

Total Application Security with the SecureSphere Dynamic Profiling Firewall

A Practical Approach to Defending the Application Data Center

Web applications such as e-commerce, supply chain management and online banking have become the backbone of business process in nearly every segment of the economy. Unfortunately, by making data more accessible, Web applications have dramatically increased the business risks associated with that data. Web applications are vulnerable to a complex mix of threats, including targeted Web-based hacking from individuals outside the organization, unauthorized internal database breaches, and worms of internal or external origin.

Traditional security infrastructure technologies such as network firewalls, deep inspection firewalls and intrusion prevention systems, while effective against simple application threats, do not protect against the most serious attacks threatening application data centers today. This paper provides an analysis of the application data center threat environment, followed by a description of how Imperva's SecureSphere™ Dynamic Profiling Firewall™ provides a comprehensive and completely automated platform for securing these important IT assets.



Web Application Threats

Web applications are vulnerable to a wide variety of threats from both internal and external sources. These threats include targeted Web hacking, unauthorized direct database breaches and worms.

Targeted Web Hacking

Targeted Web hacks involve an individual attacker located outside the perimeter firewall, attacking the Web interface of an application. While targeted attacks sometimes take advantage of known flaws in commercial software such as IIS and Apache, hackers typically have no advance knowledge of a specific vulnerability within the targeted application. Therefore, targeted Web application hacks require a systematic process of reconnaissance and exploit construction that focuses on the custom application logic linking the Web interface to back-end databases. It's these vulnerabilities in custom code that can lead attackers to specific valuable data such as customer credit card numbers, personal identification information and the like. Two classic examples of targeted Web attacks are cookie poisoning and SQL injection.

Cookie Poisoning

Cookie poisoning attacks involve manual modification of an application cookie to gain unauthorized information or to impersonate another user. For example, many applications assign cookies to identify users as they move from one area of a Web site to another. By guessing (through trial and error) legitimate values and manually changing their cookie to that of another user, attackers can hijack another user's session.

SQL Injection

SQL injection attacks exploit input validation vulnerabilities to insert arbitrary SQL commands into a Web application for execution by a back-end database. Using this method, hackers can gain access to the entire contents of a database.

There are many more examples of common vulnerabilities in custom application code that can be exploited with serious consequence. For more information, see http://www.imperva.com/application_defense_center or <http://www.owasp.org>.

Internal Database Breaches

Web application databases are also vulnerable to direct internal attacks by renegade employees or partners. This type of attack may be accomplished in a number of ways, including unauthorized access to default accounts, brute-force password attacks, general infrastructure attacks against the network protocol stack, or through the use of generic (unauthorized) database clients to execute malicious commands.

For example, a help desk administrator might use a generic SQL client such as SQL Plus to steal customer information or even delete an entire database table.

Worms

Worms are malicious programs designed by hackers to attack previously known vulnerabilities in commercial software platforms. While worm attacks do not generally expose proprietary or confidential information (as targeted Web attacks do), they do result in denial of service by crashing target systems or consuming available bandwidth, bringing business operations to a grinding halt.

Worms are usually designed to take advantage of multiple mechanisms to first penetrate corporate security defenses and then propagate from host to host. Typically, worms will infect corporate desktop machines via e-mail attachments and then spread to other internal hosts by scanning for open ports on local IP addresses. Worms will also attempt to traverse perimeter firewalls by masquerading as legitimate Web traffic on Port 80.

Total Application Security

For any organization serious about protecting their Web applications, building a comprehensive security architecture to defend against targeted Web hacking, unauthorized internal database breaches and worm attacks is a multi-dimensional challenge. No single technology protects against all three threats. Network firewalls, deep inspection firewalls, intrusion prevention systems and dedicated application firewalls are all necessary to provide complete security. This section analyzes the strengths and weakness of these existing enterprise infrastructure security technologies with respect to the application attack vectors described in the previous section.

