



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

# Network Security

## IN2101

Prof. Dr.-Ing. Georg Carle  
Dipl.-Inform. Ali Fessi

Institut für Informatik  
Technische Universität München  
<http://www.net.in.tum.de>



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

# Network Security

## Chapter 13

Attack prevention,  
detection and response



# Part I: Attack Prevention

- ❑ 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 (if the world was an ideal one...)
    - “Verify” source of traffic (e.g. with exchange of “cookies”)
  - Widely distributed DoS: ???



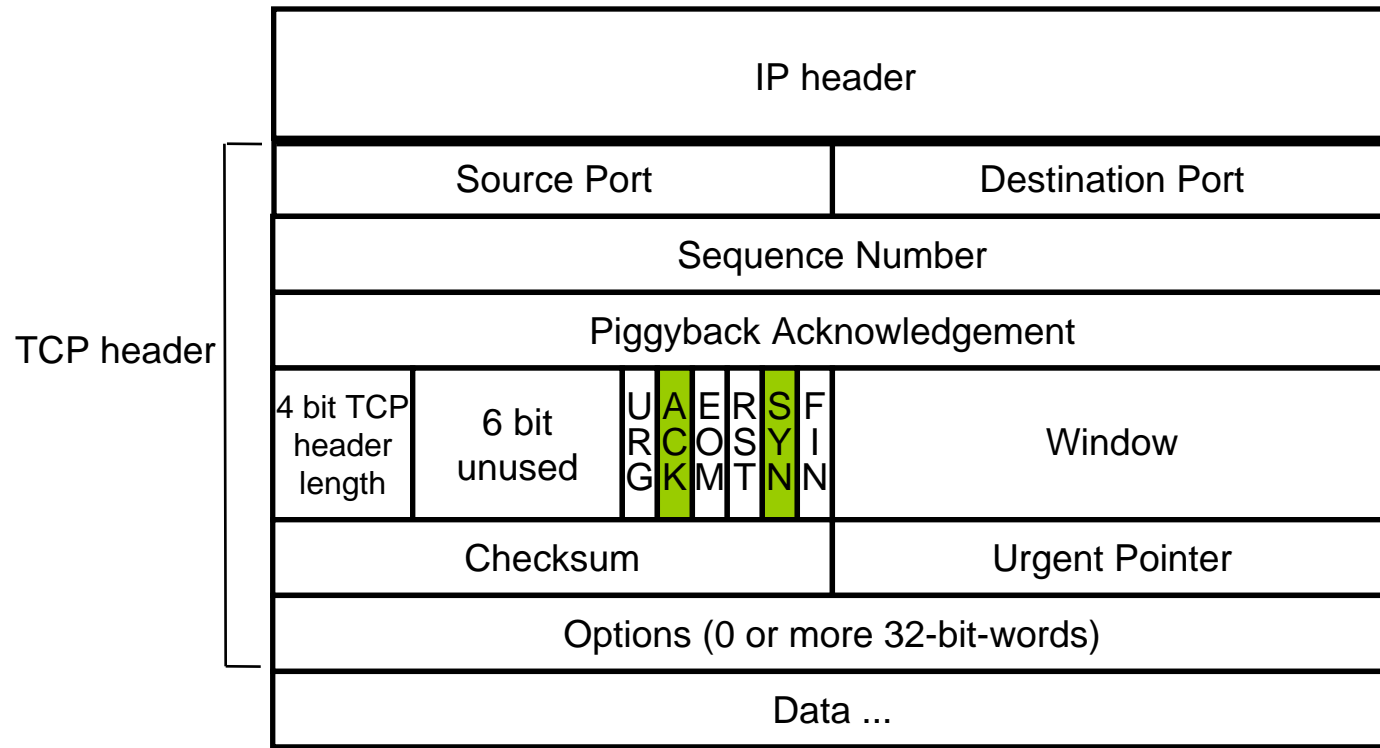
## Ingress/ Egress Filtering

- ❑ Goal:
  - Reduce the address space that can be used by the attacker by filtering the packets at the edge of the network
  
- ❑ Ingress filtering:
  - Incoming packets with a source address belonging to the network are blocked
  - Incoming packets from the public Internet with a private source address are blocked
  
- ❑ Egress filtering:
  - Outgoing packets that carry a source IP address that does not belong to the network are blocked



## Example: TCP SYN Flood Attack (1)

- The TCP protocol Header:

