waratek

WHITE PAPER

Runtime Application Self Protection (RASP)

Fast Accurate Runtime Protection

Application Protection in a software-driven world

We live in a world where even our toothbrushes and toasters use software to meet our daily needs. Yet, we still rely on 1980's vintage cybersecurity techniques to protect enterprise applications and to look for indicators of malicious attacks.

Time consuming and risky physical patching is still the primary approach to addressing known security bugs. And, we still guess if an attack is actually an attack using heuristics that are prone to false positives, slow application performance, and require a significant amount of manual intervention in the form of whitelists and blacklists as well as routine tuning.

The reality is simple - these approaches increasingly do not work. If you need proof the status quo is no longer effective, look no further:

- In 2017, a record number of cyberattacks resulted in the highest amount of stolen data in the 18 years that breaches have been tracked.
- Gartner predicts that 99% of successful cyberattacks will be the result of software vulnerabilities known – but unpatched – for at least one year.
- NIST adds a new software flaw to the National Vulnerability Database every 30 minutes on average.

Application software vulnerabilities – left unpatched - can be used to steal small and large amounts of data from a single organization. A 2018 report from The Ponemon Institute and IBM shows that it takes only a matter of days for a hacker to exploit a vulnerable application, but it takes 197 days on average for the attack to be discovered and more than two months (69 days) before the attack is contained and blocked.

The application security market is filled with very effective tools to help you identify software flaws, but tools that help you fix those flaws quickly and easily are rare. Traditional application security tools also do not address related issues such as timely patching and out-of-support applications that represent security risks and compliance roadblocks.

So, if traditional cybersecurity approaches are not stopping or slowing the number and severity of attacks, where do we go from here to improve application security?

What is Runtime Protection?

Are all offerings the same?

Traditional Approach vs Innovative Techniques

The term Runtime Application Self-Protection or RASP refers to a relatively new class of application security solutions that can be implemented in an application's runtime by one of two basic methods: the same traditional approaches based on heuristics and used by Web Application Firewalls (WAF) that have significant negative side-effects; and, a deterministic approach that takes advantage of the compilation pipeline to offer solutions that are more accurate, easier to install, simpler to operate, and has an ultra-low performance impact.

Traditional approaches typically require 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.

Compiler-based runtime protection solutions use the "just-in-time" (JIT) compiler of Java and .NET platforms to correct known vulnerabilities and insert security rules that block known and Zero Day attacks. This same technology can also be used to create a virtual container to virtually upgrade out of support Java applications using a guest/host approach.

The distinction between traditional and compiler-based runtime protection is an important factor when deciding between implementation approaches.

Let's review each of these implementation methods before highlighting why Waratek's runtime protection solution is the choice for leading companies around the world.

The Traditional, WAF-style Approach

The most common form of application security today is the WAF, a derivation of the network firewall dating back to the 1980s. Most RASP providers have adopted the same techniques as WAF when instrumenting the runtime.

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.

However, this implementation demands prior knowledge of the application and a great deal of developer attention to configure and maintain. This can generate significant costs in terms of time and financial expense when implementing RASP, greatly reducing the scope of the protection and ability of resource constrained teams to deploy.

Arguably the most common problem with WAF and WAF -style RASP is the inaccurate nature of the heuristics used to detect attacks and protect applications. The sheer volume of false positives generated can be a major source of confusion and distraction from efforts to deal with real threats.

Instrumentation also leads to increased performance overhead and demands a significant amount of time be devoted to on-going tuning. The more heavily the code is instrumented, the greater the performance overhead and maintenance burden.

> **"EVENTUALLY RUNTIME APPLICATION SELF-PROTECTION (RASP) WILL TAKE OVER WEB APPLICATION FIREWALL (WAF) AS THE BEST WAY TO COMBAT WEB APP ATTACKS. THEY HAVE DEEPER KNOWLEDGE THAN WAFS OF THE APPLICATIONS THAT THEY PROTECT, AND THEY CAN VIRTUALLY PATCH VULNERABILITIES AND WEAKNESSES."**

