Lecture 07b - Intrusion Detection Systems (IDS)

What is intrusion detection?

- It generall means we have failed...
  - ... but at least we might know why we failed.
  - ... and maybe find out what happened.
  - ... and maybe stop it happening again.
- **Intrusion detection** is detecting the *circumvention of a policy.*
  - In particular security policies but we don't necessarily need to separate them out in particular.
  - This means we failed to enforce a policy.
- As already mentioned, intrusion detection is closely related to system auditin, and underlying both we need to record what is going on in a system.
- **Intrusion detection systems** monitor the behaviour within a system with the mindset that there may have been intrusions.
  - That is, the mechanisms for enforcing our policies might not work all the time.
  - Remember, access control related to both policies and mechanisms.
- We need to distinguish between intrusions and legitimate behaviour.
- Some types of intrusions are significantly more visible than others.

Categories of Attackers

- Attackers roughly fall into 1 of 3 clases.
  - **Clandestine:** these try to avoid the intrusion detection or auditing system.
  - **Masqueraders:** these pretend to be a legitimate user.
    - So if Oscar had guessed Alice's password, Oscar can masquerade as Alice.
  - **Misfeasors:** these are legitimate users who are misuing the privileges they have.
    - These can often be difficult to determine.

Masquerader vs. Misfeasor vs. Legitimate

- All have the password for a legitimate account.
- How can we distinguish between them?
- Maybe on the basis on where or when they log in.
- Likely primarily on the basis of what they do once they are logged in.

Misfeasor? Hard to know...

- As a lecturer, I have access to SMP so I can look up (partial) academic records and contact details for UOW students.
  - *Any* UOW student.

- How would an auditing system determine that I was accessing records inappropriately?
- What records should I be allowed to look at?
  - In other words: how do we define inappropriately? Or is there a subset of records I shouldn't look at even though I have read access?
- Should I be allowed to check the suitability of students enrolled in my subject?
  - But I want to check the suitability of a student who wants to take my subject?
    - And, of course, I don't trust them.
- If I was a degree coordinator too, should I only be allowed to access students within that degree, or within my subjects.
  - But sometimes degree coordinators have to advise students who are considering changing degree.
- What does the university say about this?
  - One policy of relevance is the Acceptable Use policy.
- We would like the IPS to be as automated as possible, but discretion is unlikely to be handled well.

A note on IDS vs IPS

- Intrusion detection can be "extended into prevention."
  - This is active extension is then an Intrusion Prevention System (IPS) or Intrusion Detection and Prevention System (IDPS).
- So, for example, an IPS may detect port scanning, which is allowed behaviour, but may be indicative of attack under way, so may actively work to block the attack.

> IPS technologies are differentiated from IDS technologies by one characteristic: IPS technologies can respond to a detected threat by attempting to prevent it from succeeding.
>
> NIST 800-94 guide

"Normal system behaviour"

- Denning (1987) described three characteristics that allow us to determine systems that are behaving normally, or ones that are not.
  1. Interactions between subjects and objects follow statistics.
  2. Sequences that are likely to circumvent the policies would not be part of typical sequences of behaviour.
  3. Processes have a specific set, or a set of allowed actions.

Models

- Behaviour inconsistent with one or several of those characteristics may be taken as evidence of an attempt of an attack.
- A model or strategy allows us to characterise a sequence of states or actions.
- We have two main strategies:
  - The first is **anomaly detection**, for when observed behaviour differes from the typical behaviour for a user.

    - This requires statistics on typical user behaviour, for individual users.
  - The second is **signature** or **misuse based**, when observed behaviour indicates an attempt to inappropriately use resources.
    - This requires that we know typical attack vectors or paths.
- Attack tools such as operating system specific rootkits can, for example, be used to run different versions of standard software, such as `ls`.
  - These variants would typically breach point 3 of our "normal behaviour."
- There is a third strategy, "**Stateful Protocol Analysis**."
  - This is tied to the principle underlying the user of Stateful Inspection Firewalls, that it is possible to identify appropriate behaviour in the context of a particular protocol.
  - It's effectively a subset of anomaly detection that uses vendor supplied "profiles of benign protocol traffic."
    - The distinction is this used universal vendor traffic, whereas generally anomaly detection is based on the organisation specific traffic.

