ONAPSIS

WHITEPAPER

5 Reasons Why You Need **Application Security Testing** for Business-Critical Applications

Business-critical applications are at the center of every enterprise organization. These applications manage increasing complexity throughout the enterprise to drive core functionality for the business — from supporting manufacturing and supply chain management to human resources and sales. Complexity requires customization, and enterprise organizations employ large software teams to write custom code to support and streamline critical business processes. Traditionally, to protect these business-critical applications, enterprise organizations often employ a "defense-in-depth" security model (i.e., applying layers of technology to protect critical systems). But unfortunately, not enough consideration is given to the security of the custom code development for the applications themselves.

According to Forrester Research, applications have become a key attack vector for enterprises. More than one-third of external attacks are due to a software exploit.[1] Successfully exploiting a vulnerability at the code level of these systems can allow an attacker to execute a wide range of malicious activities. For example, threat actors can impact supply chains and manufacturing processes, redirect financial payments, or compromise highly sensitive data which can result in compliance regulation issues as well as data loss.

43% OF ORGANIZATIONS ARE EMPHASIZING SECURITY IN THE DEVELOPMENT OF NEW APPLICATIONS.[2]

Enterprises are beginning to react to this challenge. Forty-three percent of Ponemon Institute survey respondents noted that they emphasize security in the development of new applications.[3] This is a step in the right direction, but it means that the majority of enterprises are still not building applications with security in mind. This is particularly concerning for business-critical applications since they are a high-value target for threat actors. The need for a solution tailored to protect these business-critical systems is more urgent than ever before.

Here are five reasons why you need application security testing (AST) capabilities specifically designed for your most business-critical systems.

[1]

Balance Security With the Speed of Development for Digital Transformations

Digital transformation projects were being gradually implemented over the last decade. However, beginning in 2020 with the onset of the COVID-19 pandemic, these projects were accelerated to warp speed. Greater demands to implement more digital interactions for customers, employees, vendors, and partners have arisen in every industry. According to a global survey of executives, companies accelerated the digitization of their customer and supply-chain interactions as well as their internal operations by three to four years.[4]

This shift favored expediency over security and, consequently, has left business-critical applications at greater risk. Worse still, these applications have migrated from an on-premises model to cloud and third-party-hosted models, leaving them more exposed. Additionally, as software development cycles have shortened to accommodate this acceleration, security has become more of a cursory checkbox or an afterthought. This can result in a never-ending cycle, where security can't quite catch up to the pace of application development for the business.

[1] Forrester Research, The State Of Application Security, 2021: Applications Remain A Key Attack Vector, But Signs Of Hope Emerge by Sandy Carielli with Amy DeMartine, Melissa Bongarzone, and Diane Lynch; March 23, 2021
[2,3] Ponemon Institute, Reducing Enterprise Application Security Risks: More Work Needs to Be Done; February 2021
[4] McKinsey, Digital and Strategy & Corporate Finance Practices: How COVID-19 Has Pushed Companies Over the Technology Tipping Point — and Transformed Business Forever; October 2020

36% OF IT AND SECURITY PRACTITIONERS SAY THERE IS NO COLLABORATION BETWEEN DEVELOPMENT AND SECURITY TEAMS.[5]

Yet trying to insert security into the development of business-critical applications poses challenges due to manual review processes and a lack of automation. AST tools for business-critical applications such as SAP require a large number of unique capabilities, including support for specific code, languages, frameworks, and transports. SAP development also requires integration with specific change management systems such as SAP ChaRM, which allows you to track change requests and transport requests within SAP. Given these realities, it's no wonder that security testing is often bypassed completely. According to a Ponemon study, 36% of those surveyed report there is no collaboration between development and security teams, and 29% say there is limited collaboration.[6]

[2]

Eliminate Blind Spots When Working With Contractors and Third-Party Developers

Hiring IT staff, especially application developers who have experience with business-critical platforms like SAP, can be challenging. According to United States labor statistics, by the end of 2020, the global talent shortage amounted to 40 million skilled workers worldwide — and this shortage is expected to persist.[7] Enterprises continue to hire outsourced consultants, contractors, and system integrators to fill the gap. According to a Harvey Nash/KPMG CIO survey, 41% of organizations have plans to increase their spending on software outsourcing.[8]

However, bringing on additional resources to meet project deadlines for development is not without its challenges. Organizations must validate the work of these third parties. In-house application development leaders need visibility and automation capabilities for third-party (and internal) code and transports so they can ensure that corporate standards are met, code is high quality and secure, and security checks aren't interfering with their team's ability to meet project timelines.