Source: Amy DeMartine, Forrester

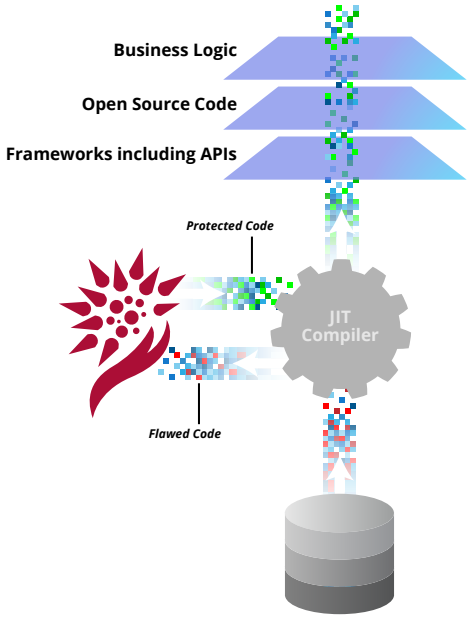
The Compiler-based Approach

How a RASP provider detects tainted code (code that has malicious logic injected within it), without generating any false positives or false negatives determines the degree of precision and overall simplicity of configuring runtime protections. Depending on the approach, common attack vectors can be mitigated using simple, generalized rules and false positives can be eliminated.

Compiler-based RASP does not demand direct interaction with the application code. No prior knowledge of the application is required, and there is no recompilation step or any other form of post-processing that forces a restart.

For example, the JVM or CLR can be altered to allow/deny/replace the execution of code using the compiler pipeline without application downtime, source code changes or any routine tuning. Once compiled into memory, the patch or security rule continues to execute until the patch/rule is removed.

When compared to the complex nature of a WAF or WAF-style RASP and the overhead of managing large numbers of rules, the business benefits of a compiler-based approach are clear; high accuracy, low maintenance, ultra-low performance overhead, and virtually no disruption of service once deployed.



Waratek
Dublin Office
8 Harcourt Street
Dublin, Ireland

US Office
117 Towne Lake Pkwy, Ste 210
Woodstock, GA 30188

What are the issues facing Application Security Teams today?

Vulnerability Response (Patching)

In a report on the 2017 Equifax cyberattack, the US Government Accountability Office noted that data thieves infiltrated the credit reporting agency's system via a known, unpatched vulnerability – two days after the flaw was publicly announced. The shrinking time between the emergence of flaws and exploits, and the sheer number of new vulnerabilities reported each day directly conflicts with patch management best practices and quality assurance testing.

The Ponemon Institute's recent study on The State of Vulnerability Response points out that spending on headcount for patching is on the rise. Ponemon calls this the "Patching Paradox" – the belief that more staff dedicated to vulnerability response will equal better security when the underlying barrier is actually too many manual processes and too many siloed tools.

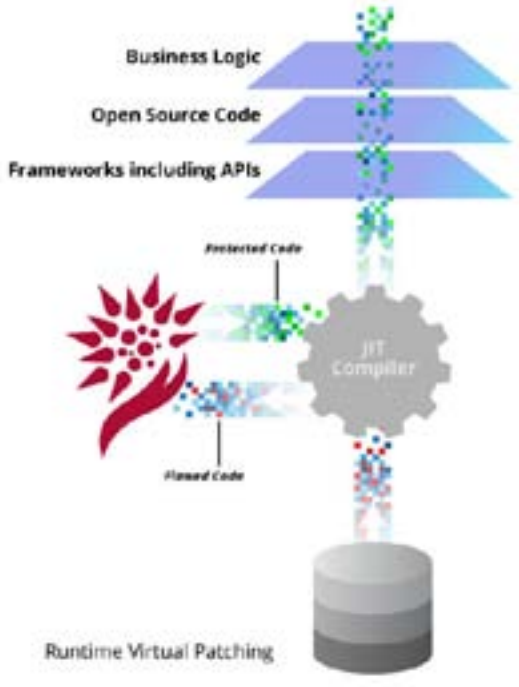
The Ponemon study concludes that the best solution is to rely more on automation to replace manual processes along with using tools that complement, not complicate, efforts to improve patching.

Traditional virtual patching, also known as virtual shielding, is used by Web Application Firewall (WAF) and most RASP providers, as a way to quickly protect applications against known CVEs. However, traditional virtual patches still leave you vulnerable to attack since these tools do not fix the flawed code and often result in false negatives and false positives. Routine tuning is also required for a WAF-style virtual patch to remain effective against attacks.

ACCORDING TO PONEMON AN AVERAGE ORGANIZATION SPENDS 321 HOURS PER WEEK DEDICATED TO PATCHING. AT AN AVERAGE RATE OF $63 USD PER HOUR FOR SECURITY ENGINEERS, THAT WORKS OUT TO BE MORE THAN $20,000 PER WEEK AND MORE THAN $1 MILLION PER YEAR.

