

# Intrusion Detection

Security 1 (CM0475, CM0493) 2019-20  
Università Ca' Foscari Venezia

Riccardo Focardi

[www.unive.it/data/persone/5590470](http://www.unive.it/data/persone/5590470)  
[secgroup.dais.unive.it](http://secgroup.dais.unive.it)



# Introduction

## Intrusion detection

**Intrusion:** unauthorized act of **bypassing** the security mechanisms of a system

**Intrusion detection:** analysis of information from a computer or a network to **identify** possible intrusions

# Introduction

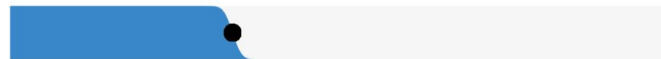
## Intruders

### [Verizon Data Breach Investigations Report 2019](#)

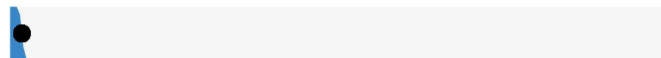
69% perpetrated by outsiders



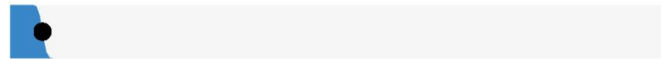
34% involved Internal actors



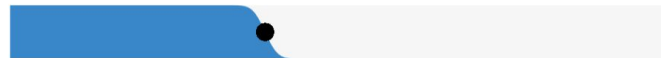
2% involved Partners



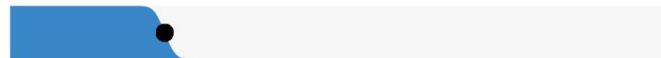
5% featured Multiple parties



Organized criminal groups  
were behind 39% of breaches



Actors identified as nation-state or state-  
affiliated were involved in 23% of breaches



0% 20% 40% 60% 80% 100%

# Introduction

Causes and tactics

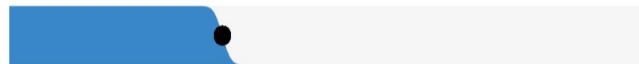
[Verizon Data Breach](#)

[Investigations Report 2019](#)

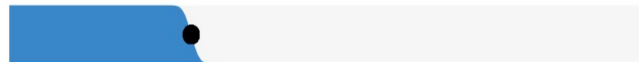
**52%** of breaches featured Hacking



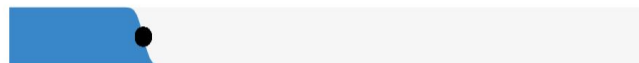
**33%** included Social attacks



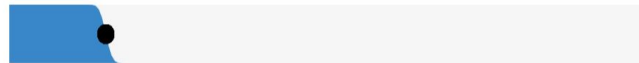
**28%** involved Malware



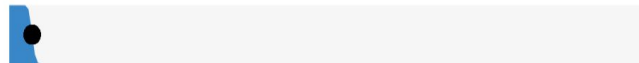
Errors were causal events in **21%** of breaches



**15%** were Misuse by authorized users



Physical actions were present in **4%** of breaches



0% 20% 40% 60% 80% 100%

# Introduction

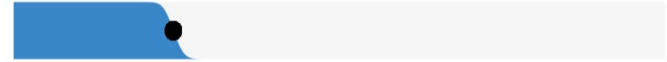
Other ....

[Verizon Data Breach  
Investigations Report 2019](#)

**71%** of breaches were financially motivated



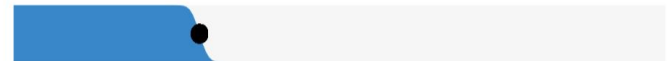
**25%** of breaches were motivated by the gain of strategic advantage (espionage)



**32%** of breaches involved phishing



**29%** of breaches involved use of stolen credentials



**56%** of breaches took months or longer to discover



0% 20% 40% 60% 80% 100%

# Classes of intruders

**Cybercriminals:** individuals or members of an organized crime group with a goal of **financial reward**

**Activists (a.k.a. hacktivists):** individuals or groups motivated by **social** and **political** causes

**Examples:** Anonymous, LulzSec, WikiLeaks, ...

**State-sponsored organizations:** groups of hackers sponsored by governments to conduct **espionage** or **sabotage** activities

**Others:** hackers motivated by **technical challenges** or by peer esteem and reputation, usually advancing the **state-of-the-art** in hacking techniques

# Intruder's skills

**Apprentice:** has minimal technical skill, primarily uses existing attack toolkits. Also known as “**script-kiddie**”. Comprises the largest number of attackers