Network Firewalls

Network firewalls traditionally provide network layer access control and attack protection services. They have been uniformly deployed at the network perimeter to protect against Internet threats. However, more recently, mature security architectures also call for network firewalls to be deployed in front of critical internal network segments to protect against internal threats. Network firewalls are critical, particularly in front of application data center segments. Not only do they protect against internal network-layer hacking (network scanning, telnet, etc.), but they also prevent the spread of worms from corporate desktops to application servers via non-essential ports.

However, while network firewalls do a good job of preventing network-layer attacks and worm propagation via non-essential ports, firewall rules explicitly allow essential protocols such as HTTP and SQL (and usually a few others) unrestricted access to the application data center. Over time, the hacking community has learned to use these rules to their advantage by embedding attacks into otherwise legitimate application layer protocols. Code Red and SQL Slammer are examples of worms that spread via protocol-compliant communications. SQL injection and cookie poisoning are similar examples of targeted Web hacks that go completely unnoticed by firewalls. As long as an attack is carried out via commonly allowed protocols and meet basic protocol requirements, firewalls are an ineffective defense.

Deep Inspection

The broader security industry has responded to the need for a deeper understanding of application layer behavior with a technology commonly referred to as Deep Inspection. Deep Inspection looks at the contents of a packet's data payload and compares it to a list of known attack patterns (or signatures) that are derived from documented vulnerabilities in commercial software. Deep Inspection technology can also enforce protocol restrictions to protect against known protocol vulnerabilities.

Since virtually all worms to date have been based on known vulnerabilities in commercial software, Deep Inspection can be an effective worm defense and, therefore, a useful component of a comprehensive application data center security architecture. It has been implemented in the form of dedicated Intrusion Prevention Systems (IPS) and is also being integrated into Deep Inspection firewalls that combine network firewall with Deep Inspection capabilities, providing organizations with a choice of standalone or integrated solutions.

Unfortunately, Deep Inspection is ineffective against targeted Web hacks and unauthorized database breaches. Targeted Web hacks prey on vulnerabilities of custom, internally developed application logic, while database breaches rely on database configuration and access control flaws. Since each attack is unique to each targeted application, there are no known signature or protocol restrictions to apply to these threats¹.

¹ Although some Deep Packet Inspection systems claim to defeat certain targeted attacks such as SQL injection and cross-site scripting, these claims should be treated with caution. Deep Packet Inspection products rely on individual signatures to look for patterns that are commonly used as part of an SQL injection or cross-site scripting attack. These patterns, however, are words such as "union," "select" and "script," which are prone to false positives since they commonly appear in normal Web site content. Therefore, these signatures are usually not enabled, leaving the application open to these attacks. Even if these signatures are enabled, they can be circumvented using well-known evasion techniques.

Application Firewalls

A new category of products called application firewalls is emerging to protect against vulnerabilities in custom, internally developed Web applications and databases. Application firewalls are network-based security devices that sit in front of application servers, identifying behavior that falls outside a normal application behavior profile. All such behavior is deemed to be malicious and blocked². Application firewalls are typically specific to a single application protocol such as HTTP or SQL.

Web Firewall

Web application firewalls rely on profiles of normal URLs, parameters, parameter lengths, parameter types, session IDs, etc. that make up a Web application. If a user attempts to access a URL or enters a parameter that is not part of the Web application profile, that user may be blocked.

Database Firewall

A database firewall relies on a profile of normal SQL Queries from different user groups. A database administrator, for instance, may be allowed to access certain tables with certain queries, while Web application servers are authorized for other queries. If a developer attempts a query that accesses the “Customer” database table using the “delete” SQL command, and that query is not part of the developer’s allowed profile, then the developer will be blocked.

