# Network Intrusion Detection Systems: Important IDS Network Security Vulnerabilities

*Must Know Information on Realizing and Overcoming the Risks*

*Authored by:*
Simon Edwards
Technical Evangelist
September 2002

This document provides an overview of some of the key problems that arise in Network-based Intrusion Detection System (nIDS) deployments. These deployment issues result in missed intrusions, network degradation, and lost business. The document further demonstrates how a network-based load balancer with session-based flow mirroring developed by Top Layer Networks can overcome many of these deployment pitfalls.

## Executive Summary

The Network-based Intrusion Detection System (nIDS) has become a critical component of an organization's security strategy. However, deployment of network-based intrusion detection brings with it a number of potential pitfalls, which can compromise security. An ideal network-based intrusion detection deployment must provide 100% network intrusion coverage and ensure network availability. The following are areas that are problematic for most IDS deployments.

### 100% Network Intrusion Coverage

Many deployments result in missed intrusions and network vulnerabilities. The environments that are especially susceptible to missed intrusions are:

- **Heavy traffic networks.** In these environments the high amount of traffic overloads the IDS sensor and intrusion traffic is missed.
- **Switched networks.** In these environments a nIDS needs to see the traffic on each switch segment. In switched networks there is no ideal location to connect a nIDS -- and switch SPAN ports can't keep up with all the traffic on the switch. Deploying nIDS on each segment is cost prohibitive in many environments, thereby leaving segments unprotected.
- **Asymmetrical networks.** In asymmetrically routed networks the traffic can traverse multiple paths before it reaches the nIDS and the nIDS will only see parts of the conversation (flow); thus missing an attack. A nIDS needs to see a complete conversation (flow) in order to determine if an intrusion is present.

### Network Availability

Network availability is critical. As businesses scale their networks to accommodate growth and increase resilience, they must also look to scale the security infrastructure to maintain and ensure network availability.

- **Scalability.** As a network infrastructure grows or changes, the nIDS environment must accommodate the growth with minimal cost and change control. In most initial nIDS deployments, scalability is an afterthought, leaving the network open to intrusions and network degradation.
- **Resilience.** Just as redundant network elements are deployed to ensure high network availability, redundant nIDS are required to ensure intrusions don't impact network availability.

### Deployment Issues Resolved – 100% Network Coverage and High Availability

These deployment issues can be addressed with a load-balancing device that understands traffic conversations (flows) and that can intelligently augment and forward traffic to nIDS to ensure that the network intrusions are detected and the network is available.

## 100% Network Intrusion Coverage - Challenges

NIDS deployment issues impact a nIDS ability to detect intrusions and provide 100% coverage against network intrusions. The following sections will overview the challenges with nIDS in terms of performance, switched, and asymmetrical routed networks.

### nIDS Performance Issues

Performance is probably the most controversial issue surrounding nIDS because it is difficult to measure. Here are some of the elements one needs to consider:

- The hardware and software that you're using to run the sensor on (Linux, Windows, Solaris or hybrid)
- whether the nIDS is using Pattern Matching or Protocol Analysis, or a combination of the two
- how much encrypted traffic is running
- what packet sizes you are using and
- what type of policy you are running.
- how many alerts are being generated
- how many responses you are triggering to each alert

There are two mainstream versions of a nIDS available on the market: a 100MB sensor (capable of monitoring up to 100MB/s) and a Gigabit sensor (capable of monitoring anywhere from 300MB to 800MB). Although both types of nIDS use the same high-level logic to look for attacks (either Protocol Analysis, or Pattern Matching, or a combination of both) the major difference between the two is in the method used to analysis the packets once captured.

A major difficulty is that true performance statistics are very hard to obtain, especially in a lab. However a recent test by NSS Labs is probably one of the best (http://www.nss.co.uk/ids/index.htm) The issue is not *how many* attacks that a nIDS can detect that is the most important factor (and often the only bench mark used in lab tests), but *how effectively* the nIDS can pick out one attack in a mass of normal background traffic. It is often not the mass of attacks that a nIDS has problems dealing with, but the proverbial "finding a needle in a haystack". This becomes especially difficult when SSL (Secure Socket Layer) traffic is involved, because the nIDS cannot read encrypted traffic. It wastes valuable CPU cycles *realizing* that it can't do anything with the traffic and then discards it!

A second core performance element to consider is the size of packets. In tests, nIDS vendors usually look at an average packet size of 1024 bytes, however if the packet sizes are smaller, the nIDS will run a lot slower (e.g. consider the negative impact when monitoring a large DNS server).

A third key driver in how fast a nIDS can run is the actual policy that is running on the nIDS. Typically nIDS have hundreds of attack signatures that they are looking for at any given time. The more signatures they are looking for in a stream of data, the longer it will take to look at the next stream. This is more critical for pattern matching based systems than those that utilize protocol analysis.

There are usually too many variables to determine the exact performance of a nIDS, but typically on a 10/100MB sensor one can expect around 60-80MB/s monitored and on a Gigabit sensor it's approximately 400-600MB/s. That's approximately 60-80% utilization on a 100MB segment and around 40-60% utilization on a Gigabit segment.

This makes it appear that one cannot monitor an entire segment. However it's important to remember the environment in which these devices were originally designed -- the important phrase here is *segment*. A nIDS is designed to monitor individual segments, such as off a *hub*. A hub is a device that uses *broadcast* technology (e.g. if you have 5 PCs connected to a hub, and one PC talks to another, all the PCs connected to the hub will hear the communication). So simply connect a nIDS to one of the ports on the hub, and you will be able to monitor all of the communication that goes on in that segment. Typically with this sort of architecture you can expect to achieve a 40% utilization of the 100MB/s available. This means that the 40-60% utilization you achieve on a 100MB sensor is actually not a large problem. The problems really start when you increase this utilization through the use of switches.

**Implementation Difficulties in Switched Environments**

Heavily switched environments cause the biggest headaches for security professionals trying to implement nIDS.