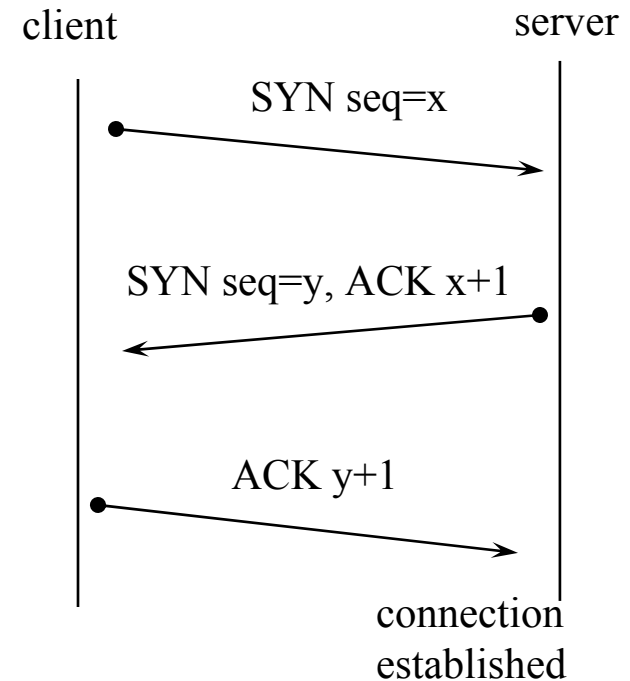






## Example: TCP SYN Flood Attack (2)

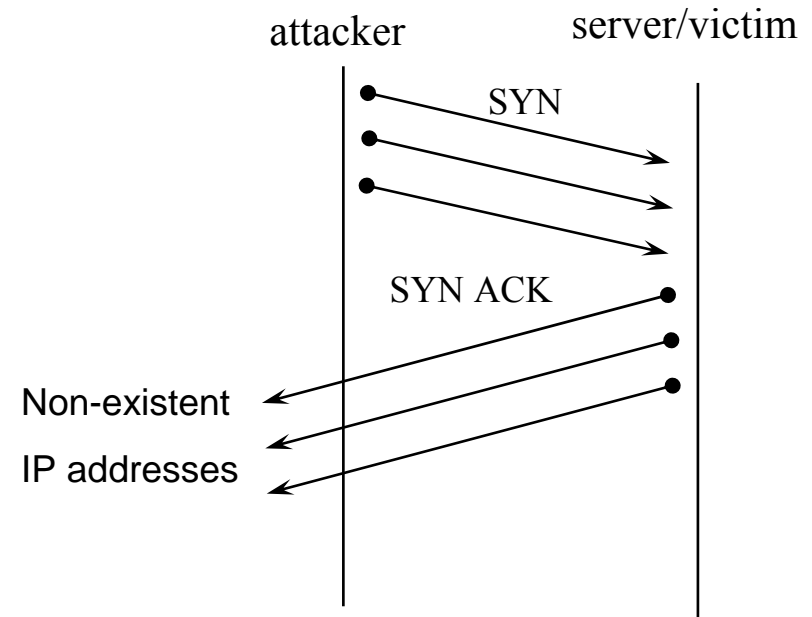
- TCP 3-Way Handshake:
  - The client sends a 'TCP SYN' message
    - seq number =  $x$  (chosen by the client)
    - ACK flag = 0
    - SYN flag = 1
  - The server sends a 'TCP SYN ACK'
    - seq number =  $y$  (chosen by the server)
    - ack number =  $x + 1$
    - ACK flag = 1
    - SYN flag = 1
  - 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)

- ❑ The attacker floods the victim with SYN packets with spoofed IP addresses.
- ❑ The victim answers with SYN/ACK packets and waits for a responding ACK packet.
- ❑ The server stores half-opened connections in a backlog queue.
- ❑ 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.
- ❑ Mostly, victims are faced with multiple attackers
- ⇒ Distributed Denial of Service (DDoS) attack.





