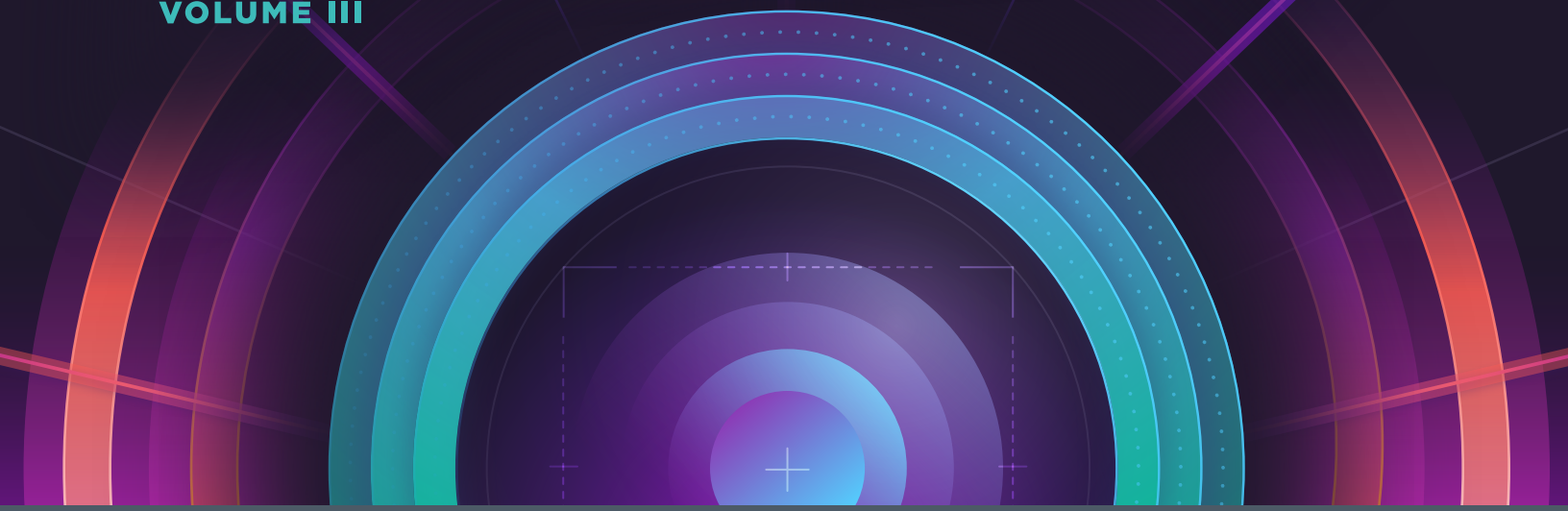
DZone

THE DZONE GUIDE TO

Proactive Security

Apps, Environments, and Messaging

VOLUME III



BROUGHT TO YOU IN PARTNERSHIP WITH

CLOUD coreo

NowSecure™

SECURITY INNOVATION

SYNOPSYS®

zscaler

DEAR READER,

When we last looked at the state of application security in 2016, we wondered what sorts of challenges 2017 would bring with it. With the year nearly over, it's clear that for all the emphasis on changing quickly, the software industry (and if you recall, every company is a software company now) needs to adapt faster to new threats and vulnerabilities from both rogue attackers and national governments.

In April, we learned that the National Security Agency (NSA) had exploited a Windows vulnerability that could allow it to execute code on a target's computer. This was not an isolated incident either, as the NSA has a regular practice of "stockpiling vulnerabilities" that it can later exploit, without alerting the relevant business stakeholders that their software is flawed. Even though Microsoft released a patch for this vulnerability in March, not everyone had upgraded and were hit with a surge of ransomware attacks.

The first of these was WannaCry, which held systems for a ransom in bitcoin. Thankfully, WannaCry had a flaw that essentially functioned as a kill switch to stop the spread of the malware. Despite that, hackers managed to get over $100,000 in bitcoin in ransom payments. In June, NotPetya (named to differentiate it from a wave of attacks in 2016 by Petya malware), held more systems for ransom, mostly focused on the Ukraine, leading several experts to believe it originated solely as a political attack against that nation. Because of this surge in attacks, 40% of DZone readers now consider ransomware to be a top security threat, compared to only 19% who thought so in 2016.

