[WP-003]

Whitepaper: Hiding an Intrusion Detection System (IDS)

A Theoretical Discussion on How to Play "Hide 'N Go Peek"

Version 040308

March 2004

Author: Bob Radvanovsky, rsradvan@unixworks.com
Contributors: Roger Kizior, Rick Larkin, Joe Fox

(A special thanks goes to those listed for being my "sounding board" on this project.)

Limited Liability Statement

In no event shall the author(s) be liable for any found errors contained herein or for any direct, indirect, special, incidental or consequential damages (including lost profit or lost data) whether based on warranty, contract, tort, or any other legal theory in connection with the furnishing, performance, or use of this material.

The information contained within this document may be subject to change without any notice. No trademark, copyright, or patent licenses are expressly or implicitly granted (herein) with this whitepaper.

Documentation pertaining to any security-related technical or proprietary information, its data and all information provided and contained within this document is considered proprietary in nature and subject to copyright protection and is intended solely for use by its owner. Additionally, this documentation is solely for the purpose of discussing managed and timed proxy servers that are heterogeneous to any networked environment, and are not dependent upon any specified architecture, hardware platform or its software.

The name "LINUX" is a registered trademark of Linus Torvalds.
The name "UNIX" is a registered trademark of The Open Group. [ref: http://www.opengroup.org/legal.htm#trademarks]
The name "Sourcefire" and "Snort" are registered trademarks of Sourcefire, Inc. [patent pending] / [ref: http://www.snort.org]

The name "Windows" is a registered trademark of Microsoft Corporation.
The name "Apple" and "Apple Macintosh" are registered trademarks of Apple Computer, Inc.
The name "SMC" and "SMC Networks" are registered trademarks of SMC Networks, Inc.
The name "EtherExpress", "EtherExpress Pro" and "Intel" are registered trademarks of Intel Corporation.

All other product names mentioned herein are trademarks or registered trademarks of their respective owners. NOTE: Any names not outlined or mentioned above are fictional in nature; as such, any relation to any name or trademark (if any) is purely coincidental.

Introduction

This document is an abstract notion in that it may be possible to (quite literally) "hide" an intrusion detection system on the secured-side of any given network. Without going into detailed lecture specific to monitoring both external and internal network traffic, intrusion detection systems are seeing a reintroduction into commercial networks as viable network management tools. Reasons for IDS environments are due to a combination of recent world events and the increasing number of cyber threats from Internet connectivity upon which the business community has grown dependent. IDS has mutated into an even more useful tool through the combination of some self-intelligence, firewall rule prevention and another newly introduced technology called "intrusion prevention systems" (or "IPS"). Product manufacturers of this technology have several variations for their name, but essentially, it is actually two technologies combined: an IDS console (and its sensors), and a firewall. Many IT professionals who have worked with and been around since the first inception of IDS technologies, perceive "IPS" as just another spin-doctored marketing facade to push more questionable products that may not necessarily be in the best interests in today's heterogeneous networked corporate environments.

As IPS has been noticed by corporate executives, IDS environments are being revisited, too. With common network attack signatures being recorded and analyzed faster than ever, IDS environments are gaining popularity again, but this time, not necessarily with the larger mega-corporations. It is gaining popularity with the small or medium sized companies that do not have a large or deep budget to support their enterprise, unlike most large corporate networks. Many industries, such as healthcare and finance, have recently come under regulatory scrutiny. They will soon be required to comply with very complex and strict privacy laws recently passed by the United States Congress. Thus, small or medium sized businesses that work with or support these two industries are starting to see IDS as one possible tool for their compliance use.

IDS in the Smaller Corporations

Since purchasing one IDS console/server configuration could potentially consume the entire year's budget for many small companies; this document will emphasize the utilization of an Open Source solution: SNORT. SNORT, as of this writing, is a freely available product from SourceFire, Inc. Several other applications have been developed over the years that work in conjunction with SNORT, thus providing additional capabilities for a better, more effective management of an IDS environment. SNORT has been cumbersome and difficult to work with in recent versions, but has made great strides in its management capabilities, mostly through several third-party products: SNORTCENTER and ACID. Utilizing a MySQL database back-end as the primary storage mechanism, SNORTCENTER and ACID combined, provide a very effective front-end for SNORT. Recently, a group of developers have taken away the difficulty of installing and configuring the entire environment and created a much-waited package distribution. It's called SENTINIX. This package is built using GNU/Linux (aka Debian) and includes several packages: SNORT, SNORTCENTER, ACID, PHP, Apache, MySQL, Postfix, Mailscanner, SpamAssassin, Nessus, Cacti – with more to be added in the future.

