Application Delivery Security: Integrated "Any App" Security

Most enterprises and organizations are familiar with application security and Web Application Firewalls (WAFs), realizing that secure application access and delivery of web-based applications are necessities. There are many factors that are driving this push, most notably led by the "webification" of the older client/server architecture-model applications.[i] The single-purpose WAF market is progressing beyond the early-adopters' stage towards standardization and maturity, with a substantial amount of growth still available to active vendors in this market. Frost & Sullivan estimated that the WAF market will mature to approximately $100 million in 2007, with a sustained growth rate of 70%.[ii] While this is a great burgeoning market for the singular facet of ***web*** application security, it is still relatively small when compared to the ***entire*** application security market. The IDS/IPS market, for example, is another single-purpose market within application security and that market is forecasted to achieve an impressive $1 billion intotal revenue for 2007, with a decent 23% revenue growth rate.[iii] While WAF products and the overall WAF market are quickly gaining acceptance, the gap in market valuation between the WAF market and other application-focused security devices reveals one critical issue: point products create niche markets. Where does a device that provides both web application firewalling and IDS features fall? What about a product that covers other aspects of application security, such as email, FTP, instant messaging, and VoIP? Would products of this type need to define their own markets as well? Point products are critical for nascent markets because they identify and fulfill a need that previously wasn't seen. But point products can only survive for so long, until their core technologies cross product boundaries and are integrated into similar products. The network firewall and IDS/IPS market is an excellent example; although network firewalls are primarily responsible for isolating network segments, almost all border firewall devices today include some concept of signature matching to identify attack patterns. Both network firewalls and IDS/IPS devices have moved beyond their originally intended, single-feature functions — connection management and attack detection, respectively — to shared-services devices that handle each function as well as, and usually better than, their single-feature forefathers.

Though not prevalent on a mass scale yet, WAFs are quickly following suit. There are three major transitions in the WAF market happening today that will soon make stand-alone WAFs antiquated products: Application Delivery Controller (ADC) integration, inclusion of security beyond web protocols, and multi-layer security beyond just the application. These changes are a natural transition for WAFs because they represent how applications should be handled on the network: holistically.

What makes this minor-notation ("dot") release seem so major?

While many of the features represent technical or usability advances, this release signifies a fundamental shift in the way BIG-IP ASM interacts with the network, users, and applications. BIG-IP ASM expands its capabilities to that of an Application Firewall. Instead of being limited by notions of what constitutes a web application, BIG-IP ASM examines application traffic for security issues regardless of delivery.

All of the rich features that are part of Web 2.0 sites powered by AJAX (Asynchronous JavaScript and XML), Ruby on RAILS, and JSON (JavaScript Object Notation) technologies now are under the protective umbrella of Application Delivery Security. Applications delivered over the web are not just applications that use the Web 2.0 frameworks; Sales Force Automation (SFA), Enterprise Resource Planning (ERP), and Voice over IP (VoIP) are some examples. Due to their visibility, complexity, and the implementation cost of these types of applications, data security breaches can be a great financial risk for organizations than might come from a "standard" web application.

With this release, the focus is no longer on “positive” and “negative” security models. Those models imply that security breaches can be categorized by simple and static patterns which can be matched or blocked. BIG-IP ASM’s focus is on the actions and business logic of applications because there is more risk in *abusing* an application’s inherent functionality than there is in *breaking* it outright. Consider online shopping sites; breaking a shopping site is not nearly as damaging or potentially profitable as it is to buy the most popular items for pennies on the dollar surreptitiously.

By taking a wider view of the issues of application security, BIG-IP ASM changes the way it secures applications. At its core, that’s what makes this minor release so major.

Application Delivery: Controlling Security

Applications are nothing without networks, and likewise for networks without applications. It is a huge oversight to treat applications as if they are self-contained – there are many factors to consider when delivering an application to an end-user, a device, or another service. Web applications are typically very rich user environments, and the “webification” drive is primarily responsible for this rapid adoption in feature-rich web applications, often referred to as Web 2.0. This remote application renaissance forced the proprietary services and models available to client-server applications to be replicated bit-for-bit in the web browser. This rich-technology shift eventually created what we call the Application Delivery Controller (ADC). ADCs are responsible for everything involved in delivering the application outside the data center—from basic server load balancing and global application access and distribution to intelligent application switching; from VLAN to cookie management; from full proxying of all IP data to payload parsing and control of binary protocols and data. Owing its genesis to the basic load balancing market[iv], ADCs paved the way for integrated application delivery solutions, sparking the realization that there are many pieces required to provide highly available and optimized applications. Each one of these functions typically occurs in the same part of the data center, so why relegate individual appliances to a single task when one appliance is capable of managing multiple tasks concurrently?

