Radware's Uri Dorot on Blocking Attacks on Applications Through Their Third-Party Services

radware

iSMG
INFORMATION SECURITY
MEDIA GROUP

URI DOROT

Dorot is in charge of positioning, messaging and sales enablement of Radware's application protection solutions and services. He joined Radware in 2021 and has years of product marketing experience from leading companies in the cyber domain.

In this video interview with Information Security Media Group, **Uri Dorot** says that we should not simply be monitoring supply chain attacks on third-party services and applications, but also taking proactive steps to prevent the attacks.

In this video interview, we will explain:

- Why these attacks have become more of a problem today and exactly what the threats are;
- How formjacking works, with current examples;
- Protective measures, including the benefits of consolidating your application protection solutions.

Radware will also be at RSA Conference on April 24-27 in San Francisco. You can click here if you would like to book a meeting and find out more.

SUPPLY CHAIN ATTACKS TODAY

TONY MORBIN: Supply chain attacks have shot up the board agenda, but many people still don't understand that in today's complex environment where applications are scattered across platforms reliant on different APIs and third-party JavaScript services, they still remain liable to regulators and others in the event of a failure. Can you explain why these attacks have become more of a problem today and exactly what the threats are?

URI DOROT: Before we talk about the threats, it's important to understand the modern application environment or architecture. Just a few years back, applications were simpler to define and protect. They were built in a monolithic structure using three-tier architectures, and they were hosted in local data centers with one main entry point. Most of the content was composed by the web server, and the browsers were just in charge of rendering the content. Protecting such a perimeter just required putting in your ADC and your WAF at the gate of your application, which is quite simple.

Today, modern applications are distributed across many environments – on-premises, virtual clouds, private clouds, and public clouds such as AWS, Google and Azure, and they have many entry points. On top of that, there is complexity when it comes to using microservices like Kubernetes. Applications also rely on a lot of third-party content and services, either in the form of API connections to third-party services or JavaScript-based services with a third-party or fourth-party plug-in integrated as well. And a lot of the content used to be generated by the web server, while much of it is now being generated by and composed by the browser. This brings two new challenges to protecting applications.

The first challenge is that the applications have many entry points today. Let's say an application is a house. Before, you had one door and one window. Now, you have five doors and 30 different windows through which malicious actors can penetrate that house. The second challenge is on the protection side. Big enterprises today do a decent job at protecting their data centers. They all use DDoS protection tools and web application firewalls, so penetrating the data center of a big enterprise is not as easy as it was before. Hackers try to reach for more low-hanging fruit, such as exploiting vulnerabilities on third-party applications or services and JavaScript services that are unmonitored or not as monitored as the main data center.

Those are the two main reason why we see a surge in attacks on applications, and the attacks are done by not necessarily directly attacking the applications but by abusing APIs, JavaScript services and such.

WHY CURRENT DEFENSES FALL SHORT

MORBIN: Can you talk us through a couple of the threats in the expanded threat landscape and explain why the current defenses are not protecting us against client-side attacks, including formjacking, and maybe provide examples?

DOROT: The problem with protecting against supply chain attacks is that the environment is not as monitored. By environment, I mean the data path between the application's end user browsers and the third-party JavaScript services. Let's say 30 types of JavaScript services are used in an average website – things like Google Analytics, Adobe Analytics, social media-embedded content, buttons, video players, advertising iframes,

“Most traditional security tools and web application firewalls are sitting in the data path between the end user and the application, and they’re not monitoring the third-party connections.”

stock management tools, payment services like Tranzila, platforms like WordPress and services like Magento. All of the many tools and services embedded in each application are legitimate providers and companies, but the connection established between an end user and the third-party service within the application is unmonitored. Most traditional security tools and web application firewalls are sitting in the data path between the end user and the application, and they’re not monitoring the third-party connections.

Formjacking is one of the most common supply chain attacks. Here’s what formjacking is: As we said, an application uses many third-party JavaScript services. A malicious actor can exploit a vulnerability in one of those services and hide a malicious malware or inject the malware into that service. Later on, a legitimate user of the application will send a request to the application, which will send that user an HTML form, initiating a call to the relevant third-party service. If that service is infected, it will respond by injecting a malicious script into that form. It will change one of the destination parameters. And the information the end user types into the form will be collected and sent to the attacker’s server. This happens behind the scenes. It is completely seamless to the legitimate end user and to the application because the form is still being sent to wherever it needs to be sent and the user can continue with whatever they wanted to do. There’s no immediate impact on the performance of the application.

The problem is that traditional tools cannot see this because they’re not monitoring that data path. And organizations are still liable for protecting their users’ personal data. The regulators don’t really care whether the application was breached directly or the data was stolen through a third-party service. The most famous example is the British Airways incident a few years ago where over 400,000 records were stolen and the regulator hit British Airways with a fine of 20 million pounds. They’re still battling ongoing litigation in the hundreds of millions of pounds. To this day, it’s not completely solved.