**Journeyman:** modifies and extends existing tools, finds **new** variants of vulnerabilities

⇒ Harder to detect than “kiddies”

**Master:** high-level technical skills. Can find new (**0-day**) vulnerabilities and develop **new** attack toolkits. Typically employed by state-level organizations

⇒ Very hard to detect and stop

# Examples of intrusions ([NIST SP 800-61](#))

Remote server **compromise**  
(e.g., getting root access)

Web server **defacing**

**Password** cracking

**Leakage** of credit card numbers and  
credentials

Accessing **sensitive data** without  
authorization

Packet **sniffing** on a network

Credential theft through **phishing**

Using unattended, **logged-in**  
**workstation** without permission



# Intruder behaviour (1)

**Target acquisition and information gathering:** attacker **identifies** and **characterizes** the target system

- **examine** corporate website
- use network exploration / **scanning** tools such as DNS lookup and NMAP
- identify potential **vulnerable** services
- **interact** by email

**Initial access:** is the initial **access** to the target system by the attacker, based on previous phase

- exploit a **vulnerability**
- guess weak **credentials**
- install **malware** by phishing

# Intruder behaviour (2)

**Privilege escalation:** attacker exploits a **local vulnerability** to increase privileges

- **search** for local vulnerabilities
  - install **sniffers** to capture administrator passwords
- ⇒ exploit local vulnerabilities or administrator passwords to gain elevated **privileges**

**Information leakage and system exploit:** **leak** sensitive data and use local data to **access** other systems

- **scan** and examine files
- **transfer** sensitive data outside
- use guessed or captured passwords to **access** other target systems

# Intruder behaviour (3)

**Maintaining access:** enable **continued** access to the system(s)

- install remote administration tools and **rootkits** with **backdoors**
- use admin **password** to access
- modify or **disable** intrusion detection systems

⇒ **hide** presence

**Covering tracks:** remove **evidence** of attack activity

- use **rootkits** to hide installed/modified files
- remove **logs**

# Intrusion Detection System (IDS)

**IDS:** Hardware or software that analyzes information from a computer or a network to **identify** possible intrusions

**Sensors:** **collect** data that might contain evidence of intrusion

- network **packets**
- **logs**
- **syscall** traces

**Analyzers:** receive input from sensors and **determine** if an intrusion occurred

- guidance on possible **actions**
- stores data for **future** analysis

**User interface:** **displays** results of analysis (possible intrusions) and allows for system configuration

# Why shall we bother about IDSs?

1. If an intrusion is detected **quickly enough**, then the intruder can be **identified** and **ejected** from the system before too much damage is done or too much data are compromised.  
  
In case of immediate reaction, damage can be **fully prevented**
2. An effective IDS acts as a **deterrent**, reducing the attack attempts
3. Intrusion detection enables the **collection** of information about intrusion techniques that can be used to **strengthen** system and network **security**

