WhatWorks in Application Security: How to Detect and Remediate Application Vulnerabilities and Block Attacks with Contrast Security

CONTRAST SECURITY

WhatWorks is a user-to-user program in which security managers who have implemented effective Internet security technologies tell why they deployed it, how it works, how it improves security, what problems they faced and what lessons they learned. Got a story of your own? A product you'd like to know about? Let us know.

www.sans.org/whatworks

ABOUT CONTRAST SECURITY

Based in Los Altos, CA, Contrast Security is the world's leading provider of security technology that enables software applications to protect themselves against cyberattacks, heralding the new era of self-protecting software. Contrast's patented deep security instrumentation is the breakthrough technology that enables highly accurate assessment and always-on protection of an entire application portfolio, without disruptive scanning or expensive security experts. Only Contrast has sensors that work actively inside applications to uncover vulnerabilities, prevent data breaches, and secure the entire enterprise from development, to operations, to production.

Further information can be found at www.contrastsecurity.com or by following Contrast Security on Twitter at @ContrastSec

ABOUT THE USER

The number of applications being developed and deployed to meet business needs continues to increase, while agile development methodologies drives the shortening of development cycles and testing time. These factors are driving increases in application-level vulnerabilities that are quickly detected and exploited by adversaries. Traditional approaches for dynamic and stations application security testing are often too slow to meet the accelerated pace of today's business environment and don't provide any way to protect vulnerable applications until they can be patched.

During this SANS WhatWorks webcast, Josh Bentley, Application Security Manager at Liberty Mutual, will provide details of his selection and deployment of Contrast Security to enable discovery and inventory of application vulnerabilities and the use of Runtime Application Self Protection (RASP) to shield business critical applications awaiting patching. Join SANS Director of Emerging Security Trends John Pescatore and Josh to hear details on the selection, deployment and experience using Contrast Security. The webcast will contain a discussion of lessons learned and best practices as well as detail the metrics used to demonstrate the value of improved application security and faster security response.

ABOUT THE INTERVIEWER

John Pescatore, Director of Emerging Security Trends, SANS Institute

John Pescatore joined SANS as director of emerging security trends in January 2013 after more than 13 years as lead security analyst for Gartner, 11 years with GTE, and service with both the National Security Agency, where he designed secure voice systems, and the U.S. Secret Service, where he developed secure communications and voice systems "and the occasional ballistic armor installation." John has testified before Congress about cyber security, was named one of the 15 most-influential people in security in 2008 and remains an NSA-certified cryptologic engineer.

Q Tell us a little bit about your background and your role at Liberty Mutual?

A I have been with Liberty for about nine years now. My current role is as an Application Security Manager. I'm a liaison to our international companies working to further all kinds of different security related programs, kind of like a Program Manager.

Q So, what were the business problems that caused you to start looking at solutions like Contrast?

A We have been using all kinds of different testing methods, whether it's SAST (Static Application Security Testing) or DAST (Dynamic Application Security Testing) or external pen testers. Since a major business goal is to speed up the development process and shorten time to market. I had goals to be able to also integrate into agile development frameworks and our CI/CD (Continuous Integration/Continuous Delivery) pipeline in the future. Our needs were security tools and processes that are more accurate and faster.

We were trying to shorten the cycle time of security testing, from identification to fix. I was looking for something that would give us a much quicker turnaround, be more accurate, and then also be able to integrate to our CI/CD pipeline and then shift left.

I was attracted to their next generation AST (Application Security Testing) technology. Contrast's technology instruments the application using the Contrast agent and it is watching all the time. It has visibility beyond the common application scanning technology into the applications runtime environment.

Q What was driving that need to move faster? Was there a move in agile development, moving to more cloud apps, both?

A Both of those were factors. We've got efforts to move things to the cloud. We've got efforts to make things a lot more agile. In order to increase competitive advantage, everyone wants to be able to develop products quicker and be able to stay relevant by being able to create something faster.

Q You were using some dynamic and static analysis type testing tools when you decided you need to be able to move more quickly and increase accuracy. Did you put out an RFP? Did you bring in tools and test them?

A Well, our company has a lot of tools on the shelf. We've had a lot of different vendors providing us tools and solutions and I've had exposure to a lot of them. As I attended different conferences, I just kept my eye out for new technology and I would investigate when I saw something that was different than what we were already using. That is what led me to focus on Contrast's product.

Q What did you feel was different about the Contrast solution compared to other standard approaches?

A I was attracted to their next generation AST (Application Security Testing) technology. Contrast's technology instruments the application using the Contrast agent and it is watching all the time. It has visibility beyond the common application scanning technology into the applications runtime environment. Next generation technology also enables applications to be self-protecting. I thought that was quite appealing and it proved to be better for our use cases. Fewer false positives as well, as it gave us better true positives. In addition, it gave us additional application telemetry as well.

Q Did you pick out some candidate applications and run the other tools against them and then install Contrast and see how it compared?