The biggest challenge to implementing application firewalls is building and maintaining an accurate profile over time. A profile for a single application may contain thousands or even millions of variables (URLs, parameters, cookies, SQL queries, etc.). To make matters worse, application developers change these variables on a daily basis. Given this level of complexity and change, expecting a human administrator to manually create and maintain an application profile is unrealistic. Any practical application firewall must completely automate the creation and ongoing maintenance of an accurate profile. Unfortunately, most application firewalls have not adequately addressed this fundamental challenge. Instead, they force administrators to manually configure and tune profiles to an extent that does not scale in real-world application environments.

Practical Issues: Management and Availability

Comprehensive protection against all the attack vectors described above requires up to four different products: a network firewall, an intrusion prevention system, a Web firewall, and a database firewall. Each of these solutions introduce incremental management overhead that, when combined, can result in overwhelming total cost of ownership (TCO). Product selection, purchasing, training, deployment, configuration, ongoing policy management, logging, monitoring and reporting functions must be duplicated for each device. Therefore, it is important to evaluate the management characteristics of each solution to avoid an architecture in which TCO outweighs security benefit.

Application availability poses a similar challenge. Each inline security device introduces a single point-of-failure risk that threatens the availability of the Web application. In the application data center, this risk is compounded by extreme sensitivity to downtime (e-commerce downtime, for example, can be very expensive) and the need for multiple inline security devices. To eliminate this risk, it is critical to also evaluate the availability characteristics of each component. In many cases, redundant systems are required for each protected network segment, although in some cases fail-open and sniffer-type products may offer more cost-effective and scalable alternatives.

² This general technique of allowing all behavior defined as “good” and blocking everything else is frequently referred to as a “positive security model.” It is also used by basic network firewalls. Deep Packet Inspection / signature detection solutions, on the other hand, use a negative model which blocks behavior defined as “bad” and allows everything else.

SecureSphere: The World's First Dynamic Profiling Firewall

SecureSphere, the world's first Dynamic Profiling Firewall, provides Total Application Security™ for enterprise Web applications by protecting them from all critical threats, including targeted Web hacking, worms and unauthorized internal database breaches. Imperva's Dynamic Profiling™ technology serves as the foundation for a suite of security services that address each of these threats without manual configuration or tuning. These services include the following:

- Dynamic Web Firewall
- Dynamic Database Firewall
- Web Worm Profiling
- Deep Packet Inspection
- Network Firewall

In addition, Imperva's Correlated Attack Validation (CAV) technology provides a critical link between these SecureSphere security services by correlating user activity over time across multiple services. By basing security decisions upon multiple activities rather than a single event, CAV is able to identify attack reconnaissance activity with a level of accuracy that is not possible via independent security services.

SecureSphere appliances can be deployed in sniffer, inline/fail-open and inline/high-availability modes, depending upon cost, security and phase of deployment requirements. A three-tiered management architecture enables centralized policy distribution, logging, monitoring and reporting that scale across large, multi-application environments.

Network Deployment Architecture

SecureSphere consists of two main components: the G4 Gateway and the MX Management Server appliances. The SecureSphere G4 Gateway is deployed in the data path of critical application servers to block Web, worm and database attacks. The MX Management Server provides centralized policy management, logging, monitoring and reporting.

