SECURING INFORMATION RESOURCES USING WEB APPLICATION FIREWALLS

Petr A. BARANOV
Associate Professor, Department of Innovations and Business in IT,
National Research University Higher School of Economics
Address: 20, Myasnitskaya Street, Moscow, 101000, Russian Federation
E-mail: pbaranov@hse.ru

Eldar R. BEYBUTOV
MSc Program Student, Faculty of Business and Management,
National Research University Higher School of Economics
Address: 20, Myasnitskaya Street, Moscow, 101000, Russian Federation
E-mail: eldar.beybutov@gmail.com

This paper provides an overview of core technologies implemented by comparably new products on the information security market – web application firewalls. Web applications are a very widely-used and convenient way of presenting remote users with access to corporate information resources. They can, however, become single point of failure rendering all the information infrastructure inaccessible to legitimate clients. To prevent malicious access attempts to endpoint information resources and, intermediately, to web servers, a new class of information security solutions has been created.

Web application firewalls function at the highest, seventh layer of the ISO/OSI model and serve as a controlling tunnel for all the traffic heading to and from a company's web application server(s). To ensure decent levels of traffic monitoring and intrusion prevention, web application firewalls are equipped with various mechanisms of data exchange session «normality» control. These mechanisms include protocol check routines, machine learning techniques, traffic signature analysis and more dedicated means, such as denial of service, XSS injection and CRRF attack prevention. The ability to research and add user rules to be processed along with vendor-provided ones is important, since every company has its own security policy and, therefore, the web application firewall should provide security engineers with ways to tweak its rules to reflect the security policy more precisely.

This research is based on broad practical experience of integrating web application firewalls into the security landscape of various organizations, their administration and customization. We illustrate our research into available filtering mechanisms and their implementations with exemplary product features by market leaders.

Key words: information security, web application firewall, application server protection.

Citation: Baranov P.A., Beybutov E.R. (2015) Securing information resources using web application firewalls. *Business Informatics*, no. 4 (34), pp. 71–78. DOI: 10.17323/1998-0663.2015.4.71.78.

Introduction

Nowadays, many companies and businesses have an information security policy which assumes remote access to their information resources (calculation powers, cloud services, data storage). By saying «remote access», we mean access over the Internet. This could be made through letting all Internet users gain certain types of access to resources or through letting only identified corporate users remotely use the company's resources. Direct access to information resources is both inconvenient for users and comparably insecure because of the lack of a single access point which hinders security policy implementation. The solution is well-known and it is called Application Servers, or application-layer intermediate nodes. External users gain access to these nodes using the regular web browser, interact with unified interface and put queries to it. These queries are afterwards translated by the Application Server into more specific queries to internal information resources and, after getting a response from these resources, the Application Server transforms them into an easy-to-understand view and shows it to the external user in his/her web browser. The scheme is transparent and, once all components are installed and set up, has predictable and controllable technical support expenses. Information security breaches often lead to increased technical support expenses. In these terms, to gain control over technical support funds the company management must be sure that the security level of the single access point – the web application – is high enough to prevent malicious attempts at access and use of company data from getting through the Application Server to the data hosting infrastructure. There are national [1, 2], industry branch [3, 4] and corporate [5, 6] standards of writing secure web applications. Application development in accordance with these standards is a labor-intensive, expensive and hard-to-scale procedure. It does not guarantee safety of the result if software developed by a third party is applied. Information security officers need a versatile and configurable tool to control traffic flowing through the web application server and it must be able to prevent data endpoints and the application server itself from receiving and processing maliciously crafted traffic and queries.

Web application firewalls present a solution for the problem described. There are numerous vendors offering a variety of products who claim their products have all the mechanisms needed to provide security on top of standard web application rules. Speaking of Web application firewalls (WAF), people often become confused because of the different associations they have in mind on what features such a specific tool should contain. Even key functions of WAF are sometimes misunderstood. In this research, we would like to introduce a WAF's typical functionality and the defense mechanisms that are essential for WAF in the modern state of the industry. The research is based on practical experience of integrating WAFs of different vendors into existing company information infrastructures.