A We did. We selected a few different applications from a couple different development teams. These applications being tested already using other tools. We just used the tools we had and compared the results to what we found using Contrast.

Q Since this is different than the way these other tools work, tell us how you go about installing or running Contrast.

A Contrast is an agent-based solution, kind of like the Application Performance Monitoring tool we use, New Relic. A lot of people are familiar with that, where it is integrated and instrumented to become part of your application, running in run time. So, it's not a separate tool or a separate system. Overhead is quite low, even when you turn on the blocking/prevention mode.

An agent can be added and configured very quickly and then it is part of the application and it is looking for vulnerabilities whenever the application is running. The application teams automatically do security testing when they do any of their other normal testing. That was really huge for us.

Contrast also gives us the ability to have an accurate process inventory and gives us telemetry, a view into the application stack or architecture of the application that we've never been able to see automatically diagramed before. That was another key bonus to using this tool. When we implement this tool and show it to developers, we're trying to convince them that this is not just another security-only tool that they have to schedule more time on their project calendar for. This is a tool that developers can use to provide value to their work, and it integrates into the normal processes that they're already doing. With Contrast monitoring within their application, it allows for very quick results and even real-time notifications that vulnerable code was added and is running.

Contrast also gives us the ability to have an accurate process inventory and gives us telemetry, a view into the application stack or architecture of the application that we've never been able to see automatically diagramed before. That was another key bonus to using this tool.

When application teams asked to have new applications tested or they said, "Hey, we heard about this tool, and we want to use it. What does it take?" it was really a pleasant conversation to have because we could tell them, "We can get you results really quick." That was important because it's been known that security isn't always the first priority for developers working toward deadlines.

Application teams that were thinking about security would ask "Will it prevent us from meeting our deadlines? Will we make our timelines?" We found we could say "Look, we need about an hour of your time. In the first ten to fifteen minutes, we'll quickly explain what Contrast is, we'll show you how to deploy the agent, and then we'll spend the rest of the time looking at the results and give you training in the portal that'll allow you to see your results." That's fast and usually helped get over any doubt about impact to deadlines and schedules.

The Contrast tool became something that sold itself, that became a demand-driven tool instead of me on the security side having to go around telling people that they must do it. It picked up some good headwind as far as the implementation went.

The Contrast tool became something that sold itself, that became a demand-driven tool instead of me on the security side having to go around telling people that they must do it. It picked up some good headwind as far as the implementation went.

Q Once you've gone through that with the developers and given them the Contrast tool, do they then get an account on the SaaS platform and they're looking at results directly?

A Well, we went through different phases. Initially, we just wanted them to get experienced, to understand, "Hey, Contrast is a value-add to my development process. It gives me a lot of telemetry about the application and the stack. I've got different charts. It's got some flows that it shows us data real time as its going through the system." So, there were some neat things that they got to pick up right there.

But, a part that really is kind of the icing on the cake that made it, to me, a much better tool is that we weren't just giving them this new tool that was quicker, more reliable, more accurate. It can be much more than that. That's where I really, really think Contrast sets itself far and above the others right now, by integrating itself into the Software Development Life Cycle (SDLC) or the CI/CD pipeline and a lot of other different areas.

We started to add integrations like Jira, Jenkins, and the developers Integrated Development Environment (IDE) on their workstations with Eclipse. This brought results directly to the developer while they are fully shifted left and still coding in Development.

If we were to start at the very far left in the SDLC, the developers are using an IDE, and they're using Eclipse. What we really wanted to do is to take away the ability of our developers to say, "Oh, well, that's a security thing." We wanted to make it so that they realized, no, those are theirs, and they show up in the same defect trackers they were already using.

They already had bug trackers like Jira. So, what we were able to do is add the agent to their application environments. And all of these feed into one application portal, and it starts providing data there and back to the integrated systems.

As the developers are coding and they typically run their applications through testing, with Contrast we automatically give them notifications of any security issues and send the details to their development platform. Now, to me, that's fantastic because now we've got developers at the very beginning as far left as you can go being told right away, "Hey, what you just did there, that was a security issue. It is a defect. That was a vulnerability. Don't migrate to the next environment until you address it." They are getting on the job training. The Contrast tools are helping them see a closer association between security defects and their actions or code that they will have to fix. There's accountability there, and there's training right at that moment of development – that's a huge shift left.

This way, they addressed their code defects when they recognized that it was something they had done, instead of them getting all the way down, two months later when they go to deploy something and they are told of some abstract list of "Security issues" and are reluctant to address "Security's problems." Now we're connecting accountability directly and immediately to the developer who learns to be a better coder. We also have the next phases covered, where if they go into QA, and these issues pop up, we're sending these issues over to Jira with all the rest of the code defects. And an application manager can send it back and say, "You've got a list of things here in your bug tracker that are defects you need to fix."

