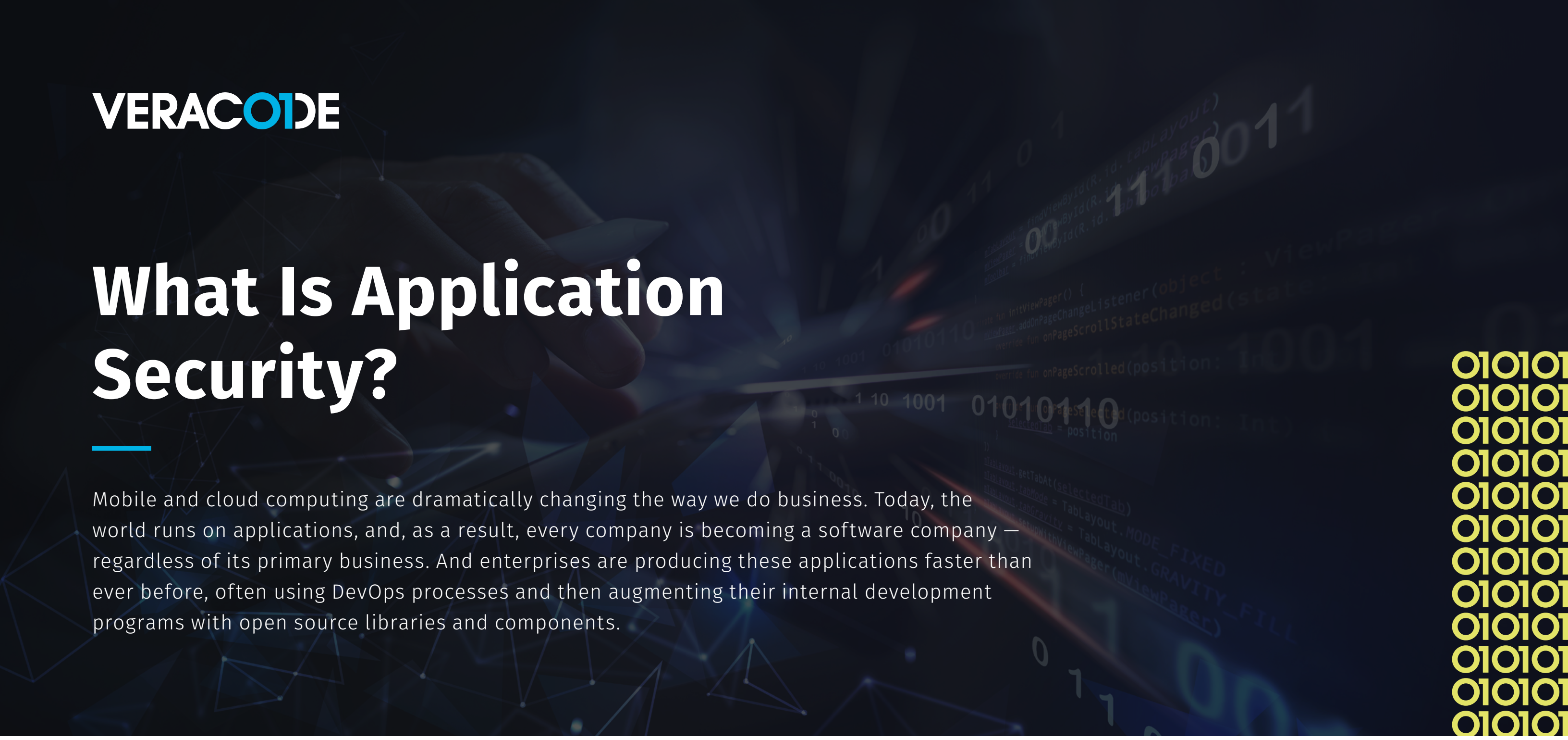

What Is Application Security?

Mobile and cloud computing are dramatically changing the way we do business. Today, the world runs on applications, and, as a result, every company is becoming a software company — regardless of its primary business. And enterprises are producing these applications faster than ever before, often using DevOps processes and then augmenting their internal development programs with open source libraries and components.

Every enterprise's application landscape is now both business-critical and rapidly expanding.

