1 Intrusion Detection System deployment Methodology

Its almost ten years since the firewall became the foremost security tool for network administrators. And for the last five years, most people agreed that it was not enough.

Increasingly IDS system have been used to augment security, and they can make a significant contribution to the security regime when deployed correctly.

They enhance security because:

- IDS provide detection to the firewall regime which is mainly prevention based.
- IDS add the inspection of application data and session data whilst firewalls concentrate on network protocol exposures.
- IDS can aid in the processing of log data, which needs to be inspected.

IDS can be ineffective because:

- They are typically installed by VARS who don't really understand them
- They aren't tailored to fit the environment to detect unauthorised traffic
- They overloaded by inappropriate signature
- They are not linked to manual procedures.

Today's businesses rely on their networks to provide vital and sensitive information to where it is needed. IDS can help companies achieve this but only if they are implement correctly. This methodology aids this.

2 My Methodology

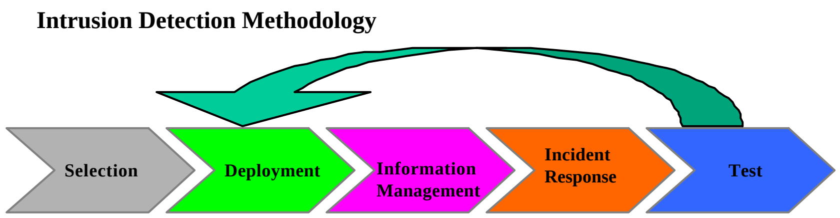


Figure 1 Diagram of the IDS methodology

The methodology shown in Figure 1, comprises a number of identifiable steps.

1 Selection

2 Deployment

3 Information Management

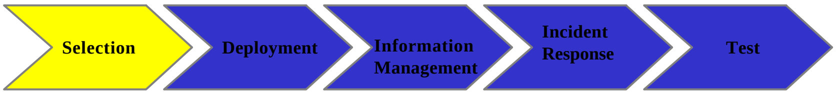
4 Incident response

5 Testing

This methodology is flexible and it can be used with a product has already been selected or a *Greenfield* situation. It was designed for a typical E-banking point solution and has been used successful in an enterprise wide "FIXED Perimeter" situation. In a flexible "holey perimeter" situation (see IDS implementation strategies www.loud-fat-bloke.co.uk) some alteration may be required.

This document covers all these stages briefly, but as this is a deployment methodology it concentrates on stage 2, deployment.

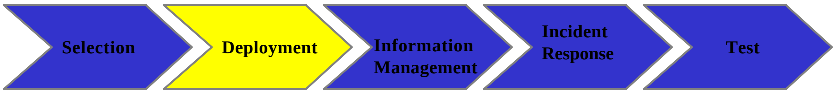
3 Selection



This is not a security product selection methodology – however it might be worth mentioning some of the Key selection criteria which may be used:

- Type of systems architecture in use – Most IDS work with common Unix or Windows – but if a significant part of infrastructure is based on mainframe computing there may be an issue that the IDS does not cover your major assets.
- Type of network architecture in use – Again most IDS work with 10 BaseT Ethernet but Today's businesses networks include ATM, Gigabit Ethernet or Token ring. Many products do not support ATM or Token ring (including one produced by a Token ring manufacturer!!). Can it be adapted to work with an increasingly common switched back-bone. The use of TAP technology is effective in small networks but soon becomes costly and administrative overhead. Spanning ports are limited to about half-a-dozen ports.
- Ease of customisation – this particularly important as a good IDS should be able to interface with a network management system like Tivoli, Openview or Unicenter. It must also be able to receive messages from applications and communicate with unusual devices.
- Need for customisation – Some seem to arrive in kit form and won't interact with firewalls or routers
- Deployment platform – Is a firmware appliance version available. Is one required due to unattended operation or like of onsite system skills etc.
- Scalability – two key features have to be considered here – the ability of the console two manage more than say twenty data collectors and the ability of the database to store the data.

4 Deployment



4.1 Background

Sensors are the key elements of the IDS system, which are capable of identifying patterns of suspicious network traffic and questionable user activities. The effectiveness of the sensors depends on their internal design and, even more importantly, on their position within the corporate architecture.

Generally, sensors can be classified into two categories (Network sensors and Host sensors)

4.1.1 Network Sensors

Network sensors monitor a defined network segment. Such sensors use a knowledge base with known attack signatures. They are capable of detecting a range of network attacks (e.g. port scans, SMB probes, etc). Such sensors, are placed in critical points within the network architecture that allow them to actively monitor and identify malicious network traffic.