# Detecting intruder behaviour

Honest and malicious behaviours differ ... but they also **overlap**

**False positives:** honest users identified as intruders (**loose** interpretation)

⇒ False alarms

**False negatives:** intruders identified as honest users (**tight** interpretation)

⇒ Missed alarms

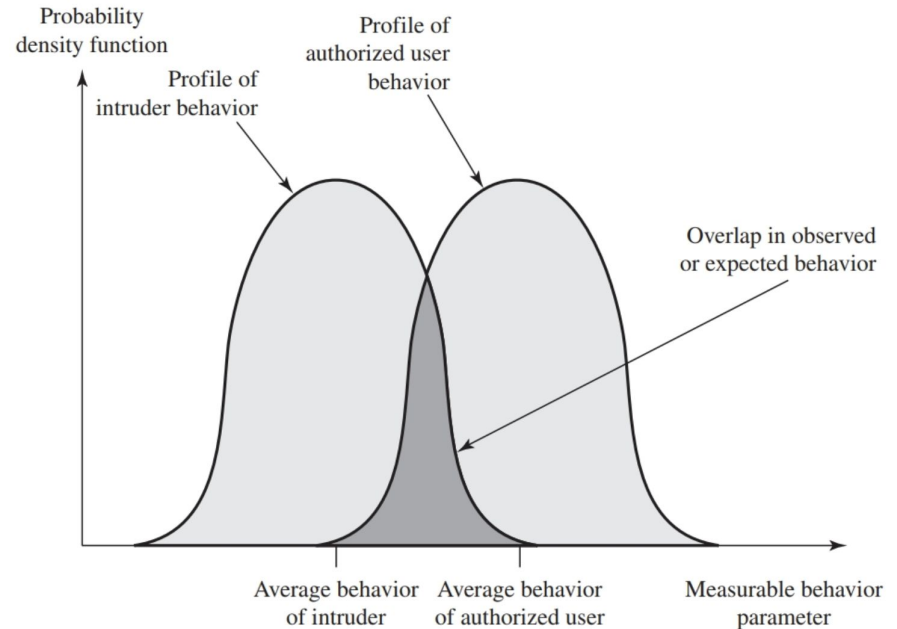


Figure from Lawrie Brown, William Stallings. *Computer Security: Principles and Practice*, 4/E, Pearson.

# False positive paradox

**Base-rate fallacy:** mind tend to **ignore** base-rate when **more specific** rate information is provided

**Example:** **breathalyzers** with 5% false positive rate (and no false negatives)

- Assume 1/1000 drives drunk
- If test on (random) Bob is positive what is the probability that **Bob is really drunk?**

**Answer:** 95% ?

**No!** only ~**2%** !!!

- 1/1000 gives **true positive**
  - 5% of 999/1000 = 49.95 give **false positive**
- ⇒  $1 / (49.95 + 1) = 1.96\%$  of positive tests is really drunk!

(does not hold in case of **other evidence**.... like driving zig-zag!)

# IDS base-rate fallacy

Systems with **few intrusions** (with respect to the false positive rate) present the **base-rate fallacy** issue

## Example:

- 1/10000 **malicious** behaviour
- 5% false positive rate

⇒ 0.2% of positives **will be true**

IDS becomes **useless** with too many false positives

**No** trivial solution:

⇒ It would be necessary to make detection **extremely tight** introducing **false negatives**



# Analysis approaches

**Anomaly detection:** involves the collection of data relating to the behavior of legitimate users so to create a **model** of user behaviour

- current observed behavior is **analyzed** with respect to the legitimate user model
- classified as *intrusion* when **difference** is over a threshold

**Signature or heuristic detection:** also known as misuse detection, uses

- a set of known malicious data patterns (**signatures**)
  - attack rules (**heuristics**)
- ⇒ This approach can only identify **known attacks** for which it has patterns or rules (**no 0-day!**)

# Anomaly-based detection

A **model** of honest user is built from sensor data, collected in a *training phase* (no intrusion)

Approaches:

**Statistical**: statistical profile of observed metrics

👍 Simple and efficient

👎 Non-flexible (which metrics?)

**Knowledge based**: rules that classify legitimate behaviour

👍 Robust and flexible

👎 Difficult to develop, requires experts

**Machine learning**: classification model, automatically built

👍 Flexible and automated

👎 Training expensive, accuracy not yet optimal

# Signature and Heuristic Detection

**Signature-based:** match known malicious **patterns** (large enough to minimize false positives)

**Example:** anti-virus

- 👍 **Fast**, widely accepted
- 👎 Continuous review of malware and attacks to **create the signatures**
- 👎 Inability to detect new, **0-day** attacks

**Heuristic-based:** **rules** that identify intrusions or suspicious behaviour, often derived by analyzing existing attack tools

- 👍 **Fast**, widely accepted
- 👎 Rules are **specific** to the machine and operating systems
- 👎 If rules are known, attackers can find ways to **circumvent** them

# IDS classification

**Host-based IDS (HIDS):** Monitors the events occurring in a **single host**, such as process identifiers and the system calls they make

**Network-based IDS (NIDS):** Monitors **network traffic** for particular network segments or devices and analyzes **protocols** to identify suspicious activity

**Distributed or hybrid IDS:** Combines information from a number of sensors, often both host and network-based, in a **central analyzer** that is able to better identify and respond to intrusion activity