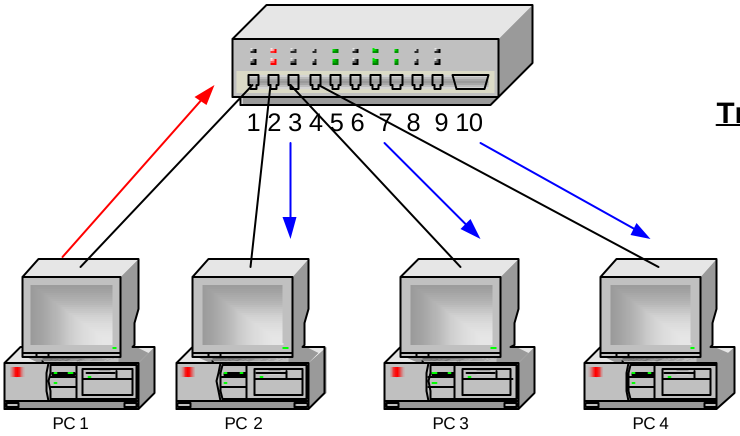


**Diagram One**
**Traffic flows through a hub**

PC1 wants to communicate to PC2, but the request is broadcast on all ports,
so if PC4 was used as a nIDS it would hear all communications

First, let us look at why switches are different. A hub only understands basic Layer 2 information, and broadcasts all packets to all ports, thus only achieving low bandwidths (See above Diagram One). A switch understands Layer 3 & 4 information, and therefore knows the IP address/s of the devices connected to it. For example (as in Diagram Two), if PC 1 wants to talk to PC 2, then the data is only sent down Port 2, and not to the other ports on the switch.

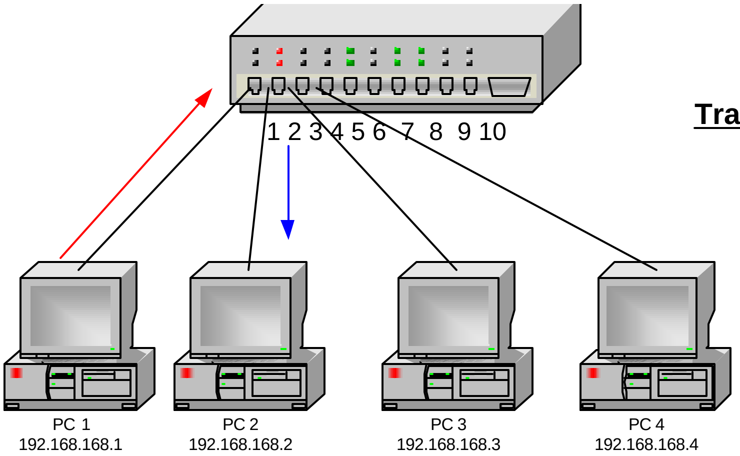


**Diagram Two**
**Traffic flows through a switch**

A switch understands Layer 3 or 4 information (i.e. IP address), so when PC 1 sends a request to PC 2, the switch knows that the packets are deemed for 192.168.168.2 and only sends it there. A nIDS seated on PC 4 would not be able to "hear" the conversation, so not be able to monitor for attacks

The issue is how to connect a nIDS so that it can listen to all the communication on the switch. The answer lies in what Cisco calls *SPAN* ports (www.cisco.com\warp\public\473/41.html) or what other vendors also call *Mirror Ports*. The principal is the same in both. You set one port, on the switch, to take copies of the other traffic from other ports. So in Diagram Three we have set port 10 to mirror all traffic from ports 1-3.

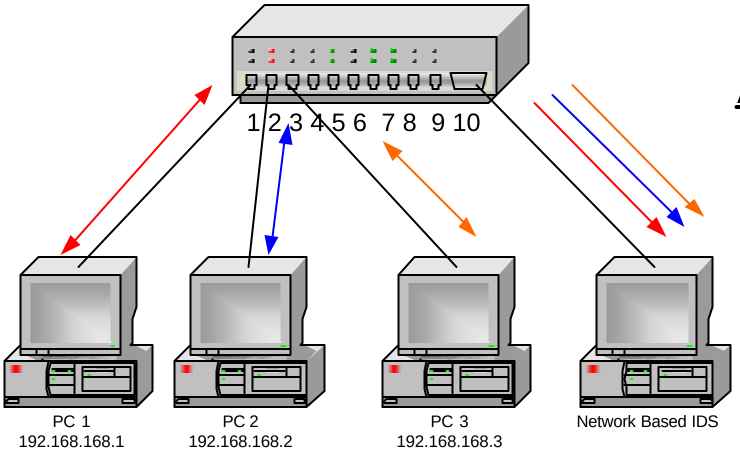


**Diagram Three**
**A switch using SPAN or Mirror Ports**

Here Port 10 has been configured as a SPAN or Mirror Port, so all traffic emanating from Ports 1-3 will be sent to Port 10.
Placing the nIDS on Port 10 will therefore allow you to monitor all communications from Ports 1-3.

However, as you start to use the SPAN ports on your routers and switches you begin to get into bandwidth and utilization problems. Even in a low bandwidth environment, where one has 10 ports on a switch running at 10MB/s (10% utilization), the SPAN port will be running at 100% utilization and therefore the nIDS connected to the other end will only be capable of monitoring 40-60% of the traffic. But more to the point, NOT monitoring the rest. Crank the utilization rates up to 50% on the 10 ports and the problem gets even worse, as now you're trying to squeeze 500MB/s down a 100MB/s port -- it doesn't fit! In addition, exact figures have not been tested, but it is generally assumed that using a GB port for mirroring will pull down the overall performance of the switch by 10-20%.