There have been some other cases. Last year, there was a big breach, a formjacking attack, on the Segway store, where hundreds of thousands of records were stolen. There was an interesting incident with Tupperware, where they were using an outdated version of Magento in which a certain vulnerability wasn’t patched and also they suffered a formjacking attack where many credentials were stolen. When we talk about the rise in supply chain attacks and the risk involved with them, it’s mainly formjacking. Some people call it skimming or e-skimming. It also known as Magecart attacks.

THE RADWARE APPROACH

MORBIN: How are we protecting against these attacks, including the limitations? And also, can you explain how Radware helps its clients improve their defenses? How does the implementation work in practice on different platforms? What are our defenses?

DOROT: Many organizations that we speak to are not aware of the magnitude and the risk of these problems. They blindly trust the big-name third-party service providers because they are legitimate companies. They trust their security. When these attacks happen, they don't take an immediate toll and they don't impact your application performance. It's not like a DDoS attack that brings your website down. If there's a data breach, you might only find out about it weeks or months later.

Some organizations use tools that are based on bots and crawlers that continuously crawl their applications, looking at all the scripts, requests and responses coming from all of the third-party services. They check for abnormalities or malicious scripts, and they do a good job in reporting and alerting, but they miss the protection part of things, the enforcement. In order to have full protection, you need to find a way to incorporate those tools with another protection solution that can actually do something with that information and then decide whether you want to block certain providers or scripts. But when you have two different tools responsible for protecting you against something, from an enterprise point of view, it's difficult to know who to approach in case something doesn't go right.

Let's say there's a data breach. Who was responsible? Who's going to help you? Who's liable for that – the tool that was in charge of detecting it, the tool doing monitoring or the tool that was supposed to enforce protection? When you get into integrations like that, there are always latency issues. There are a few solutions that try to provide protection and also do detection and enforcement. Our cloud application protection service, which is in a one-stop shop for all sorts of things to protect your app, is not just about client-side protection. It's a cloud platform through which we provide our WAF, bot management solution, application-layer DDoS and API protection tools. They're all under one portal. We call it 360-degree protection.

One pillar of cloud services is client-side protection. When an end user enters the application and receives a response from the app, it goes through our cloud protection services. We send back a detector to the browser side that, from that moment on, monitors all the requests initiated from that user for the protected app. It doesn't look at anything else on the browser. It monitors the data path between that specific end user and the third-party service. We do that by using various proprietary machine learning-based algorithms that create a baseline or allow list of legitimate requests and parameters in the requests going out to the third-party services. That prevents any data leakage to a legitimate service with an illegitimate parameter that would send the information to an unknown IP.

Our solution can map your entire third-party services ecosystem. It can also show you your third party's third parties – your fourth and fifth parties. We provide alerts if we see anomalies, and you can choose whether you want to block a service or not. What makes our solution unique is that we are capable of providing surgical enforcement. We understand that many of these services are vital

“We understand that many of these services are vital to your business continuity and to the operational aspects of your application – you cannot take them down altogether. So, we block only those nefarious scripts that are anomalous, incorrect or at a high risk level.”

to your business continuity and to the operational aspects of your application – you cannot take them down altogether. So, we block only those nefarious scripts that are anomalous, incorrect or at a high risk level.

It’s part of our entire application protection suite, so it works in sync with our WAF and our bot protection. When you look at reports and visibility, you see the full picture. We have tools for protecting your APIs connections, data center and web servers and for protecting against bots. So you get an understanding of your entire security – your application security or protection posture and all of the connections – and you see the entire perimeter right in front of you on the screen. It’s our answer to how applications are scattered all over the place. Client-side protection is super important.

KEY TAKEAWAYS

MORBIN: What are the key things that viewers should take away from today’s discussion?

DOROT: First, make sure that you are familiar with your application ecosystem and what type of services and solutions are embedded in your applications. Many times, security personnel are not aware of everything within the application because they are not the ones developing it. Make sure that you have visibility. You cannot protect something that you don’t see. And look for the right tools to do that.

Second, even if you have API connections with or JavaScript third-party services of the most robust or big enterprises or services out there, don’t trust them blindly. Don’t trust your security with them. The regulator doesn’t care. It’s your responsibility. You are in charge of your application and your end users’ data.

About ISMG

Information Security Media Group (ISMG) is the world's largest media organization devoted solely to information security and risk management. Each of our 28 media properties provides education, research and news that is specifically tailored to key vertical sectors including banking, healthcare and the public sector; geographies from North America to Southeast Asia; and topics such as data breach prevention, cyber risk assessment and fraud. Our annual global summit series connects senior security professionals with industry thought leaders to find actionable solutions for pressing cybersecurity challenges.

(800) 944-0401 • sales@ismg.io

BANK INFO SECURITY® CU INFO SECURITY® (Just for Credit Unions) GOV INFO SECURITY® HEALTHCARE INFO SECURITY®

infoRisk TODAY CAREERS INFO SECURITY® Data Breach TODAY (Prevention. Response. Notification.) CyberEd.io

ISMG
INFORMATION SECURITY
MEDIA GROUP

902 Carnegie Center • Princeton, NJ • 08540 • www.ismg.io