Beveiliging van netwerken en computers

CHAPTER 6
INTRUSION DETECTION

PROF. DR. IR. ELI DE POORTER

eli.depoorter@ugent.be

GHENT UNIVERSITY – IMEC

IDLAB

http://idlab.technology | *http://idlab.ugent.be*

imec IDLab INTERNET & DATA LAB

GHENT UNIVERSITY

Overview

- Intrusion detection
  - **Introduction**
  - Audits
  - Practical approaches
  - Honeypots
  - Limitations

"An ounce of prevention is worth a pound of detection"

2

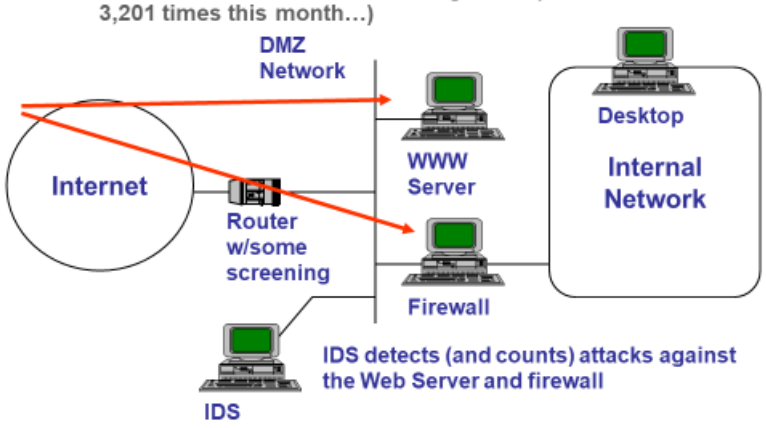
What is an intrusion?

- Difficult to define
  - **Not everyone agrees**
  - **This is a *big* problem**
    - **How about someone telnetting your system?**
      - ✓ And trying to log in as "root"?
    - **What about a ping sweep?**
    - **What about them running a port scan?**
    - **What about them trying to download code on your webserver?**

3

Attack Detection

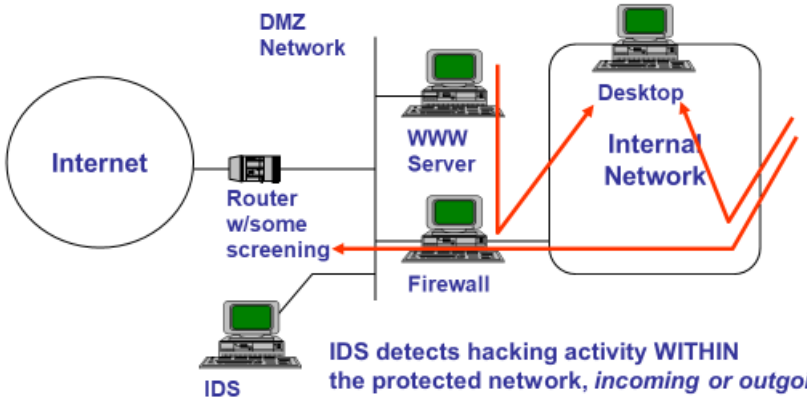
- **Attack Detection (AD)**
  - **Placing an IDS outside of the security perimeter records *attack level***
    - If the perimeter is well designed the *attacks should not affect it!*
      - ✓ Will generate a lot of noise and be ignored quickly
    - Still useful information for management ("we have been attacked 3,201 times this month…)



4

Intrusion detection

- **Intrusion Detection (ID)**
  - **Placing an IDS within the perimeter will detect instances of clearly improper behavior**
    - Hacks via backdoors
    - Hacks from staff against other sites
    - Hacks that got through the firewall
  - **When the IDS alarm goes off, it's a red alert**



5

Attack vs intrusion detection

- Ideally do *both*
  - Realistically, do ID first then AD
    - Or, deploy AD to justify security effort to management, then deploy ID (more of a political problem than a technical one)
  - The real question here is one of staffing costs to deal with alerts generated by AD systems

6

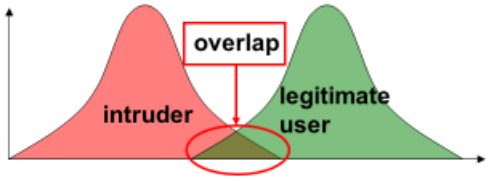
Intrusion detection

- Goal of intrusion detection
  - Detecting an attack
    - During the attack
    - After the attack
  - Doesn't replace preventive measures
    - However a useful complement
    - Catching attacks that could not be prevented

7

Intrusion detection

- **Assumption**
  - **Intruder behaves differently from legitimate user**
    - **Difference should be measurable**
    - **In reality: some overlap between both behaviours**
      - ✓ Implying false positive and false negative decisions
  - **Ideal detection probability**
    - **0% false negatives**
      - ✓ Don't let an attack pass undetected
    - **0% false positives**
      - ✓ A False Positive is when a system raises an incorrect alert
      - ✓ It's easy to achieve this: *simply detect nothing*

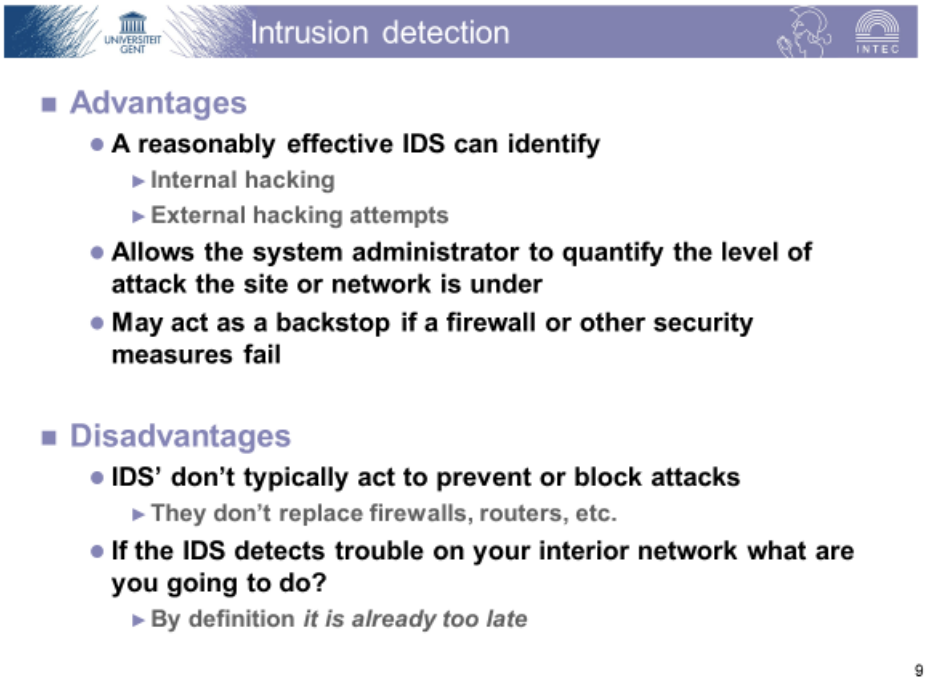


8

A false positive decision occurs when legitimate use is considered an intrusion. If the number of false positive decisions becomes too large, it is almost impossible to discern the real intrusions from the large quantity of false alarms.

A false negative decision occurs when an intrusion is considered legitimate use. This intrusion will then typically not be detected.

In practice false positive alerts are the more problematic issue of intrusion detection systems, as legitimate use is much more common than intrusion attempts.