Waratek's first-of-its-kind Runtime Virtual Patching is fundamentally different. A runtime virtual patch is the functional equivalent of a physical binary patch that is applied while the application runs with no source code changes and no tuning required. The known CVE is fixed in the compilation pipeline of Java and .NET applications, reducing the time-to-patch across an enterprise to a matter of minutes.

A single patch administrator can download and deploy routine and out-of-cycle runtime virtual patches across an entire application estate in a matter of minutes – work that according to Ponemon now takes an average of nearly 17,000 work hours per year. And, Waratek Patch can save 75% - 90% compared to the average costs of traditional patching programs reported by Ponemon.



Waratek
Dublin Office
8 Harcourt Street
Dublin, Ireland

US Office
117 Towne Lake Pkwy, Ste 210
Woodstock, GA 30188

waratek.com

What are the issues facing Application Security Teams today?

Known and Zero-Day Attacks

Cross-site scripting (XSS) remains the number one attack vector for web applications while SQL injection (SQLi) bug rates came in second, according to a review of two trillion lines of code by CA Veracode. That same review revealed that 87.5 percent of Java applications scanned for the 2018 report had at least one vulnerable component that could be exploited.

Known attack vectors require a significant amount of application security teams' time and resources. But, Zero Day attacks are the worst case scenario for any business. Zero Day vulnerabilities can be exploited continuously until they are identified and stopped.

Traditional application security solutions like WAF and WAF-like RASP tend to rely on instrumentation or filter-based approaches. They use pattern matching, regular expressions, whitelisting and blacklisting, and other heuristics to provide protection that is limited in scope and generally can only protect the business logic layer of an application's code.

These common approaches are time and resource intensive to install and operate and they can also lead to increased performance overhead. The unwanted byproduct of such heuristic approaches is often false positives.

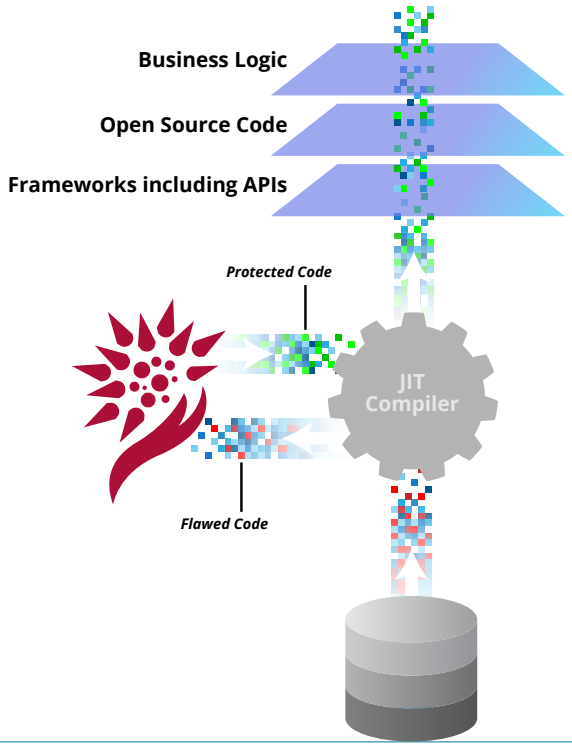
87.5% JAVA AND 85.7% .NET APPLICATIONS CONTAIN AT LEAST ONE VULNERABLE COMPONENT.

SOURCE: VERACODE STATE OF SOFTWARE SECURITY 2018

A compiler-based runtime protection solution, however, can provide highly accurate, low-impact security by using the JIT Compiler to disable the vectors that are commonly exploited by known and Zero Day Attacks. For example, Waratek can be configured to deny the resources required to open network connections or to access databases and files - either entirely, if the application has no need of them, or only allowing access to the specific resources the application requires.

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

Waratek's compiler-based approach generates no false positives and detects and protects against known as well as Zero Day attacks using a series of out-of-the-box settings that are fully configurable. Waratek protection solutions can 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 and CLR platforms.



Waratek
Dublin Office
8 Harcourt Street
Dublin, Ireland

US Office
117 Towne Lake Pkwy, Ste 210
Woodstock, GA 30188

What are the issues facing Application Security Teams today?