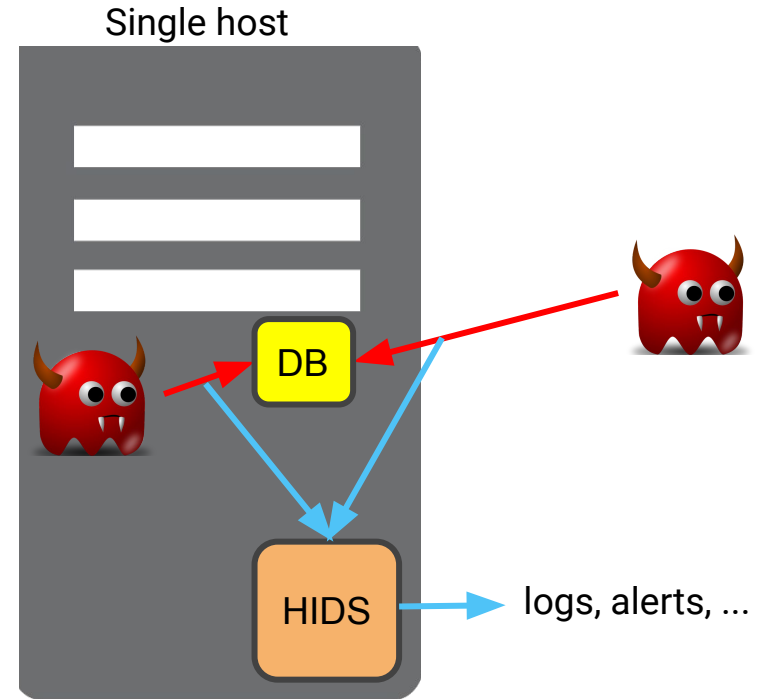
⇒ sums up the **advantages** of multiple HIDS and NIDS

# Host-based IDS (HIDS)

**HIDS**: an IDS running directly on a host to protect its application

⇒ **detects** intrusions, **logs** suspicious events, send **alerts**

⇒ detects both **internal** and **external** intrusions



# HIDS sensors (1)

**System call traces:** sequence of **syscalls** invoked by processes

Syscall traces provide accurate information about the **interaction** of processes with the OS

**Anomaly-based:** create **models** of honest syscall traces

**Heuristic-based:** **rules** that detect suspicious syscall invocation

**Log files:** modern systems already log **events** which can be directly used as sensors for HIDS

- 👍 **Less overhead** than syscall traces
- 👎 Less information, **lower detection rate**
- 👎 Might be easier for the intruder to **manipulate**

# HIDS sensors (2)

**File checksums:** compare crypto **checksum** with stored ones. Look for changes to important files

- 👍 Easily detects **integrity** attacks
- 👎 **Overhead** managing checksums
- 👎 **Complex** to configure: which files to monitor to reduce false positive while detecting intrusions?

**Example:** [Tripwire](#)

**Registry access:** monitor access to the **registry** (Windows OS **specific**)

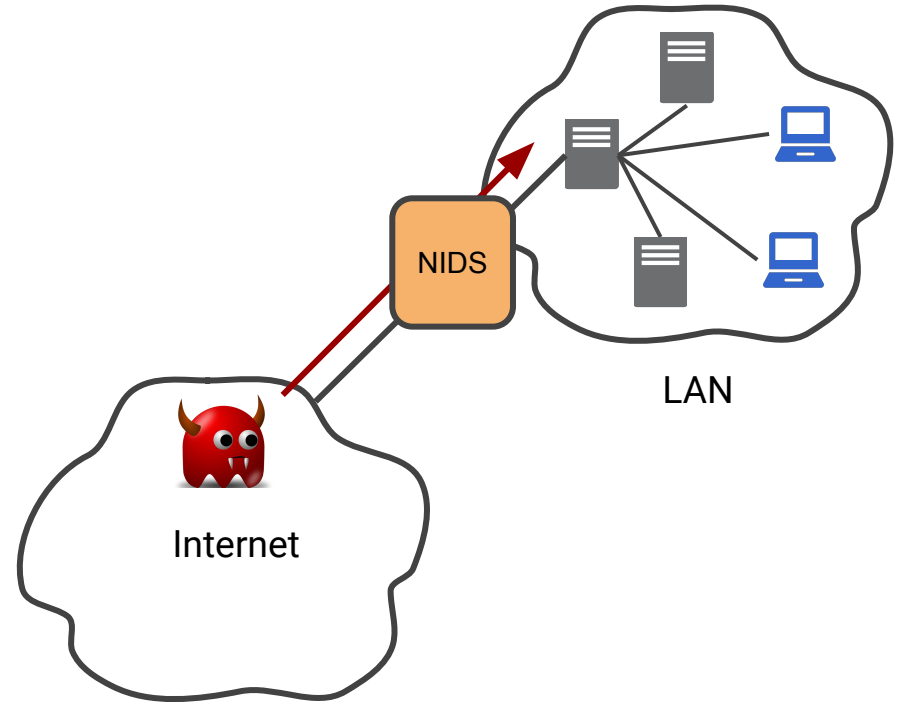
**Files:** Signature-based HIDS that look for known **signatures** such as in anti-virus programs (file system, attachments, ...)

