Deploying ISS Realsecure in a Large Scale Environment Part 1: Implementation Architecture

by Richard Reybok and Michael Engle

last updated Monday, April 17, 2000

Relevant Links

Deploying ISS Realsecure in a Large Scale Environment (Part 2)
Rich Reybok and Michael Engle

Background

Welcome to the first half of a detailed description of a methodology used to deploy the Internet Security Systems (ISS) RealSecure intrusion detection product. It is aimed at those of you who own or are thinking of purchasing the software and have questions about deployment in a large-scale environment. This part will serve as an introduction to the general topic but will also dive right into deployment specifics. The next part will talk about the manageability of the deployment and how to use all the data you will be generating. For those of you using other ids systems, some of the information presented may be of value to you as well. Some of the specific details may not apply, but certainly the concepts and thought-processes are similar.

What is an enterprise?

Enterprise rollouts mean different things to different people. When we speak of enterprise in the context of this article it is referring to a network engine deployment of over 100 and a host agent deployment of greater than 1000 in a mixture of Unix and NT hosts. Your mileage may vary. You might be smaller, or you might be larger. Most of what is described here scales equally well to organizations of many sizes. The most important advice you can receive is to know your environment, know what you need to monitor and know your traffic and user patterns.

The environment

To get an idea of what you can expect from this article, the following table details the specifics of this example environment.

Number of Network Engines	40
Number of IP Addresses Monitored	20,000
Number of Host Agents	1000
Platforms of Host Agents	Solaris 2.x, Windows NT 4.x
Data Gathered	500MB/per day
Number of true incidents detected	3/month (average)

Implementation Architecture

The implementation of RealSecure consists of the network engines, the host agents, and console placement. The network engines are probably the most familiar to people with knowledge of IDS infrastructures. The host agents are relatively new to the mix, and few vendors have yet to adopt this technology. It is important in your design to plan for both of these as they are very complimentary and help alleviate some of the costs of deploying network engines. Finally, console deployment is one of the biggest challenges facing an ISS install. The consoles handle all of the policy updates, database downloads, and reporting mechanisms. If you do not plan from the beginning, your deployment and support costs will quickly run up.

Deployment

This article assumes you have already defined your requirements for deploying an intrusion detection system. This process should have been done during your product evaluation stage. For example, if you have a requirement of having host agents on SUN Solaris, or you need to deploy network engines to FDDI segments, you need to make sure the product does this before purchasing. Incidentally, if these are your requirements, you picked the right product.

Network Engines

Network engine deployment is a complex topic and it is impossible to talk about all the issues related to it here. What follows is the thought processes we went through in planning our initial rollout. This is by no means the only way to do it, but is what worked for us in our specific environment. Your needs from a security, manageability, and supportability standpoint may be significantly different and you will need to apply what we present here to your environment.

Costs of deployment

Usually your rollout is constrained by a fixed budget to work with. You not only have to consider the cost of the software itself, but also the hardware it is going to run under, the support costs associated with maintaining the hardware, and also the costs of actively monitoring the security alerts from the multitude of engines deployed. Unless you are Bill Gates and you are wiring up your electronic house to watch for Linux hackers, you probably know full well what I am talking about.

The first thing you need to know is how much can you spend on your deployment. You can reduce your costs by making the decision to go with PC Hardware and Windows NT as your server platform. I don't care about the Unix vs. NT philosophical debates. The point is once you rip NetBT off the NT box you can reasonably secure it to the point where it is not a liability. In the long run you might be able to get 2 or 3 extra engines with the cost savings. A machine with dual processors, two NICs, 256MB of ram, and a 9.1 GB disk is really all you need. The one drawback is that you will get better performance by running your engines off a Solaris box. There is an issue with the NT NDIS driver and its associated buffer size. Solaris' is a bit larger and therefore can handle somewhat higher network utilizations. In testing, we found that the cost savings justified the slight performance hit.

Where to put them

Once you have your fixed count of machines, the next step is finding where to put them. There are the usual suspects, the DMZ(s), backbone(s), WAN connections, etc. Those are typically the easiest to identify and will not be a cause of great concern. Finding suitable locations for the rest of your engines is the more problematic issue.