Goals of Intrusion Detection

- Before we look at strategies in more detail it's useful to identify the goals of intrusion detection systems, and to talk about accuracy.
  1. **Detect** all intrusions, internal and external... meaning it needs to be;
  2. **Dynamic** in the sense of capable learning of taking into account current attacks and/or user behaviour. It also needs to be...
  3. **Timely**, in the sense of providing information at a point in time at which it is still useful. This isn't very useful though unless we have...
  4. **Clear and consie reports** of the results of the analysis. And then there are going to be problems unless the reports are...
  5. **Accurate** and known to be accurate. An incorrect report isn't good, but a correct report that we don't believe isn't good either.
- Note: I guess we could split off **assurance** as a sixth point, the "known to be accurate."

False Positives and negatives...

- These are to do with the likelihood of getting a result which is wrong.
- The interpretation in the context of intrusion detection systems is a little different from that of authentication.
- A *false positive* is when we make a match but "shouldn't have."
  - In an intrusion detection system we will be aware of a false positive, which we wouldn't have been with authentication in teh context of a false positive.
- A *false negative* is when we don't make a match but "should have."
  - This time it's the intrusion detection system where we are unaware, and the authentication system where we are aware.

Anomaly Intrusion Detection

- We are going to illustrate this in terms of the user behaviour aspect, although generally it is simply the system behaving differently.
  - We could represent these notes in terms of system characteristics.
- We need a way to characterise user behaviour, in a measurable way.
- We can look at measurements associated with individual activities, or with many activities, and we can also look at the actual mix of activities too.
- Types of Anomaly Detection
  - **Statistical:** compares behaviour against measured statistical behaviour.
  - **Knowledge based:** uses an expert system with reference to a set of rules modelling legitimate behaviour.
  - **Machine learning:** uses suitable training data to determine a classification model.
- We are going to focus on statistical anomaly detection.
- Typically the measurables for individual activities can be characterised by the time density and event length distributions.
  - How many at a time? And each for how long?
- Measure might include CPU usage, number of processes, number of files opened, typically length of file open time, or process run time.
- Depending on the user, some of the characteristics may be more significant than others, even within the same system.
- In terms of activity mixes, this might be something distinguishing between something like:

  ```
  edit file, close file, compile file, run process, edit file, close file, compile file, run...
  ```

  and...

  ```
  edit file, edit file, edit file, edit file, edit file, edit file, edit file,...
  ```

Statistics: A Threshold Model

- The simplest statistical model for modelling anomalies is a threshold model.
  - If more than a certain number of something happens an alarm is triggered. Or
  - If less than a certain number of something happens an alarm is triggered.
- Login attempts for example

Mean and Standard Deviation

- The threshold can be turned into slightly more than a count, particularly if we are dealing with a continuous measure rather than a discrete one.
- Effectively, we can specify an anomaly as being a value more than a certain number of standard deviations from the mean.

The χ^2-test

- We can use this "goodness-of-fit" test to determine the consistency of a frequency distribution, a sample, with a probability distribution, our theory of the population statistics.
  - The number of occurrences of outcome x_i: N_i
  - The number of trials: n.
  - The probability of outcome x_i in any given trial: p_i
- Two versions are used:

$$\chi^2 = \sum_i \frac{(N_i - np_i)^2}{np_i}$$

$$= \sum_i \frac{{N_i}^2}{np_i} - n$$

How good?

- A large goodness-of-fit statistic indicates a lack of agreement between the observed and expected frequencies.
- An unusually small goodness-of-fit statistic may indicate that the data may have been 'rigged', which is also of interest.
- For k possible outcomes we consider a fit to be poor if:

$$\chi^2 \geq k - 1 + 2\sqrt{2(k-1)}$$

- A poor fit, or a rigged fit, could be taken as the basis for raising an alarm.
- To be more precise the value determined allows a confident level to be specified, based on the number of degrees of freedom $(k-1)$.
- A p value of, 0.3 for example, would mean there is a 30% deviations are due to chance only.
  - In this table if the measured $p > 0.05$ you would accept the sample as consistent with the population.

