowledge-based Detection
- Anomaly Detection

Anomaly Detection (1)

- Anomaly detection systems include a model of “normal system behavior” such as:
  - normal traffic dynamics
  - expected system performance
- The current state of the network is compared with the models to detect anomalies.
- If the current state differs from the normal behavior by a threshold then an alarm is raised.
- Anomalies can be detected in
  - Traffic behavior
  - Protocol behavior
  - Application behavior

Anomaly Detection (2)

- A formal definition: [Tapidor04]
  - An anomaly detection system is a pair $\delta = (M, D)$, where:
    - *M* is the model of normal behavior.
    - *D* is similarity measure that allows obtaining, giving an activity record, the degree of deviation (or likeness) that such activities have with regard to the model *M*.



Source: [Tapiador04]

Simple Anomaly Detection

- Performance Metrics of your system
  - E.g. number of requests

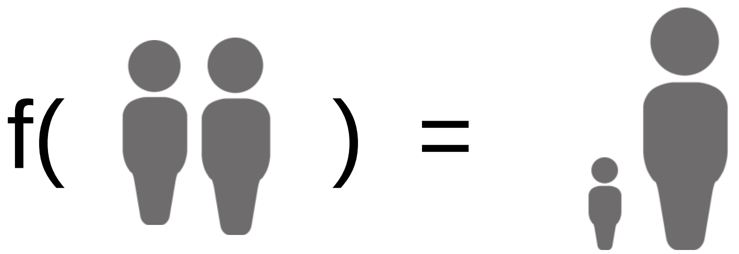
- Define a normal operational interval for the metric.
- Anomaly if metric outside of interval („fixed threshold“).
  - E.g. number of requests > 200 requests per second

- Cons:
  - Legitimate change of system over time, e.g. usage increases over the years (→ success is no attack)
  - No inclusion of periodic changes (e.g. daily and weekly changes in use) and trend changes (usage increases 8 % in year) as above

Other options

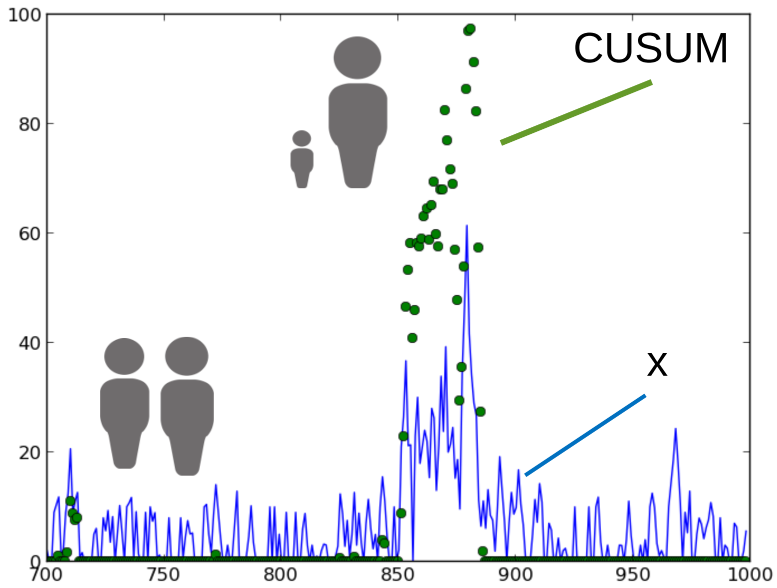
- Time series (of performance metrics) → Change detection in time series
  - The assumption is that an attack changes the system comparably rapidly.
  - A resource depletion attacker will not slowly increase bandwidth for a year until succeeding.

- Change detection
  - Ignore single outliers
  - Respond quickly once multiple values indicate change
  - Basis usually a function that amplifies the change.