This can turn into a departmental struggle because the Security Officer is now infringing on the domain of the Infrastructure Department. The Security Officer is not only "stealing" one of Infrastructure's valuable GB ports, but also affecting the overall performance of the switch. This may sound like a comical scenario but it is one that goes on every day in major enterprises. Some infrastructure vendors would like to see security becoming part of the infrastructure, but they are forgetting that the objectives of Infrastructure are very different from that of Security. For Infrastructure, availability is everything; but for Security, it is making things secure that is paramount, and often these two objectives clash heavily.

Let's assume we are able to work around the Infrastructure Department and are using a SPAN port to monitor our 10 ports at 60% utilization. The SPAN port is ejecting 600MB/s to our GB sensor, but the sensor can only monitor 400-600MB/s. So it is going to start loosing packets. This means we are *not* getting 100% monitoring efficiency.

**Implementation Difficulties in Asymmetrical Routed Networks**

As networks become more mission critical, the need increases to provide as much redundancy as possible. It is this thinking that has led to the growth of Asymmetrically Routed Networks. In its basic form it is used where an enterprise wants to add maximum redundancy to the front end routers on a network, while also making them work smarter, and share the loads between the routers.

For example, Diagram Four is a simple illustration of an asymmetrically routed network where four routers are used in: Active/Active, Active/Active configuration --as the data leaves the network towards the Internet, it can go one of four ways.

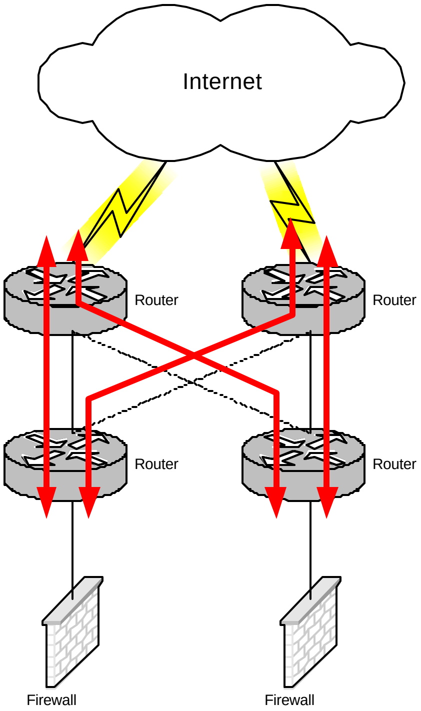
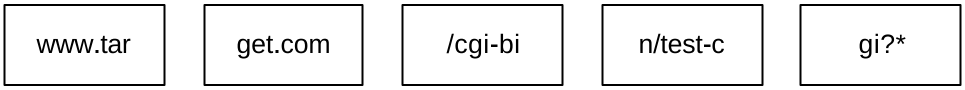


**Diagram Four**
**Asymmetrically Routed Networks**

These four routers have been configured to route asymmetrically (active/active), therefore a stream of data could travel one of four paths.

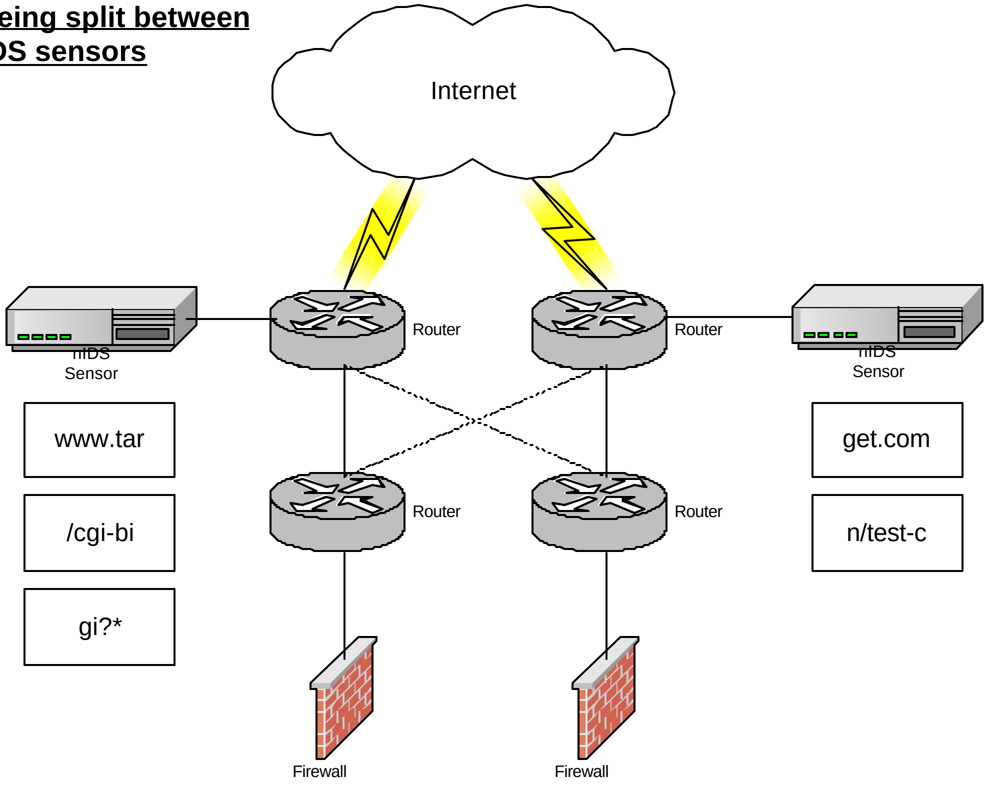
A nIDS can only work properly if it sees all the packets in a stream of data. For example, if we look at a simple CGI bin exploit on a web server, a hacker could enter in: http://www.target.com/cgi-bin/test-cgi?* to get a list of all the files and directories in the scripts directory. This stream could be split into 5 packets (see Diagram Five).

**Diagram Five**
**HTTP stream split into packets**



Within an asymmetrically routed network, this stream of data could be sent any one of 4 ways -- even if one has connected a nIDS to a SPAN port on each of the *front* routers, (see Diagram Six) and the data was distributed equally to each of these routers, half the packets would go to one nIDS and half to the other - so neither would pick up the attack.