The ideal Intrusion Detection System (IDS) will notify the system/network manager of a successful attack in progress:

- With 100% accuracy
- Promptly (in under a minute)
- With complete diagnosis of the attack

Ideally, an IDS will categorize/identify the attack

- Assess the target's vulnerability
- Notify the administrator
- If the vulnerability has a known "fix" it would include directions & recommendations for applying the fix

Unfortunately, this is not realistic. In practice the best you can hope for is to get information on probable attacks, as well as which types of attacks are being performed. Nevertheless, even when this information is not provided real-time, such a system can potentially save millions of euros by catching intruders before any real harm is done.

Intrusion detection

- **Usefulness of intrusion detection**
  - **If detection is sufficiently rapid**
    - Identify and expel intruder before any damage is caused
  - **But even in case of detection at a later stage**
    - Evaluate and repair damage as quickly as possible
  - **Deterrent**
    - Risk for attacker to be exposed
  - **Collection of information about attack techniques**

10

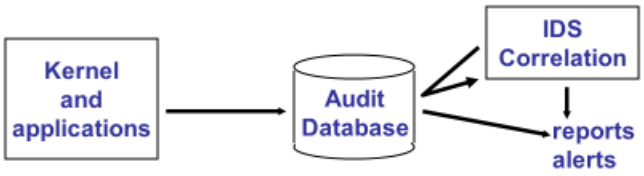
Overview

- **Intrusion detection**
  - Introduction
  - **Audits**
  - Practical approaches
  - Honeypots
  - Limitations

11

Audit records

- Fundamental part of IDS
  - Records of user activity: input for IDS
  - Used for dissecting user operations in elementary steps
  - Audit based IDS post-process audit trail (and other) information
    - Activity is first logged *then* post-processed
    - Batch oriented approach allows for virtually infinite correlation if enough data is present

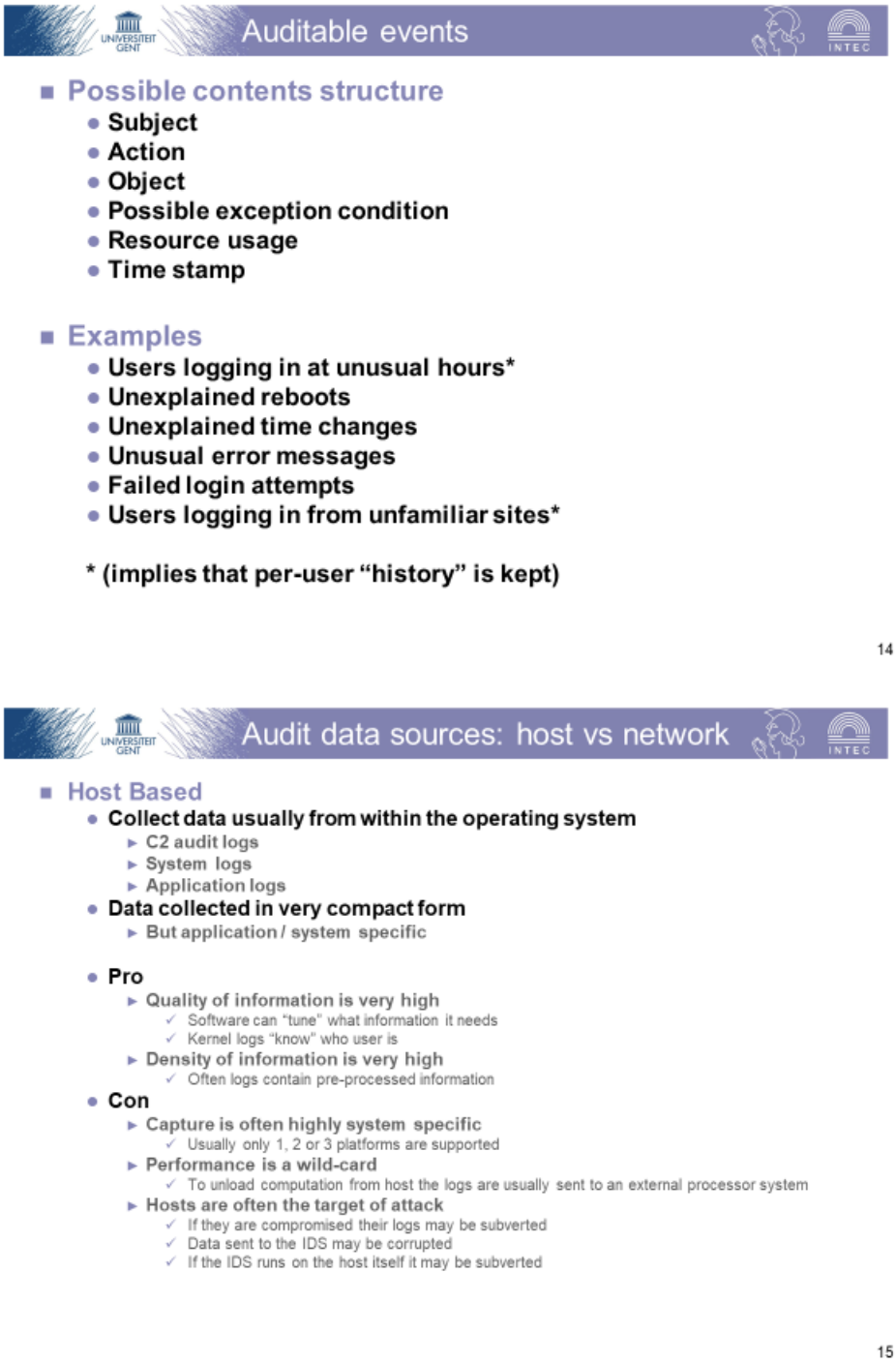


12

Audit data

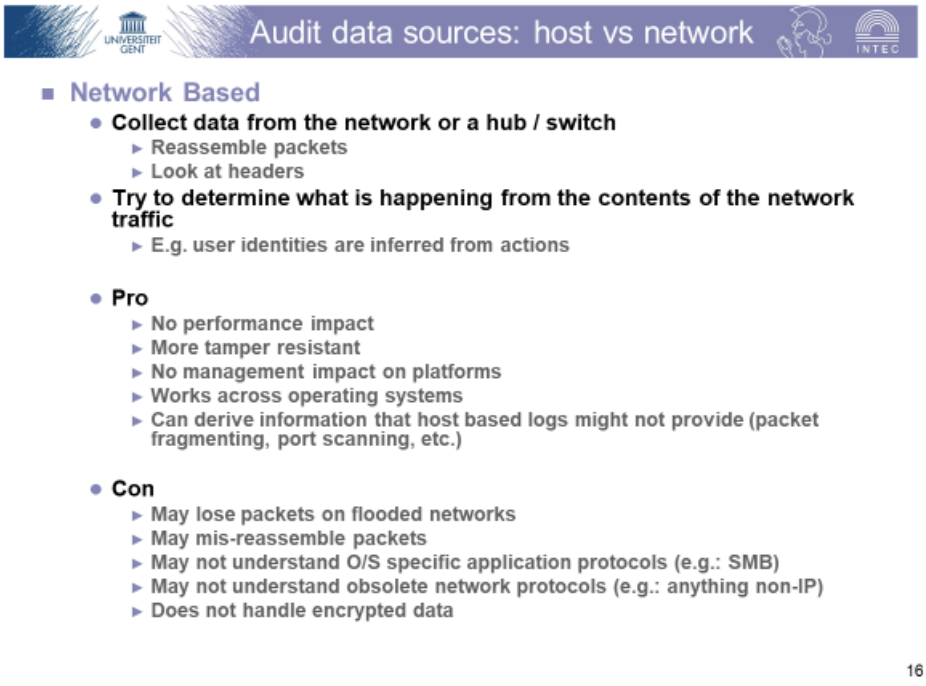
- Types
  - "Native audit records"
    - Component of most operating systems
    - Doesn't require additional software…
    - …but isn't always in most adequate format
  - Detection specific audit records
    - Collect IDS specific information
    - Possibly system independent
    - Drawback: 2 audit programs on same system
- Determining what is a good audit probe point (what & where to record something) is a difficult problem
  - Orange book on "Trusted Computer System Evaluation Criteria" includes 23 probe points within UNIX kernel and applications
  - Most vendor specific solutions are not compatible with each other

13



Intrusion detection systems are of two main types: network based (NIDS) and host based (HIDS) intrusion detection systems.