4.1.2 Host Sensors

The knowledge base of such sensors consists of suspicious and potentially abusive computer usage patterns which are performed at a local host level (e.g. attempts to delete log files, deleting system files, etc). They are also able to generate adaptive rules for each user. Initially they establish the normal usage patterns (e.g. types and mix of jobs being run, normal usage hours, etc). Given these usage patterns any activity that falls outside the norms will be considered suspicious.

For the purposes of this report we will initially focus on network sensors and then cover the deployment of host sensors .

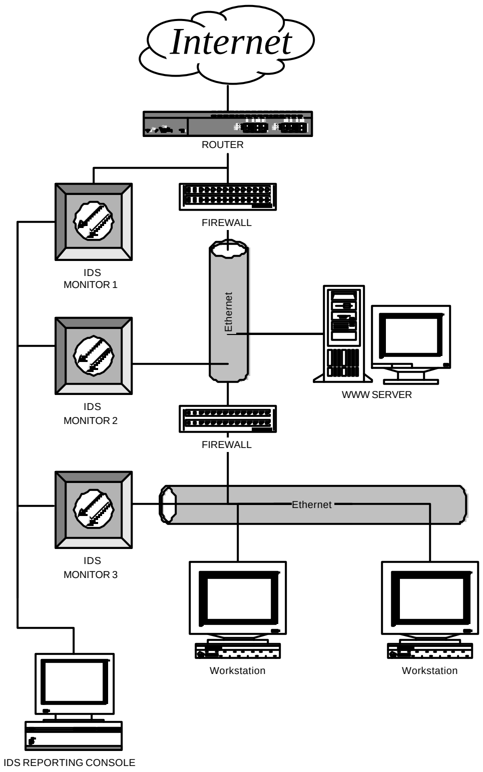


Figure 2 IDS Sensors in a typical corporate network

4.2 Step 1: Planning Sensor Position and Assigning positional risk

As described above, Network sensors monitor a defined network segment. The positions they are deployed in are determined by two fundaments:

- Reason 1 - The network segment contains assets that require protection and are at risk from attack;
- Reason 2 - The network segment would give a sensor the ability to predict an attack or defend against an attack.

Figure 2 shows three classic positions of the IDS sensors.

Sensor 2 – This is the ideal position for a sensor. The network segment it is on contains servers that require protection (reason 1). However, the DMZ is traditionally considered as an intermediate stepping-stone to the main network – correspondingly, a sensor could be justly positioned for pre-emptive reasons (reason 2). *Sensor 3* is justified by reason 1 entirely. Sensor 1 is justified by reason 2 and probably provides no more security functionality than the firewall logging and alerting functions already provide.

WARNING *Many people suggest placing Sensors in front of the firewall. This is because, they suggest, it is important to know what attacks are being thrown at you but are blocked by the firewall. This is not a security reason, it is PROBABLY not your job to police the internet and report on breadth and variety of attacks. However, Such information might be extremely useful to have for any boardroom battles or cost justification, as the comparison of before firewall stats against those from the sensors behind it carry a powerful message. However, in such a position it is very likely that it will causes dozens of false alarms from attacks that will not effect you. It is vitally important to configure before firewall sensors so that they just log stats. Perhaps consider a honeypot if you need to monitor hacker behaviour – I believe they are dangerous in a corporate (i.e. great for research environment) because they attract unwanted attention and could lead people to believe that your organisation has weak defences.*

There is a positional threat-rating associated with each sensor depending upon the position within the network. This roughly relates to the value of the assets on the network segment. Correspondingly, an attack detected at sensor position 1 (where there are no assets) would represent a potentially lower threat to the organisation than if it was detected in the DMZ at position 2 after the perimeter router. Similarly, attacks registered at position 3, are very serious as they have bypassed both the perimeter and the internal firewall and are taking place on the internal network where there are high value assets. Table 1, shows the three sensors with their associated risk level.