You want to try and hit as many of your network entry points as possible, the SMTP backbone, your dial-in infrastructure, and any place you terminate a leased line connection. You can probably catch enough events here that will help justify a further deployment. The SYNFloods, E-mail viruses, and port-scans that are in constant motion on the Internet will probably show up quickly in these fairly unprotected zones. Once you've got those sealed, start selecting the networks based on the criticality of the hosts that are placed there. These hosts might be your mainframe IP gateway, human resource and payroll systems, and other critical business systems. Hopefully you have a detailed information risk plan and disaster recovery scenario that identifies these hosts. If not, try interviewing your business line technology and support managers and get their opinions.

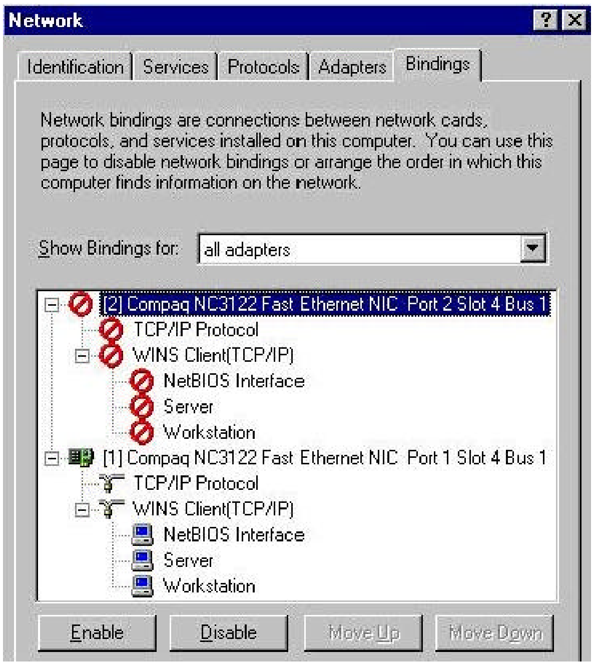
If you have any extra engines left after this stage, try hitting the segments of your power users. This includes the SA community and any in-house developers if applicable. These are the guys who are probably reading articles like this and downloading the root-kits and scanning software to "test" on their local machines. Then try looking at networks that house your senior management, or other users who may be potential "targets", of a disgruntled employee, or of theft of inside information. The more data you can piece together from your engines during the forensics stage the easier time you are going to have limiting the damage exposure and ultimately seeking prosecution.

Special Configuration Considerations

There are a couple of specific things you need to do to the engines before you start deploying them everywhere. I will continue to talk in terms of NT-based engines, but similar theories apply to Solaris. First of all, make sure you apply the most recent high-encryption service pack for NT 4.0. This will allow you to take advantage of 128-bit encryption between the engines and consoles.

I mentioned earlier that the box itself would require two NICs. This is to facilitate a couple of deployment considerations. Basically, one NIC will act as normal with only TCP/IP bound to it (on a completely separate IP segment from NIC 2), this will be used for all console to engine communication. The second NIC will have no protocols bound and will be used to do the packet capturing (Figure 1.). This comes in handy for a number of reasons. First, it will help "hide" the engine from detection from the local segment it is listening on. Second, it will allow for easier deployment to segments that are separated by a firewall. What you do is place the "listening" NIC on the protected segment, and the communication NIC on a private, internal IP wire. This reduces the need to punch holes in your firewalls, while still maintaining a high degree of security. If you secure these boxes, i.e. no CDROM or floppy drives, hardened administrator username and password, no local NT source files, and you lock them up in a room then there is little risk with this method.

Figure 1:



Network bindings are connections between network cards, protocols, and services installed on this computer. You can use this page to disable network bindings or arrange the order in which this computer finds information on the network.