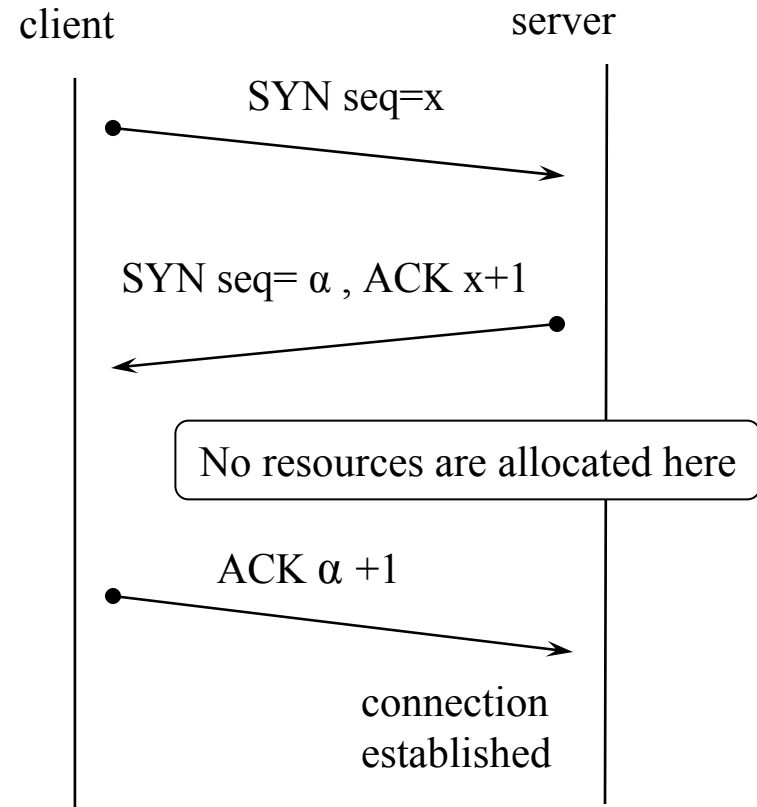
## Example: TCP SYN Flood Protection

- ❑ 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
- ❑ TCP proxies:
  - TCP connections are intercepted by the TCP proxy.
  - When the 3-way handshake is complete, the connection is forwarded to the server.
    - ⇒ TCP connections are slower.
    - ⇒ Use only when an attack is assumed.
  - The sever remains safe. However, in case of an attack, legitimate users still can not connect.
    - ⇒ Only a “fuse”. Does not solve the real problem.
- ❑ SYN cookies (see subsequently)
- ❑ Anti-spoofing features



## 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  $\alpha$  such as:
  - $\alpha = h(K, S_{SYN})$
  - $K$ : a secret key
  - $S_{SYN}$ : source addr of the SYN packet
  - $h$  is a cryptographic hash function.
- ❑ At arrival of the ACK message, the server calculates  $\alpha$  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  $\alpha$  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 cryptanalysis: 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:  $t \bmod 32$ , 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  $t$ .



# Attack Prevention, Detection and Response

- ❑ Part I: Attack Prevention
- ❑ Part II: Attack Detection
- ❑ Part III: Response Mechanisms



## Part II: Attack Detection

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



# Introduction

- ❑ Prevention is not sufficient in practice:
  - Because it is too expensive to prevent all potential attack techniques
  - Because legitimate users get annoyed by too many preventive measures and may even start to circumvent them (introducing new vulnerabilities)
  - Because preventive measures may fail:
    - Incomplete or erroneous specification / implementation / configuration
    - Inadequate deployment by users (just think of passwords...)
- ❑ 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)