First of all, we will define a list of the defense mechanisms for a WAF which must be present. We will describe every mechanism, its features and how it works. To illustrate the way some mechanisms function, we will make comparisons of their implementation for different WAF vendors. As examples, we will mainly focus on solutions from leaders in the area their solutions prove to be quite representative (for more about WAF market leaders see [7]). The question of «which WAF is the best» has no answer, and we have chosen these vendors just to highlight variations of implementation of a single product ideology in different practices.

Let's list the most necessary defense mechanisms for any WAF:

✦ protocol check;

✦ signature analysis;

✦ machine learning of access identifier formats;

✦ injection and XSS protection (mostly proprietary);

✦ user-defined rules of illegitimate queries detection;

✦ «denial of service» attack prevention;

✦ integration of the information security landscape.

We will now go through all of these mechanisms and explain their capabilities.

1. Protocol check

Protocol check is a passive protection mechanism against potential threats exploiting non-typical use of HTTP protocol features. Firstly, it involves HTTP header check for compliance with RFC. But RFC doesn't have all the rules and restrictions to ensure security of the traffic and vendors invent their own restrictions that, of course, do not interfere with the work of legitimate users, once implemented.

Protocol check is one of the primary mechanisms. Its main role is to leave the intruder as few opportunities as possible to exploit possible internal vulnerabilities. The HTTP transactions are limited through the following checks:

✧ RFC requirements;

✧ header and parameter's length and number;

✧ time limits;

✧ JSON and XML entity checks;

✧ illegal values detection.

2. Signature analysis

Signature analysis is one of the eldest technologies of ensuring the security of applications. It is still widely used and its effectiveness is proven through the number of devices and solutions based on it (e.g. antivirus software, classic firewalls, spam-filters).

Modern trends in the evolution of information security threats show that nowadays the majority of intruders do not develop their own maliciously-oriented software. During a typical attack, a malefactor applies ready-to-use hacking means previously created by a third party (for examples see [8]. Moreover, the usage intensity for these means is so high that public web applications suffer automated attacks nearly all the time. Theoretically, mechanisms involving a machine learning process that creates a normal behavior model can render signature analysis useless. Based on this fact, some WAF vendors do not rely fully on the signature analysis mechanism in their products and do not invest much in signature update procedures (for example instead of analyzing User-Agent and other signature-based methods, Wallarm WAF implements behavioral fingerprinting schemes to determine the tools used in the attack [9]).

But practice shows that in some cases this protection mechanism is irreplaceable. For example, during the machine learning period signatures prove to be very useful due to the fact that they ensure a «clean» data environment for anomaly-detection software. They also ensure the overall security level for the leaning period. Because of this, a decent WAF solution must have a wide and relevant signature database applicable for all types of web applications.

3. Machine learning

Machine learning of access identifier formats is one of the key features for products of WAF-class. The main concept here is creation of a normal behavior model based on URL, parameters and cookies. Once a model is created and tested, the comparison of live traffic against it could prevent both known and unknown vulnerabilities from being exploited. Let's try to estimate the efficacy of machine learning.

It must be said, that the technology's efficiency is hard to estimate by means of a mathematical statistical algorithm because the algorithm is usually proprietary and is not disclosed by solution producers. Live traffic imitation poses certain difficulties as well.

Nonetheless, we are able to compare the following features of the algorithms:

- flexibility of learning parameters;
- resulting data optimization.

In a solution from F5, for example, for every web application being protected a profile is created. This profile specifies the beginning and ending of learning thresholds. There is a possibility to list an interval of trusted IP addresses. It is very important, since queries from such IP addresses would represent an invaluable contribution to a concept of the normal traffic model. As for machine learning improvement, the replies from web applications are also analyzed: the parameters used in reply forms are more trusted for WAF than those created by a remote client.

