Global Information Assurance Certification Paper

Brad Skrbec
GSEC Practical
Version 1.4b
Option 2
December 11, 2003

Know Thy System: Making Peace with Your Inner Rogue

Abstract

This case study presents a company's attempt to defend the potential damage done by unmanaged or poorly managed network-attached equipment within its own internal network.

This paper will discuss why those "rogues" must exist, difficulties that arise from their existence and the processes that can be put in place to mitigate the risk and maintain a safe business computing environment. We will show the effect of such processes on a large organization.

Introduction

Recently, large numbers of advisories veritably flow out of the CERT organization, keeping those in security and information protection on their toes. Adding kindling to this bonfire of security issues, many companies have a high percentage of "rogue" workstations and servers connected to their networks. For the moment we'll define rogue as a workstation that does not come under the administrative control of the central IT group. This term will be refined and internalized to fit the organization's needs as this paper progresses.

To understand the scope of the problem, we first need to look at the old environment and a definition of the problem. Next, we'll look at what parts of the existing environment that must be changed versus those that must stay in place. Finally, we'll look at the processes put in place to overcome the problem and their benefit to the organization.

The Environment

This case study addresses a segment of the computing environment of a Fortune 500 company. The entire network is comprised of hundreds of thousands of individual pieces of computing equipment widely dispersed around the world. The smaller subset detailed here is related only to a specific telecommunications business unit of that company. It is a sizable network segment, comprised of

more than 20,000 pieces of computing equipment. A high percentage (90%) of that network is located locally in the buildings of the business unit, but there are still many pieces of equipment that are either off-premises, or in different offices. There are also workstations that are either semi-permanently located in employees' homes or laptops that travel between home and the office.

Beyond the objective measure of the network by sheer number, there is a qualitative difference in the three distinctive segments of the network. The core network is comprised of static servers that remain powered on and (if we do our job correctly) available to users. The occurrence of movement and retirement of core pieces of the network is not frequent, and many pieces of equipment never move during their entire service record. The central IT organization has sole administrative responsibility for these servers.

The second subset of hosts is that of the user equipment. This entails laptops and desktop workstations that reside at the desks of the user community. Again, the true administrative work on these hosts is the responsibility of the IT organization; however, there are a few differences from the previous subset. First, IT is not the only authority on these boxes. Many users have administrative rights on these boxes, so that they can install software, drivers, and perform other assorted administrative tasks. Secondly, this segment of the network is highly mobile, with laptop users carrying their equipment off-premises to do remote work at home, conferences or other company sites. These users often need to connect to the company network through their own external network connections, so they frequently configure things like printers and other external devices as their work environment changes.

The final subset is where the true rogues reside. The company has a broad number of products that use embedded computers and/or contain workstations as part of the end salable product. To simulate these end-user environments for development and testing several labs have been created. Each of these labs has its own set of requirements, and those requirements frequently prevent the hosts housed in the labs from following the standard image that the IT organization uses to create its servers. These labs are sometimes secured labs that have connections to our business partners, and are firewalled off of the main network backbone. These rogues comprise approximately 6000 hosts, or 30% of our network.

Is a Rogue Evil?

As stated before, the general consensus of the IT group on the definition of a rogue was any host that we did not specifically control. To the general IT way of thinking, this is an inherently bad (evil?) situation. As administrators, we want to be in control of our "world". By its very definition, the rogue prevents that control.

So your first question might be: "What's the big deal? Why are rogues really a problem?" The CERT Coordination Center advises a seven step method for securing your network in the paper *Securing Networks Systematically – the SKiP Method* [1]. Let's compare the first five steps of that process against what the rogue allows us to do.

- *Step 1: Select systems software from a vendor and customize it according to the organization's needs.*
  With a rogue, systems software selection is done by the customer group or rogue owner. They choose the O/S that best fits their need.

- *Step 2: Harden and secure the system against known vulnerabilities.*
  Having surveyed the environment, I would conservatively estimate that 80% of the rogue systems that are installed on our network possess the out-of-the-box configuration of the operating system. Not a single system parameter has been altered.

- *Step 3: Prepare the system so that any anomalies can be identified and analyzed for potential problems.*
  Step 4: Detect any anomalies that could indicate a system intrusion.
  These steps would imply that some host or network-based intrusion detection systems (HIDs or NIDs) were put in place to detect anomalies. The engineering groups typically have other project priorities that are so high that security measures such as these don't even make it onto their radar. Since the security practitioner is of the IT organization, he is not often sought out for his expertise. The subtle implication here is that the system in question is often an integral component of a larger system that we sell to our customers. The security practitioner should realize here (with a shudder) that these inadequacies in security are sometimes passed on to the customers.