Beset on all sides by rogue actors and some of the best-equipped agencies of our governments, how can the global business community counteract these threats? The answer is to catch vulnerabilities sooner in the SDLC and to patch them before they affect a single user. Shifting security concerns left towards developers, creates an additional layer of checks and can eliminate common vulnerabilities early on through simple checks like validating inputs and effective assignment of permissions.

In this edition of DZone's Guide to Proactive Security, you'll not only learn techniques and principles behind shifting security left, but also how security applies to operations teams in articles like a checklist on Zero Trust Networking. You'll also learn about your enemy by learning how a UPX packer, a common tool for distributing malware, works to obfuscate the true nature of the program it's compressed.

With so many possible avenues for attack, we at DZone feel it's everyone's responsibility to protect their software against those who would harm our businesses and our users. We hope this guide will help educate you against existing threats and safeguard your organization from future ones. Happy reading!

BY MATT WERNER
PUBLICATIONS COORDINATOR, **DZONE**

TABLE OF CONTENTS

3 Executive Summary
BY MATT WERNER

4 Key Research Findings
BY G. RYAN SPAIN

6 A DevOps Approach to Building Security
BY JEFF WILLIAMS

11 Diving Deeper into Security

12 3 iOS App Attack Vectors and How to Strengthen Your Defenses
BY KATIE STRZEMPKA

16 Baking Security Into the Development Lifecycle
BY BOAZ SHUNAMI

18 Infographic: Secure Strategies at Club Code

20 Application Security for Modern Operations Teams
BY JAMES WICKETT

24 Packers: How They Work, Featuring UPX
BY CHRIS LAMB

29 Checklist: How to Go Zero Trust Like Google's BeyondCorp
BY IVAN DWYER

30 Executive Insights on Proactive Security
BY TOM SMITH

34 Solutions Directory

42 Glossary

PRODUCTION
Chris Smith
DIRECTOR OF PRODUCTION

Andre Powell
SR. PRODUCTION COORDINATOR

G. Ryan Spain
PRODUCTION COORDINATOR

Ashley Slate
DESIGN DIRECTOR

Billy Davis
PRODUCTION ASSISSTANT

MARKETING
Kellet Atkinson
DIRECTOR OF MARKETING

Lauren Curatola
MARKETING SPECIALIST

Kristen Pagàn
MARKETING SPECIALIST

Natalie Iannello
MARKETING SPECIALIST

Miranda Casey
MARKETING SPECIALIST

Julian Morris
MARKETING SPECIALIST

BUSINESS
Rick Ross
CEO

Matt Schmidt
PRESIDENT

Jesse Davis
EVP

Gordon Cervenka
COO

SALES
Matt O'Brian
DIRECTOR OF BUSINESS DEV.

Alex Crafts
DIRECTOR OF MAJOR ACCOUNTS

Jim Howard
SR ACCOUNT EXECUTIVE

Jim Dyer
ACCOUNT EXECUTIVE

Andrew Barker
ACCOUNT EXECUTIVE

Brian Anderson
ACCOUNT EXECUTIVE

Chris Brumfield
SALES MANAGER

Ana Jones
ACCOUNT MANAGER

Tom Martin
ACCOUNT MANAGER

EDITORIAL
Caitlin Candelmo
DIRECTOR OF CONTENT AND COMMUNITY

Matt Werner
PUBLICATIONS COORDINATOR

Michael Tharrington
CONTENT AND COMMUNITY MANAGER

Kara Phelps
CONTENT AND COMMUNITY MANAGER

Mike Gates
SR. CONTENT COORDINATOR

Sarah Davis
CONTENT COORDINATOR

Tom Smith
RESEARCH ANALYST

Jordan Baker
CONTENT COORDINATOR

Anne Marie Glen
CONTENT COORDINATOR

Want your solution to be featured in coming guides?
Please contact research@dzone.com for submission information.

Like to contribute content to coming guides?
Please contact research@dzone.com for consideration.

Interested in becoming a dzone research partner?
Please contact sales@dzone.com for information.

Special thanks to our topic experts, Zone Leaders, trusted DZone Most Valuable Bloggers, and dedicated users for all their help and feedback in making this guide a great success.

