RSAC Innovation Sandbox 2022 FINALIST

SOC 2 COMPLIANT TYPE II

Araali Networks

Prioritized Risk Mitigation for Cloud Environments

The Time for Resilient Shielding has Arrived

Keeping systems and software patched and up to date has become a constant mantra within cybersecurity circles. The majority of cybersecurity incidents occur due to unpatched systems and software.

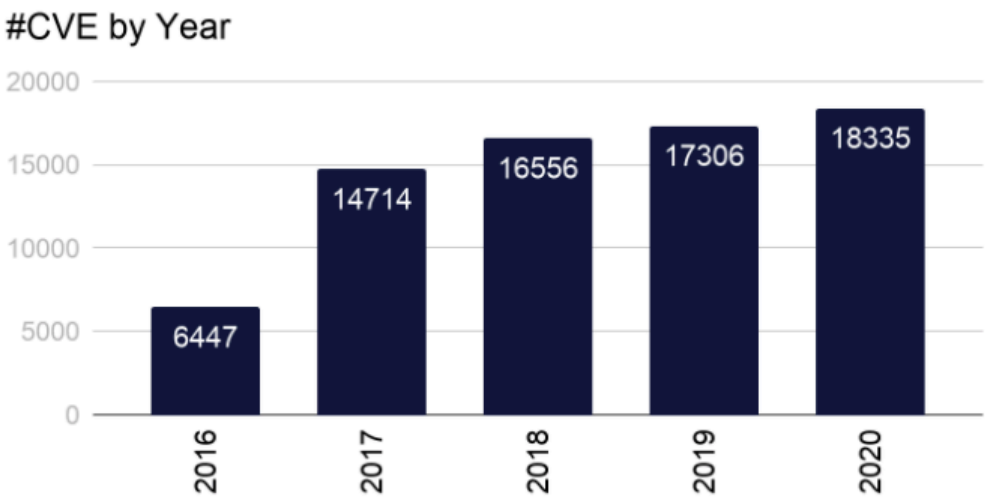


Fig: The growing volume of vulnerabilities, 20K in 2021 (4K high and 13K medium)

According to a study conducted by the Ponemon Institute, 60% of breaches in 2019 were linked to a vulnerability in which a patch was available, but not applied. In fact, according to the UK's National Cyber Security Center, the one thing that an organization should do to protect from cyberattacks is implementing updates and patching for known vulnerabilities. That's because threat actors are constantly probing organizations for known software and system vulnerabilities to exploit. As an example, the Wannacry ransomware crypto worm is still used today four years after it was first implemented as systems to this day still remain unpatched for its targeted vulnerability.

The Challenges of Traditional Patching

We all know that keeping systems and software fully patched is essential to protect against malware and cyberattacks. So why don't IT departments do it? Well for one, we 'know' a lot of things. We know that we should exercise three times a week, stay away from sugary drinks and go to the doctor for annual checkups. While we intend to do these things, we often don't for a variety of reasons. It's the same when it comes to the traditional physical patching of systems and software. Some of the reasons why IT tends to procrastinate patching efforts include the following:

The list of systems and software that need patching has become overwhelming

- Organizations are completely dependent on their invested technologies and can't afford for systems and applications to be taken offline for necessary patching
- Because there are so many moving parts that must work cooperatively within today's systems, patches must often be implemented in a testing environment first to discover introduced glitches and incompatibilities
- The network can slow to a crawl for organizations with a large on-prem presence whenever a critical patch is needed across the entire IT estate
- Because IT staffs are so stretched today, they must prioritize patching efforts while also juggling other projects that bring greater value to the organization

While it would be nice if enterprises could simply implement automatic patching of 'everything' in the same fashion that many personal computer users do, it's just not possible for today's complex enterprise environments.

The challenge of remediation

35-64 days

Mean time to remediate medium and critical vulnerabilities during which teams take unnecessary risk by running vulnerable software in production

The patching race

Patching is a race between defenders and threat actors. Once a patch is released, it's a matter of who gets there first. Defenders need to patch everything perfectly, and on time; while attackers need only find a single vulnerable system to establish their attack beach head. Even for IT departments that are vigilant about patching, they often find themselves waiting for the designated release dates such as 'Patch Tuesday.' It is no surprise why the attackers have the advantage.

