Introducing the Next Generation Web Application Firewall

What you need to know before choosing a WAF

Whitepaper
February 2015
# Summary

1. **Introduction** ........................................................................................................3  
   1.1 Customer Requirement ..................................................................................3  
   1.2 Introducing the Next Generation Web Application Firewall ......................4  
2. **Application Security vs Network Security** ..................................................5  
   2.1 Intrusion Prevention Systems .......................................................................5  
   2.2 Next Generation Firewall ..............................................................................5  
   2.3 WAFs and network security controls complement each other ...................6  
3. **Ease of Configuration** .......................................................................................7  
   3.1 Workflow ........................................................................................................7  
   3.2 False Positive Management ...........................................................................8  
4. **Ease of Deployment** .........................................................................................9  
5. **Web Application Security** ...............................................................................10  
   5.1 Black and white listing ................................................................................10  
   5.1.1 Black listing ............................................................................................10  
   5.1.2 White listing ............................................................................................10  
   5.2 Scoring model ...............................................................................................11  
   5.3 Advanced Detection Engines ......................................................................12  
   5.4 User Behavior & Analysis ...........................................................................12  
   5.5 Browser Security ...........................................................................................13  
6. **Web Access Management** .............................................................................14  
   6.1 Authentication ...............................................................................................14  
   6.2 Single Sign On/Off .......................................................................................14  
7. **Web Application Delivery** .............................................................................16  
   7.1 High Availability ............................................................................................16  
   7.2 Server Load Balancing ..................................................................................16  
   7.3 Caching ..........................................................................................................16  
   7.4 Compression ..................................................................................................17  
8. **Advanced Use Cases and Scenarios** .............................................................18  
   8.1 Virtual Patching .............................................................................................18  
   8.2 Forensic Analysis ..........................................................................................19  
   8.3 Application and Network debugging ...........................................................19  
9. **Conclusion** .......................................................................................................20
1. Introduction

The requirements for effectively protecting a fully Web-enabled IT have evolved. Attackers clearly focus their attention on the weaker and easier to exploit application infrastructure nowadays. Defenders are adjusting to the threat: most large organizations have invested in secure coding techniques, application security testing and have deployed Web Application Firewalls (WAF). A lot of medium-size and smaller organizations are now considering these investments.

As an application security expert, DenyAll often gets asked why WAFs are the best way to protect the application-layer, and why network security tools are not sufficient. Customers also inquire about what makes us different and better than others.

This document captures our answers. It starts by listing the requirements for effective, modern application security, explaining the differences between network security and application security, and goes on to highlight the features required in a Next Generation WAF to effectively secure modern web applications and services.

1.1 Customer Requirement

Several technologies need to be mastered and integrated together to turn application security into a true business enabler, which meets the growing demand for automation and simplicity, combined with lower cost of ownership and greater security effectiveness. Such a solution needs to:

1. Be deployable easily, wherever the applications live: in the DMZ, inside the network, close to the internal Web applications and services, or in the cloud;

2. Offer advanced detection capabilities, to block known and zero day attacks without generating false positives and to circumvent evasion techniques. This requires innovative filtering techniques, beyond the basic capabilities provided by attack pattern signatures and whitelisting;

3. Provide the ability to secure XML, SOAP and REST based applications, like a fully featured Web Services Firewall;

4. Understand the applications’ specific nature and vulnerabilities, in order to fine-tune the security policy and become an integral part of the application development lifecycle;

5. Be able to discover and profile non-protected applications, with a view to automating the provisioning of an ad hoc WAF policy for these applications;

6. Simplify user access to applications and to enforce the authentication policy that matches the needs of the business, without requiring any changes to the applications;

7. Provide an ergonomic environment ensuring WAF administrator productivity in their day-to-day tasks.
1.2 Introducing the Next Generation Web Application Firewall

With its application security technology portfolio and focused expertise, DenyAll is ideally positioned to bring to market the first Next Generation WAF, which matches the above requirements. DenyAll Web Application Firewall defines a new standard, because:

1. It can automatically discover unprotected applications, profile them, identify their vulnerabilities and provision ad hoc policies, to ease the burden of administrators;

2. It learns how applications work, identifies how attacks could be carried out and provides guidance to administrators on how to fine-tune the security policy;

3. It uses grammatical analysis and sandboxing technologies to identify the nature of incoming requests, ahead of eventually interpreting their content using signatures and heuristics (scoring) technology, in order to block complex, zero day attacks and evasion techniques;