Q Once you on the security team have done the integration with the Contrast SaaS application and the development environment, then, as long as the developers are installing the agent with their code, then all this integration with the reporting and the Jira tracking all happen automatically?

A Essentially yes. In the IDE a plugin is added by the developer. And once setup they don't need Security involved in the remediation loop. There's a little bit of work you're going to have to do on the Jira side to build out workflows with defect tickets if you want custom work flows. We've also done additional configuration, adding the agent to newly built application servers going to the cloud by adding the agent to AMIs (Amazon Machine Images). So, when we go through the pipeline to generate a new server and application environment, we automatically have the full image built up with the Contrast agent part of the application.

All they have to do is some minor tweaking to complete configuration and onboarding of the application with Contrast Security.

...a part that really is kind of the icing on the cake that made it, to me, a much better tool is that we weren't just giving them this new tool that was quicker, more reliable, more accurate. It can be much more than that. That's where I really, really think Contrast sets itself far and above the others right now, by integrating itself into the Software Development Life Cycle (SDLC) or the CI/CD pipeline and a lot of other different areas.

After that we're also now integrating into Jenkins to watch and query the results from the checks that are in the SaaS application. It's going to say, "Hey, I'm here at a final gate. There's going to be a deployment of this application." Jenkins then pulls the application details out of the Contrast Team-Server and says, "Hey, do you have any bugs that are critical or high, for example, or is it okay to go to deployment?" And it either comes back and says, "Yeah, it's good," or, "No, it's not good. It has critical or highs," or whatever the thresholds are that we want to configure. Jenkins can break or stop the build if the threshold standards are not met.

Q When you say compatible, is this mostly for Java environments, or what would be not compatible? Legacy apps or different languages?

A Contrast was initially built, I believe, for Java. So, it works very well with Java. I will let Contrast speak to what languages they support.

Q You have mostly talked about the Assess side of Contrast products. They also have the Protect side. Are you using Protect?

A We are using Protect. We found it very effective. The Runtime Application Self Protection (RASP) functionality provides additional protection for the application in a production environment. If somebody were to be trying to exploit code, the dangerous code path is blocked. This is another example of how Contrast's technology is a little more advanced and more next generation, using RASP functionality. It seemed to have fewer false positives and unwanted application interruptions then a normal Web Application Firewall (WAF). It is a little smarter, and it says, "Only if somebody is actually exploiting this vulnerability, doing something malicious, sending in data that actually is malicious," will Contrast actually block.

Q If you're going to turn Protect blocking on, is there some sort of QA testing you have to detect any false positive blocking? How much tuning is required?

A There is a level of QA testing and tuning, but I have found it's not nearly as difficult to tune, as for example, other web application firewalls. Sometimes those rules tend to require a little bit more understanding of the tools UI and scripting for tuning and it can be a little bit more difficult to analyze for potential impacts. Contrast tuning requirements are quite easy to spot and it is easy to use the tool to put in rules if there is something that needs to be tuned.

You can tune quite readily with the tools that they give you within the Contrast portal. And so, tuning is pretty quick. Oftentimes, if someone's paying attention to their app and they're watching it and working with it in Team-Server portal, it's actually a quick turnaround to be able to turn something from monitor and alert mode to block mode.

We are using Protect. We found it very effective. The Runtime Application Self Protection (RASP) functionality provides additional protection for the application in a production environment.

Q How long have you been using Contrast operationally?

A We did a POC (Proof of Concept) before the end of last year and started to roll out operationally the first few months of this year. So, about ten months now.

Q Are there any lessons learned you can pass on? Something you'd do differently just getting started, based on what you know now?

A Our enterprise is large with many varying departments and groups with development teams and this causes some complexity in managing that much data. I'd probably add some more formalization to the onboarding process to allow us to use naming conventions, tags etc to assist in reporting. There's two different methods for getting started. One is onboarding it with configurations preset and some of those being the application name and then the application's grouping, which allows different users to have different visibility. Because we're a larger enterprise, we've got a lot of rules that allow certain people to see certain apps and others to not see others. We try to segregate based on enterprise market, country, application teams, things like that.

If somebody were just to onboard the agent without putting in the right configurations or application name, they potentially can't see it because they don't have the groups that allow them to see the application. They would ask an admin to help add the needed permissions. But, if I wanted to look at it across all eighteen different countries, I might just see thousands of applications, and it doesn't have the right context because I don't know how those applications are related.

So, going back, I would probably have a little bit more of a plan to make sure that the right names and groups are being applied up front so that there's no changes later to go back and rename or regroup things if they weren't done initially, maybe even to the extent of putting in a control or required fields.

Q You mentioned next generation technology, a fairly complex product. How have you found the support from Contrast?

A It's really good. They've got some very smart people in their support group. We work with a lot of different vendors, a lot of different tools. Contrast's support team is very knowledgeable. They understand the tool. They understand the product. They get to very detailed inner workings of how Contrast works and how our applications work so they can tailor solutions right to what we might need if we have questions.