Degrees of Freedom (*df*)	**Probability (p)**										
	0.95	0.90	0.80	0.70	0.50	0.30	0.20	0.10	0.05	0.01	0.001
1	0.004	0.02	0.06	0.15	0.46	1.07	1.64	2.71	3.84	6.64	10.83
2	0.10	0.21	0.45	0.71	1.39	2.41	3.22	4.60	5.99	9.21	13.82
3	0.35	0.58	1.01	1.42	2.37	3.66	4.64	6.25	7.82	11.34	16.27
4	0.71	1.06	1.65	2.20	3.36	4.88	5.99	7.78	9.49	13.28	18.47
5	1.14	1.61	2.34	3.00	4.35	6.06	7.29	9.24	11.07	15.09	20.52
6	1.63	2.20	3.07	3.83	5.35	7.23	8.56	10.64	12.59	16.81	22.46
7	2.17	2.83	3.82	4.67	6.35	8.38	9.80	12.02	14.07	18.48	24.32
8	2.73	3.49	4.59	5.53	7.34	9.52	11.03	13.36	15.51	20.09	26.12
9	3.32	4.17	5.38	6.39	8.34	10.66	12.24	14.68	16.92	21.67	27.88
10	3.94	4.86	6.18	7.27	9.34	11.78	13.44	15.99	18.31	23.21	29.59
	Nonsignificant								Significant		

Multiple variables:

- A collection of measures, $(m_1,\ m_2, \ldots,\ m_n)$ and associated weights $(w_1,\ w_2, \ldots,\ w_n)$.
- Each user has an appropriately determined base profile $(M_1,\ M_2, \ldots, M_n)$.
- To process a system we measure the active profile $(\mu_1,\ \mu_2, \ldots,\ \mu_n)$
- We then apply a collection of distance functions D_i to determine $d_i = D_i(M_i,\ \mu_i)$.
- Our decision can then be based on a threshold, either at the level of individual distances, or in terms of a composite threshold.

$$\sum_{i=1}^{n} w_i d_i \leq d_t$$

- It could also be possible to cross-correlate the distance functions.
- The distance functions themselves are functions of the probability distributions associated with the user profile and their actual profile, that is the result of observations.
- Here goes an example:

Event	Average	Stdev	Weight
Logins	4.50	1.25	2
Total time online	287.15	42.12	1
Emails sent	65.40	30.71	1
Orders processed	150.73	20.13	1
Pizza's ordered online	2.03	1.06	0.5

Event	Day 1	Day 2
Logins	7	5
Total time online	300	280
Emails sent	60	75
Orders processed	170	190
Pizza's ordered online	2	4

Calculate $\frac{(7-4.5)}{1.25}=2$

So logins contribute $2 * 2 = 4$.

- Another modelling technique used in anomaly detection involves Markov models and predictive patterns.
  - Here, we are looking for unlikely events.

Derivation of Statistics

- What are reasonable values?
  - Abnormal implies there exists a normal value.
- IDES assumes a Gaussian distribution associated with normality.
- Experience indicates that the distribution is not typically Gaussian.
- We would expect that the statistics are based on measurements of users behaviour.

Aging of data...

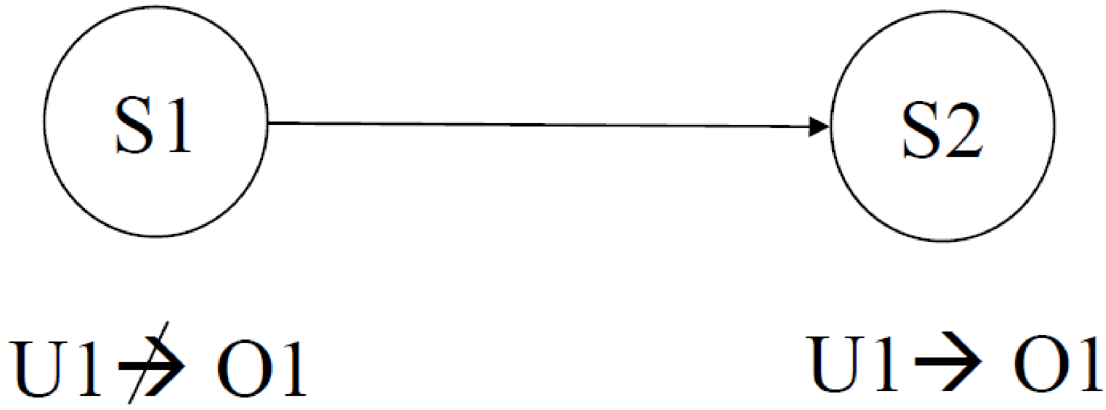
- We shouldn't really rely heavily on old statistics.
- If we are accumulating data over a significant period of time, and taking it all into account, we should weight the data as a function of time.
- Example - Intrusion Detection Expert System (IDES)
  - In IDES subjects: a user, a login session, applications, routers, ..., are represented as an ordered sequence of statistics:
    - $(q_{0,j}),\ \ldots,\ q_{n,j}$
    - $q_{i,j}$ is the the i^{th} statistic on day j.
    - Metrics are counts or time intervals.
    - The profile for each subject is updated every day.
  - IDES weights statistics to favour recent behaviour over past behaviour.
  - Let $A_{k,m}$ be the summation of counts making up the metric for the k^{th} statistic on day m.
    - That is, the total number of events contributing to the k^{th} statistic up to and including day m.
  - The statistic becomes...