4. It combines XML data flow routing with superior web services security;

5. It can handle various methods of authentication and simplify security for users with single sign on, to simplify and secure access to web applications;

6. It can analyze user behavior to identify and block abnormal activity, and it will soon be able to evaluate user reputation and restrict the access of likely hackers;

7. It can ensure in-session browser security to prevent compromised devices from turning into data leakage vectors;

8. Its workflow-based visual representation of policy provides a productive environment for administrators who need to manage applications, optimize data flows and adjust policy to every changing applications;

9. It will soon scale automatically as traffic grows, using a modular architecture, APIs and cloud orchestration technology.
2. Application Security vs Network Security

It's very important to understand the difference between application security and network security controls, and realize that they use complementary techniques and that both are required to effectively protect modern IT assets.

Network firewalls and intrusion prevention systems don't provide sufficient protections for Internet-facing websites, internal business-critical applications and Web Services. WAFs are most often the only control that is able to inspect encrypted and unencrypted inbound Web traffic at the application layer (Layer 7). A WAF not only helps in terms of compliance (with PCI DSS for example), it also enables companies to securely deploy web applications within minutes.

2.1 Intrusion Prevention Systems

Intrusion Prevention Systems (IPS) mostly operate at the network level and only have a limited ability to identify attacks inside the application level (OSI layer 7). Especially if we have a closer look at SSL encrypted traffic. Even if some IPS's are able to decrypt SSL traffic by uploading the private keys to those systems, it is a huge task as every SSL handshake needs to be captured in order to be able to decode the traffic. From a security perspective, IPS’s are only able to identify a very limited number of SQL injections and Cross Site Scripting attacks. WAFs do terminate and secure all web applications at OSI layer 7.

2.2 Next Generation Firewall

A Next Generation Firewall (NGFW) is a network firewall with the added functionalities of an IPS and the ability to take into account users’ access rights, by communicating with the LDAP directory. A lot of Next Generation Network Firewall (NGFW) vendors talk about their product being application-aware. That can create confusion about their ability to effectively secure applications.

NGFW helps control which applications employees have access to. A NGFW allows network administrators to determine how much bandwidth is allocated to web applications such as Facebook, eBay or Gmail, and may decide to restrict their use to time, while corporate applications such as SAP and Exchange will be accessible at all times with the appropriate network bandwidth. Along the same idea, a NGFW can identify applications like Skype, Lync or Twitter inside the network traffic and apply a similar type of policy. So, a NGFW does some very useful work.

However, because it is not usually acting as a reverse proxy, a NGFW cannot identify and block all attacks targeting applications accessible via a Web browser. And even if some NGFW can be configured in reverse proxy mode, they are not really built for security, but for network termination. This is the reason why NGFW only integrate few checks on application security compared to a dedicated Web Application Firewall, which is dedicated to securing application-layer traffic.
2.3 WAFs and network security controls complement each other

In a February 2014 research note titled “Web Application Firewalls Are Worth the Investment for Enterprises”, Gartner analysts noted the following:

“Threats against Web applications are well-documented. The Open Web Application Security Project (OWASP) Top Ten, CWE/SANS Top 25 Most Dangerous Software Errors and Web Application Security Consortium (WASC) Threat Classification v2.0 and Cross Reference View can help raise awareness of the threat landscape, providing elements to justify the need for technology dedicated to Web application security. However, security staff often fail to explain how WAFs can provide deeper, more-granular Web application safeguards than NGFWs and IPSs. Figure 2 highlights the differences between NGFWs, IPSs and WAFs when it comes to Web application security.”

Figure 1: Main differences between WAF, IPS and NGFW
3. Ease of Configuration

In most cases, WAFs are seen as complex to configure. Nowadays, this is not true anymore. As there are several approaches to configuring products, easiness needs to be defined a bit. Of course, it is important to automate administrative tasks as much as possible, but in no case will automation be able to build an accurate security configuration. A better way to achieve an easy configuration is to try to understand the applications and the attacks, and to build the security configuration around both.

Rather than relying solely on huge databases of regular expression based patterns, which tend to generate false positives, it is easier to add a weight to each component of incoming requests and verify that only in certain combinations a block should occur.