**Diagram Six**
**Packets being split between nIDS sensors**



A nIDS Sensor has been connected to each router, if the stream of data was exactly split 50/50 between routers, then each nIDS will only see half the conversation.

This is further compounded by IDS Evasion tools used to attack the nIDS itself. Sensors now have to often buffer and re-fragment packets, and maintain complex state tables to keep track of individual sessions if they are to have a chance of detecting a true and correct attack

This means that in any complex switched or asymmetrically routed network consideration has to be given to how and where sensors are placed. A nIDS must see the complete flow or stream of data in order to detect if an intrusion exists. In many complex environments the data needs to be re-augmented before passing it to the nIDS in order to accomplish this critical task.

## Network Availability Challenges

Resilient nIDS deployments accommodate fail-over conditions and prevent hacker attacks to the nIDS. The following section explores the challenges of deploying a resilient nIDS environment to ensure high network availability.

### N+1 nIDS Redundancy

As nIDS become more and more important, so does their own criticality. Unfortunately, these systems do not include any form of redundancy or fail-over capability. It is therefore becoming increasingly important to be able to have some form of fail-over capability in a "n+1 configuration."

Although many load balancers are available (for firewalls and servers), these devices balance traffic based upon a *packet-by-packet basis*. With a nIDS the concept of *streams of data* is more important, as the nIDS has to see all of the conversation if it is to make a proper analysis of whether the communication is malicious. A packet-based load balancer cannot provide redundancy for a nIDS, only a *flow-based load balancer* can provide redundancy.

## Challenges Summarized and Next Steps

In summary, the deployment challenges highlighted demonstrate the complexity of trying to listen to massive amounts of information while searching for the proverbial needle in the haystack.

These issues are not going away. There has been an emergence of cyber attacks designed to target the nIDS. An example is *Stick* (http://www.eurocompton.net/stick/projects8.html), a pre-defined set of attacks that send lots of small single packet attacks to overwhelm the nIDS. The nIDS becomes too busy keeping up with the small, fairly benign attack that the true larger scale attack is missed. Hackers continue to find new and increasingly complicated ways for exploiting systems, and each month the IDS vendors release their various counter measures. For more information on IDS evasion see http://online.securityfocus.com/infocus/1577

The solution to this is to use a flow-based load balancer so that the traffic is shared across multiple nIDS sensors and then no one nIDS will be overwhelmed.

Combine with this the level of reliance and criticality of nIDS and you start to see the need for IDS balancing. The next sections will look at how the IDS Balancer™ from Top Layer Networks uniquely addresses nIDS deployment challenges.

## Introducing Top Layer Network's IDS Balancer with Flow Mirroring

The concept of using specialist switches to balance traffic to various devices is not new (i.e., firewall and server load balancers). However, balancing traffic to a firewall or web server requires a very different process from that of balancing traffic for a nIDS.

With the most common method of load balancing, there are active connections from the end station to the specific servers. All connections must terminate to a VIP (Virtual IP address) and then get balanced across to specific servers. This is done via NAT (Network Address Translation) where the destination IP address is changed to the specific server and the response either goes back to the NAT device or the end station. Typical load balancing devices do not copy a flow, redirect a flow, nor allow for seamless transport of the flow, which is what a nIDS requires.

A *flow* is the unit of measure that the IDS Balancer from Top Layer Networks uses to classify and recognize conversations between hosts on a network. It can be compared to an entire conversation between two people. A packet, on the other hand, represents only a piece of that conversation. To use the same analogy, a packet is a word or phase in the conversation.

The IDS Balancer recognizes and manages the flow or conversation as a whole. Whereas a normal balancer is packet-based, and only concerned with making decisions about each packet.

The IDS Balancer treats each flow individually regardless of source and destination IP addresses. The IDS Balancer makes copies of each flow and redirects the copy to a designated group of nIDS devices. This differs from a mirrored port or SPANS port in that when you mirror with SPANS it is a one-to-one relationship. All traffic goes to one port and you cannot break out traffic to share the load. Flow Mirroring designates each flow to a specific Monitor Group, or set of groups. It is possible for the same data to be sent to multiple groups – to send the data to two different types of nIDS, or the same data to a pool of nIDS and a sniffer (for forensics).

It is this ability to maintain session information that makes the Top Layer offering so unique, and why nIDS vendors endorse the IDS Balancer as a solution to help overcome the deployment issues discussed earlier.

Let's now turn back to these challenges – 100% intrusion coverage and network availability - and look at the ways that the IDS Balancer can help address them.