The practice known as Patch and Pray

There is a lot to pray for when it comes to patching. Patching makes you reliant on other people to patch and release their binaries. For developers, it is code that they don't normally build as part of a software release. Software companies depend on bug bounty hunters and external cybersecurity teams to discover vulnerabilities within their own code. The whole process in fact requires an involved leap of faith. One must assume that the fully patched software behaves as intended and that an attacker can't simply gain influence over the patch as well. A 'patch and forget strategy' is all too common as organizations don't have the time to check on applied patches. And then there are those instances in which a patch goes bad and creates disruptions that must be remediated. If patching is so important, it is clear there needs to be a better way.

WAF style virtual patching

Some have turned to virtual patching. Virtual patching was a concept introduced by the web application firewall (WAF) industry. This WAF style of patching is about applying a mitigation signature in an external appliance (WAF) to ensure that a designated attack never reaches the application. There are a few problems with this style of virtual patching, however:

- These signatures are easily mutated and defeated by an attacker
- The appliance becomes a choke point and is typically applied at the perimeter
- There is limited capacity for these rules/signatures,
- It requires complex processing and doesn't scale adequately for cloud-native workloads.

While WAF-style virtual patching alleviates some of the weaknesses of traditional physical patching, it too has far too many dependencies while introducing its own sets of challenges.

Araali's Resilient Shielding

Fortunately, there is a new way to patch and protect today's software. This modernistic solution is known as **Araali's resilient sheilding**.

Resilient Shielding does not involve the perpetual chasing of vulnerabilities and signatures. Like its virtual patching predecessors, it is applied externally. The difference though is the way in which it creates a virtual operating environment that enforces the Principle of Least Privilege (PoLP) to all applications. In the same way that PoLP policies deny standard users blanket level local admin rights, resilient shielding locks down the application by enforcing its pre-specified behavior, thus locking any and all deviations.

As a result, an attacker can reach an application, but fail to gain influence over it. We hold the app to pre-specified behavior and lock out deviations.