Likewise, it is important to understand the latest web technologies like JSON or REST services. In order to be able to recognize attacks it doesn't make sense to manually apply enormous regular expression in order to filter all special characters and prevent false positives. It will be much easier if such a syntax is RFC validated, normalized to http default “parameter=value” pairs and then send through standard security engines.

Beside all this technology helpers, a logical way of configuration is important to better understand how to build an accurate security policy. In other words, why do not configure the actual flow of a request towards the application and backwards to the client? This approach will not require any technical understanding but provides a logical view on the application and data flow.

3.1 Workflow

The workflow introduces a complete different way of configuring web application security. Instead of starting at a technical topic, as everyone does, the workflow configures a logical way through the application flow. The configuration is as easy as moving some bricks, decision bricks and flow arrows at the right position into a workflow.

Using this easy to understand module even web security inexperienced administrators are able to configure a web application firewall with high security standard.
For administrators with a wider knowledge about HTTP security this opens a new, enormous flexibility in setting the right security level in all areas.

### 3.2 False Positive Management

As important as the security configuration itself it is to provide an easy way to adjust the configuration from a blocked requests perspective. There is no security model, especially no negative, which will not generate one or another false positive. The more accurate a security model is identifying an attack the less false positive will be generated. Please refer to the Scoring model further down in this document.

The importance in false positive management is not only how fast it is possible to create an exception but also how accurate the exception is done.

In terms of easiness the process to generate an exception should not need more tasks then selecting a blocked request from alert logs and click a button Resolve. If it then comes to the exception creation the process should not just disable the blocking rule globally. The more accurate the exception is created the higher the security level will be.

An exception should always be created

- on a single rule
- for a dedicated URL
- on the part of the request where it was identified (e.g. POST or GET parameter)

Only by combining these 3 parts it will be possible to create exceptions without decreasing the security level significantly as attacks will still be recognized in all other URL’s and all other parts of the request.

If the security configuration implements a white list, a positive security model where only defined requests are allowed to pass the WAF the process is more or less the same but no exception is created. If a request is falsely blocked while using a positive security model this request needs to be added to the white list and therefore allows the request to pass. In this case by clicking Resolve a new rule will be added to the white list. The detail level of this rule depends on the white list level is implemented. It can be just the URL or as well all parameters and a definition of their values.
4. Ease of Deployment

Deployment is an often-discussed topic while integrating a WAF. From a network perspective people often believe the integration of a reverse proxy cause a lot of additional administrative work. In fact there are just minimal changes to be done in the network and from a security perspective a reverse proxy architecture is the most secure way of deploying a WAF. If the WAF is configured the only thing important to change is at the network firewall and forward requests to the WAF instead of the real web application.

![Reverse Proxy Architecture Diagram]

Figure 4: reverse proxy architecture

A reverse proxy implementation is not the only deployment method available. In some rare cases it is mandatory to deploy a WAF in transparent mode. But be careful with default transparent deployments as it normally prohibits several WAF functionalities. Using DenyAll’s transparent implementation it is still possible to use all reverse proxy functionalities such as caching, compression, SSL termination or server load balancing. As well, compared to others, it is possible to use a mixed mode configuration and set some applications as reverse proxy and some in transparent mode. The configuration is quite simple as well, as only 4 parameters are mandatory to set.
5. Web Application Security

A lot of innovation has taken place in web application security since the early days, due to the very evolution of web application development languages and protocols and the fact that attackers have switched their attention to the application layer. Most WAF vendors only provide black and white listing functions, while DenyAll has invested a lot in alternative and complementary techniques.

5.1 Black and white listing

There are mainly two security models used in WAF’s today. A negative security model, which integrates a ‘blacklist’ of attacks patterns as its ruleset, and a positive security model, whose ruleset is referred to as a ‘whitelist’. While a blacklist allows all requests except those matched by its patterns, a whitelist will block all requests which are not defined in its ruleset. Both methods have their limitations.

5.1.1 Black listing

With a blacklist approach it will impossible to create a list of all possible patterns to match all possible attacks. Especially if you think about encoding techniques like hex-, unicode- or HTML-encoding. All can be used in combination in a single parameter which makes it quite impossible to generate patterns in all different combinations. Due to this reason attackers will have the possibility to evade their attacks and they won’t get recognized by a blacklist. Even if one would be able to create approximately all rules, using a blacklist also leads to high false positive rate as many patterns recognizing attacks for one application which in other applications are legitimate requests. Definitely the attack recognition in a negative security model requires new, innovative approaches.