**Accesses to resources:** Heuristic-based HIDS that look for known **suspicious** access requests

# Network-based IDS (NIDS)

**NIDS:** an IDS that monitors traffic at selected points on a network

Inspects **network packets** directed to (potentially vulnerable) hosts





# Types of NIDS sensors

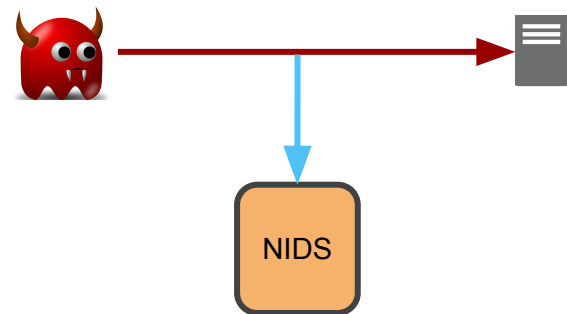
**Inline:** traffic **must pass** through the sensor

- 👍 Can detect and also **prevent** intrusions, by blocking packets
- 👎 Overhead managing and analyzing packets, might require **expensive hardware** to avoid delays

**Example:** NIDS integrated in a firewall or switch device

**Passive:** monitors a copy of network traffic

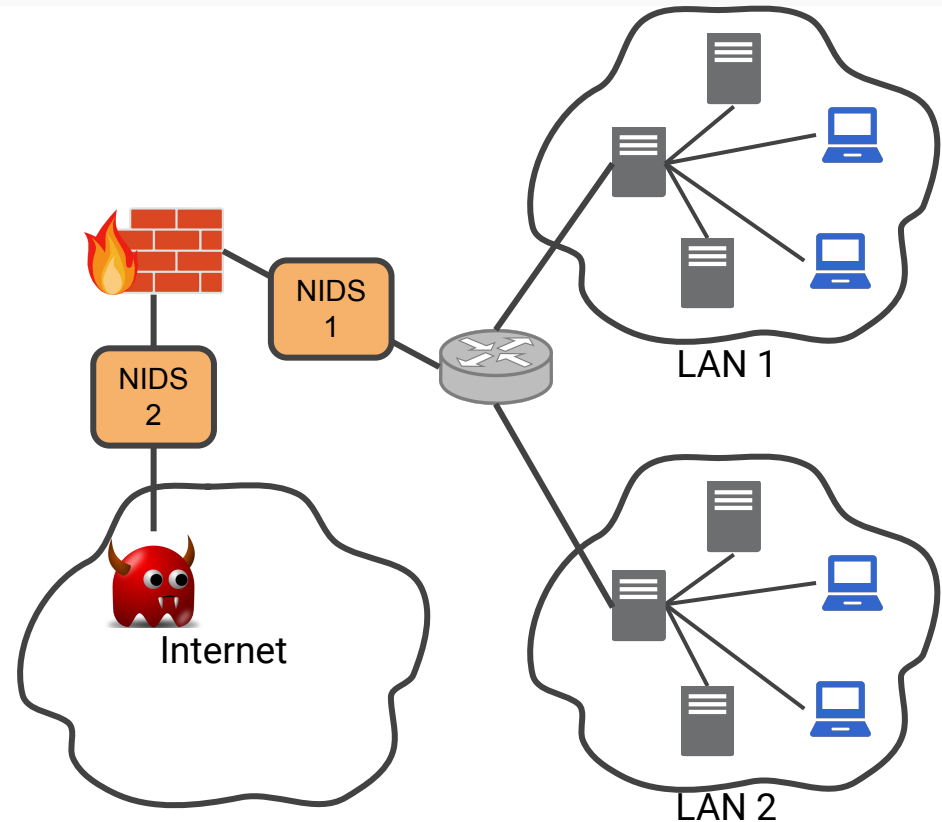
- 👍 Does not introduce delays
- 👎 Cannot prevent attacks



# NIDS sensor deployment (1)

On the **external perimeter**:

