



White Paper

Protocol Anomaly Detection

September 9th, 2004

Overview

The PORTUS Application Protection System functions as an in-line Network Intrusion Prevention System (NIPS) and firewall. PORTUS delivers in-depth protection against known and unknown forms of attack. Protocol Anomaly Detection (PAD) detects and blocks previously unknown forms of attack without the need for signatures. PAD provides Zero-Hour™ protection, which means new forms of attack are blocked the instant they reach the PORTUS gateway. One does not have to wait days for the latest attack signature to be identified and downloaded for use. Even unknown viruses and worms are stopped.

In 2003 more than 80% of the successful Internet attacks were at the application level, having passed undetected through the best of the stateful packet filter firewalls. These attacks targeted application data streams that are buried deep within the payload portion of the transmitted packets. Stateful packet filters only examine protocol headers allowing the malicious content to pass undetected. These malicious payloads are designed to exploit weaknesses in the targeted system. If the targeted system is vulnerable to the attack, it will succeed. The damage done by application level attacks has been estimated in the tens of billions of dollars per year.

Standards violations: Many programmers too often assume that clients or servers will abide by the application standards they were developed and designed upon. Unfortunately it is too often the case that these programs perform no internal checks to watch for or prevent inappropriate commands and behavior. In general many clients and their servers provide an insufficient level of checking for compliance, which leaves them open to exploitation by malicious code and attacks.

Stateful Packet Filters & Intrusion Detection Systems

Due to the shortcomings of stateful packet filters, network administrators have deployed intrusion detection systems (IDS) to supplement their firewalls. The rationale behind this approach is that the IDS will catch the attacks that have penetrated the firewall. Unfortunately IDS solutions have only met with partial success due to their inherent limitations, some of which are:

- ▶ Signature based rules that are good for detecting known attacks but are poor at blocking new forms of attack
- ▶ A robust set of IDS rules often produces large numbers of false positives, forcing the network administrators to spend an excessive amount of time in event correlation analysis and rule tuning.
- ▶ Increasing the number of rules in an IDS increase the number of false positive alarms.
- ▶ Restricting the rule set to avoid an excessive number of false positives weakens the ability of the IDS to detect real attacks. There is a delicate balancing act between generating too many false positives and letting malicious attacks to pass undetected.
- ▶ Increasing the number of rules in an IDS reduces the bandwidth of the device. At some point the IDS will not be able to examine all of the packets. If the IDS is an out-of-band device then malicious traffic will pass undetected. If the overloaded IDS/IPS is an in-line device then packets will be dropped and require re-transmission. The network then becomes sluggish.
- ▶ Polymorphic attacks such as ADMutate and multi-function attacks such as Nimda require multiple signatures exacerbating IDS performance problems.
- ▶ Activation of a robust signature based rule set can inadvertently block legitimate traffic.
- ▶ Simply detecting an attack and alerting the administrator does not stop an attack in time to prevent a security breach. The IDS must automatically alert a firewall or IPS to block the attack.
- ▶ packet level IDS that signal a SPF firewall to modify its rules will fail to block attacks that fit within a single packet. By the time the blocking rule is in effect the malicious payload will have been processed by the target system.
- ▶ packet based IDS systems will fail to recognize application level attacks that violate the applications protocol using such methods as out of sequence commands. Application state is hard to maintain for a packet based solution.
- ▶ Out-of-band IDS systems will fail wide open, which is the worse possible thing that can happen to a security system. A truly secure system will always fail shut. Otherwise attacks can and will go undetected. A far better approach is to use a fault tolerant in-line solution that has less than 6 minutes of unscheduled down time per year (99.999% availability).
- ▶ IDS systems run as part of the kernel which is dangerous. If there is a weakness in the IDS code an attacker will be able the vulnerability to corrupt the security and integrity of the underlying system. In this case the IDS will fail open, allowing subsequent attacks to go undetected.

The fundamental question a network security administrator should ask is what percentage of the attacks will my IDS detect and what percentage of the attacks will my IPS block? Unfortunately signature based systems suffer from so many false positive alarms that administrators are more concerned about the percentage of false positives. The fact that the administrators are concerned with the wrong question implies the entire approach to solving the problem is wrong. Fortunately it is technically feasible to solve all of the afore mentioned problems as well as many not mentioned. A fault-tolerant, in-line, multi-method Intrusion Prevention System can accurately detect, block, and report known and unknown attacks targeting a network while operating at gigabit speeds.

Protocol Anomaly Detection (PAD)

In order to safeguard and secure various applications and their related programs, servers and clients PORTUS utilizes Protocol Anomaly Detection. Protocol Anomaly Detection (PAD) works by analyzing application-level traffic, commands and behavior, blocking and denying undesirable or otherwise inappropriate commands.

