An Introduction to the Waratek Application Security Platform

The Transformational Application Security Technology that Improves Protection and Operations

Highly accurate. Easy to install. Simple to operate.

Waratek.com

Copyright 2017 Waratek – All Rights Reserved
The Waratek Application Security Platform: Protecting Applications through Virtualization

Protecting applications through patented virtualization technologies is the core of the Waratek Application Security Platform. Waratek not only redresses the deficiencies of traditional security implementations, such as Web Application Firewalls (WAF), but also introduces a new set of capabilities that together represent a shift in the way enterprises secure their data and embed security within their applications.

Advocates of using the runtime to protect applications draw attention to the insights, guiding principles and solid practical and economic foundations that gave rise to the concept when differentiating it from preceding approaches to cybersecurity. Since its emergence in 2012, runtime protection has seen steady growth with vendors penetrating many market segments, notably the Banking and Financial sectors, across Europe, the US and Latin America.

The original insight observed by the early runtime protection technologists was that present in the runtime of an application is all the data and context required to prevent the execution of tainted code i.e. the application runtime affords full contextual awareness. Thus, some of the most common forms of cyber-attack such as Command Injection, Cross Site Scripting and SQL Injection can be mitigated with precision, avoiding false positives and false negatives.

The definitive traits of runtime protection that distinguish it from preceding cybersecurity technologies such as WAF, Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) are:

- Precision Application Protection
- Virtual Patching
- Pre-emptive Security

This discussion paper describes the defining characteristics of runtime application protection and will demonstrate why the Waratek Application Security Platform offers the most advanced and extensive deployment of runtime application security in global use today.
Instrumentation vs Virtualization

There are two categories of runtime protection implementation: instrumented runtime protection and runtime protection by virtualization also known as containerized runtime protection. The term Runtime Application Self-Protection or RASP is also used to refer to this class of solution.

Instrumentation typically requires that specific sections of code are ‘instrumented’ to enable runtime protection for those sections of code. In its broadest sense, instrumentation results in a change to application artefacts such as source code, deployment descriptors or binaries to allow highly targeted use of runtime protections. Instrumentation requires recompilation or redeployment of the application and will result in operational intervention when the configuration is first introduced or subsequently changed.

Virtualization-based runtime protection implementations apply a virtual container using rules to govern how the application is protected. Waratek’s container-based solution has the advantage of allowing rule configurations to be completely separated from the application artefacts and has no impact on the application lifecycle or its normal operations.

See “How Waratek Works” for technical details of implementing the Waratek application security solution.

This distinction between instrumented and containerized runtime protection is an important factor when deciding between implementation approaches. Let’s review how each of these categories implement the defining capabilities for runtime protection before highlighting why Waratek’s implementation has resulted in it being the most extensive deployment of a runtime security solution in the Java space today.

Precision Application Protection

Arguably the most pernicious problem with WAFs is the sheer volume of false
positives that are generated. Beyond being a major source of confusion and distracting from efforts to deal with real threats, the imprecision of a WAF can also cause denial of service to the very resources it attempts to protect.

Precision Application Protection refers to the ability to eliminate false positives and false negatives. Runtime protection technologies can achieve this by running within the application or the application container and thus have access to all application and user data as well as the context needed to differentiate between the two.

The capability to distinguish user and application data is central to detecting tainted code (code that has malicious logic injected within it) unambiguously and without generating any false positives or false negatives.

How different vendors implement taint detection determines the degree of precision they can apply to protection and the overall simplicity of configuring runtime protections. Beyond the elimination of false positives, the most common attack vectors can all be mitigated using simple, generalized rules.

When compared to the complex nature of a WAF and the overhead of managing large numbers of rules, the business benefits are clear; reliable and accurate determination of security violation attempts are considerably simpler to implement and maintain.

Instrumentation allows fine-grained control directly in the code base. Developers have complete control over which lines of their code will be enabled, allowing them to strike a balance between protection and performance where such considerations are deemed to be important. For example, in ultra-high performance applications.

However, the cost of this implementation of precision is that it demands prior knowledge of the application and a great deal of developer attention to configure. This can generate significant costs both in terms of time and financial expense when implementing RASP and, thus, greatly reduce the scope of the protection to perhaps only critical applications.

The use case for instrumented runtime protection is, therefore, constrained to applications that, for reasons mostly pertaining to low latency or high performance, must maintain sub millisecond latency.
Containerized runtime protection, on one hand, provides fine-grained precision of the container rather than the code. For example, one can alter the JVM to allow/deny/replace the execution of code using rules that the virtualized runtime protection imposes on the container rather than the application logic.

Containerization does not demand direct interaction with the application code, it requires no prior knowledge of the application, and there is no recompilation step or any other form of post processing that requires an application or container restart.

Altering the runtime behaviour in the container does not impose any service disruption to legitimate functionality as well. For any large-scale deployment in the enterprise, the ability to abstract protection from the code and application is critical for reducing the overall effort of configuring runtime protections, avoiding operational downtime and reacting quickly to newly identified threats.

