Applied Application Security—Positive & Negative Efficiency

Introduction

After many years of purely negative security provided by anti-virus scanners, IDS/IPS, and anti-spam engines, it's refreshing to hear that the positive security model—the basis for tried and true security devices like network firewalls and ACLs—is coming back in vogue. Most recently, this positive policy re-emergence has revolved around the Web Application Firewall (WAF) and application security market. Yet with the positive security positioning comeback carries with it a very interesting point of detail: although many in the WAF space argue that the positive model is preferable, nearly all application security providers still rely on a partially negative solution. While acknowledging that a positive security model is the preferable model to secure web applications, many practitioners and vendors advocate a bilateral approach of both positive and negative security. As the application security market continues to evolve and define itself, there continues to be diverging views on which security methodology is the best option. In reality, enterprise security decisions are highly dependent on many factors, most of which are more business than technology oriented. Implementing an application security solution that is both secure and practical—while still allowing for the fluid nature of protecting dynamic applications—requires taking the best pieces of technology and business analysis and synthesizing them into an effective and efficient security solution.

What Does "Good" Security Cost?

In theory, the best security is impenetrable, but practical security does not function as a control group. In a business environment, security is a multivariate problem. What is the performance of the security? How easy is it to deploy? What impact will adding security have on the cost per transaction? Is it more expensive to build an impenetrable security system or risk covering the cost of a public breach? The quality of the security is always questioned as well, but it's never the only question. Many security-related questions come from the balance sheets, not the security engineers. Approaching security from a technical standpoint alone does not help the business; it hurts it.

Businesses constantly analyze their economic model to generate better operational efficiencies and a greater return on investment; the entire business intelligence market exists for this purpose. The driving force behind any IT security decision is an evaluation of a situation's potential risks versus the investment necessary to circumvent these risks. In the same vein, a business' security efforts should address a business problem; namely, to increase operational efficiencies. Security breaches can mar this efficiency, hurting a company's value, either in real dollars, operational downtime, or loss of customer trust. In truth, the motivation behind every IT decision (including security decisions) is a business decision. This has been stated and fully advocated by Gartner as well:

> *Jay Heiser, a Gartner vice president, said the fundamental problem with a purely technical approach is that IT security professionals have no understanding of business. Speaking at [the] Gartner IT Security Summit in London, Heiser said businesses must now mature and appoint individuals who understand the complexities of business, rather than the simplicities of security*[i].

When IT decisions become business decisions, blurring the distinction between secure value and business value, theoretical security and applied, or practical, security begin to separate. The theoretical approach places security on a singular plane, untouched by other business factors; applied security is comprised of the measure and level of actual security safeguards and

implementations needed to accomplish business goals. For a security product to be a functional part of a business' IT infrastructure, the product's applied security must be given more attention than the product's theoretical security. A solution that only strives to provide ultimate security will almost always be replaced with a solution designed to apply "good enough" security and increase business efficiencies than one intending to recreate Fort Knox. *Theoretical security is a check-box criteria, applied security becomes more of a buy-and-use criteria.*

Applied security exists at an equilibrium point between total security (theoretical security) and total functionality (no security). The choice between these two—and what to sacrifice—is made based on the most operationally efficient method for achieving that prescribed balance. To better evaluate the root ROI question when dealing with security products, the next logical question becomes "Which model, positive or negative, provides this equilibrium in the most operationally efficient manner?"

Positive vs. Negative Application Security

The two approaches to security most often mentioned in the context of application security—positive and negative—are diametrically opposed in all of their characteristic behaviors, but they are structured very similarly. Both positive and negative security approaches operate according to an established set of rules. Access Control Lists (ACLs) and signatures are two implementation examples of positive and negative security rules, respectively. Positive security moves away from "blocked," end of the spectrum, following an "allow only what I know" methodology. Every rule added to a positive security model increases what is classified as known behavior, and thus allowed, and decreases what is blocked, or what is unknown. Therefore, a positive security model with nothing defined should block everything and relax (i.e., allow broader access) as the acceptable content contexts are defined.

At the opposite end of the spectrum, negative security moves towards "blocked what I know is bad," meaning it denies access based on what has previously identified as content to be blocked, running opposite to the known/allowed positive model. Every rule added to the negative security policy increases the blocking behavior, thereby decreasing what is both unknown and allowed as the policy is tightened. Therefore, a negative security policy with nothing defined would grant access to everything, and be tightened as exploits are discovered. Although negative security does retain some aspect of known data, negative security knowledge comes from a list of very specific repositories of matching patterns. As data is passed through a negative security policy, it is evaluated against individual known "bad" patterns. If a known pattern is matched, the data is rejected; if the data flowing through the policy is unidentifiable, it is allowed to pass. Negative security policies do not take into account how the application works, they only notice what accesses the application and if that access violates any negative security patterns.