This functionality is more «boxed» when speaking of Imperva's solution. The only parameterized value is the query learning time limit. By default, it is set to 240 hours for all web applications. This means that after 240 hours of learning process of an object (application), the object is treated as «learned.» All the queries from now on will be compared in relation to the normal model.

The resulting data optimization is a vital process. The need to perform it is raised when the normal model is not formed correctly or the web application being protected has been modified by its developers. Most WAF products support this feature to be performed by hand (it is called «manual object adjustment»). Some of them provide users with automatic object adjustment.

Usually automatic measures here are implemented through a mechanism of tracking changes. It monitors the number of false positive events generated by the model. For example, for the WAF by F5, if 5 different users in a 5 minute time interval performed the same violation, the object is switched to re-learning state. Another example used in the Imperva solution tracks the number of typical model violations per 12 hours. If the number is higher than 50 for every hour throughout a 12-hour interval, the object is rendered «unlearned».

4. Injection
nd XSS protection

Positioned between the web application server and the outer network environment, WAF as a security tool has an opportunity to «comprehend» the traffic going through it, analyze it and check it for compliance with security rules. Injection attacks take place in cases where the web application sends unchecked (or not sufficiently checked) data taken from client's query to a neighboring system's command interpreter. Neighboring systems here could be databases, the operating system, LDAP-server, XPath interpreter and many others. This query transmission allows malefactors to manipulate adjacent functional systems.

Injection prevention is achieved through application of the following mechanisms:

✧ tokenization. Using a finite automaton, the query is parsed and the target system's tokens are detected. When certain (previously defined in WAF parameters) tokens are found, the query is treated as potentially dangerous;

✧ web application response control. Here the search is performed in service information of the target system's response. The information that could appear only in case of incorrectly processed output is searched for. When this data is detected in the web application's response, the whole response is considered dangerous and is not carried through the WAF;

✧ signature analysis. A signature group is created in the WAF's internal storages. Each signature describes a case of a target system's manipulation attempt. If a sample in traffic contains the signature's data, the corresponding query is considered illegitimate.

The other threat is cross-site scripting (XSS) attempts. This becomes possible if a web application's response uses client-provided data without doing proper checks on this data. XSS allows the malefactor to steal client session identifiers, make web page defaces and re-route clients to arbitrary information resources. To detect XSS, the following techniques are applied:

● tokenization. Using a finite automaton, the query is parsed in order to search for declarative programming language tokens. If tokens valid for a programming language syntax are found, the current client query is declared potentially dangerous;

● content security policy (CSP) integration. A comparably new approach in information security: the CSP header defines for each web application response the possible resource sources which can be used to construct the page being displayed by the client's browser. The main difficulty: complexity of manual description of CSP rules. Some WAFs have techniques for CSP rules to be auto-created;

● response analysis. The response's content is matched to the data received from the client. If the data matches for the query-reply pair, the data transaction is declared illegitimate;

● web application response is injected with special Javascript code intended for page display control in the client's browser. This technique is most effective at detecting DOM-based XSS attempts;

● signature analysis. A signature group is created in the WAF's internal storage. Each signature describes a case of an XSS attempt. If a sample in the traffic contains the signature's data, the corresponding query is considered illegitimate.

5. User-defined rules of detecting illegitimate queries

WAF is an information security tool that is used «on top» of the protected server. It has a wide potential of features to analyze queries that go through it. Its capabilities, therefore, are:

✦ decryption;

✦ normalization;

✦ parsing;

✦ session control;

✦ traffic inspection;

✦ security policy checks;

✦ data leakage check.

These capabilities could be applied not only within the hard-coded mechanisms of WAF, but are presented to the information security administrator to form new, user-defined security rules. This could be useful when new security instructions are introduced in the information systems. Another case is adding new rules to prevent vulnerabilities detected during the information security auditing session. Moreover, when there is a need to define user logic from scratch and reflect it in website partitions access restriction, there is little to be done without user-defined security rules on WAF.

Generally, this mechanism's scenarios of usage are limited only by the toolset provided by WAF. Let's go through some examples of implementation.