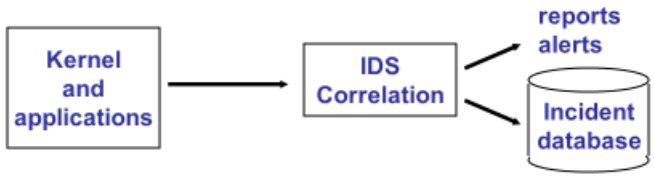
Host Intrusion Detection Systems (HIDS) run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator if suspicious activity is detected. It takes a snapshot of existing system files and matches it to the previous snapshot. If the critical system files were modified or deleted, an alert is sent to the administrator to investigate. An example of HIDS usage can be seen on mission critical machines, which are not expected to change their configurations.



Network Intrusion Detection Systems (NIDS) are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. It performs an analysis of passing traffic on the entire subnet and matches the traffic that is passed on the subnets to the library of known attacks. Once an attack is identified, or abnormal behavior is sensed, the alert can be sent to the administrator. Typical locations to install a NIDS are the subnets of the DMZ to see if someone is trying to break into the firewall. Ideally one would scan all inbound and outbound traffic, however doing so might create a bottleneck that would impair the overall speed of the network. NID Systems are also capable of comparing signatures for similar packets to link and drop harmful detected packets which have a signature matching the records in the NIDS. When we classify the designing of the NIDS according to the system interactivity property, there are two types: on-line and off-line NIDS. On-line NIDS deals with the network in real time and it analyses the Ethernet packet and applies rules to decide if it is an attack or not. Off-line NIDS deals with a stored data and pass it on an offline external process to decide if it is an attack or not.

Inline approaches

- Inline (real-time) intrusion detection approaches
  - Process audit data as it is generated
    - Typically discard audit data that it does not recognize as significant
  - Amount of correlation tends to be limited
- Inline is faster but only provides a "local" view unless a lot of data is forwarded in real-time to a central location
  - Audit is deeper but requires keeping lots of data
  - Hybrid systems exploit both: inline detection of significant events to an audit station

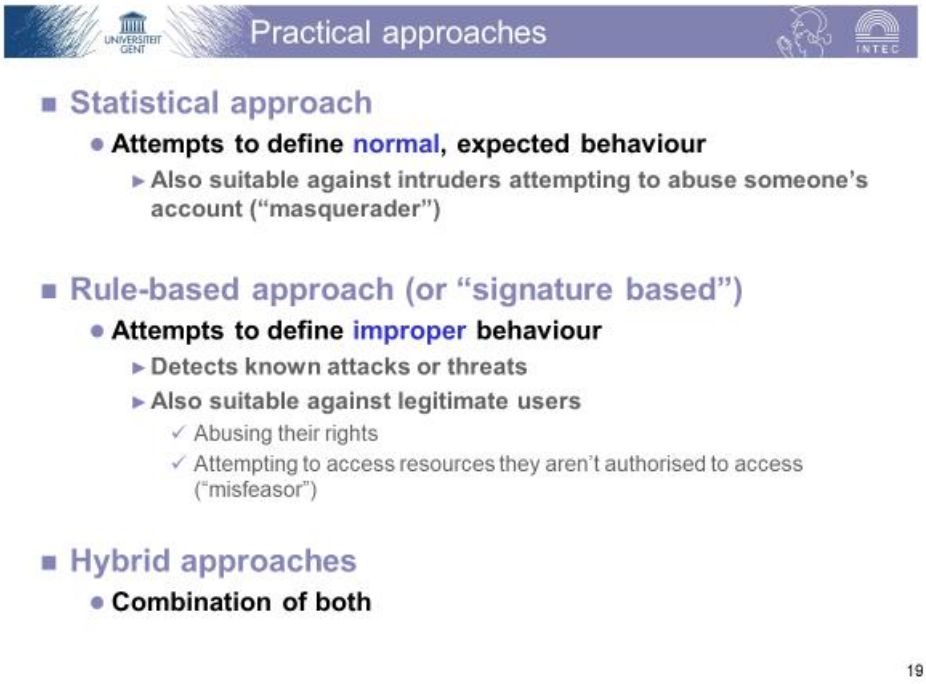


17

Overview

- Intrusion detection
  - Introduction
  - Audits
  - Practical approaches
  - Honeypots
  - Limitations

