Supply Chain Attacks: The Next Generation of Business Risk

For years, ransomware dominated headlines and discussions around the boardroom. While that threat persists, organizations are realizing they’re vulnerable in a new, complex way that extends beyond the perimeter and outside of their traditional security defenses and even pushing the boundaries of endpoint security.

Whether you want to acknowledge it or not, your business relies on a software supply chain for both home grown and third-party applications. It’s an intractable problem that was fostered through the shift to DevOps, the preponderance of open source software in enterprise applications and the need to ‘adopt and go’ quickly using open source code. The rapid timeline of transformation that occurred over the past 12 months has compounded the implications of Nth party risk. While you may have the right security controls in place, it doesn’t mean your vendors across your software supply chain does.

As our industry has repeatedly learned, an organization’s security controls cannot rely on trusting everything from the ecosystem, even from partners. Companies have to think beyond their immediate set of vendors - they must account for their vendor's vendors too.

Security Risks in the Age of DevOps

Modern business applications are more complex than ever -- a trend that will accelerate in the coming months and years as monolithic applications continue to decompose into APIs, microservices and serverless functions. Simultaneously, organizations have also changed the way they run these workloads in production environments. With more ephemeral workloads and distributed architectures, there’s no silver bullet for pre-production software analysis.

Even in the most rigorous Software Development Life Cycle (SDLC), the complexity of development means vulnerabilities will be introduced. Unfortunately, software ecosystems introduce risk that even scanning won’t identify. While security teams rely on static and dynamic application scanning tools to identify vulnerabilities before production, they’re only thwarting a fraction of Common Vulnerabilities or Exposures (CVEs) in software applications and libraries.

Vulnerabilities in the software supply chain are not new. According to Imperva research, more than 150,000 Common Vulnerabilities or Exposures (CVEs) in software applications and libraries have been reported since 2000. The scale of this problem is growing faster each year as more organizations construct their businesses on applications and microservices.

Over the past year, application vulnerability has become a growing target for motivated attackers. Imperva research shows that remote file inclusion (RFI) and remote code execution (RCE) was a leading attack vector with more than 30 million total recorded incidents during 2020. Further, cross-site scripting (XSS) attacks totaled more than 16 million incidents and SQL injections (SQLi) amassed more than 10 million incidents. The incredible volume of attacks points to an untenable situation that will create more business risk in the months and years to come unless organizations begin to address the vulnerabilities that lurk in their lines of code.

Embedding Security Within the Lines of Code
To help address the growing scale of automated attacks within the software development lifecycle, organizations need to adopt a threat model that includes all parts of the supply chain, including Nth-party code. In this model, organizations have to think differently about protection. Application protection must identify run-time application behavior, such as whether third-party code is responsible for unwanted traffic. Only by blocking unexpected behaviors will they ensure prevention of novel attack behavior, such as establishing command and control (C2) from internal applications to a remote server or the fraudulent syphoning of payment card and personal information.

Runtime Application Self-Protection (RASP) is one way for companies to provide security from within and to protect the application with more in-depth context. From attacks to injections or even weaknesses, RASP is capable of detecting and preventing attacks in real-time. Since RASP integrates security into DevOps via continuous integration, continuous delivery (CI/CD) processes, it seamlessly becomes part of an organization’s SSDLC. The technology can pinpoint attacks down to an exact line of code and automatically stop exploitation of a vulnerability, which gives organizations the time needed to patch vulnerabilities on their own schedule.

The NIST organization recognized how many security controls fail to address the challenge of mitigating software supply chain attacks, and determined that only runtime protection prevents these stealthy attacks. NIST SP 800-53 Revision 5, includes Runtime Application Self-Protection (RASP) is a recommended control [SI-7(17)] to respond to emerging threats from the software supply chain.

For more sophisticated supply chain attacks aimed at establishing a foothold and moving laterally across a network, organizations need a fast and easy way to mitigate risks. Often in these cases, a compromise is rooted in a third-party JavaScript connection. To combat it, security teams need control over the behavior of any JavaScript code embedded into web applications. Client-Side Protection (CSP) prevents online fraud from supply chain attacks like formjacking, digital skimming, and Magecart. As no one can manually triage and approve JavaScript services for execution across millions of transactions, it must be automated. CSP helps clearly identify JavaScript vulnerabilities and unwanted behavior, such as compromised code stealing and transmitting your customers’ sensitive data. Once a baseline of expected activity is identified, high-risk and suspicious behavior can be exposed and mitigated.

Supply Chain is a New Domain of Data Security

Every software and application vulnerability is a data security issue. For too long, data security was relegated as the concern of the GRC or compliance team. Today, as businesses are powered by microservices and APIs, security teams must be involved in developing a strategy that prioritizes the protection of the data itself.

Attackers are motivated to access sensitive information, regardless of its composition and structure. Therefore, organizations need to understand where data lives, if the data is classified, if the right access controls are in place, and ensure that strong tools for auditing and anomaly detection are in place. Knowledge of the supply chain is a key component of this process. Only from there will security teams know where their data is at all times across all environments, how it is used, and who has access to it in order to apply the appropriate controls.