CSO
FROM IDG
February 21, 2018 www.csoonline.com

REVIEW

How Contrast Security protects applications from the inside out

Contrast Security has one of the most elegant solutions out there for application security. We can see why it scored 100 percent on the OWASP Security Benchmark.

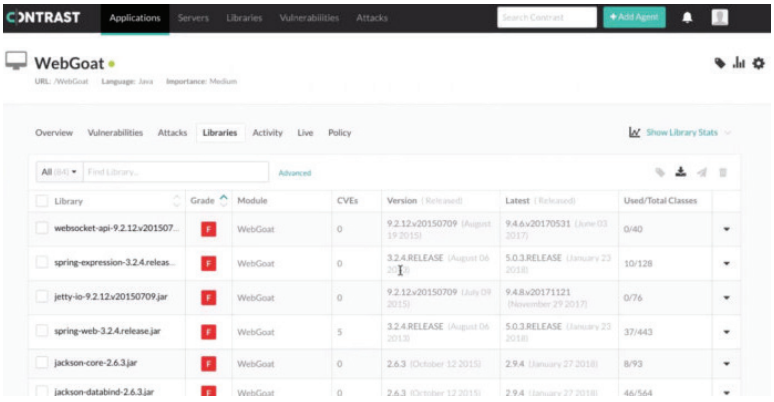
By John Breeden II

Proper cybersecurity these days requires a defense in depth. Like in military planning, relying on a single defensive line is a recipe for failure, especially in the long term. Eventually something or somebody will learn how to bypass or defeat a narrow perimeter, allowing them to freely attack everything sitting behind it.

As such, cybersecurity programs tend to look at the problem of defense from a lot of different angles, with the expectation that enterprises will employ several different types of security at the same time. This has led to a different problem: alert fatigue setting in on IT teams as all of those programs sound the alarm many times, and all the time.

The Contrast Security suite aims to change that trend in two important ways. First, it takes one critical aspect of cybersecurity today, application security, and condenses it into a single program that can protect apps from the time development first begins all the way through deployment and their full lifecycle. Second, because Contrast Security embeds agents inside each app that it is protecting, essentially becoming a part of the program, there is almost no chance of false positives. In fact, it scored a rare 100 percent on the OWASP Security Benchmark, passing over 2,000 tests without generating any false positives.

The secret sauce for Contrast Security is its use of bytecode instrumentation, a feature in Java used to help integrate programs and application features during development. Only here, Contrast Security uses it for the purpose of cybersecurity, specifically embedding an agent into an application, which will thereafter be directly monitored and protected from the inside out. In a sense, it turns any type of normal application into one that is designed to focus on security. But don't worry, all the normal business-focused tasks for the app will still function.



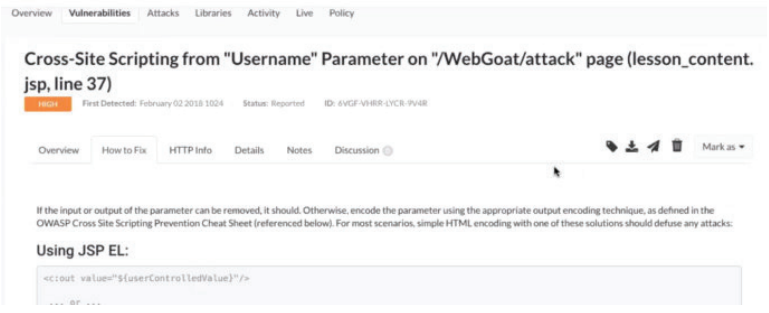
John Breeden II/IDG

Because Contrast Security uses bytecode instrumentation to insert agents into applications, it can alert development teams whenever they trigger a vulnerability as the code is being programmed or tested, with no scanning required.

The only restriction to the Contrast Security plan is the language that the app is written in, which needs to support bytecode instrumentation. Java and .NET do, and so do most other popular languages like Ruby and Python. But if you use some sort of odd or unique language to program your apps, you might be out of luck.

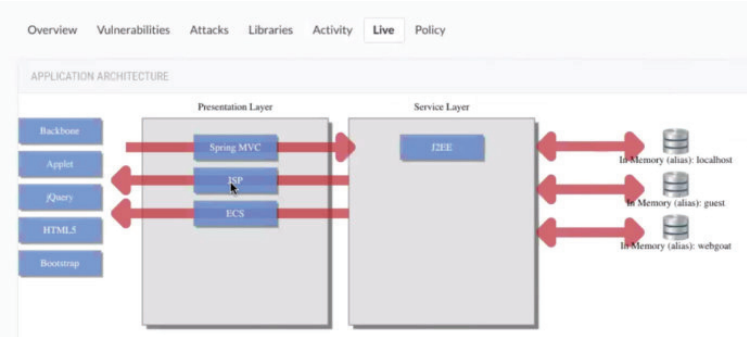
Once the new app is implanted with the Contrast Security agent, development and programming can continue as normal. However, the agent will alert developers whenever they write code with some sort of a vulnerability. It will also alert them if a vulnerability is discovered while the app is being tested. At no time does a developer need to run any sort of scan or compiler to trigger Contrast Security. Because it's a part of the program from the start, the agent will let them know whenever something breaks security policy or triggers a vulnerability.