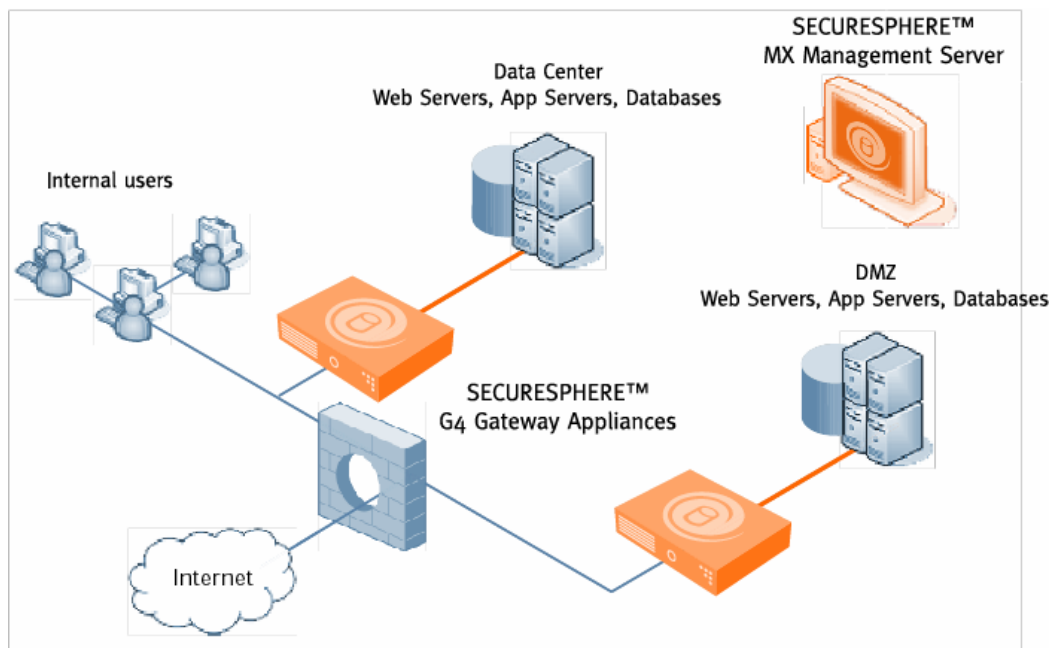


Figure 1: SecureSphere G4 Gateway and MX Management Server deployment.

Dynamic Profiling

At the heart of SecureSphere's ability to deliver automated security for changing applications is Dynamic Profiling. Immediately upon installation, Dynamic Profiling begins monitoring all interactions between users, Web servers and databases to automatically build a profile of the application's "normal" behavior. By comparing the profile to actual traffic, SecureSphere is able to identify and block potentially malicious activity of any kind. Persistent learning algorithms automatically detect changes to the application, ensuring that the profile is accurate over time. This continuously updated Dynamic Profile serves as the foundation for all SecureSphere services.

Web Firewall

SecureSphere's Web Firewall service relies on the Web elements of the Dynamic Profile to protect enterprise applications from targeted Web hacking. The Web Firewall profile elements include legitimate URLs, HTTP methods, parameters, cookies, response codes, hidden fields, etc. The profile includes both HTTP request and response information.

Based upon this deep understanding of normal user activity, any unusual behavior can be quickly recognized and blocked. For example, if a hacker attempts to manually change a session ID cookie in order to hijack a user session, SecureSphere will immediately identify it as a profile violation and block that specific user. SecureSphere's Web Firewall similarly protects applications from any attack targeting the OWASP Top Ten Most Critical Web Application Vulnerabilities (<http://www.owasp.org/>).

Database Firewall

SecureSphere's Database Firewall service relies on the database elements of the Dynamic Profile to protect enterprise applications from unauthorized internal database breaches. The database profile elements include SQL queries, user names per SQL query, IP addresses per query, etc. Based on this profile information, the database firewall can block or alert on any unusual queries from internal or external sources. For example, if a developer attempts a query to access the "Customer" database table using the "delete" SQL command, and that query is not part of the developer's allowed profile, the developer will be blocked.