Applications protocols are published in RFCs and vendor documents. The description of the application protocol can be used to check for proper or expected behavior. In fact, this is a simple and elegant approach to solving the problem. It is far easier to define and check correct behavior than it is to look for all the possible ways to define incorrect behavior. The number of applications and RFCs that define their behavior is relatively small and changes slowly. On the other hand, the number of signatures required to detect the known forms of attack requires frequent signature updates. Worse yet automatic downloading and application of new signatures can lead to unintentional applications disruptions caused by false positive alarms. There might be several ways to perform a function correctly, but there could be hundreds of ways to do it wrong. As a result signatures are constantly be added or updated as hackers discover new ways to beat the system.

The beauty of PAD is a simple check can catch many forms of attack that all depend on the same type of protocol violation. This approach has many advantages: (1) Fewer rules are required to block the attacks; (2) New forms of attack will be immediately blocked without requiring the downloading of a new signature; (3) Command blocking can depend on the current state of the application, eliminating attacks based on out-of-sequence commands; (4) Commands can be restricted based on authenticated user ID, time of day and other application state information.

For example, there is a maximum size for all SMTP commands. A simple size check will catch all attacks that depend upon a buffer overrun to perform a DoS or insert malicious code. In this case, one simple check can replace dozens of signatures used to detect different attacks that all depend on the more fundamental problem, which is the violation of the protocol. Another e-mail example is the insertion of executable code into an e-mail header line. Since all e-mail addresses consist of a limited set of printable characters, a simple check will detect any executable code. A signature based check will only find one instance of a many possible programs.

PAD in many cases requires less processor overhead. With signature based checking, the client commands will have to be compared with all the signatures. As time goes on this list will become longer and slower. PAD can be implemented so that the types of checks performed will depend on the applications state, thereby minimizing the number of checks and associated processor time.

If PAD is so powerful one might ask why hasn't it been used before. In fact protocol anomaly checking has been successfully used to protect a limited number of applications for more than tens years. The PORTUS Application Protection System (APS) has successfully detected and blocked all forms of attack against SMTP and FTP protocols for the past ten years using PAD. This is a truly remarkable record that is unequalled. . These proxies have demonstrated the validity of the approach. The ability to detect new and unknown forms of attack was demonstrated in the first quarter of 2003 when PORTUS blocked a new attack using a protocol check that was in place in 1996, seven years before the first appearance of the attack. In the past year limited protocol analysis as well as attack signatures has been added to the PORTUS reverse http proxy. The reverse HTTP proxy blocks unsafe UTF-8 characters and URIs that can not be converted to conical form.

Today the application protocol checking for SMTP, FTP, and HTTP is hard coded in the respective proxies. Application protocol checking for POP3, and NNTP is hard coded in functions called by the API found in the generalized application proxy. The current implementation works well. It has proven to be very secure (unbreached for 10 years) , very fast (multi-gigabit) and highly reliable (99.999%).

Protocol anomaly detection will catch many forms of attack that escape detection by signature based IDS. However, PAD does not detect attacks which do not violate application protocols. Viruses and worms contained in an e-mail attachment is a good example. What is needed is a hybrid solution that employs multiple forms of detection and blocking.

No single detection method can provide comprehensive coverage of all possible attacks. However, a hybrid solution employing multiple methods can achieve much better results than what is used today. The FAS solution uses the most appropriate method for each type of attack. The methods include, application protocol anomaly detection, application state analysis, stateful pattern recognition, fine grained sub-command authentication, traffic anomaly detection with traffic shaping. Anti-IDS evasion techniques such as IP packet fragmentation are automatically defeated by our architectural solution. TCP packets are assembled into messages before analysis defeating attacker obfuscation attempts, and low level network attacks.

The most effective way to protecting networks from application level attacks requires an architectural approach different from those used by existing network firewalls and IDS systems. Solutions based on architectures that deal with individual packets, such as stateful packet filters, are not well suited to protecting against application level attacks. An IPS using an architecture designed to process messages after they have been assembled from one or more packets has many significant advantages. It is much easier to design, therefore more likely to be correctly implemented. It permits a fault tolerant design that can detect, isolate and dynamically recover from both software and hardware errors, making it immune to attacks directed at the IPS itself. This architecture is immune to many low level network attacks including: those that depend on IP packet defragmentation, anti-IDS evasion, data stream obfuscation and so on.

For more information on PORTUS please contact us at:

Freemont Avenue Software, Inc.
1830 S. Kirkwood Suite 205
Houston, TX 77077
Phone: 281-759-3274
Fax: 281-759-8558
e-mail: portus@lsli.com
Web: www.lsli.com