	SENSOR 1	SENSOR 2	SENSOR 3
POSITIONAL THREAT RATING	LOW	MEDIUM	HIGH

Table 1 The list of three sensors with their associated risk

4.3 Step 2: Establish monitoring policy & Attack Gravity

Each time one of the sensors identifies an intrusion, an alert is generated and reported to the main IDS system. What is classified as an intrusion is controlled by a monitor policy. In order to minimise the detection of false positive alerts and the overhead of the IDS, its is good practice to tailor the policy. This usually consists of first tailoring a provided list of attacks monitored by the sensor so that it is relevant to your environment – i.e. if you are running a Unix environment, it is not good use of your cpu to monitor the network for winnuke or SMB-packets. Secondly, you should modify the detector to detect remarkable traffic. This process is expanded below

1 Attack Signatures:	An attack network traffic like port scanning, Evil-ping or Unicode hack exploit
2 Abnormal Traffic Alert	Traffic that due to the security regime or system environment is suspicious. Typical examples of these may be: ■ An rlogin, a common Unix utility, attempting a session in a windows network environment ■ An ssh session from the perimeter firewall to a webserver, in an environment where console only access is prescribed. ■ An attempted session with two DMZ servers

	using their external addresses in a NAT'ed environment. - Telnet from inside DMZ back into the Internal network - TFTP from Web server out to the internet - Web Browsing from the Firewall None of these events are definite proof of a hacked network - However, in many cases they are indicative of an abnormal event that must require further investigation. It may have only been caused by a new firewall administrator unaware of the rules governing webserver access or it may indicate a failure of the anti-spoofing rules (as in the last case). An investigation of front and rear firewalls will provided detailed information here. On RealSecure for example this can be captured by a "connection event" or some cases a "user event". In Snort, for example, this is a simple to/from rule. In the Cisco product this is a little harder to achieve.

Once a monitoring policy is established, the severity of each alert must be assigned. In order to minimise the detection of false positives, different gravity levels are assigned to the reported alarms. Otherwise, a high number of false positives will lead the operators of the IDS system to ignore its output, which may lead to an actual intrusion being detected but ignored by the operators.

Various network events with their conceived risk level are shown in Table 2.

NETWORK EVENTS	Attack Threat
MSADC or UNICODE HACK	HIGH
Port scanning	MEDIUM
Outdated attempt of a D.O.S. attack	MEDIUM
Telnet attempt	LOW

Table 2 Network activities with their risk level.

A key concept to this methodology is that the overall alert severity depends on the detected attack signature combined with the position of the sensor(s) which raised

ALERT SEVERITY = POSITION THREAT + ATTACK THREAT

However, it must be noted there is a strong degree of subjectivity involved. Table 3, shows a sample set of network activities together with their relationship to the sensor position. Statistics have showed that the greatest security threats come from internal abuse, therefore whenever the IDS sensors are triggered by abnormal pattern in the internal network they are classified as having greater risk level than the other sensors.

NETWORK EVENT	SENSOR POSITION			
		SENSOR 1	SENSOR 2	SENSOR 3
Port scanning	GRAVITY LEVEL	LOW	HIGH	HIGH
Telnet (attempt from outside)		LOW	MEDIUM	HIGH
Outdated attempt of a D.O.S. attack		MEDIUM	HIGH	HIGH

Table 3 Sample set of network activities with their associated gravity level, depending on the position of each sensor

In order to clarify the gravity levels shown in Table 3, we may consider an example. Assuming that a port scan is detected at the internet router (Sensor 1). Although this event does not constitute a high impact attack, it is a common initial sign that a more serious attack will shortly follow. Therefore, it can be characterised as low risk and recorded in a log file. However, if a port scan is detected in the internal network (Sensor 3), this implies that either the attacker has been successful in compromising one or more internal computer systems, or an internal user is trying to abuse the operation of the network. The latter event is clearly more harmful than the former, and therefore it is characterised as high risk.

4.4 Step 3: Reaction

Once the seriousness level of the reported alert has been established, the IDS system can respond by a set of predefined actions. Table 4, shows the associated risks together with suggested alert actions in a typical environment.

EVENT IMPORTANCE	ALERT ACTION
LOW	No further action at this stage – possibly record for forensic use. Example *Log the incident in the log files*.
MEDIUM	Flag the incident, for next working day follow up. Example: *Email firewall administrator*
HIGH	Immediately alert the operator, and/or act against the offensive connection. Example: *Page firewall administrator and sent a TCP/IP RST to initiating tuple.*

Table 4 Proposed actions depending on the level on risk.

When a high level alert is generated, then the IDS system has to immediately inform the operator than an attempt of intrusion has been detected. If the attack is considered to be harmful to the network then the IDS system can actively act against it.

4.5 Step 4: Further Action

4.5.1 Firewalls

Typically, an IDS system can respond by resetting the suspicious connection, by locking-out particular addresses in the firewall or even shutting down the firewall. However care needs to be taken, to ensure that the adverse consequences of the IDS response outbalance the impact of the detected attack.