The WAF product by F5 offers integrated programming language to define user rules. The scope of potential functions for this tool is huge, but it comes at a price: high competency requirements for the WAF administrator. Obvious difficulties could occur as well while implementing protection mechanisms requiring prompt intervention.

Toolset, offered by Imperva in their WAF, is sufficiently different from F5's. User rules here are generated by combining and correlation of a criteria set. The criteria amount totals about forty.

Doubtless, this kind of implementation provides far lower flexibility, particularly when speaking of bypassing traffic influence capabilities. However, these limitations are recouped by the relatively low threshold of competence for security administrators to manage the WAF.

6. «Denial of service» attack prevention

Ensuring availability of a protected resource is a task with the same importance level as data confidentiality and integrity maintenance. Sometimes this task is even more important for systems sensitive to constant feedback to user (emergency control, payment processing, etc.). There is a prejudicial opinion that denial of service attacks should be handled at layers lower than the application level according to the ISO/OSI model. Nevertheless, WAF, acting at the application layer, offers interesting methods of prevention for this type of malicious activity.

WAF possesses a bot detection mechanism that is capable of telling whether there is a human operating a client machine or there is an automaton generating queries directed at a protected resource. Blocking automated machines hinders botnets' denial of service attack participation. This is reached through injection of a special javascript in the web application's replies to the client. The client must answer an easy question for WAF to draw a conclusion if there is a human-operated remote machine, or not.

Let's walk through other denial of service attack prevention mechanisms available in WAFs. First of all, WAF detects nodes involved in the attack. After this, reaction measures are applied to these nodes. A denial of service attack is defined based on queries per second or time required by the web-server to reply.

A query volume control mechanism works as follows. The last minute's statistics on number of queries are compared to the last 5 minutes' statistics. If the first value is higher than 5 times the second value (or if the first value has reached a previously defined level), active reaction measure performing mechanisms are enabled. Here, both statistics for every URL and IP-address are taken into account.

Turning to the web application time-to-reply controlling mechanism, it works the same way as the query volume control mechanism. Here the average delay for a query is analyzed and its ratio against 5-minute historic time interval is monitored.

As a result of any of these mechanisms' work, the WAF can enable one of the following attack response measures, thereby decreasing the attack's success chances:

✧ replies of web application are injected with a «proof of reply processing» Javascript-task, thus, slowing legal and illegal query sources. Nevertheless, we leave an availability window for an ensured number of users;

✧ the user and web-application interaction process is interrupted with a popup window containing «captcha»-trial. After the trial is completed, the session is considered to be with a «human»;

✧ bandwidth limitations for clients sending queries to the information resource being attacked are applied.

7. Integration of the information security landscape

An information security solution's efficiency is multiplied if different security software and hardware tools are interconnected. It is therefore important that products like WAF have wide integration capabilities to «understand» other security products and solutions and use data, generated by them to enhance its own security functions. Nowadays WAFs can be combined with the following system and services:

✦ vulnerability scanners;

✦ security information and event management systems;

✦ reputation services;

✦ fraud prevention services.

It seems that the most fruitful connection here is a vulnerability scanner data exchange. A function called «virtual patching» is implemented through this data exchange. It automates application security control: a scanner uncovers vulnerabilities and makes a report. Based on the information in the report, the WAF forms rules to block activity aimed at exploiting the newly-found vulnerability.

Information security incident control systems are a key point of information security efficacy for many large businesses. Using WAF, it becomes possible to take into account and correlate the events generated by web applications to events from other information and security systems.

Reputation services are specialized in revealing suspicious IP-addresses among those from the global IP-address pool. The database contains addresses of TOR endpoints, anonymous proxy servers, phishing and spam-generating nodes. The base also holds messages from the community participants on addresses violating their own security policies.

Fraud prevention systems are used by WAF to ensure that the client-server data exchange is not interfered with by third parties and that the client is not subject to a malware attack. WAF could use fraud prevention services to check that remote clients are «clean» and therefore raise the overall security level and lower the risks of scam and fraud operations.