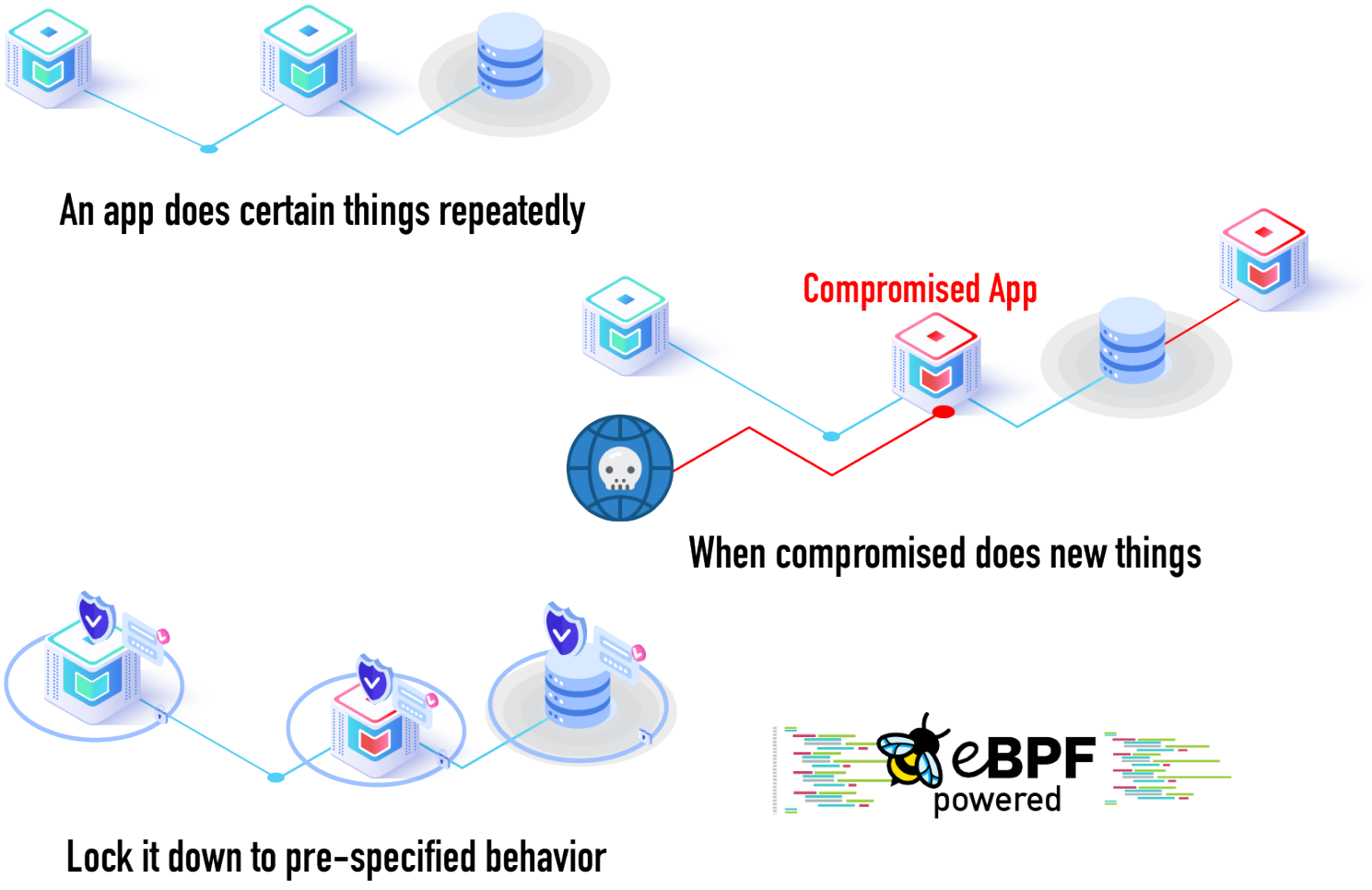
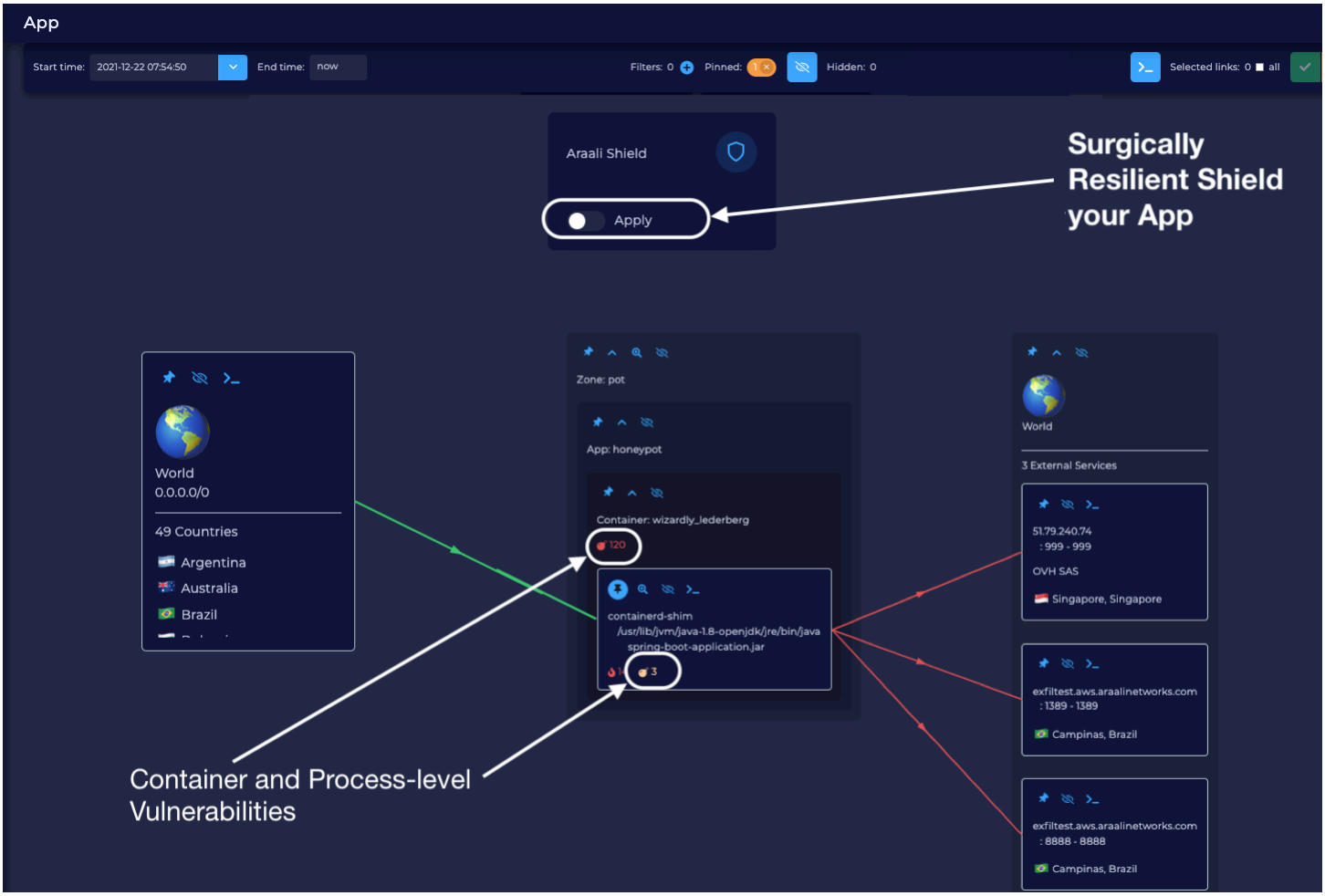