5.1.2 White listing

Using a whitelist is a very easy and secure way for small and static applications. As using a whitelist requires to define each URI with each parameter and each possible value it is a huge task to create a whitelist for bigger applications with a lot of URI’s and parameters. As well if the application is changing frequently it requires adapting the whitelist each time. If you think of e.g. new marketing campaigns to be available online, how fast can that be achieved with a whitelist?
In many cases you will hear about a so called “auto learning” process which should be able to automatically integrate new request into the whitelist. But how could a software without artificial intelligence decide if a request is legitimate or not. Experienced hackers can easily get attacks learned by such an auto learning process.

5.2 Scoring model

The scoring model relies on the capability to attribute weights to different parts of data submitted to the server. The sum of all the weights is calculated and compared to a predefined threshold. If this threshold is reached the traffic is considered as malicious and dropped.

\[
/\?id=1;select+1&id=2,3+from+users+where+id=1--' or /* comment */ 1=1 -- \\
\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
0.2 \quad + \quad 0.2 \quad + \quad 0.3 \quad + \quad 0.3 \\
\hline \\
= \quad 1.2
\]

Figure 5: Weighted, generic approach - The scoring model

This mechanism can be compared to the one implemented in most anti-spam software. Obviously the most critical aspect of the implementation is the definition of appropriate weights. Indeed erroneous values will lead to false-positive or false-negative detection and either allow malicious traffic or deny legitimate one. On the other hand, once these weights have been tested and validated there should be no need for updates, except in the case of the discovery of new attack technologies. However, sudden large scale exploitation of such technology is quite unlikely to happen without any prior knowledge or identification by any security research group. As a consequence the delay which can be allowed for an upgrade will be of no comparison with the one for a signature update which should be immediate, and is largely acceptable for critical production systems.

Last, as the security engine is completely agnostic to the protected application, it is not impacted by any kind of changes. Therefore, there is no need for manual or automatic updates.

The scoring model proved its efficiency in blocking more than 85% of new attacks with no need for updates or learning phase. Therefore it appears as a mandatory mechanism making it possible to fill gaps left opened by current models.

Moreover it is necessary to highlight that attacks which are best identified and blocked belong to the very first categories of the OWASP top 101 (Cross- Scripting and Injection flaws), thus making the model even more relevant in real-world environment.

Please refer to our white paper “Scoring Model Efficiency Report” available [here](#).
5.3 Advanced Detection Engines

Hackers and attacks getting smarter over time, linear, pattern based filters like blacklists or even scoring mechanisms won’t be able to block all modern attacks. DenyAll invested in developing new security engines, which are able to identify such advanced attacks.

Talking about modern technology like JSON it is important to not only provide a huge and complex regular expression each security engineer has to implement on his own. It would make a lot more sense to “decrypt” the JSON request to standard http parameter=value pairs and send it through default security engines.

The same applies to SQL injection techniques, which in fact can be just a single word. But blocking the occurrence of a single word will cause a lot of false positives. It will be a lot better if some keywords are prefixed with real SQL statements in order to verify it is a real SQL injection or not.

As well looking closer at command injections, common pattern based identification causes a lot of false positives, but integrating a honey pot similar technology and testing such identified commands will easily extend the security to better identify command injections and minimize false positives.

An advanced security engine are not limited to those previously mentioned and also integrate technology such as HTML security, Response Splitting, scripting language injection protection or arithmetic calculation.

5.4 User Behavior & Analysis

Sometimes attacks are not crafted very sensitive and attackers just want to crash the backend services by overloading the webserver itself. So called (D)DOS, (distributed) denial of services, attacks are not very easy to identify if they are done on http/s or XML/SOAP services. In difference to a (D)DOS attack at the network layer where no valid TCP connection is needed, each request arriving at the WAF is a legitimate request which should be forwarded to the backend.

To identify such attacks several thresholds need to be configured. One is the amount of requests a single IP is allowed to send within a predefined timeframe. But be careful if a company proxy is sending these requests as many users may use the same proxy to browse the application.

But not only backend-overloading can be identified. In many cases it is important to verify the amount of attacks per IP. If e.g. a vulnerability scanner is running to scan the application it most probably will generate an enormous amount of blocks. Why not setting a threshold on such a value like “if IP xyz generates 10 blocks in 2 seconds, block this IP for abc seconds/minutes”. Of course these thresholds can be set for various other parts of the configuration or look at a wider range or the user instead of an IP address. If user Adam forced 10 blocks in his web session, block this user for XX seconds/minutes.