One of the essential components of application delivery that is often overlooked is security. If an enterprise decides to secure their applications, often they will look at each application individually with tools such as source code reviews or application “black box” penetration testing and scanning. Although these options are excellent choices, they do not account for the secure delivery of the application. Both code and penetration test scanning make assumptions about accessing the application and behavior over a network. These assumptions typically consist of testing rules that are defined in a lab in a controlled network, with standards such as page access flows and types of browsers used. But what happens when the application is moved to the Internet, available to anyone with a web browser? Maybe the application works as expected when the penetration test user mimics behavior that was tested in the controlled environment. But how does the application respond when a user in Russia attempts to access page 37 directly with their home-grown java-based browser that inserts 15 extraneous headers into the HTTP request? These are the types of application delivery questions and use cases that are typically ignored when looking at the application as a stand-alone entity. A good analogy to this type of thinking is car safety testing. A manufacturer may test their cars in various collision scenarios, such as side impact or angled front impact, but they also need to test these same scenarios with different road conditions, such as a front impact test when it’s raining on a dirt road with a downward slope greater than 3%. The car will respond differently to the crash tests depending on external environments, just like an application will respond differently to security attacks depending on its set of external variables.

Stand-alone WAFs are a bit closer to a complete solution, but to some extent they still approach application security as if in a vacuum. WAFs typically only secure an application either immediately before the application server farm or as part of the application process itself, such as WAF technology built into the web server. This architecture limits the security benefits that the

WAF can truly bring to the application. If the WAF has exposure to the application session request after the request has already been load balanced, the WAF won't have insight to the intelligent decisions the ADC has already made, such as terminating SSL or session state and persistence information. If a malicious user mounts an attack against a web application's user state information, and the ADC has distributed one part of the attack to one server and another part of the attack to another, stand-alone WAFs may not recognize this attack because it's targeted at two different application servers. Another example is much lower on the network stack. If the web server and the WAF are connected to a switch in the data center and in their own VLAN segment, the WAF won't have insight into the secure configuration of the VLAN, nor that of the request's originator. The key problem with segmenting WAFs from ADCs is that a WAF has no ability to control or secure the conditions in which the web application request was generated or how it arrived at the WAF. It doesn't know that the car is driving on a wet, muddy mountain road, only that the car is traveling and it's about to deploy the side airbag.

An application security solution that combines both ADC and WAF security into one integrated device provides greater security by implementing intelligent application delivery heuristics with multiple lines of defense. The ADC is responsible for knowing the conditions in which the application is being delivered, such as the condition of the WAN, what type of device the user is using to make the application request, and most importantly for application security, how the user is making the request. ADCs are already handling all of these environment variables for application availability and optimization; securing the application and applying "contextual security" at the same time is a natural progression. ADCs are also in a unique place within the network to combine deep network intelligence with deep application intelligence—the two key factors for proper application security. Often, the ADC is already terminating SSL to offload that service from the application server, and managing cookies for multiple application servers so it can transparently load balance between nodes. By design, if an attacker mounts a session-tampering attack against an SSL-protected online banking application by changing user credentials in the cookie, the ADC will see this change anyway. With an integrated WAF, the ADC will be able to apply secure policy management to the connection, recognize that this is an attack, and respond appropriately. Integrated ADC and WAF capabilities provide the best of both worlds in a very elegant solution: ***Application Delivery Security***.

Moving Beyond HTTP Application Security

Integrated Application Delivery Security is the next logical step for true application security, but there are still a few additional components required to move application security into this entirely new arena. Application security, by definition, is focused on layer 7 of the OSI network reference model: the layer that carries protocol information for the specific application making and receiving the network requests. WAFs typically only secure HTTP, the predominant transport protocol on the web (hence the "web" adjective in the name). Within HTTP, WAFs usually secure standard HTML traffic, but can also add security to other language protocols that are carried over HTTP, such as XML, server-side JavaScript, and even rich content such as WebDAV and AJAX implementations. The common limitation of WAFs, however, is that they are solely focused on HTTP, yet HTTP is just one of many application protocols that fall into the OSI Layer 7 category. While they may secure HTML or XML carried in HTTP connections, they don't secure HTML that is carried via other application protocols, such as SMTP and IMAP (email), SIP (Session Initiation Protocol, used to manage VoIP connections), YMSG (Yahoo's instant messaging protocol), or RTSP (Real Time Streaming Protocol, used for rich media). Any of these protocols (and many more) are capable of transporting and understanding HTML content, yet they are not protected by typical WAFs. The fact that archetypal WAFs can only protect server-side HTTP and ignore other very common applications, such as FTP, is a huge concern in application security.