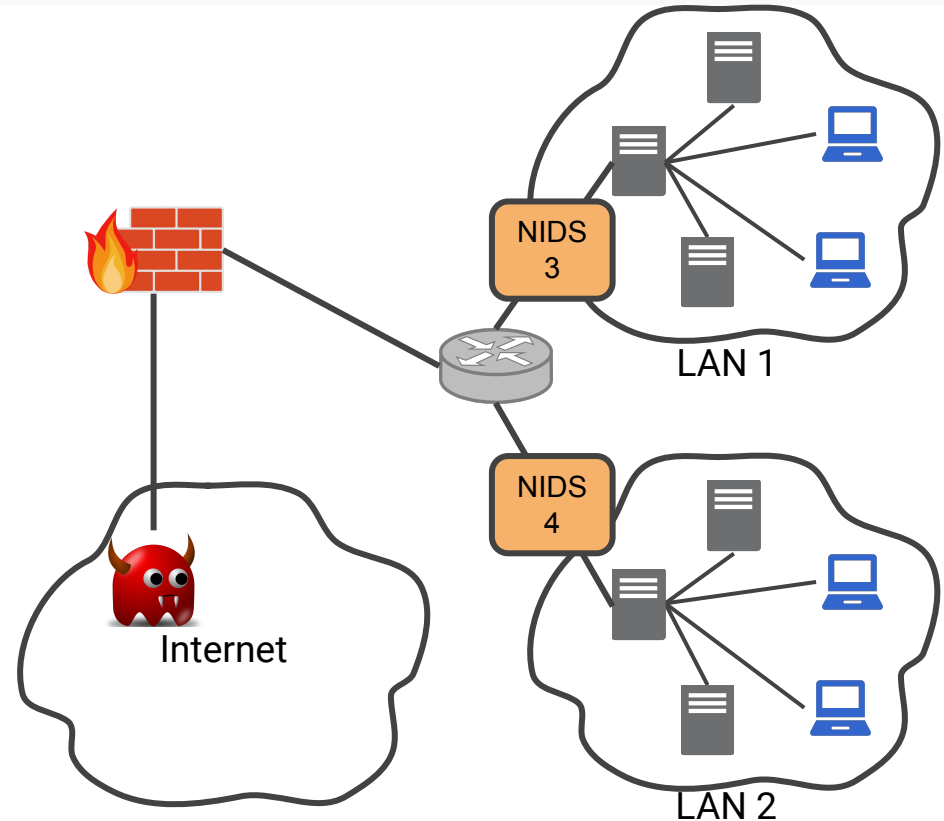
- 👍 Detects **external** intrusions
- 👍 Detects firewall **misconfiguration** (if after the firewall, NIDS 1 )
- 👍 Can detect **outgoing** malicious traffic
- 👎 Does **not** detect **internal** attacks
- 👎 High **load** if before the firewall (NIDS 2)



# NIDS sensor deployment (2)

Before the **LANs**:

- 👍 Detects both **internal** and **external** intrusions
- 👍 Detects firewall **misconfiguration**
- 👍 Can detect **outgoing** malicious traffic
- 👍 Can be configured on **specific** resources



# Anomaly-based NIDS detection

**Denial-of-service (DoS):** involve **anomalous** increased packet traffic or increased connection attempts

**Scanning:** A scanning attack occurs when an attacker probes a target network or system by sending different kinds of packets. It can be detected by **atypical** flow patterns

**Worms:** show anomalous behaviour on the network:

- propagate quickly and use large amounts of **bandwidth**
- cause hosts to **communicate** (that typically do not)
- cause hosts to use **ports** that they normally do not use
- many worms perform **scanning**

# Signature-based NIDS detection

**Application layer attacks:** patterns of attacks targeting application layer protocols

**Transport layer attacks:** unusual packet fragmentation, TCP-specific attacks such as **SYN floods**

**Network layer attacks:** spoofed IP addresses and illegal IP header values

**Unexpected application services:** detect if activity on a transport connection is **consistent** with the expected application protocol

**Policy violations:** Examples include use of inappropriate websites and use of **forbidden** application protocols

# Honeypots

**Decoy systems** to **attract** intruders

- **divert** an attack from accessing critical systems
- **collect** information
- try to **keep** the intruder long enough to be able to respond

Honeypots are not accessed by users  
so any access is an **attack!**

**Low interaction honeypot:** provide a **realistic** interaction, but does not execute a full version of services

**High interaction honeypot:** **full** operating system, services and applications

⇒ identical to production systems  
but if compromised could be used  
to **attack other systems**

# Honeypot deployment

On the **external perimeter** (Honey 1):

- 👍 **No danger** for the LAN
- 👍 **Reduces** intrusions
- 👎 Cannot trap **internal** intruders

**In the LAN** (Honey 2):

- 👍 Traps **internal** intruders
- 👍 Detect **misconfigured** firewalls
- 👎 Might attack other systems