$$q_{k,m+1} = A_{k,m+1} - A_{k,m} + 2^{-rt} q_{k,m}$$

    - t is the number of log entries or the total time elapsed since time 0.
    - r is a half-life determined through experience.
  - The exponential decay of previous values models the sensitivity to changes in behaviour over time.

Signature Based Intrusion Detection

- These look for specific sequences of events, or resource usages, or some complex condition sets, that describe a known attack.
- Such patterns are sometimes referred to as intrusion signatures.
- Once an attack has been attempted a signature can be extracted and used to detect later instances of the same attack.
- Misused modelling usually refers to misuse from *within* the system by an authorised user.
- If a set of actions by a user matches to a set of known rules, an intrusion is reported.
- Expert systems are often used to analyse the data.
- Some examples are:
  - **Intrusion Detection In Our Time (IDIOT)**
    - Monitors audit logs and looks for sequences of events that correspond to an attack.
  - **STAT**
    - This monitors the security state of the system.
    - If it goes from a less privileged state to a more privileged state, the way this transition has occurred is monitored.



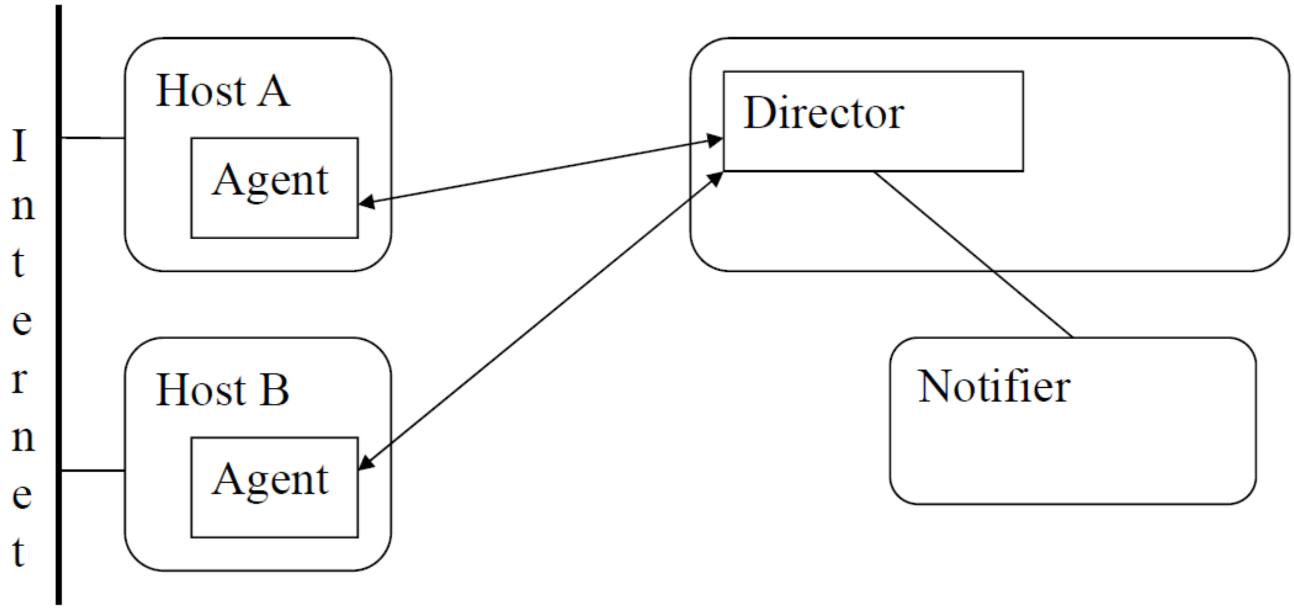
- **Network Flight Recorder (NFR)**
  - This intrusion detect tool takes packets from the network and filters them.
  - A backend writes the information generated by these filters to disk.
  - Administrators can then query this backend without impacting network performance.

