1 Predictive Intruder monitoring and prevention

This article explores the possibilities and the cost savings that could be gained by integrating IDS, vulnerability scanning and patch management.

1.1 What is intruder detection

Intrusion Detection systems (IDS) is the burglar alarm of IT security. In IT security, IDS/IPS is where the action is, So lets spend sometime on the recent changes to the IDS landscape. You may not have noticed but networks are getting faster - switched or even encrypted. This has caused problems particularly for Network IDS (NIDS) as it makes it harder for the devices to acquire important data and requires the same device to process the data quicker. Generally, this has resulted in vendors favoring HIDS (Host IDS) where client IDS software is installed on each host to be protected. However, the actual users are often less impressed with HIDS for a number of reasons.

1.2 HIDS or not a HIDS, that is the question

Many HIDS only compare a stored, known correct checksum with a newly generated checksum of key files to detected changes (known as *state monitoring*). The limitation of this technique results is reduced warning of hacking activity which in-turn provides less time to react. For example imagine a situation where you detect a hacker by discovering that a file has been changed in */etc/rc5.d* – great you caught a bad guy when he changed something he shouldn't have.

But this file almost certainly wouldn't have been his first choice of target, what about the dozen attempts to update */etc/shadow*, */etc/passwd* and */etc/hosts* that he would have tried beforehand - which your checksum based HIDS failed to detect because the hacker didn't manage to change a file. As the point of an IDS is to provide early

www.loud-fat-bloke.co.uk

warning of hacking, this approach is poor because it only warns you after the damage is done and an unauthorized change made – too late for me. Most experts will confirm that state checking tools like *TCT* are a superb way of determining what a hacker has done to you when you are in the recovery stage of your Incident Response process.

Even where the HIDS has superior event data acquisition from using a kernel mod (i.e. *LIDS*) or by links into the audit subsystem, users are complaining that some HIDS adds little over audit systems given the cost. There is just not enough granularity in the rules without having to write complicated scripts – it may not be clear to the software engineers writing these package but ***administrator logins*** are frequent events on most networks. We need to know when an administrator logs in from a strange workstation or out-of-hours but not every time he performs a normal job function from his standard work station. This doesn't mean that the technology is a *lemon*; it just means that it is not as mature as it really should be and currently it works best being supported by a sensible NIDS deployment. Now that I have just about finished the WIDZ project, I intend to spend sometime working here.

In fact HIDS have a great potential, they have the ability to directly access the machine to get patch and inventory information.

2 NIDS in your hair

Or was that Nits, the fact is both have caused some head scratching. However, most manufacturers haven't ditched NIDS , but they are having to work harder to make it work. The overhead of processing thousands of attack signatures (*signature analysis*) is huge. When common media reached speeds of 100Mbits, manufacturers introduced protocol analysis (looking for things in the right part of the right packet) instead of checking all packets for all signatures, appropriate or not (*packet greping*). This common sense approach has

helped but the emergence of the gigabit network makes congestion inevitable. However, common sense has never been a strong feature of our industry so most manufacturers have become fixated with search for techniques that have a lower resource requirements and network latency on IDS devices rather than concentrating on more important features like better detection. One good side effect however is that it has hastened a move by a few vendors to re-visit *anomalous detection*. This means establishing what traffic is not normal on your network AND that is indicative of a hacker - and then using any divergence from this baseline to trigger alerts. This will not only result in better detection but one day the NIDS will be able to use anomalous detection to isolate a hitherto unknown attack signature and send details to vendors to be included in IDS& Scanners signature databases.

Many vendors have poo-poo'ed the concept of *anomalous detection* by implying that

1) most networks are too diverse to baseline,
2) training time will be too long and that
3) it will produce too many false positives.