Executive Summary

BY MATT WERNER
PUBLICATIONS COORDINATOR, DZONE

Application security is still a major issue among software developers and their users. A single breach caused by one overlooked issue, such as the Equifax attack in September 2017, can impact millions of customers around the world. With the rise of high-profile ransomware and DDoS attacks, we've heard that more and more developers are realizing the importance of developing for security and making sure their apps are secure, but where are their priorities, and what are they doing to combat the growing list of threats? To answer these questions, DZone enlisted the help of several topic experts to share their thoughts and experiences, and surveyed over 540 audience members to learn what they're doing in their organizations.

HOLDING ATTENTION FOR RANSOM

DATA The top security threats in 2017 are phishing (49%), SQL injection (48%), DDoS attacks (43%), and ransomware (40%). Concern about ransomware increased drastically, with only 19% or respondents concerned in 2016.

IMPLICATIONS High-profile attacks such as the exploits of MongoDB in early 2017, in which over 30,000 databases were held for ransom and deleted, have drastically increased developer concerns about ransomware. Additionally, phishing is now the greatest concern, only barely taking the top spot from SQL injection.

RECOMMENDATIONS With regards to phishing attacks, the first step is employee education. Learning to spot fake domain names, common tactics to mimic professional email templates and language, and how to deal with these emails is paramount to not only protecting personal information, but also company information as well. As ransomware has become a prominent threat, ensuring that developers have the ability and knowledge to patch up any vulnerabilities early in the development process will be key to fighting these threats.

CALL TO ACTION

DATA The most impactful reasons to care about security are actual breaches at an organization (48%), customer requirements (43%), regulatory requirements (39%), and user feedback (35%).

IMPLICATIONS As demonstrated in part by the spike in concern about ransomware thanks in part to high-profile attacks on MongoDB, breaches at organizations are most likely to spur IT professionals into action. Customer requirements are also incredibly important, as being able to accommodate security requirements directly impacts the bottom line.

RECOMMENDATIONS While it is a positive development that there are major catalysts that spur developers to work on improving security, in an ideal world it would not take these events to implement better security throughout the organization. Widespread cultural change is necessary to instill the need to prioritize security throughout the SDLC. As 18% of time is spent designing for security in the SDLC, organizations have a lot of work to do.

WAS IT WORTH IT?

DATA The primary types of security testing were penetration tests (26%), code reviews (18%), source code analysis (13%), and vulnerability assessments (11%). 15% of respondents had no formal security testing in place.

IMPLICATIONS The most popular types of security testing, penetration testing and code reviews, are both quick to execute, and therefore don't slow down the development process as much as vulnerability assessments can. The perception that security slows down the time-to-market for an application is also the most likely reason that 15% of respondents have no testing schedule.

RECOMMENDATIONS While penetration testing is the most popular technique, organizations such as OWASP actually recommend against relying solely on penetration testing for application development: "Whilst it certainly has its place in a testing program, we do not believe it should be considered as the primary or only testing technique." Further, OWASP stresses the importance of a balanced approach that includes both manual reviews and penetration testing that covers all phases of the SDLC.

Key Research Findings

BY G. RYAN SPAIN - PRODUCTION COORDINATOR, **DZONE**

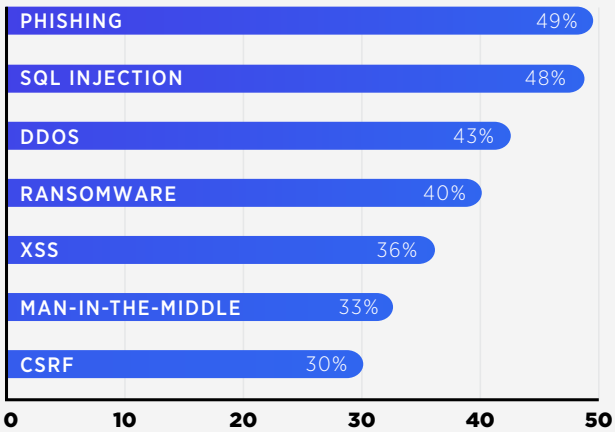
540 software professionals completed DZone's 2017 Application Security survey. Respondent demographics are as follows:

- 38% of respondents identify as developers or engineers, 16% identify as developer team leads, and 18% identify as software architects.
- The average respondent has 14 years of experience as an IT professional. 57% of respondents have 10 years of experience or more; 22% have 20 years or more.
- 33% of respondents work at companies headquartered in Europe; 39% work in companies headquartered in North America.
- 22% of respondents work at organizations with more than 10,000 employees; 20% work at organizations between 1,000 and 10,000 employees; and 26% work at organizations between 100 and 1,000 employees.
- 82% develop web applications or services; 48% develop enterprise business apps; and 30% develop native mobile applications.

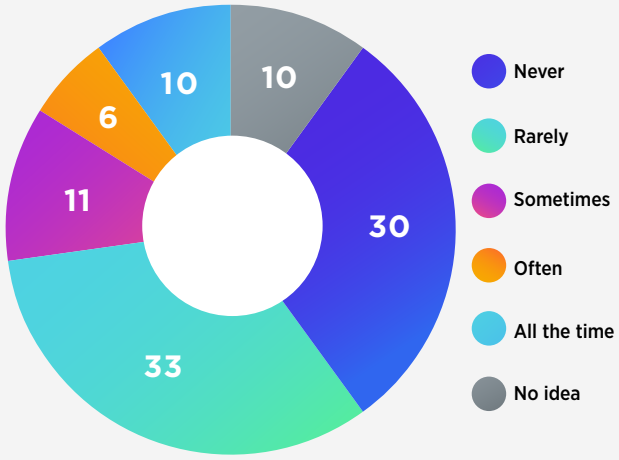
TECHNIQUES, TOOLS, AND PATTERNS

Most respondents use multiple architectural patterns to help enforce security in their applications (69% said they use three or more of the seven types of architectural patterns we asked about). Using roles was the most common pattern, with 75% of respondents using them. Sessions was also a popularly used pattern with 67% of respondents, though this pattern did see a drop from responses we saw in our 2016 Application Security — last year 75% of respondents said they used sessions. As far as secure coding techniques used by respondents are concerned, input validation was the most widely-used technique (83%), followed by architecting and designing secure policies (59%), and making permission explicit and denial default (50%). Almost all of these secure coding techniques saw statistically significant drops from last year's survey, with the exception of executing all processes with the least set of privileges necessary, which had a minor increase in usage year over year (38% to 42%). For verifying message integrity, most respondents favored authentication tokens with digital signatures (70%), with the next most-used technique for this purpose being the use of checksums (40%). The most common encryption API/implementation used was OpenSSL with 66% of respondents; other encryption tools used were Bouncy Castle (22%), crypto for Node.js (16%), and cryptlib (15%).

Which of these threats have you been most concerned about in your organization over the past year?



How often do release schedules override security concerns?



CONCERNS, THREATS, AND VULNERABILITIES

The most concerning security threat respondents had over the last year was phishing (49%), which just barely replaced SQL injection (48%) as the top threat from last year's survey. Other top concerns were DDoS attacks (43%), ransomware (40%) and cross-site scripting (36%). While most responses changed negligibly from last year's survey, concerns about ransomware more than doubled from last year to this year (19% to 40%). The most commonly seen types of vulnerability from the OWASP Top 10 (respondents answered that they have seen these vulnerabilities often or very often) are security misconfiguration (28%), sensitive data exposure (22%), and using components with known vulnerabilities (21%). On average, respondents said that 20% of deployments are made with known security vulnerabilities, and 67% of respondents say they make their customers aware of known vulnerabilities in their application. When asked how often release schedules override security concerns, most respondents answered either rarely (30%) or sometimes (33%), and most respondents thought security analysis and vulnerability fixing had either a low (38%) or medium (43%) impact on releasing on time and on budget. Finally, respondents said the following were highly impactful reasons to care about application security: an actual breach at their organization (48%), customer requirements (43%), regulatory requirements (39%), and users pointing out vulnerabilities in their software (35%).