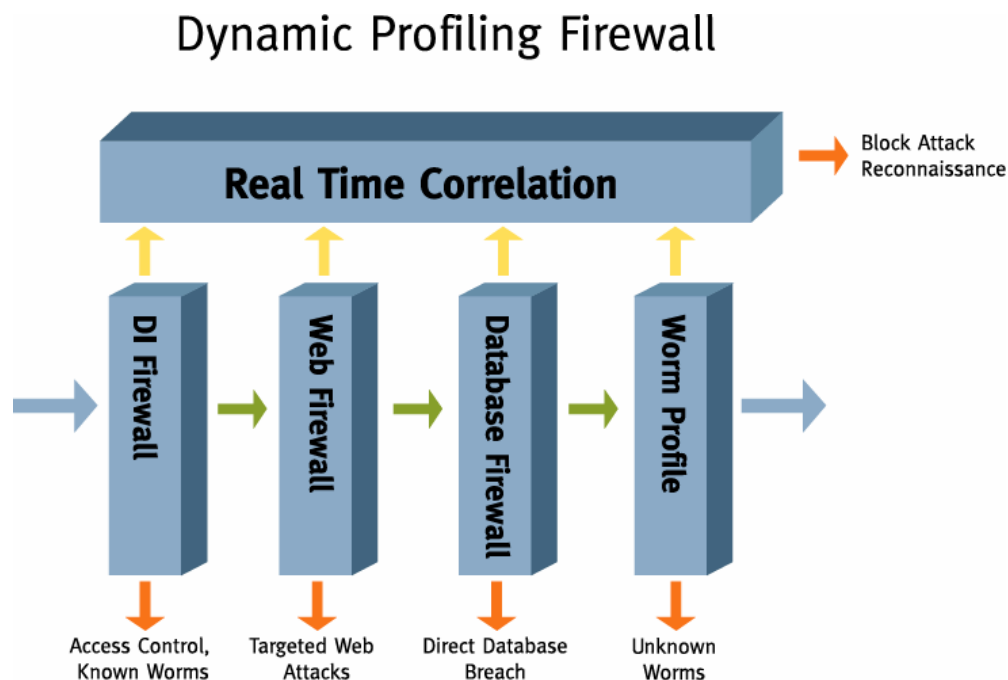


Figure 2: The SecureSphere architecture delivers Total Application Security.

Web Worm Profile

SecureSphere's patent-pending Web Worm Profile service provides an extremely accurate and completely automated defense mechanism against Web-based worm threats without the use of signatures. Accurate detection is achieved by correlating a combination of Web profile violations that uniquely identify Web-based worms. These violations include the following:

1. **Unknown URL:** Web worms are not part of the normal Web site structure.
2. **Default Directory:** To spread effectively, Web worms target URLs that exist in a large number of Web servers. These URLs correspond to default directories.
3. **Invalid Session IDs:** Web worms do not negotiate legitimate session IDs.
4. **Invalid Host Name:** Web worms do not include legitimate host names since they do not know the names of the hosts they attack. They spread by enumerating IP addresses and typically use generic text as a host name.

Any attempt to access an unknown URL in a default directory, with no legitimate session ID, and without a real host name, is blocked. This technique is highly effective against ALL Web-based worms, including the infamous "day zero" worm, since it does not rely on signatures. No advance knowledge of a particular vulnerability is necessary.

Other Web security architectures claim day zero worm defense. One such architecture uses code simulation techniques to identify potentially malicious programs. Unfortunately, this approach is prone to performance and accuracy problems for the following reasons:

- Code simulation in real time is extremely resource-intensive, resulting in prohibitive performance penalties
- Identifying executable code in traffic streams is subject to false positives and false negatives
- Identifying malicious behavior in simulated code is subject to false positives and false negatives

Since the worm profile requires the simultaneous occurrence of four independent profile violations, it delivers accurate detection with very little incremental resource requirements. It is an excellent example of the power of the Dynamic Profile as a foundation for advanced security services.

Deep Inspection Firewall

SecureSphere also includes full Deep Inspection Firewall services to protect application data center resources from known worms and network-level threats. These services include network firewall, signature detection and Web protocol compliance checks.

Network Firewall

SecureSphere's network firewall service applies network layer access control to traffic flowing into and out of protected network application segments. Both white listing and black listing are supported to prevent unauthorized access to critical servers from internal users via dangerous protocols such as Telnet, PC Anywhere, or even SQL. The firewall also plays a critical role in a comprehensive worm defense by preventing the spread of worms from internal user desktops to protected network application segments via non-essential ports.