Waratek’s Application Security Platform goes even further, eliminating the performance impact that was associated with early implementations of containerized runtime protections. Ultimately, when evaluating instrumented vs containerised precision, the benefits of Waratek make it more suitable in the vast majority of use cases.

**Virtual Patching**

Through 2020, 99% of vulnerabilities exploited will continue to be ones known by security and IT professionals for at least a year\(^x\). Understandably, keeping up-to-date with the latest patches has become a major concern for the enterprise and a significant challenge for every CISO.

Add to this shortening time between the emergence of exploits, and the sheer number of new vulnerabilities; the pressure on IT managers to rapidly patch production systems begins to directly conflict with configuration management best practices of quality assurance testing.

Zero Day Vulnerabilities are also another cause of disturbance to planned application patching and release schedules, diverting yet more resources from application development to patching. Many
organizations are struggling to keep current with the constant release of new patches and updates. At the same time, they are under pressure to maintain 100% availability of critical business systems.

The solution is Virtual Patching, which is defined as: The capability to apply both routine and emergency security patches without the need to change code and with no downtime. Waratek's Virtual Patching solution is a non-intrusive, centrally managed operation, which can dynamically patch applications without disruption to normal operations or scheduled down time.

Waratek's implementation of Virtual Patching replicates the effect of binary patches using simple rules provided to the Waratek Application Security Platform. For example, Waratek replicates CPUs issued by Oracle for Java on a quarterly basis giving businesses the option of obviating the need to apply CPU binary patches.

This is important for the enterprise as applying binary patches to hundreds or thousands of JVMs can be a difficult process requiring not only engineering resources but also scheduled down time to apply and test the patch.

For large enterprises with lengthy and complicated patching processes that demand considerable orchestration between different teams, this aspect of Virtual Patching provides considerable cost savings and reduces the overall risk associated with intervention on production systems.

Legacy Java

Containerized implementations of runtime protections can also extend Virtual Patching to protect legacy applications. However, the ability to host legacy code within a container that provides the same degree of protection as though it is running in an updated and compliant version of the runtime is unique to Waratek.

Many businesses cannot update critical legacy Java applications as they are incompatible with newer versions of the Java Virtual Machine and the cost (in time and money) of upgrading the code can also be prohibitive. The business may also find the option to upgrade is not possible due to lack of expertise.
In the past, the only option for mitigating the risks posed by such applications has been to replace the applications, over time, to reduce the costs of the technology refresh. However, until they are replaced they pose a significant risk to the business, one that is potentially unacceptable.

Waratek allows legacy applications to be run within a container attached to the latest version of Java. This means that the vulnerabilities associated with legacy code are remediated and can be patched using Virtual Patching without requiring any code changes.

**Pre-emptive Security**

Zero Day Attacks⁶ are the worst case scenario for any business. Zero Day Vulnerabilities can be exploited continuously until they are identified and existing application security technologies provide little or no protection to this class of attack.

A well implemented runtime protection solution, however, can afford pre-emptive security by disabling the vectors that are commonly exploited by Zero Day Attacks.

For example, a Waratek enabled Java Virtual Machine can be configured to deny the creation resources required to open network connections, access databases or files, either entirely if the application has no need of them or allowing creation for the specific resources and application requires.

This fine-grained control can stretch to every aspect of the JVM runtime including the loading of packages and classes. While vulnerability may technically still exist in the application logic, they cannot be exploited and alerts can be raised when illicit attempts are made to access protected resources.

Waratek takes this capability to its logical extreme by providing full Zero Day Defense. Waratek’s out-of-the-box settings protect applications at every level of the application stack - the business logic layer, third party components, frameworks and API’s as well as the JVM platform itself. Waratek protects those layers from both known and unknown vulnerabilities, including ten out of ten on the OWASP Top Ten list of known flaws,
Waratek also provides an advanced feature that profiles an application and disables any unneeded or unnecessary components of the application. Given that the vast majority of Java artefacts are third-party API’s and libraries that are never used, this feature can reduce the attack surface of an application by at least 80%.

Summary

Waratek differentiates itself from other application security providers by its sophisticated virtualized, container-based runtime protections. Waratek’s Application Security Platform is highly accurate, easy to install, simple to operate and does not impact the performance of an application.

The fundamental characteristics of Waratek’s solution are:

- No false positives backed by a $10,000 guarantee per unique event
- No prior knowledge of application code required
- No code changes needed to implement RASP
- No additional hardware is necessary
- The entire application stack is protected

The key benefits of Waratek are:

- Application Security – state of the art, precision security without slowing down the application and without false positives
- Zero Day Defense – pre-emptive defense against the most damaging attacks
- Virtual Patching – reducing the cost and complexity of patch management
- Legacy Java – providing an immediate solution to challenging sources of risk

---

1. Make Apps Self Protecting, Self-Diagnosing and Self Testing, Waratek, April 2015
2. Applying KnowIT principles when deploying Waratek AppSecurity for Java, Waratek, August 2015
5. OWASP Top Ten Project, OWASP
6. Command Injection, OWASP
7. Cross Site Scripting Flaw, OWASP
8. SQL Injection, OWASP
10. Zero-day (computing), Wikipedia