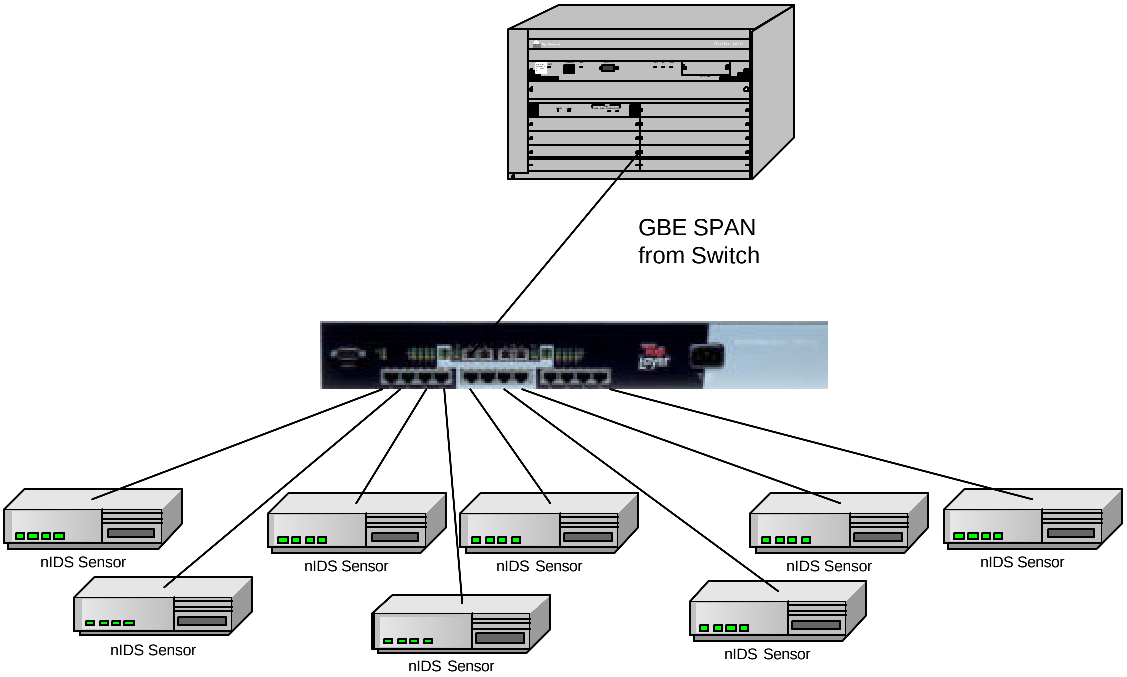
## How the IDS Balancer facilitates greater performance from a nIDS

As discussed, whether you are using a 100MB/s sensor, or a Gigabit one, you can only expect it to track around 40-60% of the traffic. And where the connections are coming from a SPAN port off a switch the traffic loads can easily exceed 80%.

The IDS Balancer uses as standard, the most common algorithm (*round robin)* to balance the traffic. This means that each flow is evenly balanced across the number of nIDS connected to it (this is best achieved with an odd number of nIDS). Other options do exist for Port Based Mirroring and Application based mirroring.

In practical terms this means that a full Gigabit per second of traffic can be sent to the balancer, and as long as you have enough sensors, you will get 100% attack recognition. In Diagram Seven we see a Gigabit Ethernet Mirror port being connected to the Gigabit port on the IDS Balancer, the traffic is then equally balanced across 8 100MB/s sensors.

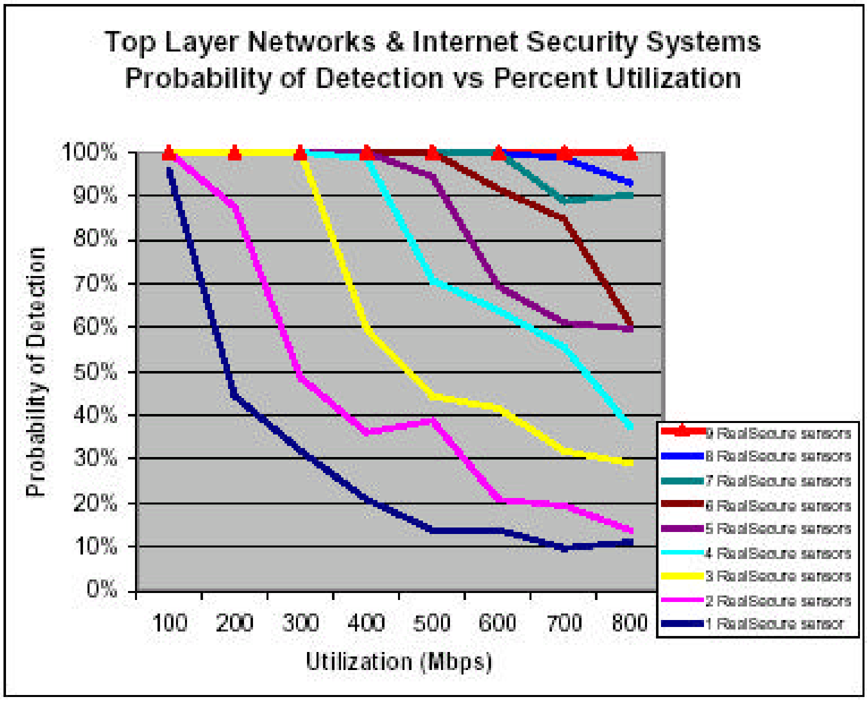
**Diagram Seven**
**Using Multiple 100MB nIDS to Monitor a GBE feed**



A Gigabit Ethernet or Fiber SPAN comes from the switch into the IDS Balancer. nIDS sensors are then connected and the traffic is balanced between them.

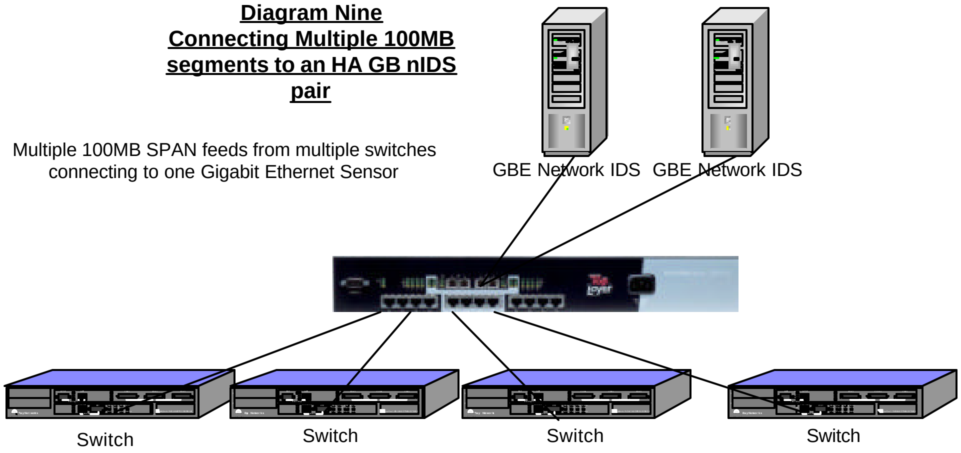
Tests were carried out with Internet Security System's Real Secure Network Sensor where a full Gigabit of traffic was sent to the IDS Balancer's gigabit port from a fiber SPAN port on a switch. Nine Network Sensors were then connected to the 100MB ports, and the following statistics were recorded (see Diagram Eight) (a full copy of this white paper can be found at http://www.iss.net/support/product_utilities/realsecure_tech_center/disclaimer.php?filedown=RealSecure_and_TopLayer_Gigabit_Performance.zip). However, it is important to note that the actual number of sensors required for a given environment will vary depending on the vendor and type of hardware used.