In stark contrast to single-point WAFs, intelligent ADCs have been able to apply application delivery management to a laundry list of protocols for some time; ADCs are frequently used to provide highly available and optimized access to SMTP mail servers or instant messaging centralized servers, for example. As we've seen above, ADCs are just now starting to come around and include security tools such as embedded WAFs. Even so, traditional WAFs don't know how to secure non-HTTP traffic, so even if an intelligent ADC and a standard WAF merge, there isn't much benefit beyond securing HTTP sessions and traffic. To take application security to the next level, the ADC has to be able to truly extend the Application Delivery Security paradigm to include many application types and protocols. The basic load balancing market didn't evolve into ***Web*** Delivery Controllers; it evolved into ***Application*** Delivery Controllers. In order to fully implement Application Delivery Security, application security must be extended beyond those protocols only associated with traditional web and HTTP. A true Application Delivery Security appliance must be able to add the same contextual security value to any application-layer protocol at the same time that contextual availability and optimization are addressed. If an ADC can support content-based routing for XML messages between application end-points, it must also be able to apply security to those messages at the same time. Likewise if an ADC is already providing highly available access to multiple mail servers, it should be able to secure SMTP traffic as well, both in transport (such as SSL termination) as well as at the destination (protocol cleansing and method enforcement, for example).

Multi-Layer Security

Similar to the shortcomings of only addressing one specific application protocol within the application layer, WAFs by design only recognize and secure the application layer, layer 7 of the OSI model. WAFs effectively ignore the other six layers, which together with the application layer are responsible for end-to-end delivery of the application. Securing the lower layers, layers 2, 3, and 4, is not new technology. However, these security tasks are usually relegated to network-centric point-products. Basic network switches that support VLAN tags are able to secure the data link layer (layer 2) allowing segmented traffic to flow within the same upstream connection from one switch to another. Network firewalls provide security for the network and transport layers (layers 3 and 4, respectively) which include protection against IP-based attacks such as ICMP broadcast floods and source routing attacks, and TCP attacks such as fragment attacks and SYN/ACK floods. Gateway firewalls are near-perfect socket management devices, controlling what traffic can access which services on specific IPs within the network. SSL VPN appliances protect the secure transport user session and data (layers 4, 5, and 7). In order to provide complete delivery security for a web application with each of these single devices, an enterprise would need at least four independent devices: a network firewall to allow HTTP traffic to enter the network destined for port 80 (layer 3 and 4 security); a switch to manage the VLAN that the application server is connected to (layer 2 and 3 security); some type of ADC to merge and manage VLAN segmented traffic from the firewall and provide access to the application server itself (layers 4 through 7); and finally a WAF to protect the HTTP session data (layer 7). None of these individual devices have visibility, much less a sophisticated understanding, of any of the networking or application components or layers they aren't specifically designed to secure. At a minimum, adding any form of application security will require these four unique devices and steps; there is no point in securing the application at layer 7 if it's prone to TCP synflood attacks at layer 4. But most of these functions are already managed by some type of ADC, including the integrated WAF, so why should we keep these technologies in separate devices? An intelligent ADC does have insight into all of the layers, and is required to make application delivery decisions based on any number of data points from any of the OSI layers, including security. The ADC is already at the correct location in the data center; adding application security intelligence simply becomes another application delivery data point.

Conclusion

There are many unique challenges and pieces to application security that most devices today aren't able to provide. Most enterprises still want to provide application security with just a Web Application Firewall, ignoring the rest of the application security puzzle or leaving those duties to existing security appliances and controls. While technically a viable option, this single-point architecture will start to unravel as security is required for non-HTTP application protocols, such as email, instant messaging, and VoIP. In order to provide a true, robust, and complete application security solution, three critical issues must be addressed:

1. ADC Integration: Security for data center applications should be addressed at both inception (secure code) and during transport (application delivery). Integrating application-layer protection into the ADC, which already includes deep application knowledge, is a requirement for total application security, health, and safety.
2. Extended Application Support: HTTP and HTML are extremely prevalent and important protocols to secure, but they're not the only protocols traversing data center networks. As the protocols in Layer 7 begin to converge and collapse, supporting single protocols or single protocol pairs, such as only HTML and only HTTP as they travel together, won't be a viable security solution. ADCs with integrated application security will need to support fluid application data exchange, between any applications, with any protocol.
3. Multi-Layer Security: The OSI networking model is a nested model: every layer, up or down the stack, depends on the preceding layer. In order to secure data in layer 7, data and connections for layers 2-6 will also be required. Layer 7 data is really just a layer 4 payload. Securing the lower layers as well is a necessary function for integrated Application Delivery Security.

Single point security solutions don't have a complete view into the application and how it is managed for delivery on the network to the end user. Each device only looks at securing that step at its own respective OSI layer. As the market has shown with firewall and IDS/IPS integration, the standalone WAF product will soon go the way of the dodo; WAFs have paved the way to a completely new market, but will soon be replaced with a more complete solution: ***Application Delivery Security***.

---

[i] For more information on the move away from client-server architecture to HTTP-based applications, see Protecting your Internal Resources with Intranet Application Firewalls, The ISC(2) Journal, Volume 15 Issue 5 2006

[ii] Frost & Sullivan, World Web Application Firewall Markets, August 2005

[iii] Frost & Sullivan, World Intrusion Detection and Intrusion Prevention Systems Markets, May 2006

[iv] For more information on the evolution of the ADC, see Load Balancing 101: The Evolution to Application Delivery Controller