Signature: Misuse versus Specification

- Misuse modelling looks for states that are known to be bad, specification based-modelling looks for states not known to be good.
  - Black list vs. white list.
- Specification-based detection: determines whether a set of instructions violates the specifications of the system.
- Specification based detection relies on traces or sequences of events.
- A trace policy takes a set of selection expressions and applies them to the system trace of interest.
  - E.g. if a filter is a tuple (proc, prog, host, user)...
  - (any, vi, any, john) will produce a list of subjects with program vi and user john.

Architecture of an IDS/IDPS

- The architecture involves:
  - **Agents:** gather information from loggers/sensors and likely to perform some analysis.
  - **A Director:** gather information from agents and perform some analysis.
  - **Notifier**



Agents

- An agent collects data from sources, including log files, a network or other processes.
- The agents pre-process information before it gets to the director.
  - For example, agents may transfer failed login attempts from a log, and discard successful logins.
- The director can, however, request more information from the agents.
  - This is particularly important where the correlation between events across agent domains is significant.

How do agents gather information?

- Host-Based information gathering (HIDS/HIDPS):
  - Looks primarily at log files, in the context of a host and likely particular applications.
  - Analyse them to determine what to pass to the director.
    - The events to look for are those relating to the goals the IDS.
- Network-Based information gathering (NIDS/NIDPS):
  - Monitor network traffic.
  - Detect network-oriented attacks such as DoS.
  - It is impossible if network traffic is encrypted.

Combining Sources

- In Unix, an application level log will be significantly different to a system level log.
- From the system level view, application logs are insufficient.
- From the application level view, system logs are insufficient.
- An agent, or the director, determines the appropriate level, or maps the information to that level.
- One advantage of combining information from multiple sources is the possibility of anomaly detection, used in anomaly based IDS.
  - HIDS and NIDS use misuse detection.

- Consider this example of correlating information from multiple logs:
  - Events:
    - Alice normally logs in during the day to perform system maintenance.
    - Alice occasionally logs in during the late evening to write reports.
    - One day, Alice logs in during the late evening and changes the system kernel.
  - The agent provides logs of both login times, and the commands executed separately.
  - All appear normal, but looking at them all together we see anomalous behaviour.

Why use Agents?

- Traditional IDS have one point of failure, the director.
- To overcome this we need an IDS where multiple components would function independently, yet still be able to correlate information.
- If one agent has been attacked, the others would still continue to monitor the network.
- What is the cost?
  - There will be an overhead in the communication.

Director

- The director further analyses information, using an analysis engine.
- The director can instruct the agents to:
  - Collect or send more information.
  - Process data differently.
- The director usually requests more information when it detects an attack.
- An agent can obtain information from a set of hosts, in this case it can act as a director with respect to those hosts.
- A director usually runs on a different system, so attackers can't compromise it at the same time as the system we are trying to protect.

Notifier

- Notifies the appropriate party regarding reports received from the director.
  - Email would be a standard reporting mechanism.
- Graph-oriented user interface.
- The notifier can be responsible for coordinating IDS residing on firewalls to block attacks over the network.
- If an attack is identified the notifier can instruct other IDS's to counteract the attack.

Combining Host and Network Monitoring

- The Distributed Intrusion Detection System (DIDS) was one of the early models, developed in the late 1980's, early 1990's by the US Department of Defence.
  - It is an example of a combined HIDS-NIDS.
- Host only or network only monitoring tends to be ineffective.

- Logging into a system without a password wouldn't be detected by network monitoring.
    - However, subsequent actions may be detected by host monitoring.
- A DIDS director monitors agents that are attached to hosts as well as monitoring network traffic.
- Agents scan logs for events of interest and report them to the DIDS director.
- The DIDS director performs analysis using an expert system.
- Note that combining HIDS and NIDS pretty much implies a distributed system.

DIDS Director Processing

1. Raw data
2. Relevant entries are extracted
3. Captures all events related to a single user.
4. Adds contextual information, such as the proximity of other events.
5. Looks at events in relation to context, thus identifies threats and suspicious activities.
6. Assigns a score derived from layer 5 to the security of the network.