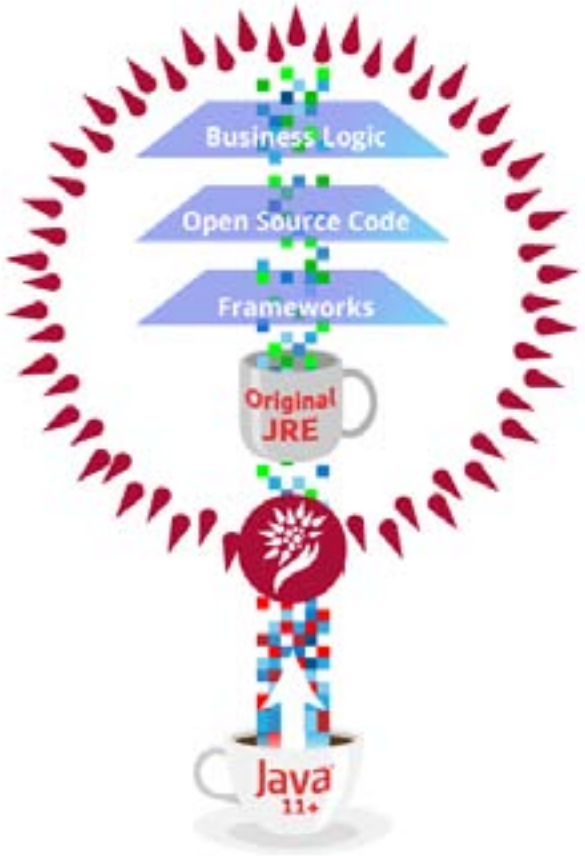
Out-of-Support Applications

Businesses often struggle to update out-of-support Java applications that are incompatible with newer versions of Java. Historically, the only option for mitigating the risks posed by such applications has been to rewrite or replace the applications at significant cost in time and money. However, until they are replaced they pose a significant security and compliance risk to the business.

Compiler-based implementations of runtime protections can virtually upgrade out-of-support Java applications without source code changes. Waratek wraps the entire application stack – including the original Java Runtime Environment (JRE) - in a virtual "guest" container on top of a current JVM that acts as the "host."

The virtualization-based guest / host architecture allows the application to continue to run using its original business logic, but protected by virtual patches and a current version of the JVM without the time, risks, or costs of rewriting the application. This arrangement also serves as a compensating control for purposes of compliance, further reducing costs and the pressure to immediately rewrite older applications.

This unique, patented approach to application protection has been in production worldwide for more than three years without generating a single false positive.



JAVA ADOPTION IN 2018:
LESS THAN 5% HAVE ADOPTED JAVA 9 OR 10.
10.6% ARE USING JAVA 7 OR EARLIER
SOURCE: BAELDUNG

Summary

Some of the world's leading companies use Waratek to patch, secure and upgrade their mission critical applications. Waratek is a pioneer in the next generation of application security solutions. Using patented technology, Waratek makes it easy for security teams to instantly patch known Java and .NET flaws with no downtime, protect their applications from known and Zero Day attacks, and virtually upgrade out-of-support Java applications – all without time consuming and expensive source code changes or unacceptable performance overhead.

Waratek is one of CSO Online's Best Security Software solutions of 2017, a winner of the RSA Innovation Sandbox Award, and more than a dozen other awards and recognitions.

The key benefits of Waratek are:

- Runtime Vulnerability Patching – reducing the time, cost, and complexity of patch management without downtime
- Runtime Application Self Protection – state-of-the-art protection against known and unknown attacks without slowing down the application and without false positives
- Legacy Runtime Upgrade – providing an instant solution for security and compliance risks linked to out-of-support Java applications at a fraction of the cost without source code changes

Resource limited application security teams can self-manage Waratek's solutions or rely on Waratek and authorized partners for installation, operation, custom rules, and virtual patches.

"BECAUSE OF WARATEK'S SOLUTION THE ATTACKS WERE IMMEDIATELY STOPPED, AND THE SOLUTION AUTOMATICALLY ALERTED US TO THE ATTEMPT. ALL OF THE HACKER'S MALICIOUS SCRIPTS FAILED, WHICH TOOK US TO A NEW LEVEL OF CONFIDENCE — THE WARATEK VIRTUAL PATCH IS PROVIDING THE PROTECTION WE NEED, BETTER AND FASTER THAN WE EVER THOUGHT POSSIBLE."

HEALTHCARE COMPANY, CHIEF INFORMATION SECURITY OFFICER

Waratek
Dublin Office
8 Harcourt Street
Dublin, Ireland

US Office
117 Towne Lake Pkwy, Ste 210
Woodstock, GA 30188