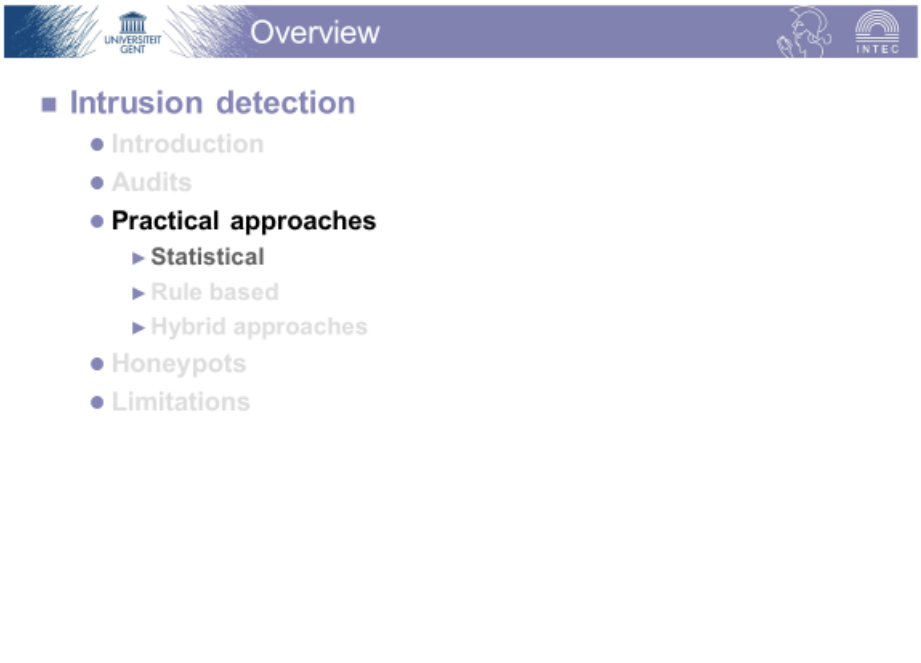
18

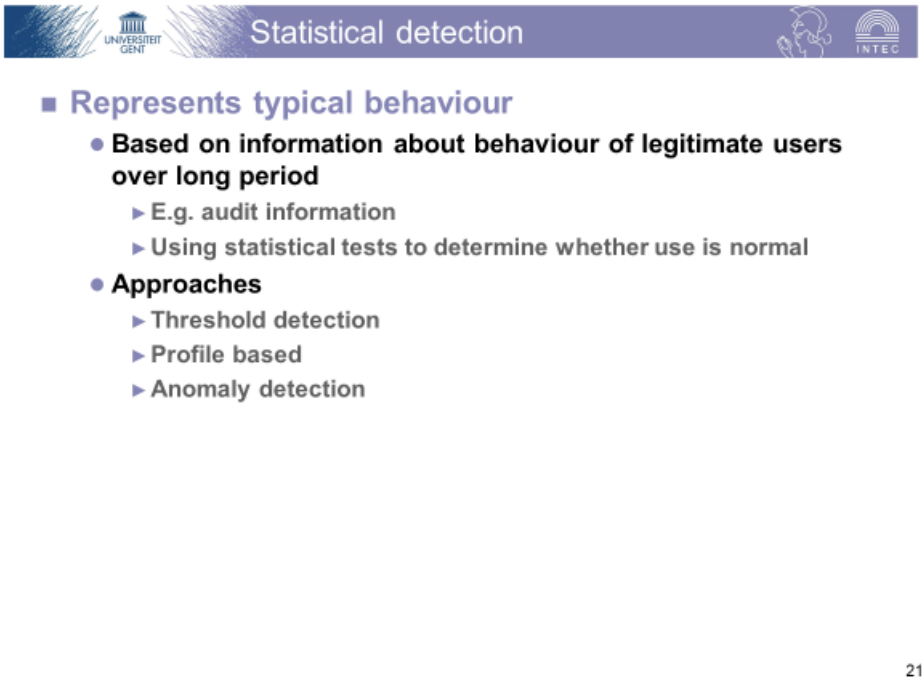


Statistical anomaly-based IDS. An IDS which is anomaly based will monitor network traffic and compare it against an established baseline. The baseline will identify what is "normal" for that network- what sort of bandwidth is generally used, what protocols are used, what ports and devices generally connect to each other- and alert the administrator or user when traffic is detected which is anomalous, or significantly different, than the baseline. The issue is that it may raise a False Positive alarm for a legitimate use of bandwidth if the baselines are not intelligently configured.

Signature-based IDS. A signature based IDS will monitor packets on the network and compare them against a database of signatures or attributes from known malicious threats. This is similar to the way most antivirus software detects malware. The issue is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to the IDS. During that lag time the IDS would be unable to detect the new threat.

In practice both approaches are often combined to be able to detect a variety of attacks that is as large as possible.





The advantage of statistical approaches is that little prior knowledge is required. The detection system "learns" what normal behaviour is and searches for deviations from this normal behaviour. It is not dependent on system specific properties and vulnerabilities and can therefore also be easily transferred between different systems.

Statistical detection: threshold detection

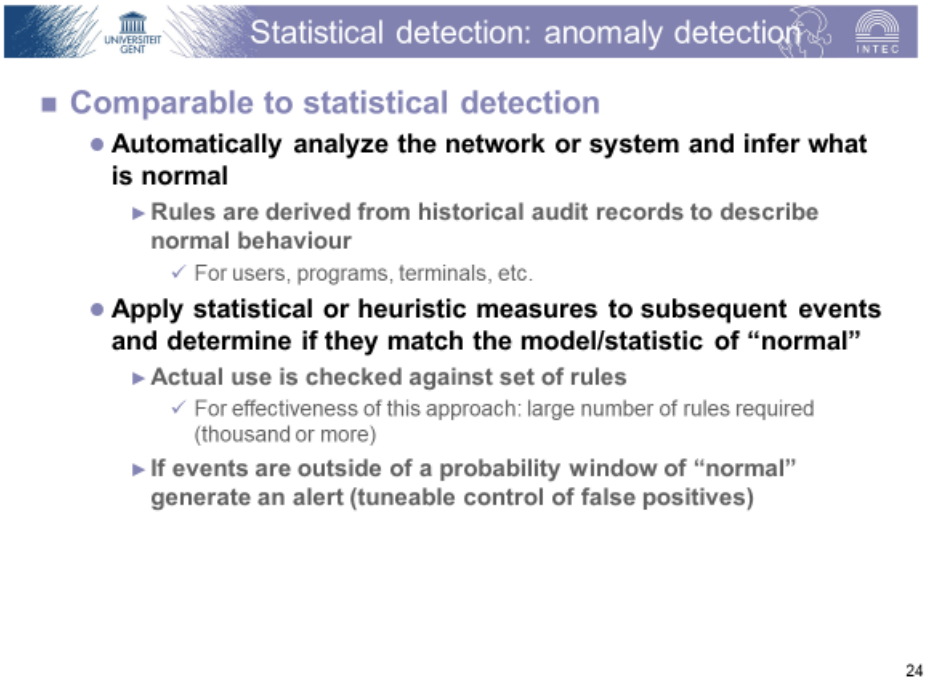
- Defines (user independent) thresholds for frequency at which various events occur
  - Counting number of occurrences of specific event within some time span
  - Intrusion presumption when threshold (limit for reasonable number) is crossed
- Not very efficient of itself
  - Many false positive and false negative decisions
  - Still useful in combination with other techniques

22

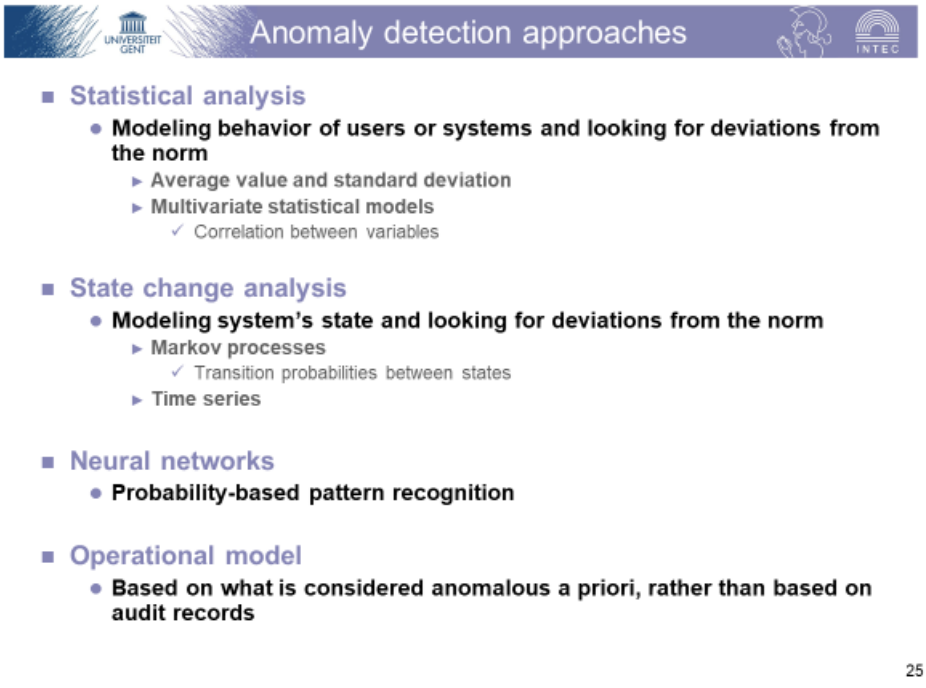
Statistical detection: profile based

- Concept
  - Defines activity profile for each user or user group
  - Detects possible changes in this activity
- Approach
  - Designer defines metrics representing user behaviour
  - Analysis of audit records over time interval produces activity profile for average user
    - Definition of typical behaviour
  - Analysis of actual audit records allows comparison with typical behaviour
    - Detection of possible intrusion
- Possible metrics
  - Counter: positive, strictly increasing
    - E.g. counting number of login attempts by user within 1 session
  - Gauge: positive
    - E.g. number of connections for 1 user application
  - Interval timer: time span between events
    - E.g. timespan between login attempts
  - Resource usage: over some time span
    - E.g. CPU time for program execution