Signature Detection

SecureSphere provides full Snort-based signature detection to protect applications from worms that target known vulnerabilities in commercial infrastructure software (Apache, IIS, Oracle, etc). The Snort™ database is enhanced with contextual information from Imperva's Application Defense Center (ADC) that includes attributes such as affected systems, risk, accuracy and frequency. Using this context and SecureSphere's Signature Center wizards, users can quickly isolate custom signature dictionaries for their specific environment. The ADC automatically updates signatures over the Internet. The ADC also provides its own advanced signatures, which are primarily designed to provide critical inputs to SecureSphere's Correlated Attack Validation engine as a mechanism for detecting reconnaissance related to targeted hacking.

Protocol Compliance

SecureSphere protocol compliance checks ensure that HTTP protocols meet RFC and expected usage requirements. For example, SecureSphere can check for malformed URLs, abnormally long URLs, abnormally long header lines, and many other protocol anomalies. By ensuring that the HTTP protocol meets guidelines, protocol compliance prevents worm attacks on vulnerabilities in commercial Web server implementations.

Correlated Attack Validation

SecureSphere's Correlated Attack Validation (CAV) engine tracks user activity over time and across SecureSphere security services (Web firewall, database firewall, signatures, etc.), looking for links between events that individually appear innocent but, taken together, definitively indicate malicious attack reconnaissance.

For example, the word "union" (an Imperva™ ADC signature) in a URL might suggest an SQL injection attack. On the other hand, it could simply mean that a new URL including the word "union" has been added to the Web site. Therefore, while SecureSphere's Deep Inspection Firewall service detects the presence of the word "union" in the URL, this single event alone is not enough to confirm an attack.

However, if the same packet triggers a Dynamic Web Firewall violation (since it includes a parameter length violation) and a Dynamic Database Firewall violation (because it is an unknown SQL query), CAV would correlate the three violations, correctly conclude that an attack is in progress, and block the attack.

By basing security decisions upon multiple activities rather than a single event, CAV is able to detect attacks and attack reconnaissance with a level accuracy — and automation — that would not be possible with independently operating security services.

Deployment Options

SecureSphere G4 Gateway appliances can be deployed in sniffer, inline/fail-open, and inline/high-availability modes. Each mode has advantages depending upon cost, security and scalability requirements. None of these modes require changes to IP addresses, domain names or routing information.

Sniffer Mode

In sniffer mode, SecureSphere G4 Gateways monitor traffic via a mirror port on a network switch or test access port (TAP) device. Traffic is blocked via TCP resets, firewall commands or application logouts. The advantages of sniffer mode are reliability and ease of deployment. It introduces no single point of failure so that network architects have more flexibility to design large-scale gateway deployments without the cost and management overhead of real-time fail over and performance monitoring. In addition, sniffer deployment and upgrade procedures require zero network downtime. For these reasons, SecureSphere can be piloted on production networks in sniffer mode without any downtime or risk to applications.

Inline Mode / Fail Open

In inline/fail-open mode, the G4 Gateway is configured as an inline bridge between external and protected network interfaces. In addition to blocking attacks via TCP resets, firewall commands or application logouts, inline gateways add the ability to drop malicious packets inline. In the event that the gateway becomes unavailable for any reason (hardware failure, software failure or even power loss), the G4 Gateway's fail-open feature ensures that applications remain available until a backup unit can be installed.

Inline / High Availability

SecureSphere's inline/high-availability mode functions the same as inline/fail-open mode except that in the event that the primary G4 Gateway becomes unavailable, all traffic is routed to a backup unit. Inline/high-availability mode ensures the availability of both the gateway and the application it protects.

MX Management Server

SecureSphere's MX Management Server is the focal point of a three-tiered management architecture that allows organizations to automatically manage dozens of G4 Gateways simultaneously.