Conclusion

Web application firewalls are a new product on the market in relation to traditional firewalls and anti-virus software. We have indicated the most-important features WAFs provide to information security engineers. With many vendors offering different WAF solutions on a broad market, understanding the WAF core functions and mechanisms is crucial to anyone who wants to build an even and balanced information security landscape in a company. Apart from these core functions and algorithm implementations, every solution possesses additional capabilities and offers new methods of ensuring web application information security. We intend to cover these methods and make a comparison of the most advanced products on the WAF market in our next article. ■

References

1. Russian Federal Agency on Technical Regulating and Metrology (2013) *Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Kriterii otsenki bezopasnosti informatsionnykh tekhnologii. Chast' 3. Komponenty doveriya k bezopasnosti* [Information technology. Security techniques. Evaluation criteria for IT security. Part 3: Security assurance components]. Moscow: Standardinform (in Russian).
2. Russian Federal Agency on Technical Regulating and Metrology (2008) *Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Sistemy menedzhmenta informatsionnoi bezopasnosti. Trebovaniya* [Information technology. Security techniques. Information security management systems. Requirements]. Moscow: Standardinform (in Russian).
3. Bank of Russia (2014) *Rekomendatsii v oblasti standartizatsii Banka Rossii. Obespechenie informatsionnoi bezopasnosti organizatsii bankovskoi sistemy Rossiiskoi federatsii. Obespechenie informatsionnoi bezopasnosti na stadiyakh zhiznennogo tsikla avtomatizirovannykh bankovskikh system* [Bank of Russia standards recommendations. Russian Federation's Bank system organizations' information security. Information security maintenance in automated bank systems lifecycle] (electronic resource). Available at: http://www.cbr.ru/credit/Gubzi_docs/rs-26-14.pdf (accessed 29 September 2015) (in Russian).
4. Payment Card Industry (PCI) (2015) *Payment application data security standard. Requirements and security assessment procedures. Version 3.1* (electronic resource). Available at: https://www.pcisecuritystandards.org/documents/PA-DSS_v3-1.pdf (accessed 29 September 2015).
5. *Cisco Secure Development Lifecycle (SDL)* (electronic resource). Available at: http://www.cisco.com/web/about/security/cspo/csdl/index.html (accessed 29 September 2015).
6. *Microsoft Security Development Lifecycle* (electronic resource). Available at: http://www.microsoft.com/security/sdl/default.aspx (accessed 29 September 2015).
7. Beybutov E.R. (2015) *Obzor rynka zashchity web-prilozhenii (WAF) v Rossii i v mire* [Web Application Firewall (WAF) market review in Russia and over the world] (electronic resource). Available at: http://www.anti-malware.ru/reviews/web_application_firewall _market_overview_russia (accessed 29 September 2015) (in Russian).
8. *Exploit database*. Exploits for web applications (electronic resource). Available at: https://www.exploit-db.com/webapps/ (accessed 29 September 2015).
9. *Wallarm blog* (electronic resource). Available at: http://blog.wallarm.com (accessed 29 September 2015).

ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ РЕСУРСОВ С ПОМОЩЬЮ МЕЖСЕТЕВЫХ ЭКРАНОВ ДЛЯ ВЕБ-ПРИЛОЖЕНИЙ

П.А. БАРАНОВ
кандидат технических наук, доцент кафедры инноваций и бизнеса в сфере информационных технологий, Национальный исследовательский университет «Высшая школа экономики»
Адрес: 101000, г. Москва, ул. Мясницкая, д. 20
E-mail: pbaranov@hse.ru

Э.Р. БЕЙБУТОВ
студент магистратуры, факультет бизнеса и менеджмента, Национальный исследовательский университет «Высшая школа экономики»
Адрес: 101000, г. Москва, ул. Мясницкая, д. 20
E-mail: eldar.beybutov@gmail.com

Данная работа содержит обзор основных технологий, реализуемых в относительно новых для рынка решений информационной безопасности продуктах — межсетевых экранах веб-приложений (веб-экранах). Веб-приложения представляют собой удобный и широко используемый способ предоставления доступа удаленных пользователей к корпоративным информационным ресурсам. Однако, сервер (серверы) веб-приложений может стать единой точкой отказа, таким образом, прерывая доступ легитимных пользователей к ресурсам информационной инфраструктуры организации. С целью создания инструмента противодействия злонамеренным попыткам доступа к защищаемым информационным ресурсам предприятия был разработан новый класс решений по защите информации — межсетевые экраны уровня приложений (веб-приложений).