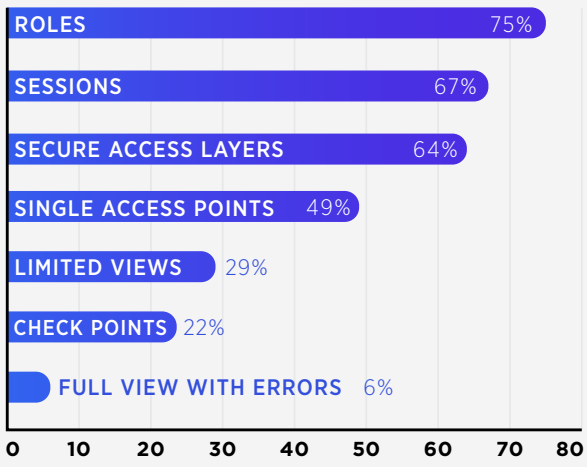
SECURITY TESTING

The main types of application security testing done by respondents are penetration tests (26%), security code review (18%), source code analysis (13%), and vulnerability assessment (11%); however, 15% of respondents said they have no formal security testing. 28% of respondents primarily determine sufficient security testing by acceptably addressing an attack surface, and 27% determine sufficient testing once the threat model has been appropriately handled, but 53% of respondents don't believe that security testing is sufficiently covered. On average, respondents estimate that 20% of their time spent working on application security is spent in the testing phase; this average is second only to implementing security in the development stage of the SDLC.

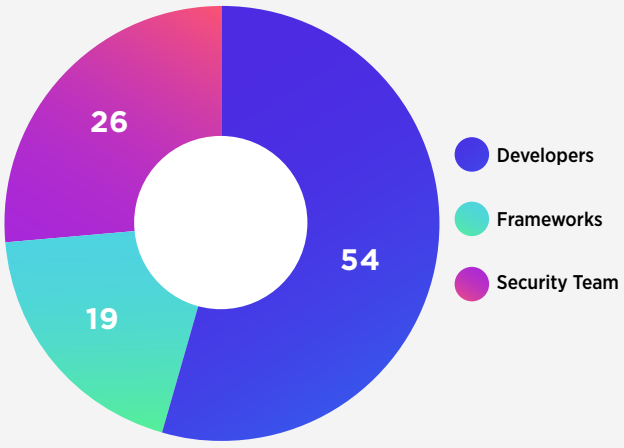
SECURITY TRAINING

The majority of respondents said that security training at their organization either happened on an ad-hoc basis (38%) or never occurred at all (26%). Regular yearly (15%), semi-annually (12%), or quarterly (9%) security training was much less common. Respondents who do have security training in their organization rely mostly on computer-based training (71%) over in-person training (50%). 64% of these respondents said their organization uses custom security training materials, while 48% utilize the OWASP Top 10, and only 16% use the SANS Top 25. Regarding the OWASP Top 10, 10% of respondents said they are very familiar with it, and 29% said they are somewhat familiar; another 39% said they have heard of it but are unfamiliar, and 21% haven't heard of it at all.

What architectural patterns do you use to enable application security?



Who should be primarily responsible for security?



A DevOps Approach to Building Security

QUICK VIEW

01 We can reinvent security by following "The Three Ways" of DevOps.

02 Understanding your "security work" is the first step.

03 Building a culture of experimentation and learning is instrumental to making systems more secure.

BY **JEFF WILLIAMS**
FOUNDER AND CTO, **CONTRAST SECURITY**

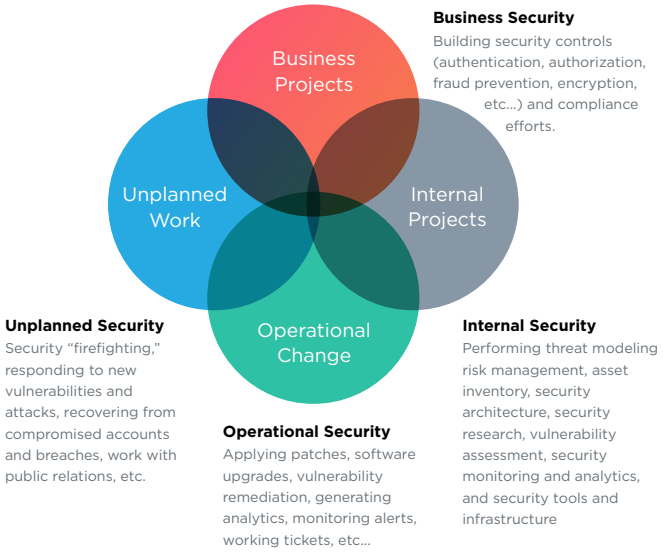
DevOps has been posting some impressive numbers lately, including 46x more frequent deployments, 5x lower change failure rate, 96x faster mean time to restore service, and 2x more likely to exceed business goals. Security has been posting numbers too, but they're terrifying. For example, the average application has 26.7 serious vulnerabilities, and the average breach costs $4 million.