All profiles, system configuration and policy information is stored centrally and automatically distributed to multiple gateways with a single click. Alerts, logging and monitoring data for dozens of G4 Gateways is automatically consolidated by the MX Management Server and presented to the administrator in a single unified view. Logs and events can be organized and grouped based on a variety of parameters (firewall, signature, profile, correlated events, etc.), providing administrators with detailed forensic information. Even specific user activity (identified by session ID or IP address) can be sorted and tracked over time for forensic analysis.

Reporting

The MX Management server integrates a graphical reporting tool that allows administrators to generate a wide range of reports to support audit initiatives, executive decision-making, and future deployment planning.

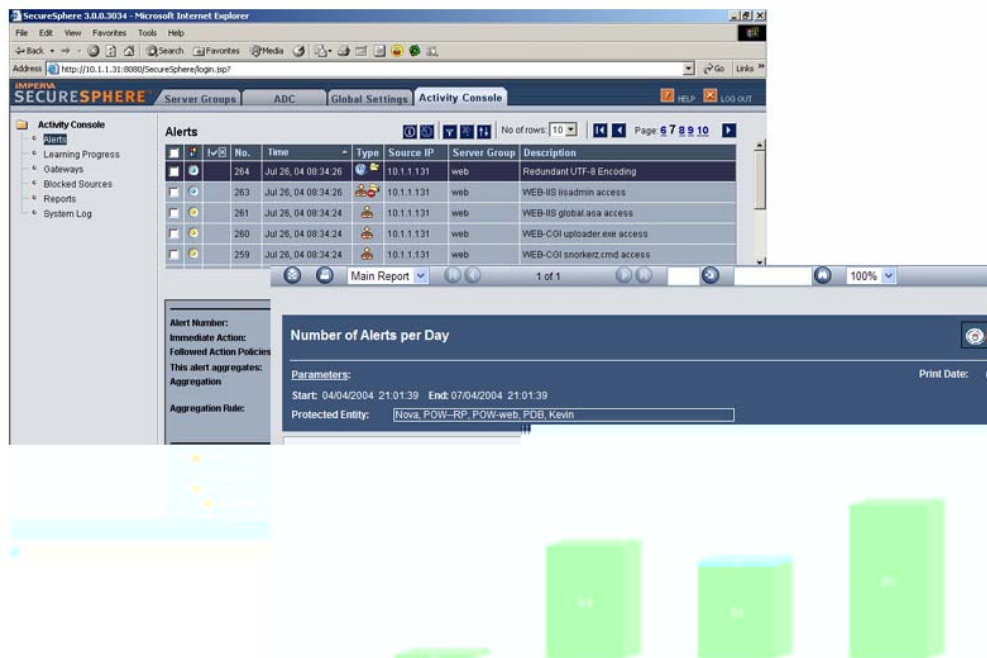


Figure 3: SecureSphere MX Management Server displays and reports.

Summary

The SecureSphere Dynamic Profiling Firewall provides a comprehensive and completely automated platform for securing application data center environments. Deep Inspection firewall, Web firewall, database firewall and unique Web worm profiling technologies combine to defeat targeted external Web hacking, internal database breaches and worms. Imperva's Dynamic Profiling technology eliminates the need for manual system configuration, while Correlated Attack Validation ensures unparalleled accuracy in the detection of attack activity. Finally, the MX Management Server provides a suite of tools that dramatically simplify the management of enterprise application data centers.

With SecureSphere, organizations that conduct business online can now deploy a practical solution that ensures Total Application Security for the entire enterprise application data center.



US Headquarters

1065 East Hillsdale Blvd.
Suite 109
Foster City, CA 94404
Tel: (650) 345-9000
Fax: (650) 345-9004

International Headquarters

12 Hachilazon Street
Ramat-Gan 52522
Israel
Tel: +972-3-6120133
Fax: +972-3-7511133