Together, the last two arguments seems bizarre, have these vendors every used their own products?? Usually, most sites have to spend a large amount of time tuning the ids and still are left with an unacceptable level of time-consuming false positives. I have dug out some of the data I had from the last job I did on the best selling NIDS, all of the data was *False-Positive* because there was no hacking occurring. But from the data, it was clear that most of the problems occurred from the non-specific nature of the rules. For example, on this site they had IIS and IPLANET servers so both sets of rules were enabled. However, this meant that some of the traffic directed towards the IPLANET server will triggered IIS events. And no the open-sources-bible thumpers can't feel smug here either, Even if you

bother (most don't) to tailor the config of the fabulously flexible Snort and accurately set the variables $HTTP-SERVER $HTTP-PORTS to the correct values, the above situation will be true in a multiple web server environment. The result is that every IIS related MSADC & Jill attack that is mis-directed to a IPLANET server will result in a high priority alert.

The situation is still worse with datagrams or context attacks that may be launched in an initial tcp packet. In this case, Snort will fire an alert for an attack on server that doesn't exist. This kind of false alarm represents the majority of the alerts most IDS produce. A simple pre and post processor could significantly insure that alerts were only produced for machine that really existed and that alerts were of a suitable priority if you really were vulnerability.

Another feature that holds great promise is Snorts' *activate/dynamic*. This feature uses one rule, the *activate rule*, to define malevolent traffic. The subsequent *dynamic rule* can used to log a predefined number of packets from the original host. This means that after an attack you have a complete session trace – a feature only available in a few commercial IDS. But with a bit of fiddling, this can even be used to set up a basic DEFCON scheme so that your IDS automatically increases its monitoring levels. Normally, a sensor will run with a low level of monitoring in-place until the activate rule triggers a more rigorous set of rules.

Another exciting feature for NIDS is the *Crypto-network* tap – these devices act as "*bumps-in-the-wire*" and allow encrypted traffic, say SSL to web servers, to be decrypted so that an IDS sensor can access attack information in clear text. This solves a major problem with NIDS whilst any security risk is minimized because no *Crypto-key* information or decrypted cipher text ever leaves the device. We are currently experimenting porting an IDS to such a HSM device.

3 Trends in Vulnerability scanning

Look at the numbers in figure 1, over 10 new vulnerabilities reported each day.