23

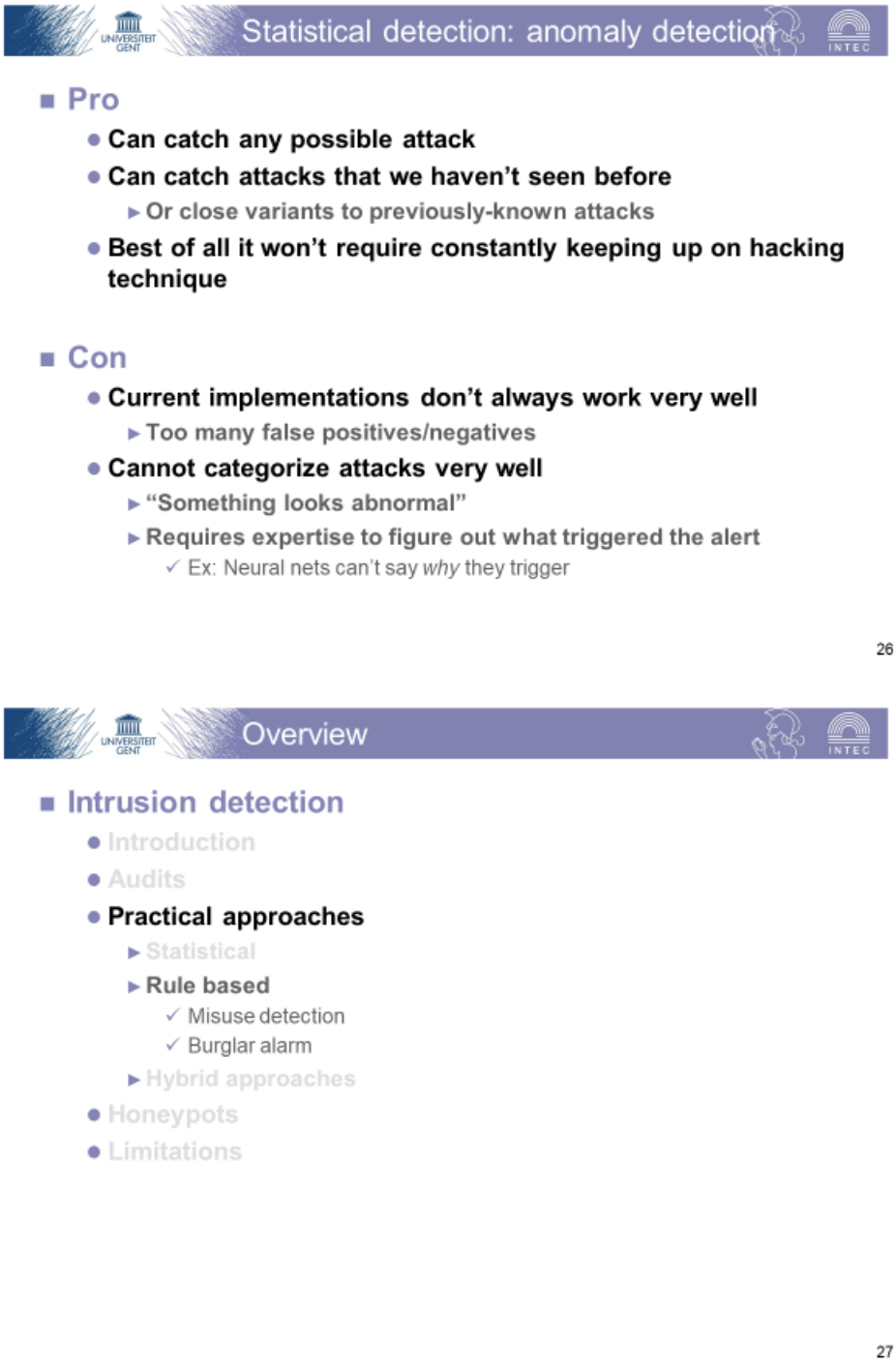


Anomaly detection approaches automatically derive rules to detect deviations from earlier system behaviour patterns. Here too, no prior knowledge of system vulnerabilities is required. In contrast to the previous approach, the system itself is modelled, rather than user behaviour.



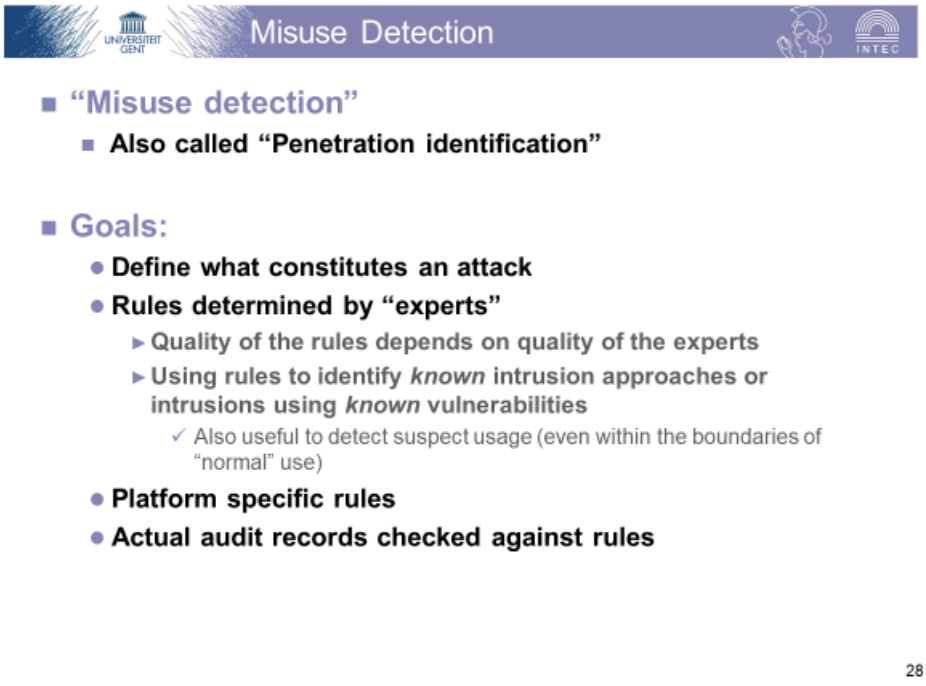
In order to determine what is attack traffic, the system must be taught to recognize normal system activity. This can be accomplished in several ways.

- A common method is to define what normal usage of the system comprises using a strict mathematical model, and flag any deviation from this as an attack. This is known as strict anomaly detection. Average values and standard deviations are probably the easiest statistics to compute, but probably not very useful of themselves, whereas other statistical tests can yield more reliable results.
- A possible use of Markov series could be the observation of transitions between some instructions. Similarly, time series can yield useful information, if some events follow each other too rapidly or too slowly compared to normal usage. Statistical tests to analyse this also exist.
- Most current systems utilize artificial intelligence type techniques. Systems using neural networks have been used to great effect.
- Finally, an operational model can be used when one has an idea beforehand of what is anomalous behaviour, without needing to analyse audit records. An obvious example is a large number of login attempts within a short period of time, which is a possible sign of an intrusion attempt.



In contrast to statistical approaches that define typical behavior, rule-based systems attempt to define set of rules determining whether behavior is an abnormal. Two main approaches are:

- Misuse detection or penetration identification, which detects known attack vectors.
- Burglar alarms, which detect deviations from policies.



In a misuse detection approach, we define abnormal system behavior first, and then define any other behavior, as normal behavior. It stands against anomaly detection approach which utilizes the reverse approach, defining normal system behavior and defining any other behavior as abnormal. In other words, anything we don't know is normal.