**Diagram Eight**



The above solution was based on connecting a Gigabit SPAN feed to a number of 100MB/s sensors. It is also possible to do this the other way around. Using a term known as "Inverse Multiplexing" it is possible to run multiple low bandwidth SPAN connections from switches, and mirror this to one or more GB sensor. This gives a number of advantages in terms of the reduced numbers of sensors that are actually need to be managed and maintained.

**Diagram Nine**
**Connecting Multiple 100MB segments to an HA GB nIDS pair**

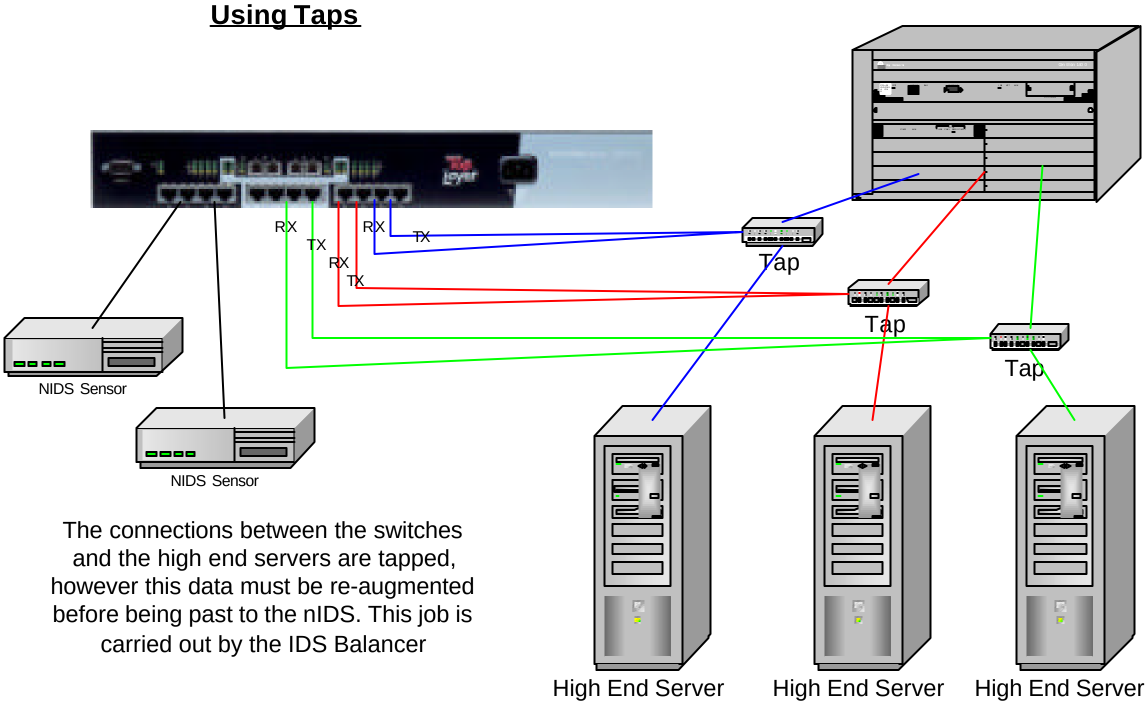

Multiple 100MB SPAN feeds from multiple switches connecting to one Gigabit Ethernet Sensor
GBE Network IDS
GBE Network IDS
Switch
Switch
Switch
Switch

Using the IDS balancer not only ensures that you are achieving 100% attack recognition on 100MB or 1000MB/s links, but it also allows for greater flexibility in deployments. Using the IDS Balancer (see Diagram Nine above), it is possible to monitor multiple low bandwidth connections with a GB HA pair, allowing you to monitor more for less. This provides a much higher return on investment for existing or new implementations.

**Increased Performance in Switched and Asymmetrically Routed Networks**
Flow mirroring is essential in switched and asymmetrical network environments when data from a conversation could originate from different sources, such as when the data is gathered from a Tap or router configured to route asymmetrically (see Diagram Ten).

As discussed earlier, SPAN ports are often not a good solution to pulling traffic off a heavily switched environment. One solution around this is to use *Taps*. In Diagram 10 we can see that three connections to high-end servers have been tapped and these connections past back to the IDS Balancer

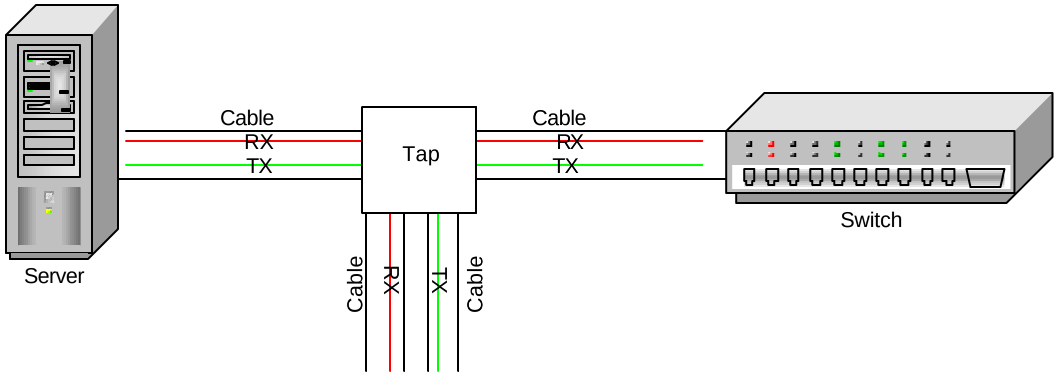
**Diagram 10**
**Using Taps**



The connections between the switches and the high end servers are tapped, however this data must be re-augmented before being past to the nIDS. This job is carried out by the IDS Balancer