41% OF ORGANIZATIONS HAVE PLANS TO INCREASE THEIR SPENDING ON SOFTWARE OUTSOURCING.[9]

Employing outside teams further increases the time and cost of the code-testing process. Few AST tools focus specifically on code for business-critical systems (e.g., ABAP for SAP), so the alternative is frequently a manual testing process that is both highly prone to error and labor-intensive. This adds cycles and drives up spend, especially concerning when you're paying outsourcing rates. Compressed project timelines and budget constraints have led many organizations to either overlook security in their development process, risking exposure or rely on costly, ineffective manual efforts.

A recent study by Forrester noted that security professionals are attempting to implement AST tools. However, only 31% are doing so in testing.[10] Clearly, implementing security earlier into the development cycle is common sense, but this is far from what's observed in practice.

[5-6] Ponemon Institute, Reducing Enterprise Application Security Risks: More Work Needs to Be Done; February 2021
[7] Forbes, Is There A Developer Shortage? Yes, But The Problem Is More Complicated Than It Looks; June 8, 2021
[8-9] Forbes, Analyzing the Software Engineer Shortage; April 13, 2021
[10] Forrester Research, The State Of Application Security, 2021: Applications Remain A Key Attack Vector, But Signs Of Hope Emerge by Sandy Carielli with Amy DeMartine, Melissa Bongarzone, and Diane Lynch; March 23, 2021

[3]

Deep Clean Your House and Keep It Clean

Just like deep cleaning your house and keeping it clean require constant effort, such is the state of cleaning up your custom code and creating a security baseline to keep it clean. Clean code is code that is easy to understand and follows secure coding best practices to minimize the risk of vulnerabilities. It is critical to scan existing custom-built code to identify and fix vulnerabilities as well as validate the quality. Creating clean code with the right processes and security can be challenging when faced with the overwhelming demand to build new functionality quickly and get it to production as fast as possible.

THE AVERAGE CUSTOMER'S SAP SYSTEM CONTAINS TWO MILLION LINES OF CUSTOM CODE.[11]

According to Onapsis research, the average customer's SAP system contains two million lines of custom code.[12] Additionally, our research found a critical security issue as well as a critical performance issue within every thousand lines of code. This translates to roughly 4,000 critical issues per system potentially introducing risk and business disruption. That's a lot to screen for manually in quality assurance (QA) or code reviews. Ideally, organizations should be able to quickly and automatically identify these errors earlier in the development cycle, before they reach production, to secure their business-critical applications. However, due to the volume of code and the fact that most organizations run multiple systems, it is clear why manual reviews aren't practical.

Establishing a security baseline and addressing code vulnerabilities in a timely manner is becoming all the more important due to the sophistication of attacks against business-critical applications. A recent **Onapsis Research Labs** threat report found conclusive evidence that attackers target and exploit unsecured SAP applications using a variety of techniques, tools, and procedures.[13] These attacks are not simply brute-force attempts made directly against the application. Some attacks chain multiple vulnerabilities together — including new ones introduced by custom code — in order to target specific applications for nefarious purposes.

[11-12] SAP® ABAP Code Quality Benchmark E-book
[13] Onapsis Threat Intelligence Report: Active Cyberattacks on Mission-Critical SAP Applications

[4]

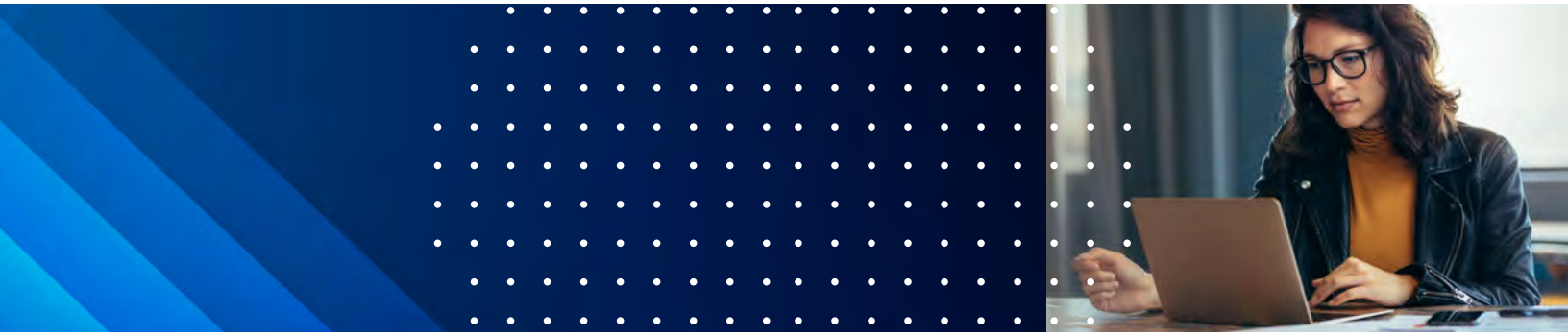
Prevent a Trojan Horse and Protect Against Vulnerable Code Transports

A critical component of custom development for SAP is the ability to transfer data from one SAP system to another. Additionally, data must be transferred from external applications and third-party software to SAP. Data transfers within SAP as well as from external sources to SAP are handled by **transports**. On average, 250 transports with up to 5,000 objects per SAP system are triggered every month.[14] Transports are all the more prevalent in large enterprises that operate several hundred SAP systems. These transports come from the company's development and QA systems as well as third-party applications.