Misuse Detection

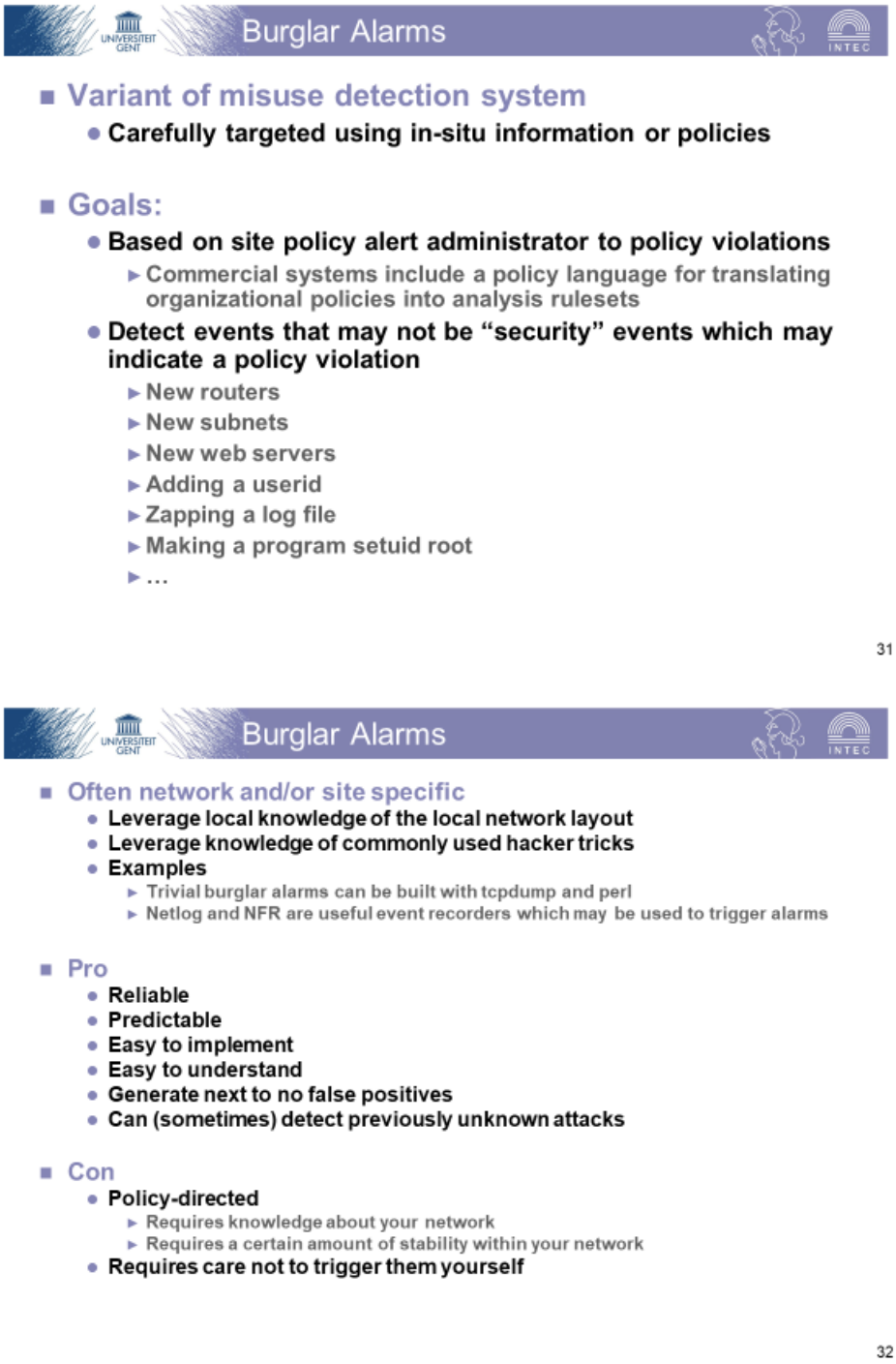
- **Possible misuse detection rules**
  - **Users don't read files in other users' directories**
  - **Users normally only access disk using higher level OS functions**
    - In contrast to malware e.g.
  - **Users don't copy system files**
  - **"Network grep"**
    - look for strings in network connections which might indicate an attack in progress
  - **Pattern matching**
    - Encode series of states that are passed through during the course of an attack
      - ✓ e.g.: "change ownership of `/etc/passwd`" -> "open `/etc/passwd` for write" -> *alert*
  - **Other examples**
    - IP Frag attack
    - Source routing
    - SATAN scan check
    - Rwhod check
    - Rlogin -froot
    - SMTP WIZ check … etc.
    - Ping flooding
    - Ping of death
    - IMAP buffer smash
    - Rlogin decode
    - TFTP get passwd check
- **Commercial products**
  - **Typically updated with rulesets as subscription service**

29

Misuse Detection

- **Misuse detection systems are similar to virus scanning systems:**
  - **Both rely on meta-rules of vulnerabilities**
  - **Both need frequent rules updates**
  - **Both are easily fooled by slight mutations in virus/attack signature**
  - **Both are fairly low in generating false positives**
  - **Pro**
    - Easy to implement, easy to deploy,
    - Easy to update, easy to understand
    - Low false positives
    - Fast
  - **Con**
    - Cannot detect something previously unknown
    - Constantly needs to be updated with new rules
    - Easier to fool

30



The ideal burglar alarm will be situated so that it fires when an attacker performs an action that they normally would try once they have successfully broken in.

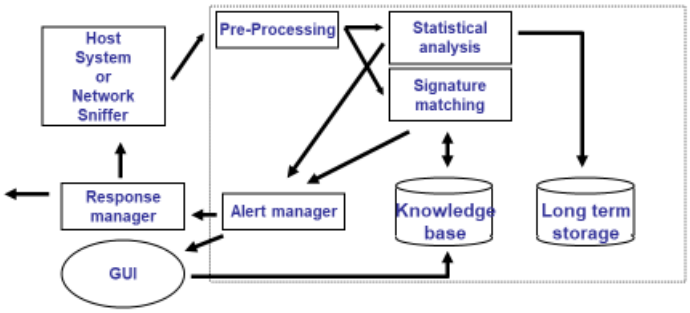
Overview

- Intrusion detection
  - Introduction
  - Audits
  - Practical approaches
    - Statistical
    - Rule based
    - Hybrid approaches
  - Honeypots
  - Limitations