- *Step 5: Respond to any intrusions.*
  Obviously, since we are not monitoring these at a system level, the only intrusions we are likely to identify are network anomalies picked up by NIDs. Since the rogues are on a firewalled/ACLed section of the network, NIDS only monitor the perimeter of the labs. No internal network or host-based detection is done.

All in all, we score badly against the SKiP method. If we examine the slightly different perspective offered by Parker[2], a security practitioner should resolve a security vulnerability by first attempting to determine if the problem can be avoided altogether. But how does the practitioner *know* if the problem can be avoided? He has no direct knowledge of the rogue system in question. What does its OS image look like? Is it patched to the current recommended levels? It's a rogue, so we don't know. So we add our first refinement to the rogue's definition. The rogue is not only a host we don't control, it is also a host that is a

black box to the IT organization. So we have no visibility to what's "inside" beyond its pretty case. This strongly violates the first principle of security specified by Cole[3]; *know thy system*.

Another difficulty with rogues is that, for reasons we'll explore later, they often become un-upgradeable and un-patchable. If we cannot upgrade or patch a system, the security team loses a layer of another aspect of Cole's model; *defense in depth*. With the recent release of the MSBlaster worm and its successor, it was easy to realize that unpatched systems were going to be a difficult feature of our network to contend with. As mentioned, some rogues are highly mobile (wireless laptops or mobile computer carts), and others are in labs to which the IT group has no access. This division of the company is sprawled over several hundred thousands of square feet of office space. Keeping track of this equipment is difficult, to say the least.

Finally, the single largest problem with this style of network is that remediation of large vulnerability exploits such as MSBlaster and Code Red generally took several months to complete. With fast moving exploits time is of the essence. These need to be addressed in days and hours, not months and weeks.

Given all of this, the answer might be, yes… rogues are a bad presence on the network. Unfortunately, it's not as black and white as it may seem.

Are Rogues Necessary?

In a perfect world, all hosts would be homogenous, and centrally controlled by a single administrative team. That team would have ubiquitous access to every piece of equipment on its network. All equipment would be imaged from a single image that had been hardened and was up to date with regard to patches, kernel modifications and service packs. To ensure that all equipment was cookie-cutter, automated installation tools such as Jumpstart or Kickstart could be used. All equipment would be connected to a central distribution framework such as SMS so that patchlevels could be kept up to current security and OS standards. All hosts would have the same applications and programs.

Sounds beautiful, doesn't it? Okay, time to snap out of dreamland. One would be hard pressed to find such a network anywhere on the planet, but the highpoints of this ideal are important to note. In the ideal world, we have centralized and consistent administrative control. We also have homogenous equipment, images and patching. These are things to strive for where it is possible. What we do with the remaining equipment is a matter for further discussion later.

So if heterogeneous networks are a problem, why allow them? There are a variety of reasons why hardware and software homogeneity is impossible to maintain in a company this size. Let's look at a few:

Engineering Variety

Different engineering problems require different engineering solutions. This means that the hardware and software that is appropriate for one project may not be appropriate for another. Many factors play into these decisions, such as performance, price and device specific requirements.

Often, even within a homogenous hardware domain, single images cannot be maintained because special peripherals or software require a particular operating system configuration or driver.

Manufacturing/Product legal regulations

This telecommunications division deals with many countries that have regulations in place to protect consumers of manufactured and software products. Among these are regulations that require a manufacturer to provide legacy support for his products, and to be able to recreate the elements of a product line for a specific number of years for support and replacement purposes. This varies from case to case, but some of the specific regulations for the telecommunications industry place the constraint of seven years of reproducibility on infrastructure equipment. Seven years can be a very long time in the computer industry, with major technology improvements moving at a rapid pace. This legacy equipment often creates rogues on the network because the hardware or software reaches a point where it can no longer be upgraded in line with the remainder of the network. A recent example of this occurred with a server that was stopped at Windows NT 4.0 with Service Pack 3. It could go no further than this because this server was part of a product that required dual token ring network cards. Installing a later service pack caused the network cards to stop functioning. Likewise, no later installation of Windows (2000 or XP) supported this configuration. So the test servers for this product were frozen configurations with no way to move forward. Hence, vulnerabilities that were exposed in these configurations could often not be patched if the remediation required a later service pack or OS level.