Веб-экраны работают на седьмом уровне модели ISO/OSI и представляют собой туннель, контролирующий и, при необходимости, модифицирующий трафик, направленный к серверу веб-приложений и от него. Для обеспечения эффективного мониторинга и обработки трафика с целью выявления злонамеренных действий веб-экраны снабжаются различными механизмами контроля «нормальности» передаваемых в рамках сессии связи данных. Такие механизмы включают проверки соответствия правилам протокола формирования сообщений, техники машинного обучения и статистические подходы, анализ сигнатур в проходящем трафике, а также более узконаправленные средства, такие как предотвращение реализации атак типа «отказ в обслуживании», инъекции XSS и CRRF-атаки. В условиях неоднородности политик информационной безопасности на предприятиях и невозможности охватить все допустимые ограничения, налагаемые правилами обмена данными, несомненно актуальной является возможность добавления пользовательских правил к тем, которые уже были реализованы производителем. Веб-экраны обладают инструментариями, позволяющими создавать правила разной степени подробности.

Данное исследование основано на широком практическом опыте интеграции веб-экранов в существующие ландшафты информационной безопасности, их администрирования и настройки. С целью иллюстрации возможностей применения фильтрующих механизмов приводятся примеры их реализации в продуктах от ведущих игроков рынка решений информационной безопасности.

Ключевые слова: информационная безопасность, веб-экран, межсетевой экран веб-приложений, защита сервера приложений

Цитирование: Baranov P.A., Beybutov E.R. Securing information resources using web application firewalls // Business Informatics. 2015. No. 4 (34). P. 71–78. DOI: 10.17323/1998-0663.2015.4.71.78.

Литература

1. ГОСТ Р ИСО/МЭК 15408-3-2013. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Компоненты доверия к безопасности. М.: Стандартинформ, 2013. 267 с.
2. ГОСТ Р ИСО/МЭК 27001-2006. Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования. М.: Стандартинформ, 2008. 23 с.
3. РС БР ИББС-2.6-2014. Рекомендации в области стандартизации Банка России. Обеспечение информационной безопасности организаций банковской системы Российской федерации. Обеспечение информационной безопасности на стадиях жизненного цикла автоматизированных банковских систем // Банк России. [Электронный ресурс]: http://www.cbr.ru/credit/Gubzi_docs/rs-26-14.pdf (дата обращения 29.09.2015).
4. Payment application data security standard. Requirements and security assessment procedures. Version 3.1. // Payment Card Industry (PCI). [Электронный ресурс]: https://www.pcisecuritystandards.org/documents/PA-DSS_v3-1.pdf (дата обращения 29.09.2015).
5. Cisco Secure Development Lifecycle (SDL). [Электронный ресурс]: http://www.cisco.com/web/about/security/cspo/csdl/index.html (дата обращения 29.09.2015).
6. Microsoft Security Development Lifecycle. [Электронный ресурс]: http://www.microsoft.com/security/sdl/default.aspx (дата обращения 29.09.2015).
7. Бейбутов Э.Р. Обзор рынка защиты веб-приложений (WAF) в России и в мире [Электронный ресурс]: http://www.anti-malware.ru/reviews/web_application_firewall_market_overview_russia (дата обращения 29.09.2015).
8. Exploit database. Exploits for web applications. [Электронный ресурс]: https://www.exploit-db.com/webapps/ (дата обращения 29.09.2015).
9. Wallarm blog. [Электронный ресурс]: http://blog.wallarm.com (дата обращения 29.09.2015).