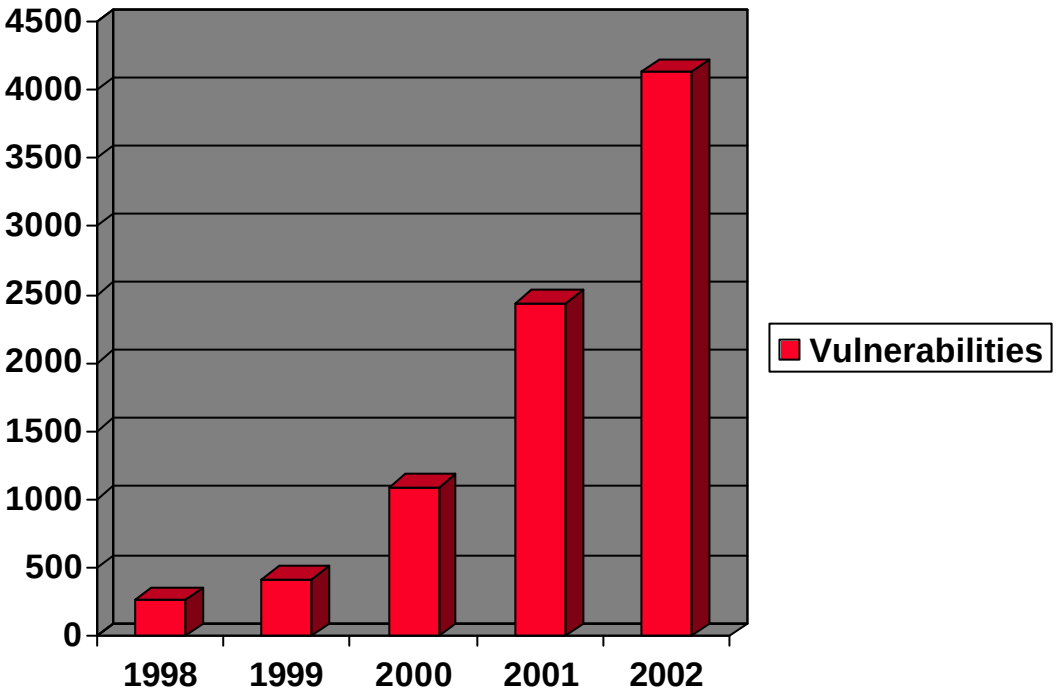


Figure 1: Vulnerabilities per year *Source Cert/cc*

Surely we should check for these vulnerabilities with the same sort of frequency. Many vendors are providing services that scan for basic infrastructure exposures on a daily basis. This allows more time and

www.loud-fat-bloke.co.uk

effort to be spent on less frequent more manual intensive application security testing.

4 Linking scanning with IDS

As more integration occurs between these two tools the combined value will increase exponentially. Both tools share a database describing potential vulnerabilities, here alone saving on integration and maintenance provide opportunity for labour saving. Instead of maintaining two, we will only need to update one. To this ends, we are already working on a tool that will link *the Nessus* Scanner with the snort IDS.