Show Bindings for: all adapters

- [2] Compaq NC3122 Fast Ethernet NIC Port 2 Slot 4 Bus 1
  - TCP/IP Protocol
  - WINS Client(TCP/IP)
    - NetBIOS Interface
    - Server
    - Workstation
- [1] Compaq NC3122 Fast Ethernet NIC Port 1 Slot 4 Bus 1
  - TCP/IP Protocol
  - WINS Client(TCP/IP)
    - NetBIOS Interface
    - Server
    - Workstation

Host Agent Deployment

The host agent is what separates an ISS deployment from some other vendors' implementations. The host agents give you the ability do a pretty good job of monitoring the overall activity on a given host. Things like who is logging in, which programs are being started, and which files are being accessed are all vital when trying to both guard against theft and tampering, and piecing together the puzzle after an event is detected.

Costs of deployment

Host agents have a completely different, and not necessarily tangible, set of costs associated with them. Generally speaking, most ISS deployments of host agents are on Windows NT and Solaris so I will stick to the issues associated with those platforms here, again this will be applicable to other platforms where appropriate.

The agents on both of these platforms are going to consume both disk and cpu. There is nothing you can do, and the more signatures you apply the more the machine is going to do. More than likely when these boxes were rolled out, there were probably not any plans for ids agents, so you will be consuming resources budgeted for real application processing. It is important to note this as some machines may not be able to support any additional load and will have to be dealt with by either upgrading the box or by revisiting it at a later time. Do note however, that the load in most cases is negligible and in a sampling taken from a real deployment we saw less than 5% CPU utilization and a memory footprint anywhere between 6 and 12 MB.

In the specific case of Solaris, there is an additional consideration to be aware of. The Solaris agent requires that the Basic Security Module (BSM) be installed prior to the agent being installed. Hopefully, you already have this installed because auditing is important to your organization. However, I have seen places where it is not installed and it is tough to retrofit auditing on a box it wasn't planned for. Issues relating to disk space and CPU once again enter the picture, in addition to a system reboot and the resulting downtime. All of this may be too much to overcome and you must either fix the box or revisit it later.

Where to put them

There is no reason, other than the cost of a license, why every machine in your firm should not have a host agent installed. If you can afford it, I highly recommend a mass global deployment. But where do you start? I like to follow the 80/20 rule when planning an initial phase of rollout. Try to hit the most visible and easy to deploy machines first, this usually turns out to be about 80% of the total projected rollout. This will give you a quick and hopefully painless infrastructure where you can begin to tune your policies and protect most of your assets. Then, over time, deal with the remaining 20% as the issues surrounding them are resolved, i.e. getting the downtime, fixing the disk space, installing auditing, etc. This last phase may take a while, so it's best not to let it delay you from completing other machines. During your rollout, try to hit your critical hosts first. The email servers, the e-commerce web servers, and anything on that disaster recovery plan you looked at during the engine deployment phase.

How to deploy

With the network engines, you had pretty much total control of the box and were dealing with significantly fewer numbers. The hosts you are deploying the agents to are generally not under your control and are subject to guaranteed service level agreements with the business lines. In addition, it is not uncommon to deploy agents to thousands of machines, making software installation difficult. For this reason, it is important to try and leverage as much software delivery mechanisms as possible. Going to each box with a CD is simply not doable without a lot of resources and time.

Thankfully, the Solaris agents come in package format and can be fed an answer file to facilitate unattended install. This should be fairly easy to script and is extremely reproducible from machine to machine. On NT, things are slightly more complicated as there is no real unattended install procedure. I'm sure many of you in these large environments have some form of software delivery to your machines. The RealSecure agents lend themselves real well to things like Microsoft SMS and Seagate WinInstall. In fact, in one deployment we pushed out 500 NT agents using WinInstall in one weekend without a single rebooted box.

What to expect next

Now that you have the engines and agents deployed, you will quickly find that an overwhelming amount of information about your network will come flooding in. The next article will go into great detail about how to handle this and keep your sanity at the same time.

Richard Reybok currently manages a Wall Street financial firm's global security engineering team where he puts his ninja security skills to the test. He has over six years experience in information security technologies and over ten years in the industry itself. When not defending the free world, Rich relaxes by spending time with his lovely wife Jennifer and two beautiful daughters, Samantha and Kaitlyn.

Michael Engle currently manages computer security and incident response for a large financial firm on Wall Street. He has over 8 years experience in the industry. In addition to computing as a hobby, he enjoys scuba diving, skiing and traveling the world with his fiancée Diana.