**Diagram Ten**

## Diagram 11 - Taps



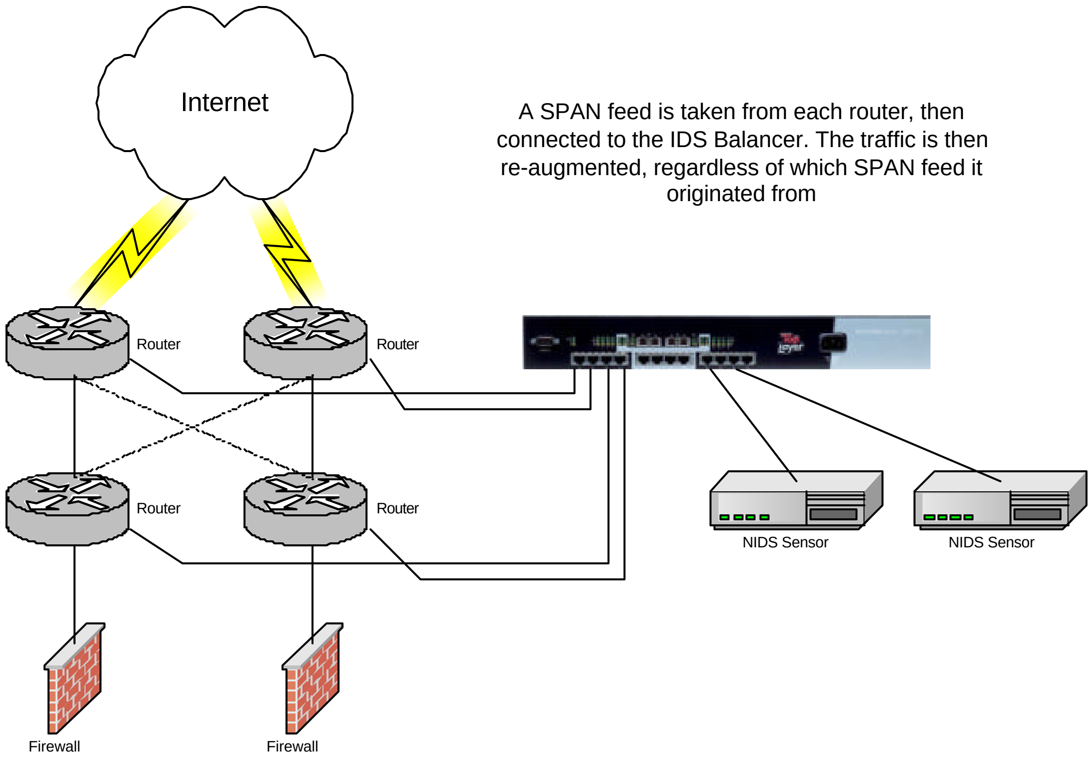
Taps are Layer 1 devices which passively connect to Ethernet or Fiber and take a copy of all the data that passes through them, typically they are designed to be fault tolerant with the main connections hardwired, so if power is lost to the unit the main connection will always be open (see Diagram Eleven).

Taps provide a number of advantages over SPAN ports. First, they have zero impact on the network or its infrastructure. Thus there is no need to change configuration of routers or switches, and therefore no performance hit to them. Second, it allows security officers to get their own copy of network traffic, which is kept away from the main infrastructure. Third, it protects whatever is at the end of the tapped connection, as only the Transmit (TX) traffic is copied, there is no way for a potential attacker to connect to devices at the end of the tapped connection (even if they know its IP address) as no data will be sent back up the connection.

Taps can offer greater flexibility to a deployment without impacting the performance of the network. However, because each Transmit (TX) cable is spliced (and so require two cables from the tap), you cannot simply connect a nIDS to each spliced TX, as you will only see half of the conversation. Therefore, these two connections have to be re-aggregated before being passed to the nIDS -- thus you need an IDS Balancer to make the solution work. TopLayer Networks can also supply a range of these taps ranging from simple copper taps to sophisticated fiber taps.
As discussed, the IDS Balancer understands and follows conversations, so regardless of which port the packets from the conversation come from, the IDS Balancer will re-aggregate the traffic before sending it to the nIDS, so that it sees *all* of the conversation.

The same follows for data coming from a router that is configured to route asymmetrically. In the aforementioned scenario of a simple Active/Active configuration, the packets could be routed through either router. Running a SPAN feed from each router to the IDS Balancer, as described in Diagram Twelve, the IDS Balancer will again re-aggregate the traffic before passing it to the nIDS ensuring that the nIDS sees *all* of the data in a conversation, and can make proper analysis of the data.

**Diagram Twelve**
**Using the IDS Balancer with Asymmetrical Routed Networks**



A SPAN feed is taken from each router, then connected to the IDS Balancer. The traffic is then re-augmented, regardless of which SPAN feed it originated from

## Redundancy

Redundancy is obviously important to any mission critical system, and using a round robin load-balancing algorithm in the IDS Balancer it is possible to configure the nIDS with n+1 redundancy (see Diagram Thirteen). This provides a level of redundancy, which is based on a simple check of the link status of a line. If a nIDS fails then traffic will no longer be forwarded to the failed nIDS and the remaining nIDS will pick up the load.

In order to determine the correct ratio of bandwidth to sensors, simply calculate the following formula :

([No of Connections x Network Speed x Full Duplex Static] / Performance of IDS) + Redundancy Factor

In cases of full duplex networks the static number is always going to be 2. This full duplex calculation is very important, as most connection between switches and high end servers are full duplex, which can double the amount of traffic that needs to be monitored.