Different Purchases, Different times

No organization purchases all of their hardware at the same time. Purchases are made as needed and as equipment depreciates or becomes unusable, it is either redeployed for lesser tasks or completely retired. A running joke in some of our organizations is that hardware never dies; it just finds new life as an office warmer. This periodic and even chaotic purchase schedule can break homogeneity in a hurry, and we often find ourselves in a struggle to get the older equipment back out of the organization. The upside of the emergence of the Y2K worry was that it gave us great ammunition to pry that old equipment out of computer rooms and labs.

Something that should be reiterated is that many of these machines are being administered by the users. These are technical engineers and many are qualified for the task, but many are not. Being an engineer doesn't necessarily make you the right person for the job, and remember that these engineers are usually fully time-allocated to do another job. So this brings another question to mind: WHY?! Why would these people want to administer this equipment in addition to their normal job? Here are a few reasons this happens:

- The IT organizations are understaffed.
  As much as we'd love to fully staff the organization, gearing up to administer and maintain 20,000 pieces of hardware would bankrupt the organization. Given the current economic climate, the organization is turning to reduction and optimization of resources, which forces us to keep admin/equipment ratios higher and to do more with less.

- The users need faster response times.
  With a shortage of administrators problem response times are slow. The customers often need very quick turn around for configuration changes, troubleshooting, or architectural decisions. This causes the users groups to seek faster methods of resolution, such as doing the administration of non-critical (and some not so non-critical) servers themselves.

- The user groups know their needs better than anyone.
  Many of the salable hardware solutions in telecommunications have very specialized architectures. Each permutation of new technology requires specialized knowledge that the IT organization cannot begin to keep up with. No one can understand the intricacies of those architectures better than the developers. What this means is that while the IT organization tends to focus on the macro-level issues (i.e. OSes, large applications, and filesystems), the engineering community focuses on the detailed micro-level issues (i.e. specialized hardware drivers and specific software configuration needs).

- The dumbing-down of OSes and applications.
  A phenomenon that has been occurring in the computer industry over its entire history is that applications and OSes have become simpler to use. Sometimes that simplicity can cause the users to overestimate their abilities. The first application that I noted this behavior with was the SQL database. Having worked for an organization whose entire existence centered around database machines and massive data warehouses, I can tell you that SQL databases are a full-time job to administer. That company had 8 full-time DBAs on staff just for the software development groups. In this company we also make broad use of databases, yet the DBA staff is extremely small. Has the job become that much less difficult? Have the tools become so sophisticated that the problems of data analysis and database design have become so easy that anyone

can do it? No. What has happened is the proliferation of computers into nearly every home combined with software like Microsoft Access. Most users have access to these applications, and they begin to believe that this is what the job really entails. Thus we have a broad collection of user administered SQL databases that eventually fall back into the overworked DBA group when the user realizes that they're in over their heads. This happens with OSes too. The typical engineer has (at least) one computer at home that they have setup and administered themselves. Sometimes this is a Windows-based piece of hardware, sometimes it is of a Linux variety. Again as the engineer maintains and administers that box, with all of the automated tools that come included, they begin to believe that this is all there is to the job. "Hey, I've administered my home PC for years, how hard could administering those servers be?" And thus, the amateur administrator is born. Rather than suffer the lag times required by an understaffed IT organization, discussions begin within their groups and they decide that they can administer their own equipment. Sometimes this works out for those organizations, and sometimes it doesn't. More than once we have bailed those organizations out when they realize how over their heads they are.

Obviously, there are many reasons customers turn to their own administrative staff. Frequently, the most successful organizations that attempt this are those that hire a professional staff of administrators, or at very least send some portion of their staff to training and make those engineers true administrators. This has happened in a few of the groups, but not most.

So what we have to this point is a pretty bleakly painted picture of the "rogue landscape". It became a problem that we had no choice but to overcome, but our choices came down to either "laying down the law" or finding a solution that would be agreeable to both the IT organization and the engineering groups.

Living with Rogues

When I first began to consider a solution for this problem, I decided that we surely could not be the only company experiencing this situation. So to start I decided to survey the field and look for the worst extreme I could find and their handling of the problem.