EACH MONTH, 250 TRANSPORTS WITH UP TO 5,000 OBJECTS ARE TRIGGERED PER SAP SYSTEM.[15]

Checking these transports can be challenging for release managers since conventional analysis tools are unable to inspect transport contents. Attempting a manual technical check can be incredibly complex, and it is generally not feasible to run preliminary checks on those transports from third parties, putting release managers at a significant disadvantage. This opens up the possibility that these transports may contain malicious or improperly configured content that could act as a "Trojan Horse" and do harm to the environment once put into production. Even simple changes through SAP transports may pose a high security risk since they can unwittingly introduce malicious content or changes into the environment, providing a gateway for data theft and data manipulation. The damage to an affected company can be considerable, ranging from financial and reputation loss to substantial penalties associated with violations of legal data protection regulations.

Complicating things even further, if an error is found within a transport already deployed to production, remediation is challenging. Transports cannot be rolled back to a prior state, so an entirely new transport must be written and deployed to production to overwrite the harmful content. This further jeopardizes the ability of application development projects to run expeditiously.



[14-15] Battling Trojan Horses in Your SAP® Transports

[5]

Understand Potential Issues Regarding Data and Security Audit Compliance

Business-critical applications are used by an enterprise to run the core of their business and many of these applications, particularly SAP, contain information that is subject to specific government and industry regulations, including financial requirements. Vulnerabilities exploited within custom code and transports can not only lead to unplanned outages, downtime, and reputational risk, but they may also result in regulatory fines and legal action. This may include data privacy laws such as General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA), as well as financial-reporting laws like Sarbanes-Oxley, if financial reporting is affected.

Enabling third-party developers and contractors to build custom code and transports for these business-critical applications increases the risk to the enterprise. Outsourced staff are often equipped with extensive authorizations in order to complete projects quickly. In light of all we've discussed, it would be very easy for a malicious insider (e.g., contractor) to hide harmful content or requests within the countless lines of code or settings and tables of a custom transport that would go potentially undetected in the SAP production system. Even if strict authorizations for non-employees are in place, there is still the risk that a contractor could bypass limitations and force changes by creating and executing commands that maliciously manipulate code or transports.

It's challenging to gain the required transparency and visibility into systems as well as oversight of both internal and external developers from many existing AST tools. It is also challenging to prove whether or not these systems are meeting compliance standards. Furthermore, the lack of reporting on the state of custom code and transports makes it difficult to prove that the SAP systems contain compliant code that can pass security audits. Even if manual testing surfaces vulnerabilities, it's not an easy task to effectively remediate these issues quickly and completely.

There Is a Better Way to Perform Application Security Testing for Your Business-Critical Applications

Introducing **Onapsis Control for Code** and **Onapsis Control for Transports**. Onapsis Control products enable application security testing, including automated code analysis and transport inspection specifically for SAP environments. Onapsis Control products provide automated assessments, integrations with development environments and change management systems, and step-by-step remediation instructions so application teams can identify and fix issues as quickly as possible. Organizations gain automation and prioritization capabilities so they can reduce investigation and remediation times, accelerate development efforts, and meet project timelines. Onapsis empowers teams to "shift left" and implement security earlier into their development process, preventing negative impacts on system security, compliance, performance, or availability.

Onapsis Control products are a core part of the Onapsis Platform, a comprehensive suite of security tools focused on business-critical applications. The Onapsis Platform is powered by the threat intelligence, research, and insights of the Onapsis Research Labs — the team responsible for the discovery and mitigation of more than 800 zero-day vulnerabilities in business-critical applications.

*Visit the **Onapsis homepage** to learn more about how **Onapsis Control for Code** and **Onapsis Control for Transports** can play an integral part enabling application security testing for your business-critical applications.*

ONAPSIS

Onapsis protects the mission-critical applications that run the global economy, from the core to the cloud. The Onapsis Platform uniquely delivers actionable insight, secure change, automated governance and continuous monitoring for critical systems—ERP, CRM, PLM, HCM, SCM and BI applications—from leading vendors such as SAP, Oracle, Salesforce and others, while keeping them protected and compliant. For more information, visit www.onapsis.com. Published 11/21.