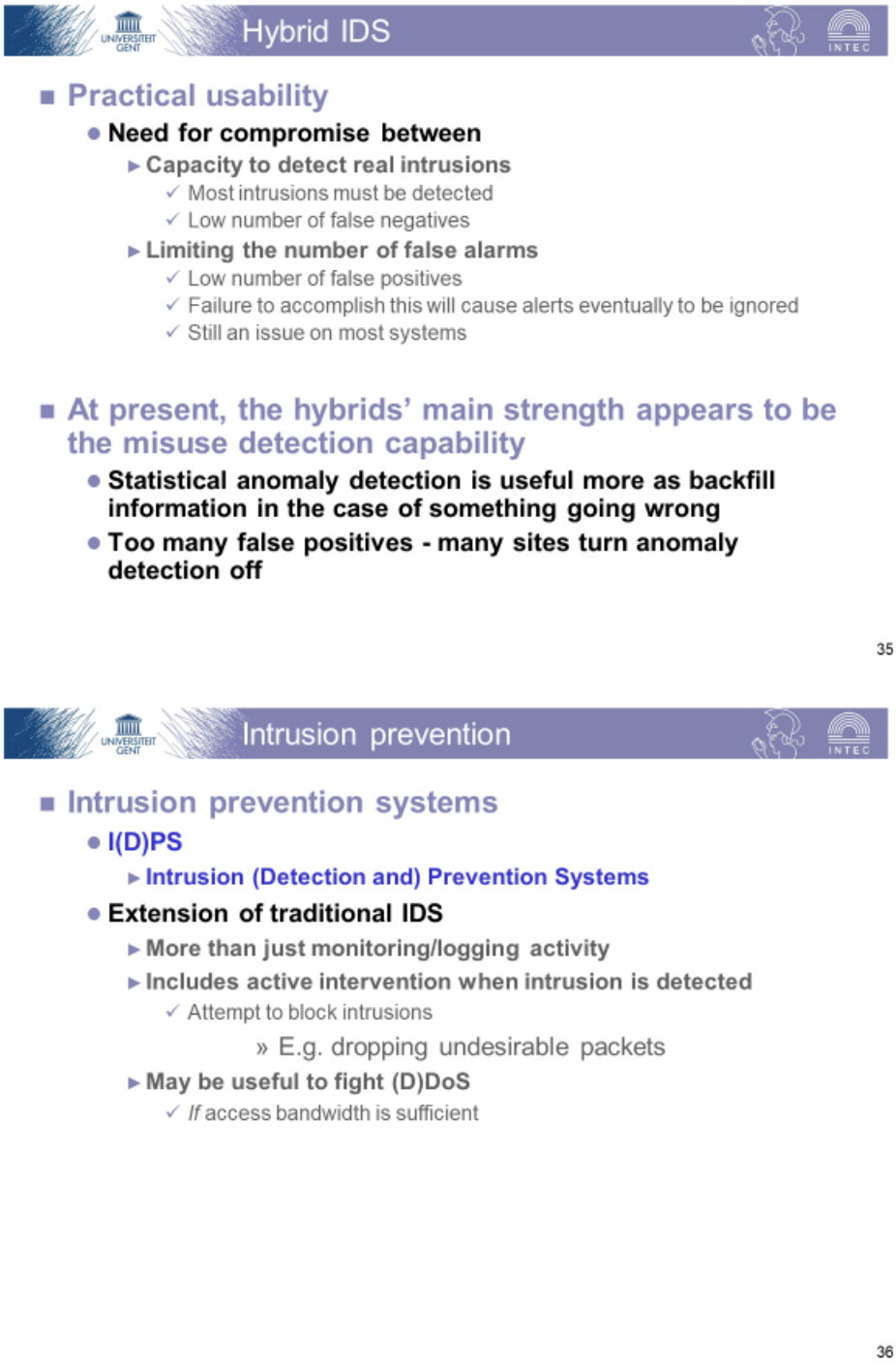
33

Hybrid IDS

- The current crop of commercial IDS are mostly hybrids
  - Misuse detection (signatures or simple patterns)
  - Expert logic (network-based inference of common attacks)
  - Statistical anomaly detection (values that are out of bounds)
  - Typical block diagram



34



In a passive system, the intrusion detection system (IDS) sensor detects a potential security breach, logs the information and signals an alert on the console or owner. In a reactive system, also known as an intrusion prevention system (IPS), the IPS auto-responds to the suspicious activity by resetting the connection or by reprogramming the firewall to block network traffic from the suspected malicious source. The term IDPS is commonly used where this can happen automatically or at the command of an operator; systems that both "detect (alert)" and "prevent".

Privacy: a problem

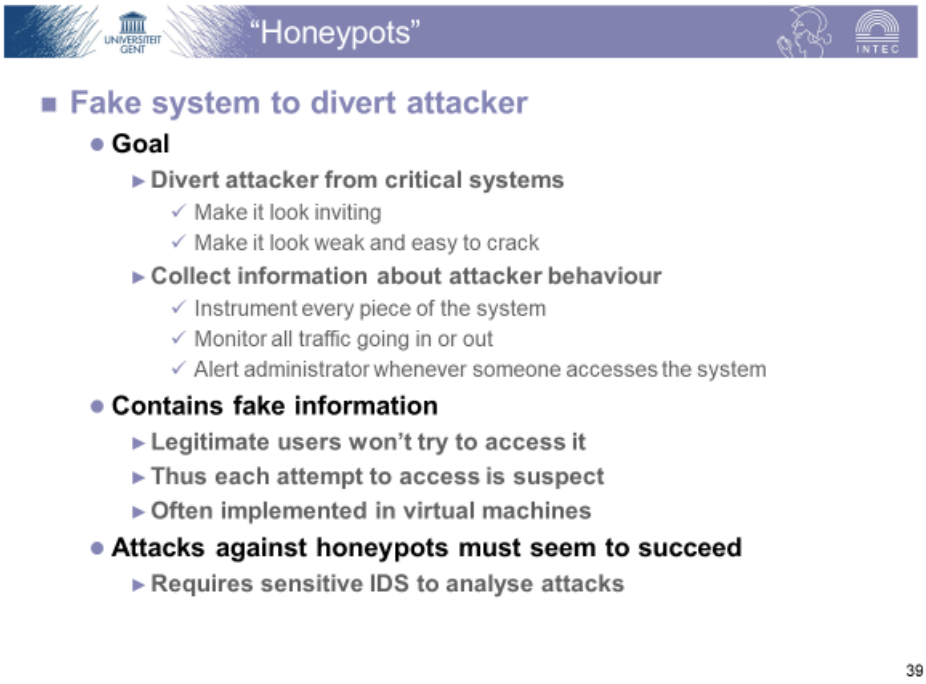
- Some governments/states mandate levels of privacy protection for employees or students
  - This may make it impossible to adequately gather data for the IDS
  - This may make it impossible to gather forensic data for analysis or prosecution
  - Is it prying if it's done by a computer?
    - What if a human never sees it?
    - What if the information is never acted upon?
  - At what point is privacy violated?
    - Looking at packet headers?
    - Looking at packet contents?
    - Looking at /var/mail/user?

37

Overview

- Intrusion detection
  - Introduction
  - Audits
  - Practical approaches
  - Honeypots
  - Limitations