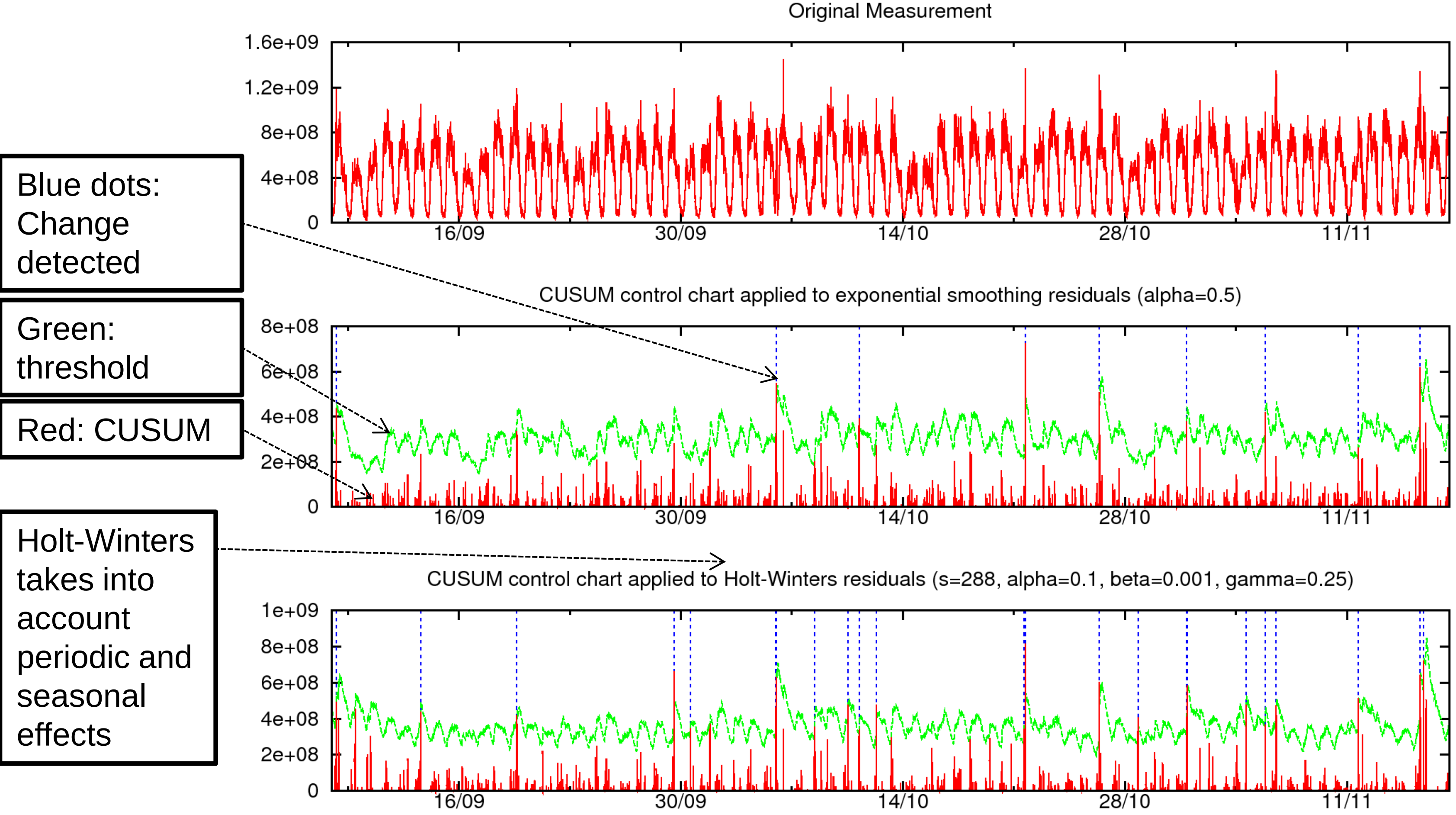


CUSUM

- CUSUM (cumulative sum) is a change detection function
  - S(0) = 0
    S(t) = max(0, S(t-1) + x(t) – m – k s)
    with x input stream and m a mean and s a standard deviation and k a factor.
  - The consequence is that
    - S = 0 whenever average or small values
    - S small whenever single or few large values occur
    - S large whenever many large values occur at some moment in time
  - Detection if S(t) > threshold h
    - h can be adaptable to a mean + k2 std dev where k2>k



CUSUM Example (Bytes in an ISP network)



- From Gerhard Münz. *Traffic Anomaly Detection and Cause Identification Using Flow-Level Measurements*. PhD thesis, Technische Universität München, June 2010.

Anomaly Detection (2)