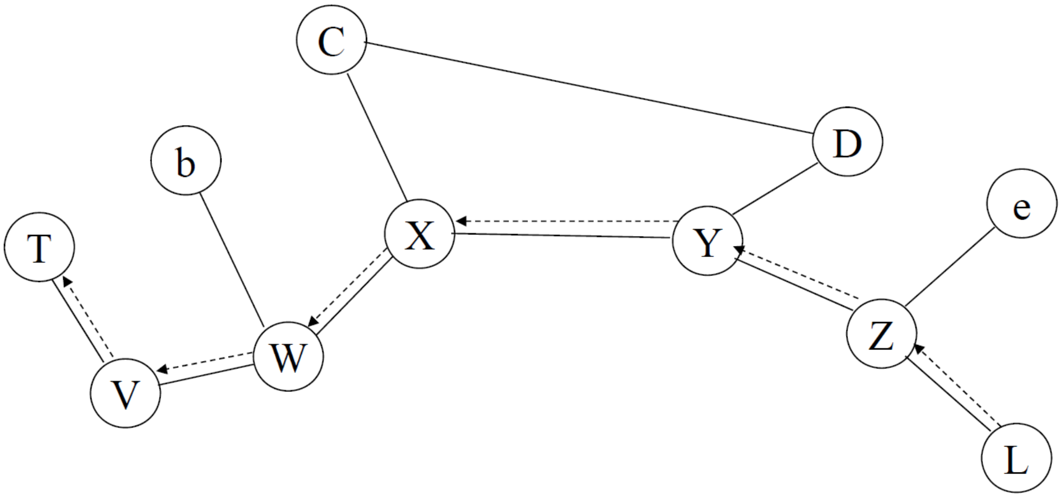
Intrusion Response

- Incident Prevention and Handling:
    - If we can we would like to try and prevent the intrusion from succeeding!
    - But we may not be able to, and appropriate handling requires that various mechanisms be in place:

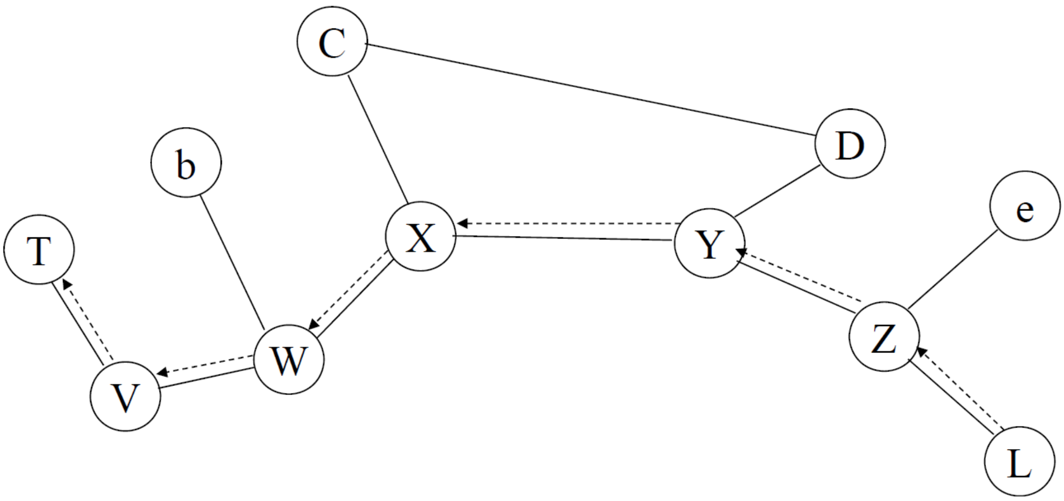
1. **Preparation for an attack**
    - Procedures and mechanisms for detecting and responding to attack must be set up before any attacks are detected.
    - Effectively you need to know in advance what to do if an attack is identified.
2. **Identification of an attack**
    - When an attack is identified the rest of the phases come into play.
3. **Containment of the attack**
    - We can try and limit access of the attacker to the system:
        - Possibly we could constrain the attacker...
        - Try and prevent the attacker from reaching their goal, if we can identify it.
        - Maybe use a "honeypot" to lead the attacker to another part of the system.
    - Passive monitoring:
        - Do nothing, but find out what the attacker is trying to do.
        - Possibly switch to more active measures later.
4. **Eradication of the attack**
    - Stop the attack:
        - Probably the easiest thing to do is to terminate the network connection or stop the process.
    - We could use a firewall to block the attacker.
    - The Intrusion Detection and Isolation Protocol (IDP) can also block connections.
        - Kahn and Zurko proposed the use of IDIP as a mechanism for blocking network flooding attacks which would result in denial of service.