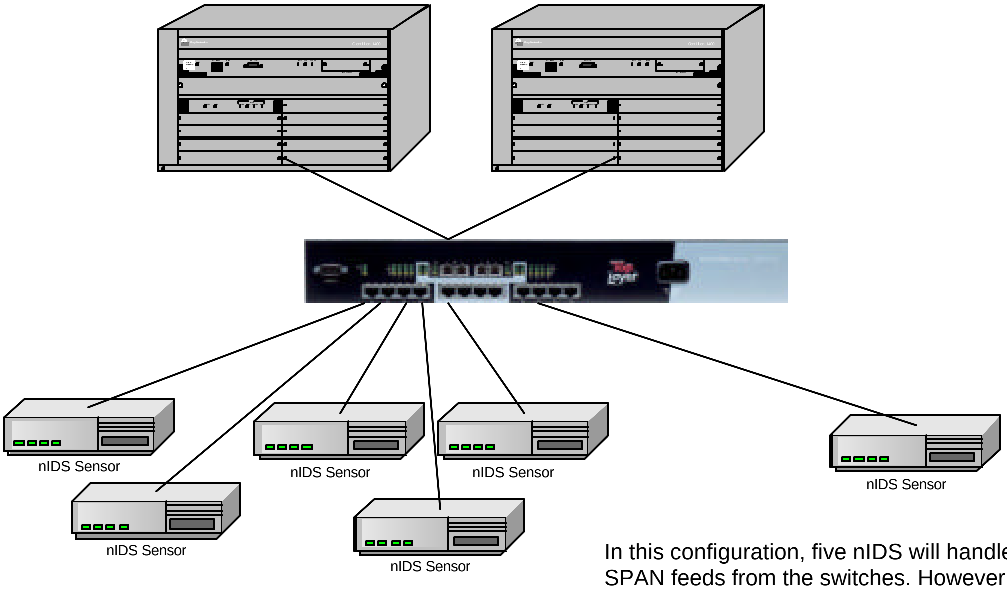
The Redundancy Factor will usually be 1. This simply means that if one of the nIDS sensors were to fail, the load could still be balanced across the existing sensors without dropping packets.

Here is an example of determining the IDS needs, where we have 4 full duplex connections running at an average of 60% utilization. Our nIDS will monitor around 60Mbps and we want to add n+1 redundancy:

([No of Conn x Network Speed x Full Duplex] / Performance of IDS) + Redundancy Factor
([ 4 x 60Mbs x 2] / 60Mbps) + 1
(480 / 60Mbps) + 1
8 +1 = 9 Sensors

If you require a 100% fully redundant environment, then the configuration in Diagram Thirteen leaves a potential point of failure in the IDS Balancer itself. The IDS Balancer can be provisionally bought with a redundant power supply. For certain requirements, then it is possible to connect two IDS Balancers together and use a heartbeat to check the status of the two IDS Balancers. However, be aware that this will require another complete set of nIDS sensors, as it is not possible with most nIDS to dual home (i.e. have two network cards in fail over) a sensor. Like with most 100% fully redundant architectures the solution can be costly, but if this level of protection is desired, then the IDS Balancer can meet the challenge.

**Diagram Thirteen**
**Using n+1 Redundancy**



In this configuration, five nIDS will handle the two SPAN feeds from the switches. However an extra nIDS sensor has been added to offer n+1 redundancy.

## Protecting the nIDS

In the last year we have increasingly seen attacks that target the nIDS, and not web servers etc. These tools, the most popular being The Stick (http://www.eurocompton.net/stick/projects8.html) work by bombarding the nIDS with lots of simple single packet attacks. The nIDS becomes overwhelmed by detecting these attacks, and also generating the alerts (which would go back to the console).

In tests, Stick was shown to generate over 250 alarms per second, and the nIDS collapsed and shut itself down after 2 seconds! Once the nIDS has collapsed, the hacker is free to launch whatever *real* attacks he chooses.

This is clearly a huge problem for a nIDS, however by spreading the attacks across multiple sensors (and by further splitting the traffic down by application), it is possible to negate this form of attack. In this way the IDS Balancer not only makes the nIDS sensor work more effectively, but it also protects it from being attacked itself.

## Summary of Benefits of using the IDS Balancer with nIDS

In summary, Network-based Intrusion Detection Systems are very valid investments in order to protect your enterprise. The IDS Balancer from Top Layer Networks offers the most comprehensive benefits in nIDS deployments:

- Ensures 100% intrusion coverage
- Easy to deploy in complex environments
- Provides the ability to monitor multiple segments with one sensor, thereby giving a greater return on existing investments
- Greater level of redundancy
- Protects the sensor from attacks to itself
- Maximum network availability

Top Layer's IDS Balancer is available in two models, one with 12x100MB ports and one with 12x100MB and 2xGB ports. For more information on the IDS Balancer this and other Top Layer security products please view our web site at www.toplayer.com or call 508-870-1300.

### *About Top Layer Networks, Inc.*

*Since 1997, Top Layer Networks (www.toplayer.com) has delivered proven network security solutions worldwide, enabling enterprises to protect against cyber threats, and scale their infrastructure to meet new, ever increasing security demands. The Company's intrusion detection and prevention products are built on a patented, ASIC-based architecture. The products are engineered to block high-volume DoS and DDoS attacks, HTTP worms, traffic anomalies and unknown attacks; improve the effectiveness of intrusion detection systems through intelligent balancing and distribution of traffic; and enhance the availability and performance of firewalls through firewall/VPN balancing technology. Top Layer Networks is headquartered in Westboro, Massachusetts with sales and support presence in Australia, France, Germany, Japan, Korea, Malaysia, Singapore and the United Kingdom.*

**Top Layer Networks, Inc. 2400 Computer Drive, Westboro, MA 01581**
**Phone 508-879-1300 Fax 508-879-9797**
**www.TopLayer.com**