With some commercial products, we are already able to correlate vulnerable systems as identified by the scanner with IDS Alerts when they are attacked. This surely means that we can already produces software that not only Correlates scanner results, but uses the information to dynamically raise the priority of any alert based on the targets sites vulnerability (as determined by the previous nights scan) – Or even cures the exposures. If only.

4.1 My blue heaven

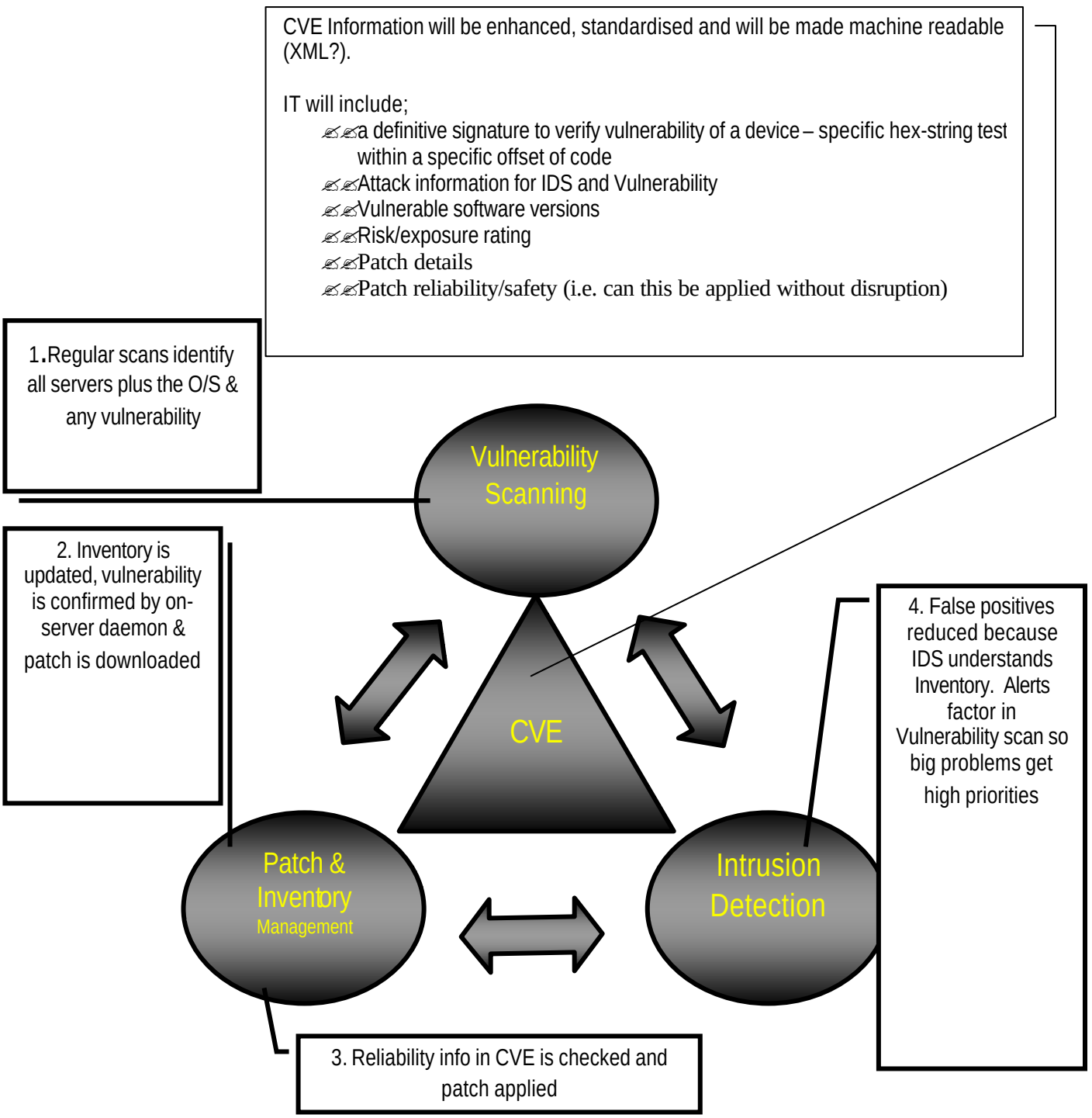
Imagine a situation where yet another stack overflow exploit is released, and is estimated to have the capability to devastate all your production servers.