- Consider a flooding attack launched against host T, by host L, along the path:

  $L \rightarrow Z \rightarrow Y \rightarrow X \rightarrow W \rightarrow V \rightarrow T$
- The flood effectively stops all traffic along that path.
  - V detects the flood, blocks traffic for T, and notifies W.
  - W detects traffic targeting T, stops it, and notifies X.



  - X detects the traffic targeting T, stops it, and notifies Y and C.
    - We refer to $C,\ D,\ W,\ X,\ Y$ as boundary controllers, they can block connections from crossing a boundary.
    - Host V is running IDIP, but isn't a boundary controller.
  - W notices the traffic for T has stopped, and it eliminates its suppression.
  - T, V, W, and b can now communicate freely again, because X has blocked the flooding.
  - Y detects the flooding, stops it and informs Z & D.
  - X detects that the traffic going through it for T has stopped, and X stops its suppression.
  - Z detects the flooding and suppresses the flooding traffic.
  - This process continues until all traffic from L to T is suppressed.



5. **Recovery from the attack**
   - We restore the system to a secure state.

6. **Follow-up to the attack**
    - Take some action external to the system against the attacker.
    - Pursue legal action against the attacker.
    - Two methods to identify attacker:
        - Thumb printing.
        - IP Header Marking.
    - Counter-attacking:
        - Only used in exceptional circumstances.

IDS Products

- SNORT: widely used...
- Anzen Flight Jacket
- Axent/Symantec NetProwler and Intruder Alert
- Check Point IPS-1 (Intrusion Prevention System)
- Cisco Secure IDS
- CyberSafe Centrax IDS
- Endian Firewall
- Enterasys Dragon IDS
- ISS RealSecure
- Network ICE BlackICE Sentry
- Untagle

SNORT: Signature Based

- Open source - freeware, but based at SourceFire where there are quite a few professional developers.
- Provides intrusion detection on Windows, Linux, and Unix.
- Requires support of WinPcap 3.1? (Current 4.1.3)
- SNORT allows sets of rules used for detection and logs:
    - Packet information.
    - Alerts
- To set something up at home ... https://techanarchy.net/2015/01/home-ids-with-snort-and-snorby/

Honeypots

- Honeypots can divert attackers from a critical system, and/or collect information about the attacker's activity.
- They can encourage the attacker to stay on the system long enough for administrators to respond.
    - We can fill a honeypot with fabricated information designed to appear valuable.
    - A legitimate user of the system will not access that location.
    - We fill the system with sensitive monitors and wait!
- Padded cells are hardened honeypots!
- A honeypot can be an individual computer, a dataset or a whole network.
    - How can it be a dataset?

        - In a medical database we could make-up some entries that shouldn't be accessed since they don't correspond to real people.
            - This could play havoc with statistical analysis of the database though so we need to be careful.
- An approximate non-digital analogy corresponds to deliberately leaving the keys in your not-especially valuable car so it gets stolen, ...
- ... but putting a tracker in too so you can find out where all the stolen cars get taken.
    - Actually putting "invisible" trackers in cars is done (Lojack tags).
- These examples really correspond more closely to related concepts...