As with any IDS environment, data gathered and stored on the IDS console server is vital to the business. The company's data retention policy (or lack thereof), will determine the level of the data's criticality and what the business must do to ensure the safety and integrity of the data. Obviously, retention of this data is necessary for forensic management as part of an incident investigation, and would be needed for problem determination, etc. The safety of the IDS console server is vitally important to the ongoing safety of the enterprise.

Since IDS servers are statically assigned an IP address, an idea came to mind. Since over 70% of all network-based attacks originate from internal sources, wouldn't it make sense to protect the IDS server from the inside and well as from the outside? Aside from protecting the IDS with a firewall (which would now make it an IDS-hybrid firewall, or IPS), what if the IDS had some additional intelligence added? One of the more common hacking methods that may signify a precursor to a possible network-based attack is a scan for open ports either for specific addresses, subnets, or all addresses for the entire network. The entire idea behind IDS is a preventative measure of passively monitoring for any attack, even ones as common as the port scan. Yet, monitoring for internal attacks are rare to non-existent.

Blackbird IDS?

Stealth is a primary concern for the IDS environment. Stealth of the sensors is necessary, but (using the analogy of mining gold) not only is the gold mine important, but so is the transportation of the raw gold ore, the processing center, and finally, the gold vault. It is that vault that contains the items that so many want. The same way that an intruder would know that his/her footsteps were heard walking into someplace where they didn't belong, an intruder would want to remove any proof of their trespassing, making any evidence of any of their infiltration attempts or network reconnaissance as valuable as gold itself.

So what am I pushing here? Quite simply, the notion that perhaps it is feasible to introduce yet-another-stealth method by utilizing DHCP or some pooled configuration of IP addresses to effectively protect the IDS server for when it comes under heavy attack. This becomes useful when an internal attack against a critical piece of the network infrastructure cannot easily migrate to another IP address easily. Or can it?

A close friend of mine and I attended a sales demonstration of a newly introduced commercial-grade firewall/IDS/VPN/all-in-one security product from one of the network security manufacturing vendor recently. Despite efforts to block external-going-into-internal network traffic, there appeared to be very little preventing internal traffic from attacking this server. The two of us looked at each other with surprise, and both of us muttered the same thing: "Hack attack from the inside!" How would this work? What measure or level would be necessary to ensure that this does not happen? Corporate management continues to think and operate at a level that it is necessary to protect and maintain only external perimeter defenses of the corporate environment at any and all costs. What these executives continue to fail with is that, although the business goals and directives continue to remain steady and consistent, the awareness of newer and emerging technologies as potential threats continues to be extremely lacking. A prime example is the introduction of the wireless access point (i.e., a wireless "router") into corporate network environments at the local, departmental, or even sub-departmental levels. If not correctly configured upon installation, these poses serious risk and, depending upon the location of the access point, could cause serious, detrimental, or perhaps fatal business consequences. This lack of awareness demonstrates the ill preparedness of most large corporations today, and how well they can not only manage and maintain their environments, but provide effective and immediate resolution in a critical scenario (as outlined above).

This resulting mindset permeates and propagates throughout the corporate social and political structure. It ensures an almost guaranteed success rate for failure in the event of a catastrophic security-related event, such as that soon-to-be-internal breach from within that corporate network environment that didn't pay attention closely to their departmental networks, such as through the injection of malicious network attacks via the not-so-widely-known access point that was interconnected by one of the local network or systems administrators. As a corporate environment grows, so does the risk of either continued or added damage to or unauthorized access to the environment. Risk assessments in today's networked environments are becoming standard for companies that support critical infrastructure (financial, healthcare, transportation, food production/processing, utilities [electric, gas, water, sewage], municipalities, etc.); however, companies want to find more effective means of not only monitoring, but automating the monitoring process down to a simplified graph or report that corporate management can easily digest.