Defining Application Security

Gartner describes application security with the analogy of a crown jewel in a treasure chest; the sensitive information is the crown jewel, and the applications are the treasure chest. To get at the jewels, attackers need to attack vulnerabilities in the application "treasure chest" — meaning it's imperative that enterprises keep that application "treasure chest" secure.

How can organizations ensure the security of their critical and fluid application landscapes? That's where application security comes in. Application security, or "AppSec," is what an organization does to protect its critical data from external threats by ensuring the security of all of the software used to run the business, whether built internally, bought, or downloaded. Application security helps identify, fix, and prevent security vulnerabilities in any kind of software application.

A software "vulnerability" is an unintended flaw or weakness in the software that leads it to process critical data in an insecure way. By exploiting these "holes" in applications, cybercriminals can gain entry into an organization's systems and steal confidential data. Common software vulnerabilities include SQL injection, CRLF injection and Cross-Site Scripting (XSS).

And almost every application has vulnerabilities. Veracode's State of Software Security Report revealed that about ***76 percent*** of all applications had some sort of security flaw. Retail businesses, government agencies, financial institutions, etc. … all are vulnerable.

> ***"According to Verizon's 2020 Data Breach Investigations Report (DBIR), web applications were part of 43 percent of breaches, more than double the amount from last year. SQL injection vulnerabilities and PHP injection vulnerabilities were the most commonly exploited."***

For More Info

About the application security landscape:

Gartner 2020 Magic Quadrant for Application Security Testing

On what a mature application security program looks like:

Everything You Need to Know About Maturing Your Application Security Program

Application Security Solutions

Although many companies first turn to tools and technologies to address application security, a successful solution will always start with a strong strategy. At a high level, the strategy should address, and continuously improve, these basic steps: identification of vulnerabilities, assessment of risk, fixing flaws, learning from mistakes, and better managing future development processes.

There are a variety of application security technologies available to help with this endeavor, but no one is a silver bullet. You need to gather the strengths of multiple analysis techniques along the entire application lifetime to drive down application risk.

The end goal for any organization should be a mature, robust application security program that:

- Assesses every application, whether built in-house, purchased or compiled
- Enables developers to find and fix vulnerabilities while they are coding
- Takes advantage of automation and cloud-based services to more easily incorporate security into the development process and scale the program

Technologies Available to Assess Applications for Security Vulnerabilities Include the Following:

Static Analysis (SAST), or “white-box” testing, analyzes applications without executing them.

Dynamic Analysis (DAST), or “black-box” testing, identifies vulnerabilities in running web applications.

Software Composition Analysis (SCA) analyzes open source and third-party components.

Interactive Analysis (IAST) analyzes code for security vulnerabilities in running web applications by an automated test, human tester, or any activity “interacting” with the application functionality. It works inside the application, unlike DAST, and only tests whatever is exercised by the functional test.

Manual Penetration Testing, or “pen testing,” uses the same methodology cybercriminals use to exploit application weaknesses.

Get more info on the different application security testing methods in *Your Guide to Application Security Solutions*

Download Guide

> *“Businesses aren’t asking for SAST, SCA, DAST — they’re asking, how do I solve my problem? And the right answer is: with a little bit of everything, depending on your environment.”*
>
> **Chris Wysopal**
> *Veracode Co-founder, CTO & CISO*

Common Application Vulnerabilities

SQL Injection exploits an application vulnerability that allows an attacker to submit a database SQL command, exposing the back-end database, where the attacker can create, read, update, alter or delete data.

Cross-Site Scripting (XSS) is an attack that occurs when “malicious scripts are injected into otherwise benign and trusted websites” (according to OWASP). XSS stems from the security weaknesses of client-side scripting languages, such as HTML and JavaScript.

CRLF Injection includes any vulnerability that enables any kind of Carriage Return Line Feed (CRLF) injection attack. It encompasses flaws involving improper output neutralization for logs and improper neutralization of CRLF in HTTP headers.

Veracode is the leading AppSec partner for creating secure software, reducing the risk of security breach and increasing security and development teams’ productivity. As a result, companies using Veracode can move their business, and the world, forward. With its combination of automation, integrations, process, and speed, Veracode helps companies get accurate and reliable results to focus their efforts on fixing, not just finding, potential vulnerabilities.

Learn more at www.veracode.com, on the Veracode blog and on Twitter.