Discussions on preferred security methods typically spawn very polarized debates. Tried and true security engineers might ardently argue the merits of the positive security model because it originates from the most "secure" place—"Only allow what I know and expect." Many any business pundits would argue that the negative model is the best as it starts in the most "functional" place—"Block what I know is bad and let everything unknown through." Both groups are correct and yet both opinions become irrelevant when projected onto applied security, because both positive and negative security are theoretical. Applied security falls somewhere in the middle of the spectrum, providing a practical balance. At some point, as the negative approach is tightened, it will take on characteristics of a more positive model, inching towards a more complete security approach. Likewise, as a positive security model is loosened to accommodate new application behaviors, wit will take on some aspects of a more negative approach, such as implementing data pattern matching, to block the more predictable attacks. As a positive policy continues to relax, it will move closer towards complete functionality. The point at which these two opposing concepts begin to overlap is where applied security starts to take shape.

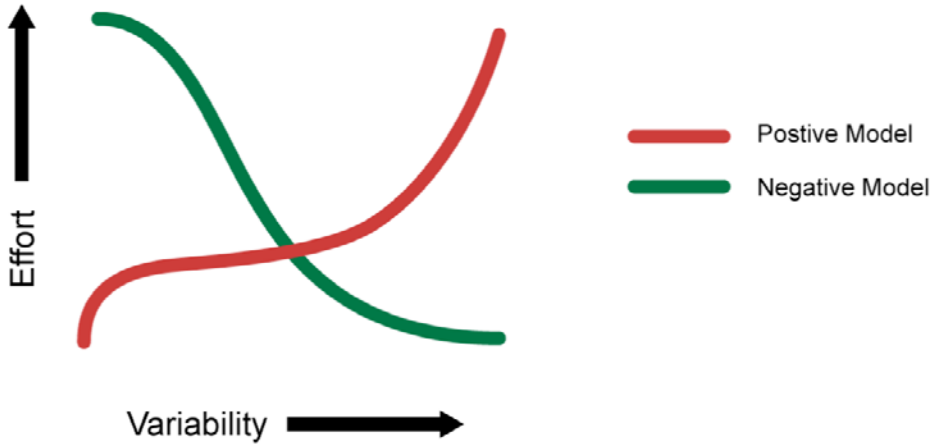
This “meet in the middle” idea suggests that from an applied security standpoint, both models are capable of achieving the same delicate balance between “security” and “functionality.” The difference between these models stems from where each begins and where they collide. This can be as simple as the number of rules required to meet the end goal, or even whether to err on the side of functionality or security if something is missed in the policy. It is clear then that, from an operational efficiency standpoint, the undiluted concepts of neither the positive nor the negative approach intrinsically provides more efficiency than the other. In some cases, the positive approach generates the least number of rules while in other cases the negative approach generates the least. It would also appear that it is the nature of the applied policy and/or the content itself which might determine the best approach. What then, are the qualities of the policy or content which makes one approach more efficient over the other?

Factors of an Effective Applied Security Model

Implementing a successful application security architecture is not as easy as deciding how much negative security and how much positive security to mix together into a hypothetical applied security blender. By design, application security devices have to have some level of application knowledge, such as the type of content delivered by the application, who is accessing any point within the application, and how to map specific policy criteria to this information. Very specific application awareness of this nature is essential in building an efficient applied security policy.

The Effect of Content Variability

Within the scope of application security, Content Variability is a measure of the content that needs to be secured and includes a number of different component pieces: the number of objects, the number of types of content, frequency of content change, and the nature of the content. A site that only has five specific objects is much less variable than a site with 500 specific objects. Within those objects, the cohesiveness of the content type is also a factor; if all 500 objects share a common format, they are less variable than a site with where all 500 objects are unique. Obviously, a site that changes only once a year is much less variable than one that changes daily. Finally, the nature of that content—for example, whether it is dynamically generated or static is a contributing factor. Essentially, variability is a measure of the site complexity. The idea of Content Variability is a single measurable value based on all of these factors. The variability of the content dictates the amount of effort needed to achieve the prescribed applied security from the chosen model.