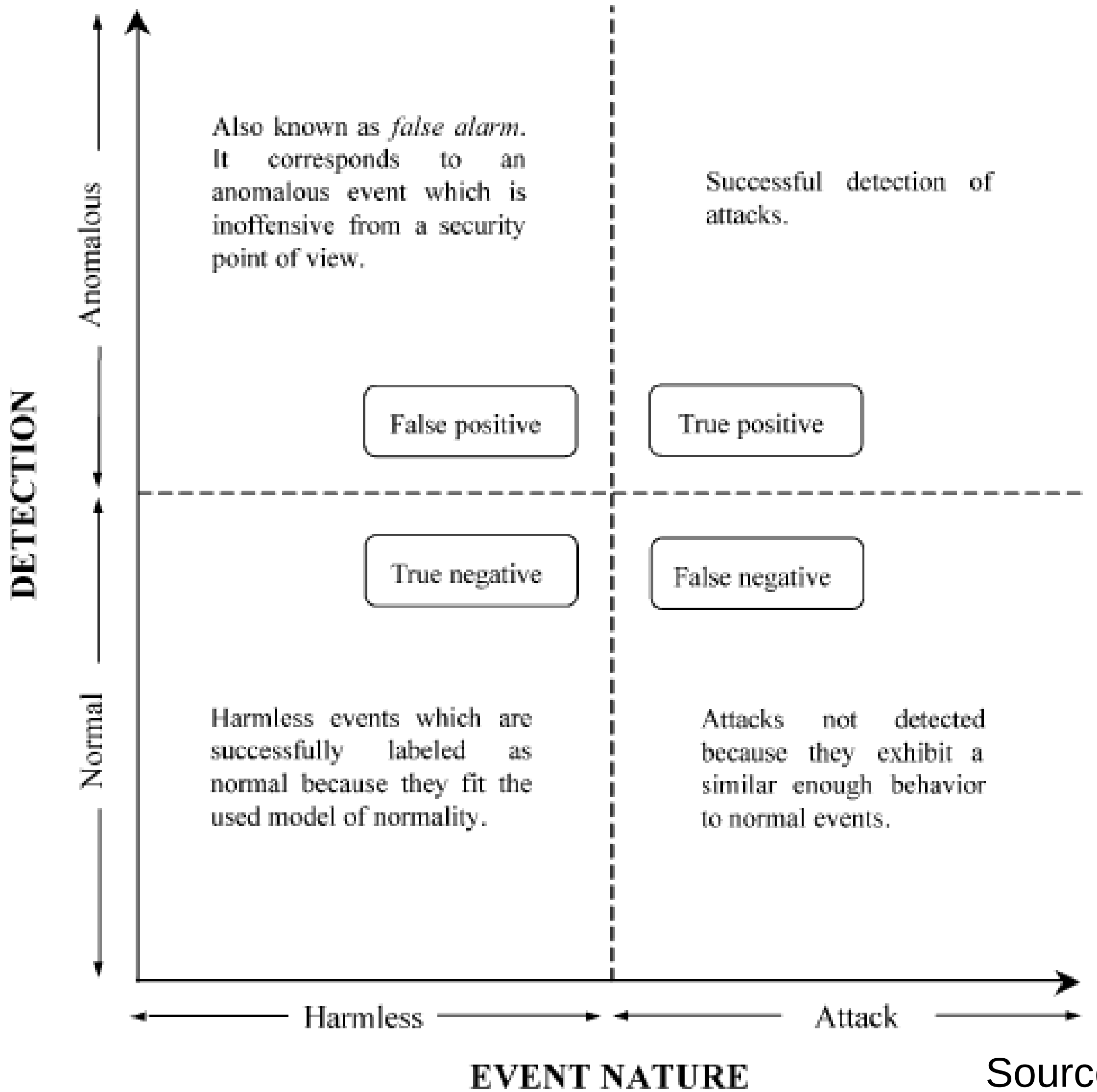
- Pros
  - Might recognize some unknown attacks as well
- Cons
  - False-positive (see definition below) rate might be high

- Definitions:
  - A *false positive* means the attack detection system raises an alarm while the behavior is legitimate.
  - A *false negative* means that an attack happens while it is classified by the attack detection system as normal behavior.

⇒ If the threshold for raising an alarm is set too low, the false positive rate is too high.

If the threshold is set too high, the attack detection system is insensitive.

Detection Quality



Source: [Tapiador04]

Anomaly Detection (3)

- Challenges
  - Modeling Internet traffic is not easy
  - Data collection issues
    - Collection is expensive, collecting the right information is important
  - Anomalies can have different reasons
- *Network Operation Anomalies*
  - caused, e.g. by a link failure or a configuration change
  - In modern data centers, migration of a virtual machine
- *Flash Crowd Anomalies*
  - rapid rise in traffic flows due to a sudden interest in a specific services (for instance, a new software path in a repository server or a highly interesting content in a Web site)
- *Network Abuse Anomalies*
  - such as DoS flood attacks and port scans

Attack Prevention, Detection and Response

- Part 0: Attacks
- Part I: Attack Prevention
- Part II: Attack Detection
- Part III: Response Mechanisms

Response Strategies

- Packet Filtering
- Kill Connections
- Rate Limiting
  - Congestion control
  - Pushback
- Tracking
  - Traceback techniques
  - Re-configuration of the monitoring environment
- Redirection

Response Strategies: Packet Filtering

- Attack packets are filtered out and dropped.
- Challenges
  - How to distinguish between legitimate packets (the „good“ packets) and illegitimate packets (the „bad“ packets).
  - Attacker’s packet might have spoofed source addresses

- Filterable attacks
  - If the flood packets are not critical for the service offered by the victim, they can be filtered.
  - Example: UDP flood or ICMP request flood on a web server.

- Non-filterable attacks
  - The flood packets request legitimate services from the victim.
  - Examples include
    - HTTP request flood targeting a Web server
    - CGI request flood
    - DNS request flood targeting a name server
  - Filtering all the packets would be an immediate DoS to both attackers and legitimate users.

Response Strategies: Kill Connection

- Kill Connection
  - TCP connections can be killed using RST packets that are sent to both connection end points
  - The RST packet requires correct sequence/ acknowledgement numbers. Otherwise it is ignored.
  - Limitation: this response is possible only for connection-oriented protocols

References

[HLM91] Heberlein, Levitt und Mukherjeeh. A method to detect intrusive activity in a networked environment. In Proceedings of the 14th National Computer Security Conference, 1991.

[Mirkovic2004] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, April 2004, pp. 39-53.

[Tapidor2004] J. M. Estevez-Tapiador, P. Garcia-Teodoro, and J. E. Diaz-Verdejo, "Anomaly detection methods in wired networks: a survey and taxonomy," Computer Communications, vol. 27, July 2004, pp. 1569-1584.