But just like any other well-intentioned plans, there are always obstacles, despite what may come from it. The internally monitored environment would need to be "smart enough" (the IDS has to be able to "think" that it is under attack) to have some form of self-preservation mechanism, yet continue to operate, collect, sort, and maintain the IDS data within its "gold vault".

How the Process Works

The proposed process is quite simple: if the IDS server receives an excessive number of network packets that appear to be malicious or have the intent of subverting the environment, the IDS shuts down its network links to the internal/secured-side of the network, whilst maintaining its connectivity with its sensor units, either remotely or locally.

After a period of time when the attacks have either subsided or migrated to some other location, the IDS server will re-establish its server connectivity to the internal/secured-side of the network. If, after reintroducing itself back into the corporate network environment, the subverting attack resumes, the IDS shuts down its network links again to the internal/secured-side of the network, issues or reassigns a new IP addressed number, either within the same subnet, or some other network location, either through a DHCP server for a "special IP address pool" that is assigned by the DHCP server, or a random "address pool" internally within the IDS server, re-establishing its network link again, but to a different location.

After the attack subsides and the IDS network links have become active, a notification is sent to either to the IDS administrator or administration group, usually via pager or internal email, in a plaintext cryptic message that states that the "for a good time, call me at 215-80" (the message is obvious that the 2 numbers are the last 2 octets from an IP address) message – or something similar to that effect. In many cases, if a more plain-as-daylight, yet obscured message is sent, one at a time, to everyone concerned, it will only appear as a violation had occurred for those individuals in which the email was sent, rather than a notification that the IDS server was just under attack, and now resides in a different virtual location of the corporate network. It's that simple!!!

All of this technology can be easily integrated with already existing technologies, thus the cost of implementing such an endeavor is much lower than introducing something that would require a much more significant investment of resources. As we promote and embrace the "Open Source Initiative", this too, can be easily implemented using already developed products that could simply be patched using several applications currently available (with permission from their authors, of course).

For this whitepaper, since SNORT was mentioned, it would have multiple network links to several locations, preferably secured. If a malicious or undermining attack commences on only 1 link, then risk is minimal, and the IDS server observes network traffic based upon the intent of the attacker. If it continues, then within the IDS server, separate from the SNORT environment, another application would monitor for port scanning activities, etc. Once a determination is made that a malicious or subverting event was occurring, this would then cause an alarm and trigger the necessary course of action, thus protecting the IDS server.

Another analogy would be similar to how Vietcong (often referred to as "VC") soldiers operated and how they met threats during the Vietnam War. When a VC member came under attack by either an enemy of the VC or threatening scenario, that soldier would dive into a small opening or hole. They would wait until the threat left. As much as the enemy of the VC or threatening scenario might have been, the likelihood that the soldier was harmed was slim to none. Once the threat diminished, the VC soldier reappeared again, usually unscathed. If the threat came back, either from the same enemy, a threatening animal, or even from a different threat, the VC soldier would utilize one of its other openings to hide, and while the threat existed elsewhere, the VC soldier would reach safety through a different opening. It is that very concept of playing a form of "shell game" with the would-be attacker that would lead them into thinking that the IDS server's links may not show up on the same network now has introduced a guessing probability into the equation. This randomness factor helps reduce the risk provided that the IDS does not reappear on the same subnet that was currently under attack!

If the would-be attacker were to accurately guess where the IDS would show up next, this would pose a serious risk to the IDS server; however, given the fact that many larger enterprise corporate network environments, which consist of tens of thousands of addresses, would make the "hide 'n go peek" that much more challenging to the attacker(s).