Modern security also requires geolocation services in order to verify no known botnet or other bad IP reputation requests are allowed to pass the WAF.
5.5 Browser Security

Web applications are not only processed at the server side, a lot of dynamic code like JavaScript or active-X controls are processed at the users browser. As well all users data also exist at the users browsers. This also includes user credentials like usernames and passwords or bank account information or security tokens in terms of online banking. This is the reason why there is such an enormous amount of trojans and malware attacking the client browser. Some of this software is very advanced and only targeting e.g. online banking applications.

To complete application security it is mandatory to extend it from server security to browser security. In order to do so DenyAll integrates it’s so called Client Shield technology with which it is possible to secure the clients browser. From a process perspective the user connects to an application and is accessing a secured area. Before logging on to the application the client shield will be downloaded.

![Image: Client Shield integration]

**Figure 6: Client Shield integration**

This can be a dll file, which controls the browser process. From that time certain security functionality is running. E.g. no plugin is loaded inside the browser, no windows hook function to the process is possible and even if an administrator tries to run a debugger on this process, the process will kill itself to not get compromised. Of course this can be extended with several additional features.

This not only runs on Windows software, where it needs no administrative rights and is also available as an exe or MSI file which also brings its own mini browser. As well there are possibilities to run it on Mac OS and Linux environments.

Additionally it is also available as a SDK for iOS, Android and Windows mobile in order to secure mobile applications as well.
6. Web Access Management

A Next Generation WAF also includes the ability to handle user authentication and single sign on for simplifying access to the protected web applications.

6.1 Authentication

DenyAll offers a wide range of authentication methods from a simple basic authentication up to NTML or SAML authentication methods.

Pre-authenticating users is a common and important possibility in order to set a higher security level. In most cases it is not allowed to access the backend application if the user is not authenticated. But in fact if the backend application is doing the authentication on its own, the user is accessing the backend server before being authenticated. This is a common problem accepted by most application owners and security officers.

Using authentication services at the WAF level prevents accessing the application before being authenticated and is solving this issue in a very easy way. Of course all credentials can be automatically forwarded towards the backend application so the user get logged in automatically.

Beside authentication DenyAll also offers the possibility of authorization with integrated or external data storages.

6.2 Single Sign On/Off

The Web Access Management module takes charge of applying the authentication policy in the Management Console to applications protected by the WAF. It allows application authentication methods to be grouped behind a single strong authentication presented to clients.

On their first request a user is presented with an external (perimeter) authentication form. Once authentication is successful, the Web SSO module will automatically handle application authentication based on the credentials stored in its internal directory or in the company directory.
The Web SSO module provides a simple integration with all web authentication mechanisms and simplifies access-control policy. This module is agentless and does not need modifications on applications.

The Web SSO module becomes an integral part of Web Security Policy in IS. It complements intrusion detection by providing a higher level of strengthening of authentication.
7. Web Application Delivery

DenyAll Web Application Firewall includes web application delivery features, which are required to safely insert application security into the application structure and data flow, without impacting stability and user experience.

7.1 High Availability

No security device is placed in a secure infrastructure without a high availability configuration. In order to provide up to 99.99% uptime of applications it is mandatory to have at least one running node and one spare node which takes over all work in case of a failure of the first node. This is a classical active-passive configuration with two nodes in a cluster.

Additionally a native active-active cluster with up to 32 nodes can provide a huge performance increase or will only set a third failover node in case of a failure of the second node.

In any way HA in active-passive mode should be considered in all installations.

7.2 Server Load Balancing

In difference to high availability, which is only working for the WAF itself, server load balancing is able to spread the load to different backend servers. Normally, load balancers are costly and configuration intensive devices. Integrated into a WAF, the cost and the configuration is minimal. The load balancing features integrated into DenyAll Web Application Firewall only implements what is needed in terms of application load balancing for web applications, it will not be able to do traffic load balancing.

Used in smaller environments it saves a lot of costs in buying a dedicated device, which most probably are heavily oversized in terms of features. Additionally no extra training costs need to be planned and configurations are done a lot faster.

But even in big network environments dedicated load balancers are not mandatory if they are only used for applications load balancing. This functionality can easily be consolidated with the integrated SLB feature inside the WAF.