- ❑ Store the signatures of attacks in a database
- ❑ Each communication is monitored and compared with database entries to discover occurrence of attacks.
- ❑ The database is occasionally updated with new signatures.
- ❑ Advantage:
  - Known attacks can be reliably detected. No “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)
```





## Part II: Attack Detection

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



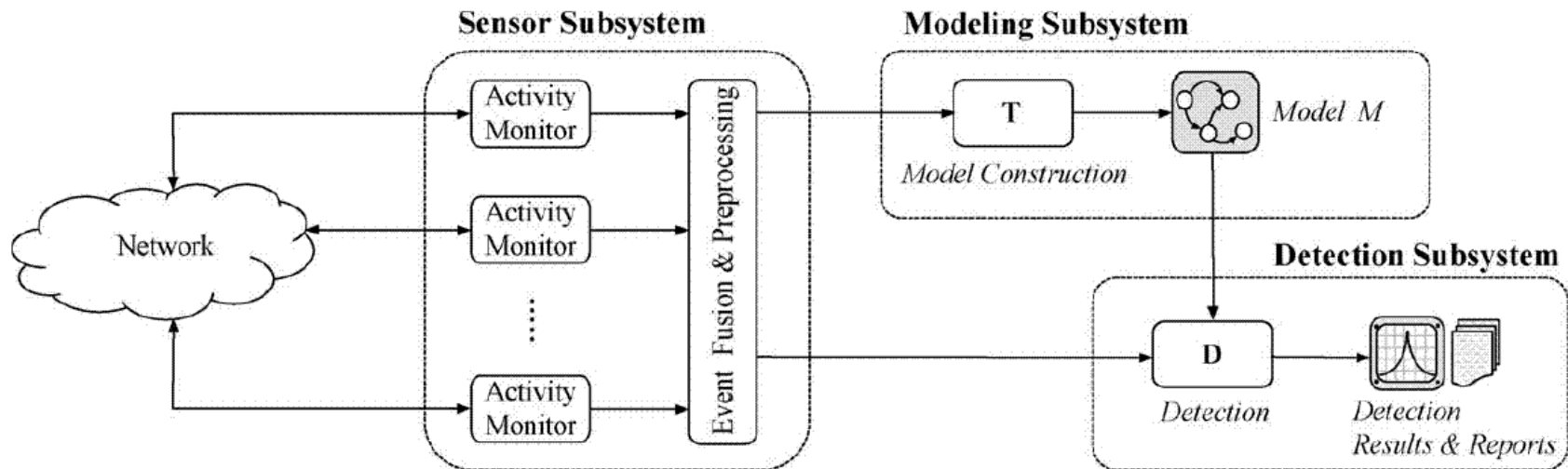
## Anomaly Detection (1)

- ❑ 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: [Tapiador04]
  - 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]

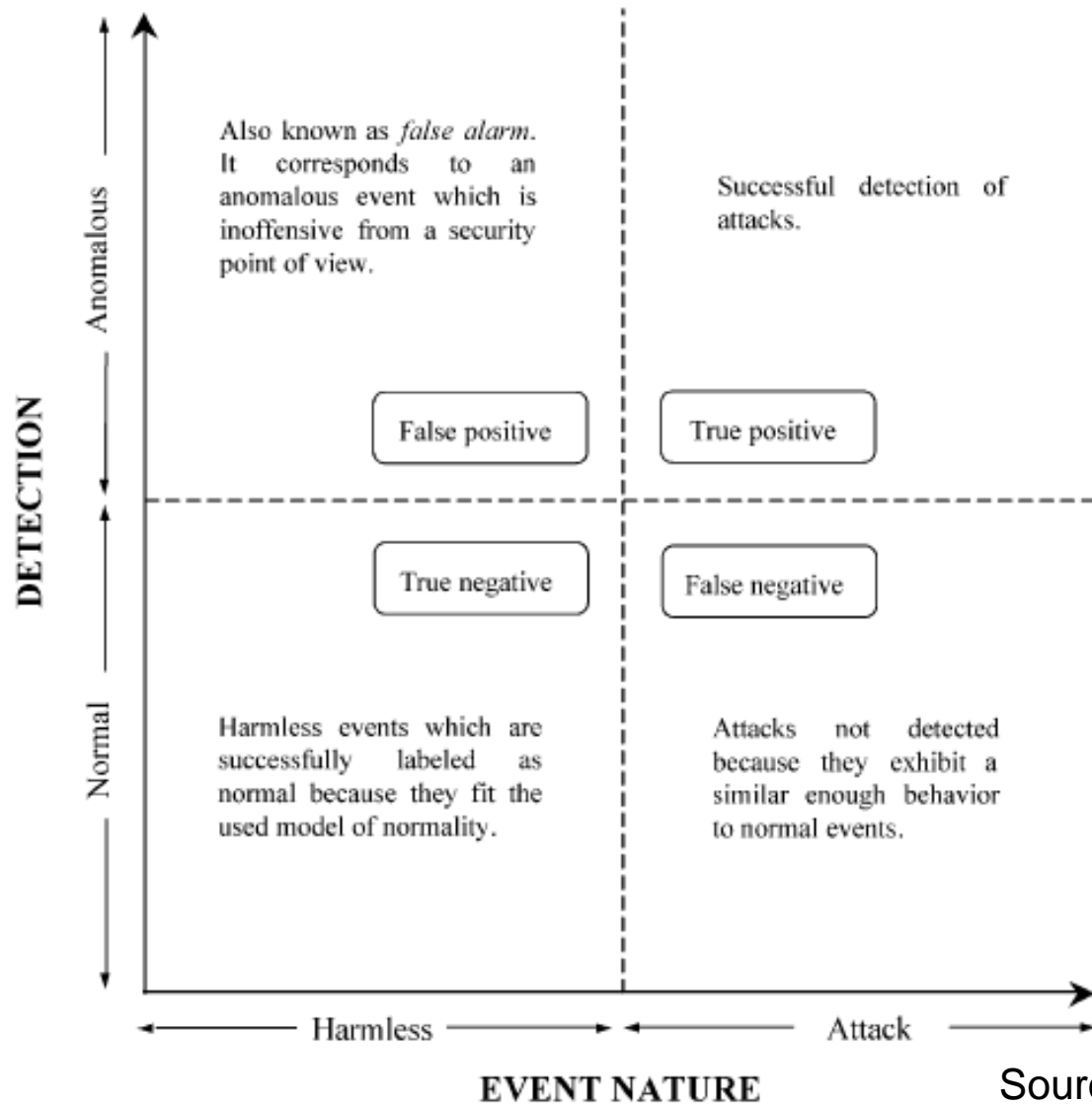


## Anomaly Detection (2)

- 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
    - Mostly no periodic behavior
    - Applications are very diverse
  - 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
- *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 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.