That part of the suite is called Contrast Assess. If all goes well there, then the second part of the suite, called Contrast Protect, may have little to do. Contrast Protect works almost the same way as Contrast Assess, and uses the same embedded agent, but it works in the production environment, looking for exploits and



John Breeden II/IDG

Whenever a vulnerability is found, an alert can be sent to the Contrast Security main console, or to whatever tools developers or IT teams are comfortable using.



John Breeden II/IDG

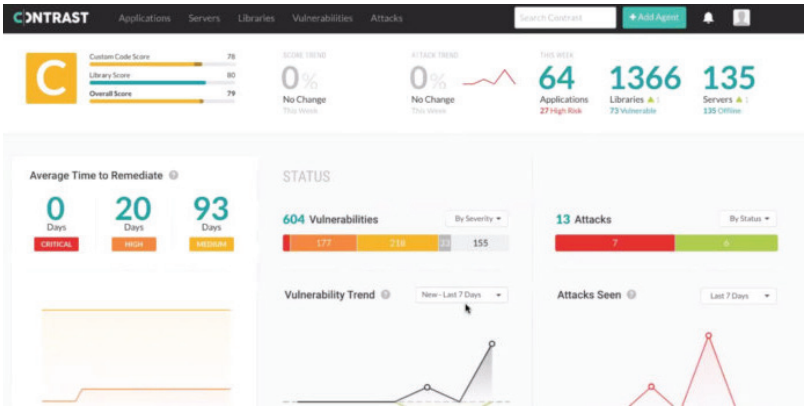
With Contrast Security agents embedded as part of the applications they are protecting, they are able to report, and visually show, every aspect of the app, including every other network asset that it touches.

unknown threats, and reporting what it finds to a SIEM console, next generation firewall, or whatever security tools an organization already has in place.

Interestingly enough, Contrast Protect can track threats that do not breach a network, which it deems as a probe, and can even tie back the attack's failure to actions taken in the development phase, thanks to Contrast Assess. So, if an attack would have gotten through, but the developers plugged that vulnerability while the app was being programmed, credit is given where it is due. Non-successful attacks don't get sent to a SIEM, unless an organization really wants them to, but are fully logged in the Contrast Security main console.

There is a lot of information tracked and graphed within the main console, which is sort of sad because some organizations may not use it. Contrast Security didn't want to require IT teams to learn a new interface or program, so it can report directly to development tools in the Contrast Assess phase. And it can alert SIEMs or other security programs in Contrast Protect mode.

If you do dive into the main interface, you can quickly see valuable information like how long it is taking development teams to fix found vulnerabilities deemed to be of high, medium and low severity. Because the agents are part of the program, Contrast Security knows when a vulnerability is found and fixed, and keeps track of all that info inside the main console. This might be of great interest to a performance auditor, or a manager tasked with streamlining and improving the development process.



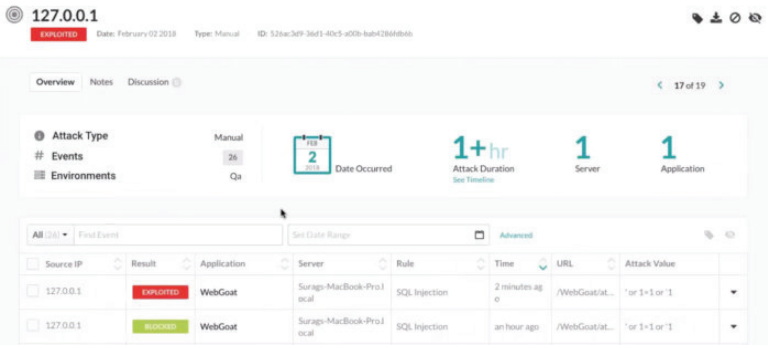
John Breeden II/IDG

If everything is working well, users may never need to log into the Contrast Security console. That's too bad, because it features a helpful and powerful interface.

The one time that most users will absolutely want to go into the Contrast Security main console is when setting up the security policies that agents will follow. While most known vulnerabilities and best practices are included as a default, an organization may have additional policies that Contrast Security can track. For example, if corporate policy requires that critical vulnerabilities get fixed within a week while apps are in development, such a rule can be set, and thereafter it will become one more thing that all agents will monitor.

Testing Contrast Security went incredibly smoothly, both in the development phases and after deployment. It's nearly impossible to trick an app into doing something other than what was intended when there is a dedicated agent intertwined with it, monitoring everything it is doing. Just the process of trying to nudge an app off task triggered an alert, no matter how light a touch was used. And because deployed apps developed in conjunction with Contrast Security agents are likely going to be almost completely secure anyway, it makes the task of exploiting them that much harder. Certainly, all the normal tools in an attacker's arsenal will fail.

Contrast Security has one of the most elegant solutions out there for application security. We can see why it scored 100 percent on the OWASP Security Benchmark. Embedding security agents inside of the apps they are protecting, and providing overwatch on the entire process from development to deployment to production, proves to be a winning strategy that is both effective and lightweight. Contrast Security would thus be a welcome addition to any organization tasked with building and maintaining its own apps.



John Breeden II/IDG

Both successful exploits and unsuccessful attempts are caught by the Contrast Security agent within protected apps. In the event of an actual breach, the alert is made a priority. Otherwise, it's just interesting information for developers programming and protecting future applications.

John Breeden II is an award-winning journalist and reviewer with over 20 years of experience covering technology. He is the CEO of the Tech Writers Bureau, a group that creates technological thought leadership content for organizations of all sizes.