... related concepts

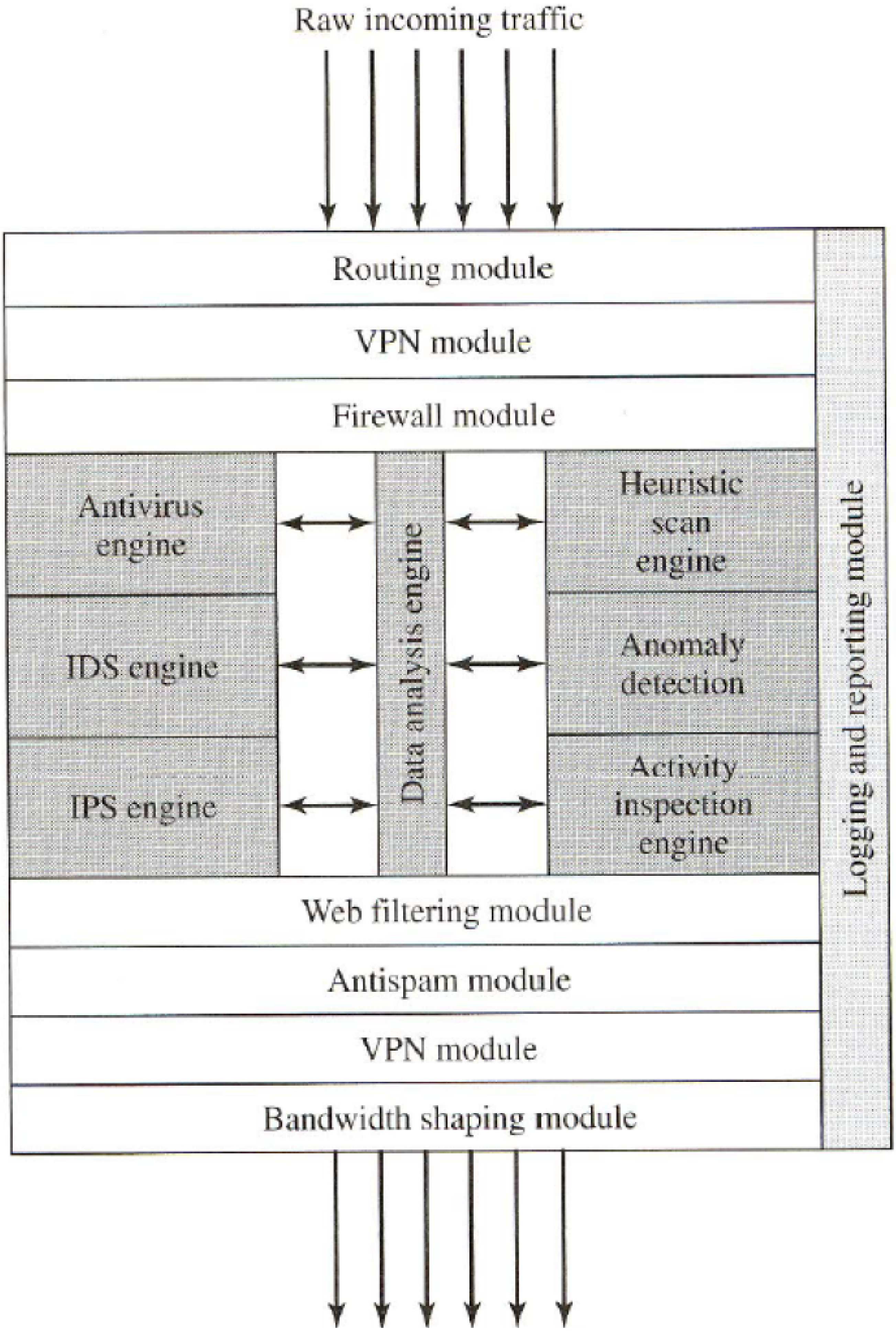
- A **honeytoken** is a non-computer honeypot.
    - It could be a fake account, the password for which has been left somewhere by "accident."
    - ... or fake data in a database which wouldn't normally be accessed.
- A **honeynet** is a collection of two or more honeypots.
    - They can be deployed in such a way as to complement each other, for example, they could run under different operating systems but interact with common databases.

Intrusion Prevention Systems

- Obviously rather than just detecting intrusions and working out what happened afterwards, we often want to prevent traffic from getting into our system.
- So, we use intrusion prevention systems.
    - Like firewalls, which will be looked at soon!
    - If you look at what we have just described, it should be clear that we are really talking about access control (again!)

Unified Threat Management Systems

- Since about 2004, the deployment of the mechanisms that we have described somewhat independently, has started to become unified.
    - In unified threat management systems (UTM).
- So we might have **firewalls**, **intrusion detection intrusion prevention**, **gateway anti-virus**, VPN, anti-spam, ..., features together.
- In addition to the consistency advantages, this simplifies the management and maintenance, since it's effectively a single device/software element from a single provider.
- There are a couple of significant disadvantages:
    - It may be a bottleneck in terms of processing and bandwidth.
    - There is a single point of failure.
        - If the UTM is compromised, you are really, really, in trouble.


Raw incoming traffic
Routing module
VPN module
Firewall module
Antivirus engine
IDS engine
IPS engine
Data analysis engine
Heuristic scan engine
Anomaly detection
Activity inspection engine
Web filtering module
Antispam module
VPN module
Bandwidth shaping module
Logging and reporting module

Clean controlled traffic

Figure 9.5 Unified Threat Management Appliance
Source: Based on [JAME06].