The organization I was looking for was one that had the highest percentage of rogue-style equipment on their network. Remember, we're defining the rogue as a piece of equipment that the IT organization doesn't administer, and to which it has no internal visibility. Without a doubt, the type of organization that best fits this profile is the broadband Internet Service Provider (ISP). Every piece of equipment outside of their central network infrastructure fits the definition of the rogue.

So how does an ISP handle this problem in the face of the current onslaught of exploits? Specifically, let's look at WideOpenWest, a large broadband vendor in the Midwest and across the U.S.

Here are some of the things that WOW does to protect it's internal network and it's customers:

1. Detection – they monitor their network segments for inordinate bandwidth patterns. This gives them the ability to shutdown problem segments if a particular port attack is found.
2. Focus on maintenance of their core network – WOW maintains their servers to current patchlevels and ensures that maintenance windows and downtimes are well publicized to their customers.
3. Secure the core network perimeter – HIDS and NIDS are placed around the WOW core infrastructure, protecting it from undetected intrusion.
4. Principle of least privilege – Servers that can be connected to by customers (i.e. POP3 and NNTP servers) are locked down other than those specific services. You cannot, for example telnet or FTP to these boxes. All unnecessary ports are locked down tight, so customers have no access beyond the server's intended purpose.
5. Customer education – The WOW homepage[4] is the standard homepage set when the service is installed. Obviously this can be changed, but customers are encouraged to check this page periodically for new information. The WOW homepage takes you to a local portal which always has new information on detecting viruses on your PC, and general tips on internet security. They also maintain links to current security updates for common operating systems.

   When your service is installed, they send one or two technicians who will not only wire your house, but will also explain the service, the WOW homepage as a resource and will install customized internet software if the customer desires.

6. Policy - The WOW Terms of Service[5], binds the customer to the contract in the first few paragraphs that contain their internet acceptable use policy. In the remainder of the policy, they require the customer to use the service in an ethical manner and give themselves the right to determine what violates that standard. They specifically disallow disruption of the service or altering of routing patterns which may affect others' use of the service. They also allow themselves the freedom to monitor customer network traffic and to penalize or disconnect customers who utilize excessive bandwidth. Again, they give themselves the right to redefine what denotes "excessive".

   Basically, WOW has given itself the authorization to take any steps necessary to protect its network, including disconnection of the customer.

They cannot directly touch your machine without your authorization, but they can watch what goes in and out of your machine and disconnect it if it appears to be broadcasting damaging content. During the MSBlaster events, WOW was selectively filtering and disconnecting infected machines until those customers got their equipment back under control. To aid customers who didn't have the technical know-how to correct the problem, most local new connections were put on hold while technicians made house calls to correct this problem.

Given this information, all that was left was to apply some of these same principles to the local environment and get those rogues under control.

The Applied Solution

To come to grips with this sizable problem we broke the solution into the following phases:

1. Identify ownership of all network attached equipment. Every piece of equipment must either have a responsible administrator/owner, or it is disconnected from the network. Unfortunately, that threat has to be followed up more times than you might expect. Disconnection does have the effect of drawing the owners out into the light.

2. Put a policy in place with the lab administrators. The policy should state that standard images are to be used where possible. Where standard images are not possible, each case should be well documented using the exceptions process. The exception process should contain dates (no matter how far out) that indicate when the exception will expire, and whether the equipment will be upgraded or retired at that time.

   When CERT announcements are made and vulnerability remediations are required, lab equipment owners are responsible for either patching for those problems, or writing the appropriate documentation for remediation exceptions. The central security organization will set the timeframes for remediation. If the remediation or exception does not take place within the allotted time, it is subject to disconnection from the network.

   The policy should also give the central IT organization full permission to scan and monitor all parts of the network including the rogue networks.

3. Focus on and secure the core network. All feasible measures should be taken to secure the perimeter between the lab networks and the core network. This includes firewalls, router ACLs, HIDS and NIDS. The goal is to minimize or eliminate any impact on the core network by any virus or worm that gets loose in the unsecured environments. If an attack is detected within the labs, depending upon the situation, the individual equipment, or in some cases the entire lab will be disconnected from the

main backbone.

4. Scan and monitor as much of the rogue environment as is feasible. The policy focused upon in #2 gave IT the right to scan those networks. Use it. Put intrusion detection systems in place to safeguard those parts of the network from the core network.