Fig: Resilient Shielding Concept

The technology behind resilient shielding

For the cloud native world, Araali installs a distributed, performance optimized eBPF-based tracing and enforcement module as a kubernetes daemonset. Behavior enforcing policies can be deployed “as code” along with the apps. Policies then follow the apps and are not permanently glued to infrastructure. Araali incorporates intelligence, centralized management and automated discovery that manage all policies.

The technical debt of legacy apps (too many vulnerabilities to patch) applies to us as well. We solve this problem by allowing a slow and guided transition into a world where every application runs with Araali’s Resilient Patch. During the discovery phase, the enforcement boundary can be very broad while application behavior is studied and evaluated. You can then tighten that boundary at any time to contain application behavior within that defined boundary. This containment not only protects the internal code but increases the resilience for all applications of the organization. While it doesn’t prevent lateral movement, it does shield the organization, preventing any possibility of command and control, remote shells, or data leaks. Resilient Shielding is not an endpoint, but a way of securing your applications to buy time so that your IT team can go about patching your apps according to priority.



Advantages of Resilient Shielding

Araali's resilient shielding solution comes with built-in monitoring and self-correction. All policies are auto-discovered and self-correcting. The automated nature of our solution removes the element of human error which is often the weakest link in patching operations. It includes a self-correction element that ensures that all behavior is captured independently of the policy assigned to it.

With Araali, you can "patch once," and "prevent forever." It guards against known as well as unknown and undisclosed vulnerabilities. As a result, zero-day attacks become a thing of the past.

With Araali's Resilient Shielding, you can harden other people's code. Because the patch is external, it doesn't matter where the code came from, or if it is a mashup from many disparate sources.

Araali's model of Resilient Shielding is known to prevent breaches emanating from application compromise. See our short demo video links below:

- Log4shell (remote code execution)
- CapitalOne attack detection and prevention (SSRF vulnerability)
- Equifax (remote code execution)

Find out more about Resilient Shielding

If you are still relying on the sole practice of traditional physical patching or encountering the challenges and shortcomings of the WAF style of virtual patching, we encourage you to learn more about resilient shielding and how it is transforming the security industry. Forget the ritual of dealing with signatures and vulnerabilities. Stop running a race against external threat actors you cannot win against. The time has come for Araali's Resilient Shielding.

Partners

awsmarketplace

aws partner network Select Technology Partner

Google Cloud Partner

Get a Free Trial:

https://www.araalinetworks.com/signup

RSAC Innovation Sandbox 2022 FINALIST

AICPA SOC aicpa.org/soc4so SOC for Service Organizations | Service Organizations™

SOC 2 COMPLIANT TYPE II