As depicted in the diagram, the higher the variability of the content, the easier it is to define a policy using the negative security model. As the complexity of the known content increases, it is easier to describe what isn't allowed rather than what is. Conversely, the opposite effect is true of the positive model; the more variable the site content, more effort is required to define those elements that are allowed. For example, let us assume that we have 10 different types of content within our site out of a possible 100 different types of content known. Because the site exhibits little variability, or is more cohesive, it is much easier to define the 10 allowable types of content than to define the 90 types of restricted content; a positive model is much more appropriate in this case. On the other hand, if the site is less cohesive, perhaps representing of 90 of the 100 different types, it now becomes more efficient to define the 10 restricted content types than it is to define the 90 allowed ones; thus a negative model is more efficient. Once again, both models are equally successful at producing a desired level of security, but the variability of the content determines which is more efficient in a given scenario. And as we map the concept of content variability back to applied security, it becomes obvious that we will take the necessary aspects from the negative security model and couple those with what is required from the positive model. The most successful implementation will come from a joint applied security policy, addressing both the security and the business needs at same time.

Rule Specificity

As the content variability affects the ability to create and maintain a security policy, the same is true of the specificity of rules used to build that policy. Rule Specificity conveys the level of detail of the protection mechanism implemented for any particular rule. For example, a rule that blocks Unicode attacks may block them from any application on one end of the spectrum all the way to only protecting Unicode directory traversal attacks against IIS5[ii] on the other end. Depending upon the specificity of a rule, many things may be allowed with a single rule (positive security) or disallowed with a single rule (negative security). But as is the problem with theoretical security, Rule Specificity itself is not an exact science. A rule that is not specific enough may block too much, creating unnecessary false positives (blocking access that shouldn't be blocked); a rule that is too specific may not block enough, creating false negatives. Content variability also impacts the efficiency of a policy by altering the level of specificity in the rules themselves. As the variability of the content increases, the ability to specifically stipulate what content is or isn't allowed becomes more time consuming. In an ideal world, every rule would be as specific as possible for the particular application it was designed to protect, avoiding false positives and false negatives. Similarly, the level of rule specificity within an application security policy can vary greatly depending on the content variability experienced by the application.

Order of Precedence

A third factor in implementing an efficient applied security policy is the order of precedence: defining which parts of the security policy are enacted before other parts of the policy. This concept is often seen in programmatic search algorithms: "match first" or "match any." Using a combination of negative patterns and positive policy rules with varying degrees of specificity is bound to create many conflicts. In order to arbitrate these conflicts an order of precedence for all rules must be defined and followed for the policy to remain coherent. This is a critical decision point for application security, because the policy must decide if it should implement a more funneled approach (parsing through the policy to weed out what doesn't match) or if it should look for the most restrictive implementation first. This order of precedence may be solved by choosing the most specific rule, whether it is positive or negative, and opening up access as data moves further through the policy. Alternately, the order may be based on implementing a given ruleset: for example, all traffic may be pattern matched first and if there are any positive matches, the data is rejected, regardless of which specific pattern was matched. No matter which method is chosen, if the policy is implemented with an incorrect order of precedence, access to the application could be blocked by a policy that tightens first. Likewise, a policy that applies rules too loosely may allow unintended access to the application. And as precedence is factored into the applied security equation, traffic volumes must also be taken into consideration. A two percent false positive error rate may be an acceptable metric in an applied security policy of an application that handles 100 connections/day, but unacceptable for a 10 million connection/day application. Regardless of the precedence methodology used it should be well defined and easy to follow to make a policy easy to audit and manage.

Conclusion—Best Practices

The problem with a purely positive policy is simply that it's merely the most appropriate model for about half of the situations in which it's deployed. The other half are unnecessarily weighed down by the fact that a negative model would be much more efficient. That is why, as a matter of best practice, every security solution should support a weighted balance of both the positive and negative methodologies. In the strictest sense of the term, negative security provides the best applied security out of the box due to the effort applied by the security vendor before the product is shipped. Focusing on known security vulnerabilities, this will block the most attacks, despite content variability. However, this does not provide security against unknown attacks or allow specific functions to be allowed. For that, positive security is required. To lessen the amount of effort needed for a given application, positive security templates should be provided by the application vendors themselves to complement the negative security.

If the goal of applied security is to reach a pre-defined posture in the most efficient manner, then the choice of model is directly related to the variability of the content itself. Somewhere between total security and total functionality is where the desired applied security level exists, and—theoretically—either security model is capable of achieving this goal. But as stated above, theoretical security can only exist in a vacuum. Applied security is a business choice and concept that moves security into real-world implementations to attain the most efficient, functional method. Neither positive nor negative security models alone can deliver the most economic solution in every situation or environment. Applied together, however—and merged with the business needs and requirements—a holistic view of both approaches can help delineate between theoretical security and applied security, enabling businesses to realize the greatest ROI from any security policy implementation.

[i] CNET News.Com. September 16, 2005. http://news.com.com/Dont+trust+security+to+techies+alone,+Gartner+says/2100-7350_3-5868906.html

[ii] Internet Security Systems. October 26, 2000. http://xforce.iss.net/xforce/alerts/id/advise68