For example shutting the firewall whenever a portscan is performed on the router, would be considered to have a more severe effect (effectively denial of service) than the original activity.

At this point, you may have a better insight into traffic patterns and can take the opportunity to enhance your firewall rules.

4.5.2 Host detectors

Host detectors can be deployed using the same methodology, it only requires a further iteration:

- To plan which servers to deploy detectors on;
- To establish what the policy should contain etc.

Even if the IDS system in use doesn't support host detectors, normal facilities like *syslogd* and *swatch* can often be deployed to great effect, the messages being interpreted by the NIDS.

Additionally, PortSentry and Tripwire provide a super alternative.

4.5.3 Application Interface

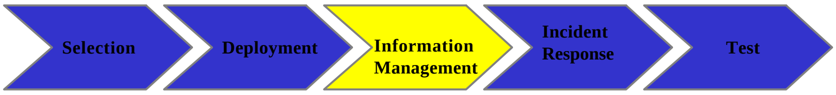
With the advent of E-commerce, more external interaction with applications is encouraged. This exposes the applications to security threats like bruteforce attacks and code manipulation. Therefore, the need for application interfaces into the IDS are becoming more important – after all, often the IDS is purchased to protect one Internet Banking application. Therefore, it is important that the IDS be able to respond, for example, to multiple requests to the bespoke application login function for multiple accounts from one address in a set period of time exactly as commercial IDS respond to repeated standard HTTP authentication requests in the same situation.

This often requires a considerable amount of tailoring but it certainly can provide useful security intelligence. Some products likes Siteminder can do this.

4.5.4 Honeypots

Honeypots are a great tool for research. Production Honeypots, those used to detect hacking traffic can be dangerous and are not included in this methodology.

5 Information Management



This stage is usually very short but is often forgotten. It deals with:

- Where is the information delivered
- What form the information is
- What time frame it is delivered in
- What form it is retained in

5.1 Console & log Management

Once the sensors are configured the IDS will potentially accumulate a huge amount of information. This information must be processed and archived. It is therefore necessary to have Log management procedures. This should define:

- When the log should be archived and cleared down
- How long it should be retained; and
- Who should have access to the information.

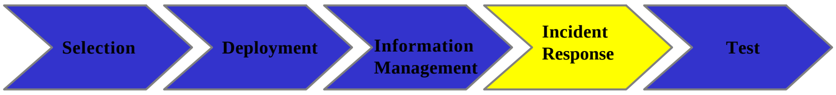
- Are any strategic management trends required from this data.

It is often necessary to purchase additional software to perform this task.

Perhaps more importantly, some consideration should be give to the position of the IDS consol. There are issue of access control (i.e. only security should be able to push a new policy onto a detector but operations or audit may legitimately want access to the alerts and event logs).

For large organisations it is good practice to report all high-severity incidents to an enterprise console which is viewed by operators 24by7. This overcomes the case whereby an attacks occur in the early hours of the morning, when the internet traffic is low, and there are minimal chances that any security staff will notice any irregular activities. Most IDS have interfaces to OpenView, Tivoli and Ca-unicenter.

6 Incident response & crisis management



There is no point in have IDS software installed if no adequate Incident response procedures are present in the organisation.

Key elements of a good Incident response procedures include:

- Early notification of potential events
- Clear escalation procedures with **defined** time limits for duration of each stage
- Automatic percolation up the stages of escalation, reversed only by formal sign-off
- Providing the people on the ground with power to make the decisions

But most of all, they should be written down.

However, designing Incident response & crisis management processes is a very specialised job – out of scope of this document.

7 Test

Selection
Deployment
Information Management
Incident Response
Test

Anything that *has to work to be useful* should be tested - especially in security implementation.

Testing should be conducted at two levels:-

- Technical

Manufacturer make a great many claims about their IDS systems. You need to establish the capability yourself. This should include:

1 Through-put – take the network to 60% utilisation using the tools of your choice(sprayd or observer.pro) – then check for dropped packets. Make sure you devise techniques for detecting dropped packets and, if possible, very unusual traffic rates(even if it is not from the IDS).

2 Avoidance – use commercial software Blade or open-source software like adm-mutate, fragrouter or ITB plus the avoidance modes of whisker or nmap.

- Covert penetration testing

Use a covert penetration testing services to test the configuration of an IDS and the reaction of your staff to the alerts. Usually this will involve, a mixture of off-site and on-site testing. Ensure that you avoid your own qwik-brew Cuban missile crisis by making sure all the right people know. Otherwise you will end up with the bad publicity you are trying to avoid.