However, we're in an ideal world so

1. Automatically, your security control console down loads the latest CVE information. This will contain machine-readable pointers to combined scanner/IDS signatures. That night your scanner scans your network automatically building an inventory of vulnerable hosts.

www.loud-fat-bloke.co.uk

2. Then the security consol downloads the patch, and assesses, based on risk/reliability information stored in the future format patch whether the patch can be applied automatically. If it is safe, the patch is applied.
3. In any case your NIDS monitors for malevolent packets containing the attack signature, if the attack is directed towards a host that isn't vulnerable the NIDS will only raise a minor alert. If it is directed at a vulnerable host, the IDS will raise a high priority alert.
4. Then the control software will make a decision on whether the NIDS or HIDS will stop the attack, either using packet modification, address shunning or TCP-reset.



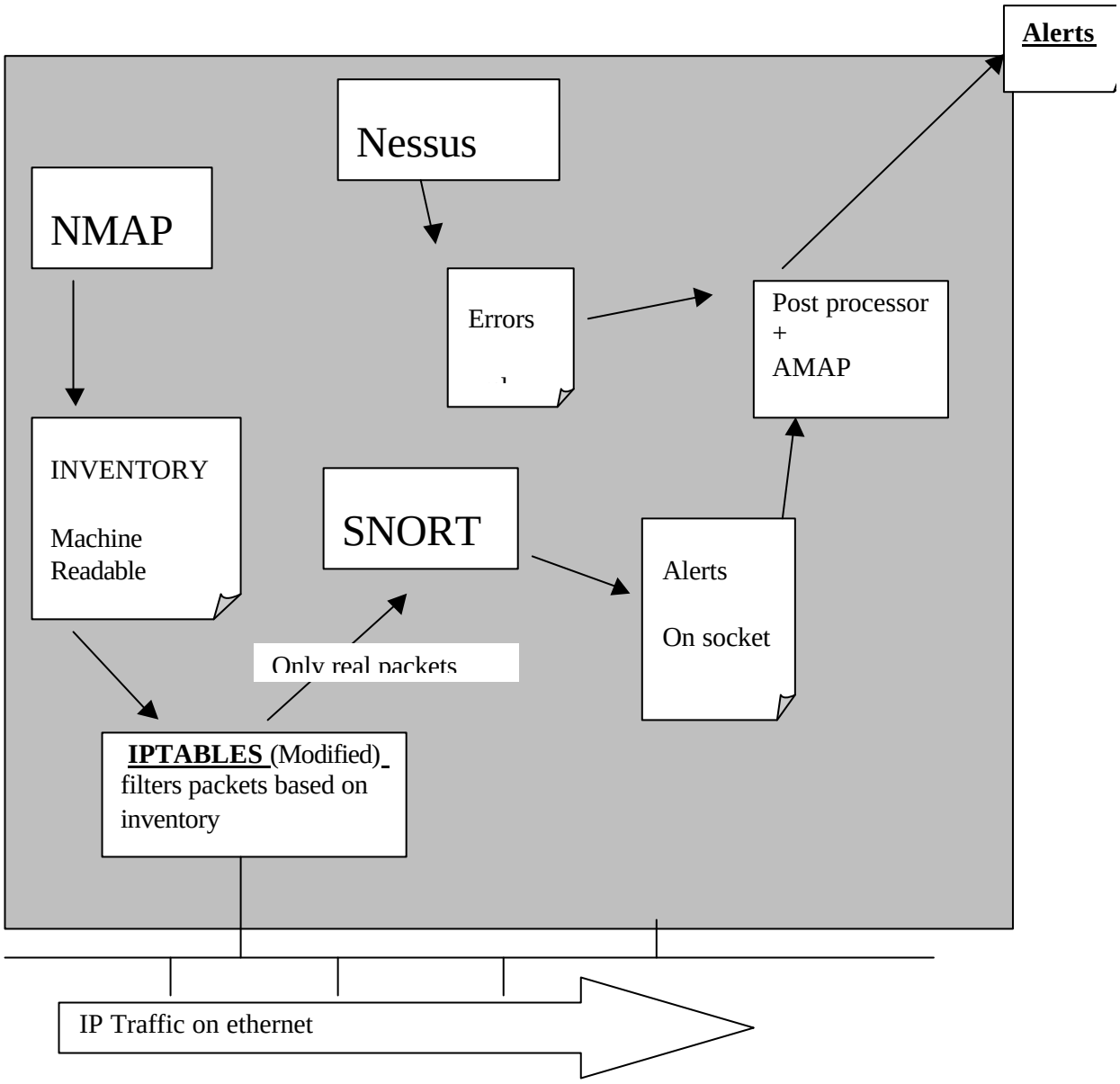
www.loud-fat-bloke.co.uk

Figure 2: Integrate Security Vulnerability management

Much of this depends on better format CVE and Patch/Advisory unification by most software vendors to providing compatible information - which will never happen.

4.2 If a jobs worth doing, its worth doing yourself

Based on my experiences with Nimrod and then WIDZ, if you just write papers about stuff nobody picks it up. If you lay down code, all the better programmers out there get motivated and build better things. So this is what I going to build on the NIDS front:



To increase throughput and reduce the number of false positives produced by probes and routing errors:-

1) Nmap is going to be used to produce a machine readable inventory
2) IPTABLES is going to be modified and used as pre-processor described above. It interprets the machine readable inventory, so that only messages that are destined for a real (i.e. active) address/port pair will be processed by the venerable snort. Load balancing across a number of machines will also be achieved by this (Did I tell you that my company suddenly found itself with 750 pc's with nobody to use them)
3) The venerable Snort will process these packets with a typical *business as usually attitude* – writing its alerts to a socket.
4) A post processor will pick this up, verify that the address exists, that the port is active and identify what it is using AMAP.
5) This will be compared to a nightly nessus scan.

If nobody bites and rallies to my support, I might add some HIDS component linked in by SSH and some form of anomalous heuristics – who can tell what I'll do if I get some free time on my hands

www.loud-fat-bloke.co.uk