Of course the WAF can be integrated in all environments with or without a dedicated load balancer. There is no limitation or prerequisites in using a WAF in combination with a dedicated load balancer.

7.3 Caching

Consequently using a reverse proxy deployment makes it very easy to automatically cache all static content. Static content is everything, which does not require any GET or POST parameters to be send along with the request. Using this feature is just as easy as ticking a checkbox. It will increase the application performance as the backend application server will not serve all this static content like images, style sheets and java scripts anymore as this will be served by the application firewall for now on. The backend webserver can now concentrate on serving the real important, dynamically generated content resulting in an performance increase.
7.4 Compression

The same as for caching applies for compression. If this function is done at the application firewall level the backend application server doesn’t need to take care about this anymore. And in fact compressing a 10 KB file inside the internal network with most probably 1GBit or even 10GBit interfaces will not speed up the application loading at all! Compression only makes sense at the client connection where bandwidth normally is a lot more limited. Therefore this feature should be placed at the application firewall level as much towards the outside connection as possible.
8. Advanced Use Cases and Scenarios

8.1 Virtual Patching

Any web application can have vulnerabilities and requires frequent vulnerability testing. DenyAll offers an own vulnerability scanner for network and application based vulnerabilities. But only providing the identification is not enough in terms of security. Of course all identified issues needs to be patched. If the vulnerable applications are an in-house development it most probably takes a few days up to a few weeks until the applications is patched.

But what if the initial developer is no employed anymore? How is able to patch this part of the application? It may take a longer time to patch this vulnerability. It even gets worse if we have a closer look on issues in Microsoft, Oracle or SAP applications, which cannot be patched by some in-house application developers. Companies need to wait until these companies themselves fixed the issue. Microsoft provides security updates each month, for SAP it sometimes takes a few months and for Oracle it sometimes need more than a year to fix a certain issue.

In fact it doesn't matter if it takes 2 days, 5 month or a year to fix vulnerability. As soon vulnerability is discovered it should be fixed immediately otherwise the application should not be accessible anymore. Imagine the risk if a CSO accepts to stay with a vulnerable application and at this time data get stolen!

Coming back to the vulnerability scanning process, which generates a report that can be uploaded towards the application firewall. At the application firewall the report is processed and vulnerabilities are fixed with a “virtual patch” at the application firewall level!

![Virtual Patching Process Diagram]

Figure 8: The virtual patching process

This easy and ongoing process enables security professionals to act very, very fast in, at least virtually, fixing vulnerabilities. This process is just a matter of a few hours from identification to fixing.
8.2 Forensic Analysis

Up to the time an application firewall was integrated into your network you had some logs at the application servers, some at the network firewall and maybe something at some proxies inside the network. Most probably attacks have never been identified or noticed so far.

By integrating a web application firewall perfectly enables you to first identify and block attacks but secondly also use the application firewall as tool for forensic analysis. As the WAF is able to store all information about the attack, the attacker, the authentication context and all GET and POST data send along with those requests it will be very easy for an auditor to run a forensic analysis on the attack.

This is very important in order to avoid future attacks and adapt applied security policies in terms of effectiveness.

8.3 Application and Network debugging

In many cases the integration of a web application firewall shows actual problems with applications or network connections and it helps in debugging those problems.

For example if an application is using redirects and those redirects are done on a hardcoded URL that represents the internal server name. Or if the backend is also using unsafe ciphers in SSL connections, the application firewall can report on this as well.

In fact integrating a web application firewall is a great tool for debugging applications and networking besides adding security.
9. Conclusion

DenyAll offers an end-to-end and an integrated approach: “Next Generation Application Security”. Based on new security paradigms and the power of the cloud, the goal is to make application security technology measurably efficient and affordable to all.

Deployed in your DMZ, behind your network firewall, DenyAll Web Application Firewall blocks application-layer attacks targeting your IT infrastructure. The result of 15 years of innovation, they combine advanced functions to effectively protect you, even against zero-day and the most advanced application-layer attacks.

DenyAll’s security approach not only includes a WAF, as you have seen in this paper: it provides a lot more features, like vulnerability scanning and virtual patching, web access management, web application delivery with load balancing, caching and compression, as well as debugging network and applications or forensic analysis.

DenyAll products can be integrated anywhere and anytime in your architecture and application development and security process.

Figure 9: Anywhere, Anytime