38

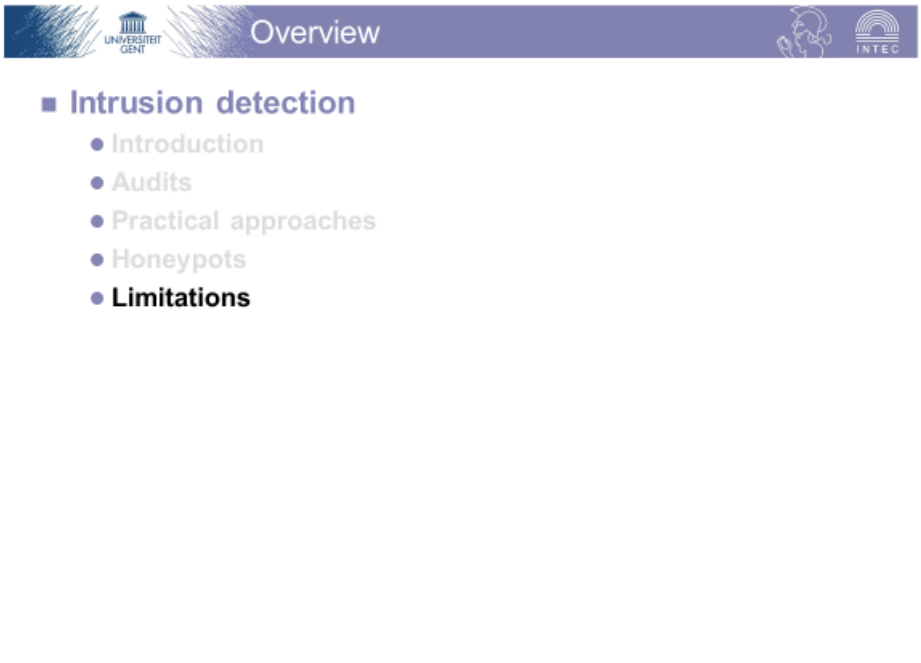


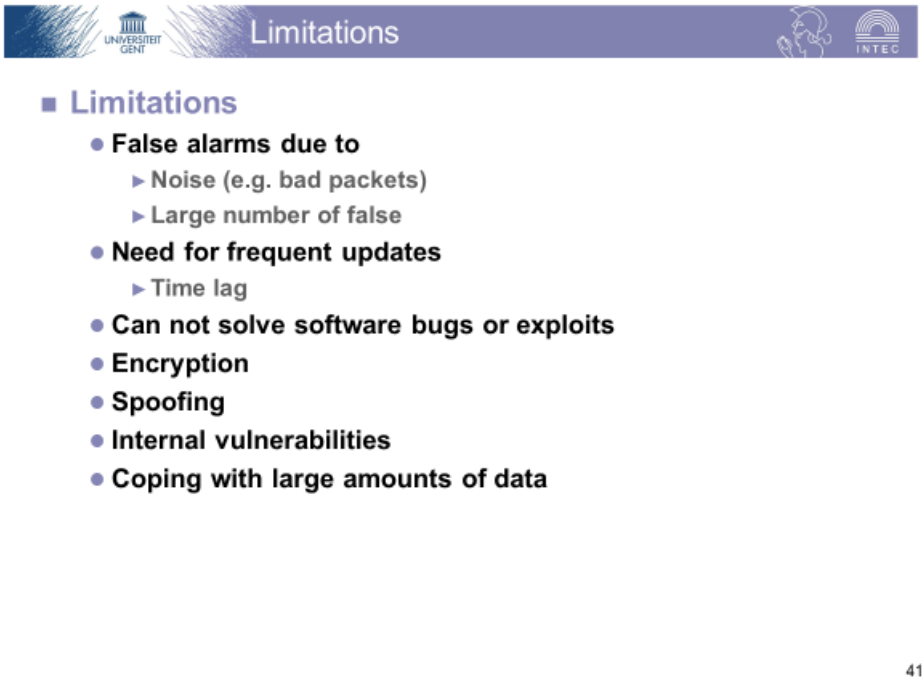
A honeypot is a computer security mechanism set to detect, deflect, or, in some manner, counteract attempts at unauthorized use of information systems. Generally, a honeypot consists of data (for example, in a network site) that appears to be a legitimate part of the site but is actually isolated and monitored, and that seems to contain information or a resource of value to attackers, which are then blocked. This is similar to the police baiting a criminal and then conducting undercover surveillance, and finally punishing the criminal. It should be noted that honeypots aren't perfect either. Meanwhile intruders have developed techniques to identify and to avoid honeypots. Here too, an arms race exists between attackers and defenders in the domain of information security.

Some example ways intruders might detect a simple honeypot include the following.

- The machine looks like it was just set up yesterday and the only thing it has on it besides default directories is a folder called "Sensitive" filled with page scans of old copies of 2600 and lists of misspelled names and address purporting to be employees of HB Gary.
- The mouse driver has the manufacturer labeled as "Microsoft SMS Solutions"
- You try to talk to the drive controller or any other DMA device and the computer begins responding like it had a lobotomy.
- The CPUID op code places value 0x02 in EAX
- You do an RDTSC timing on an instruction sequence and the resulting value is some insane number.
- You try to make an HTTP connection to cnn.com and get the error "cannot connect"
- The only printers installed on the machine have the word "generic" in their name.
- You give the command "net view" and get back the response "The list of servers for this workgroup is not currently available."

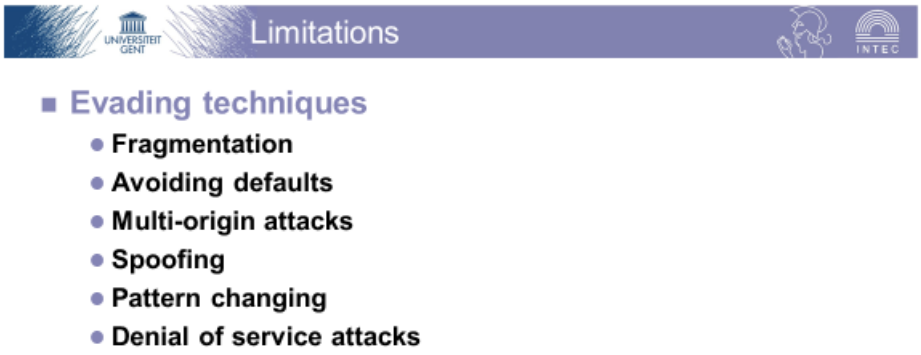
Note that more advanced honeypots are not so easily recognizable.





- Noise can severely limit an intrusion detection system's effectiveness. Bad packets generated from software bugs, corrupt DNS data, and local packets that escaped can create a significantly high false-alarm rate.
- It is not uncommon for the number of real attacks to be far below the number of false-alarms. Number of real attacks is often so far below the number of false-alarms that the real attacks are often missed and ignored.
- Many attacks are geared for specific versions of software that are usually outdated. A constantly changing library of signatures is needed to mitigate threats. Outdated signature databases can leave the IDS vulnerable to newer strategies.

- For signature-based IDSes there will be lag between a new threat discovery and its signature being applied to the IDS. During this lag time the IDS will be unable to identify the threat.
- It cannot compensate for a weak identification and authentication mechanisms or for weaknesses in network protocols. When an attacker gains access due to weak authentication mechanism then IDS cannot prevent the adversary from any malpractice.
- Encrypted packets are not processed by the intrusion detection software. Therefore, the encrypted packet can allow an intrusion to the network that is undiscovered until more significant network intrusions have occurred.
- Intrusion detection software provides information based on the network address that is associated with the IP packet that is sent into the network. This is beneficial if the network address contained in the IP packet is accurate. However, the address that is contained in the IP packet could be faked or scrambled.
- Due to the nature of NIDS systems, and the need for them to analyze protocols as they are captured, NIDS systems can be susceptible to same protocol based attacks that network hosts may be vulnerable. Invalid data and TCP/IP stack attacks may cause a NIDS to crash.



There are several techniques which attackers are using, the following are considered ‘simple’ measures which can be taken to evade IDS:

- Fragmentation: by sending fragmented packets, the attacker will be under the radar and can easily bypass the detection system's ability to detect the attack signature.
- Avoiding defaults: The TCP port utilized by a protocol does not always provide an indication to the protocol which is being transported. For example, an IDS may expect to detect a trojan on port 12345. If an attacker had reconfigured it to use a different port the IDS may not be able to detect the presence of the trojan.
- Coordinated, low-bandwidth attacks: coordinating a scan among numerous attackers (or agents) and allocating different ports or hosts to different attackers makes it difficult for the IDS to correlate the captured packets and deduce that a network scan is in progress.
- Address spoofing/proxying: attackers can increase the difficulty of the ability of Security Administrators to determine the source of the attack by using poorly secured or incorrectly configured proxy servers to bounce an attack. If the source is spoofed and bounced by a server then it makes it very difficult for IDS to detect the origin of the attack.

- Pattern change evasion: IDSs generally rely on ‘pattern matching’ to detect an attack. By changing the data used in the attack slightly, it may be possible to evade detection. For example, an IMAP server may be vulnerable to a buffer overflow, and an IDS is able to detect the attack signature of 10 common attack tools. By modifying the payload sent by the tool, so that it does not resemble the data that the IDS expects, it may be possible to evade detection.