5. Principle of least privilege[6]. The lab networks will have network access controls and firewalls in place such that the lab environment will only be able to perform tasks that are relevant to the testing and engineering taking place in the labs. Things like email and the internet will not be available in the labs unless there is a specific exception written that describes why such services are necessary.

6. Customer Education. Perhaps one of the most important tasks of all, customer education supplies information to the engineers regarding the processes that were put into place and their role in them. It also explains the need for stronger security measures in the lab environment.

This is all a considerable amount of work, and due to financial restrictions step 4 was cut back, utilizing what scanning capability we already had and putting better monitoring tools in next years budget. Customer education has also gotten off to a slow start, but is underway.

Despite the cutback in the plan, the remaining steps had a significant impact on our ability to maintain a safe network.

The Results

In the end, it can be difficult to measure the impact of such a program, since you never truly know how secure your network is until someone tries to break in. One vulnerability exploit is rarely like another. If they were all the same, the job of a security practitioner would be a great deal easier. So comparisons between exploits are rarely a useful measure. The one thing that does bear measuring is the speed with which you can deploy a patch across the organization.

Prior to the policies regarding network rogues, on February 25, 2002, CERT released advisory CA-2002-04[7], which outlined a buffer overflow vulnerability in Microsoft Internet Explorer. The deployment of those patches took three months to accomplish in my organization. That was a highly unacceptable amount of time to allow this security hole to stay in place. If the exploit had been as pernicious as a Code Red, or the MSBlaster attack, the company's network could have been crippled in that amount of time.

After partial deployment of the above processes, on July 31, 2003, CERT released the "CA-2003-19: Exploitation of vulnerabilities in Microsoft RPC Interface"[8] and "CA-2003-20: W32/Blaster worm"[9] was released 12 days later on

August 11. The organization mobilized at first word of these active exploits and patches were deployed across the entire organization in 29 days. Within the security organization we consider that still to be an inordinate amount of time. But this is a significant improvement from earlier exploits. Our goal is to reduce this timeframe to 1 week for the core network and 2 weeks for the "unsecured" portion of the network. By maturing the relatively new processes further and putting more automated tools in place I am confident this goal will be met.

Another significant occurrence related to this process was that vulnerabilities scanning which was previously only done on the core network is now done on the rogue network as well. This proactive remediation makes the equipment inherently more secure. For the first scans that were done on this part of the network, there were more than 10,000 vulnerabilities identified. That number has been cut in half in six months and work continues to reduce it to run rate activity. This could not have been accomplished without the new processes in place.

Conclusion

It is amazing what we as administrators allow within our environments, either due to political pressures, financial shortfall or just going with the flow. As security practitioners, it can be a difficult responsibility to go against "the way we've always done it". It's much easier, after all, to quietly look the other way. The end goal of this exercise was to find a way to live with our rogues, but the end result was to redefine those rogues and the policies that surround them to the point where they were a lot less roguish. They went from being virtual ghosts and gremlins on the network to being well defined members of our network. The benefits far outweigh the effort required. All in all, it would appear that sometimes someone really has to tell the emperor that he's not wearing anything.

References

[1] CERT Coordination Center, "Securing Networks Systematically – the SKiP Method". http://www.cert.org/archive/pdf/SKiP.pdf (11 Dec 2003)

[2] Parker, Donn. Fighting Computer Crime. New York: John Wiley & Sons, Inc, 1998. 253.

[3] Cole, Eric. Hackers Beware. New Riders Publishing, 2002. 719-723.

[4] WideOpenWest, Inc. "WideOpenWest Home Page". http://www.wideopenwest.com (11 Dec 2003)

[5] WideOpenWest, Inc. "WOW! Internet – Terms of Service". http://www1.wowway.com/wowStory.asp?id=1010 (11 Dec 2003)

[6] Cole, Eric. Hackers Beware. New Riders Publishing, 2002. 719-723.

[7] CERT Coordination Center. "CA-2002-04: Buffer Overflow in Microsoft Internet Explorer". http://www.cert.org/advisories/CA-2002-04.html. (11 Dec 2003)

[8] CERT Coordination Center. "CERT® Advisory CA-2003-19 Exploitation of Vulnerabilities in Microsoft RPC Interface". http://www.cert.org/advisories/CA-2003-19.html. (11 Dec 2003)

[9] CERT Coordination Center. "CERT® Advisory CA-2003-20 W32/Blaster worm". http://www.cert.org/advisories/CA-2003-20.html. (11 Dec 2003)