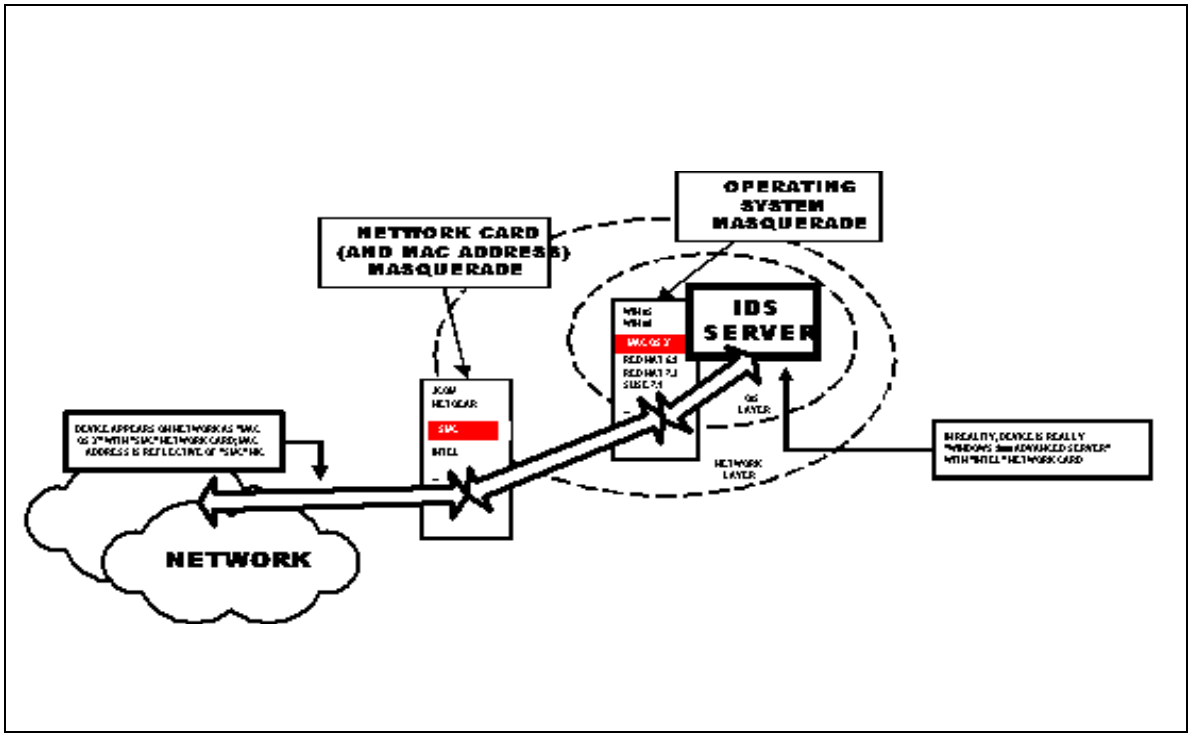
Technologies do exist which could be elaborately combined together to create a single-point distribution, thus permitting the game of "hide 'n go peek" with would-be attackers. These tools would consist of some of the more common utilities (such as DHCP), but others would be part of something of a much grander/larger scale. In a larger case, the DHCP server is one of several utilities that would be applied towards this newer way of stealthy thinking and operations. Utilizing an aggregation of commonly applied technologies would be cost-effective in its implementation, provided that the corporate management can conceive of and will agree to such a mechanism.

Addendum

One problem, however, are that small and medium-sized companies usually do not have more than a few internal network segments; many that I have encountered over the years are clumsy and awkward in design, and have multiple points of entry into internal segments, which are usually left unprotected. An alternative method of network countermeasure was that the "stealth IDS" could utilize other masquerading methods and techniques to fool and trick port scanning and/or network fingerprinting mechanisms (such as NMAP) into determining that the scanned device is operating on a different operating system and/or network interface care (NIC).

One possible example of such a configuration exists on the next page, the IDS is masquerading as a Apple Macintosh, running with the "Apple Macintosh OS X" operating system, whilst both the network interface card (NIC) is representative as a manufacturer type of "SMC" (with corresponding MAC address manufacturer header to correlate with the NIC). In reality, the IDS device may be something other than what was listed above; for this example, it is really a server running with the "Microsoft Windows 2000 Advanced Server" operating system, whilst the NIC is really an "Intel EtherExpress Pro 10/100" NIC. All of this technology is capable of properly fooling/tricking as a countermeasure to network probe attacks.

A graphical version of the masqueraded network configuration is shown below.



This is not the only method of camouflage, however, this is one possible method of protecting the IDS environment. It is with this way of thinking that we can expect "stealth IDS" environments in the future.