The similarities between the problems security is facing and the challenges that DevOps addresses in software development are striking. Many people understand DevOps as being all about automation, but it's much more fundamental. Fortunately, in "The Phoenix Project," Gene Kim created a roadmap for reinventing security called the "Three Ways."

UNDERSTANDING YOUR "SECURITY WORK"

Most organizations do security work in large chunks, late in the lifecycle, with multiple handoffs involved, and there is no clear line of sight to the business reasons for the work. Not surprisingly, security accrues a lot of technical debt, creates waste, and blocks other processes from working efficiently. While there is a need for more security experts, no amount of heroes is going to fix these fundamental problems.

FOUR TYPES OF "SECURITY WORK"



Sometimes security doesn't seem like tangible "work." People call it a "non-functional requirement" or one of the "engineering specialties," but if we can frame security work in the same way as all other development and operations tasks, as a first-class category of work, we can apply DevOps principles to get it done faster, more accurately, and more reliably. In the diagram above, we can see that security work naturally breaks down into roughly the same categories as software work.

"THE FIRST SECURITY WAY" — ESTABLISH SECURITY WORK FLOW

The First Way is about getting security work to flow from development into IT operations smoothly. How do

we get the work from the diagram above flowing? First, we have to make sure every bit of work has a clear "line of sight" to an outcome we care about.

Getting the work flowing requires structuring it to work differently than we do today. We're going to 1) make the work visible, 2) reduce batch sizes, 3) limit work in process, 4) reduce handoffs, and 5) focus on the constraint. We can rethink all of the security work above using these principles. Here are a few examples of how this approach can transform security work.

	EXAMPLE TRANSFORMATIONS	
	BEFORE	AFTER
Threat Modeling and Security Architecture	Inconsistently practiced ad-hoc paper exercise, little connection to development, not used by downstream activities, infrequently updated.	Updated with every new vulnerability, attack, and internal need. Changes tracked to epics, stories, or tickets.
Vulnerability Assesment	Monolithic pre-production security test generating PDF report. Often relies on tools without tailoring to technology and threat model. Requires human experts.	Continuous verification of presence, correctness, and use of security defenses throughout the lifecycle. Fully automated and integrated with real time notifications.
Intrusion Detection and Response	Extremely limited insight into attacks on application code, mostly based on network traffic and debug logs. Experts required to tune sensors to each application.	Application attack detection and protection are part of every application. Real time attack notification. Centralized control over application portfolio defenses.
Compliance	Annual compliance audits, minimal rigor, disconnected from development, check-the-box approach.	Automatically generate evidence demonstrating compliance continuously as part of the normal development process.
Open Source Security Management	Non-existent or point-in-time scans. Mostly reaction to highly publicized vulnerabilities only.	Continuous threat intelligence monitoring, risk analysis, and software updates. Ability to perform virtual patching to respond to newly discovered vulnerabilities.

With these transformations, security work is now focused on the outcomes that matter to the business. We've made the work explicit, broken it into digestible pieces, and simplified the processes significantly. Instead of poking holes in security and reacting, the work is now generally focused on a "positive" approach to security, focused on ensuring that the right defenses are present, correct, and used properly.

> "While there *is* a need for more security experts, no amount of heroes is going to fix these fundamental problems."

"THE SECOND SECURITY WAY" — ENSURE INSTANT SECURITY FEEDBACK

The Second Way is about establishing tight feedback loops so that the instant a security issue arises or security work goes off track, corrective action can quickly be taken. To achieve this, we need to 1) make these problems instantly visible, 2) swarm on the problem, 3) seek the cause, and 4) optimize for downstream security stakeholders.

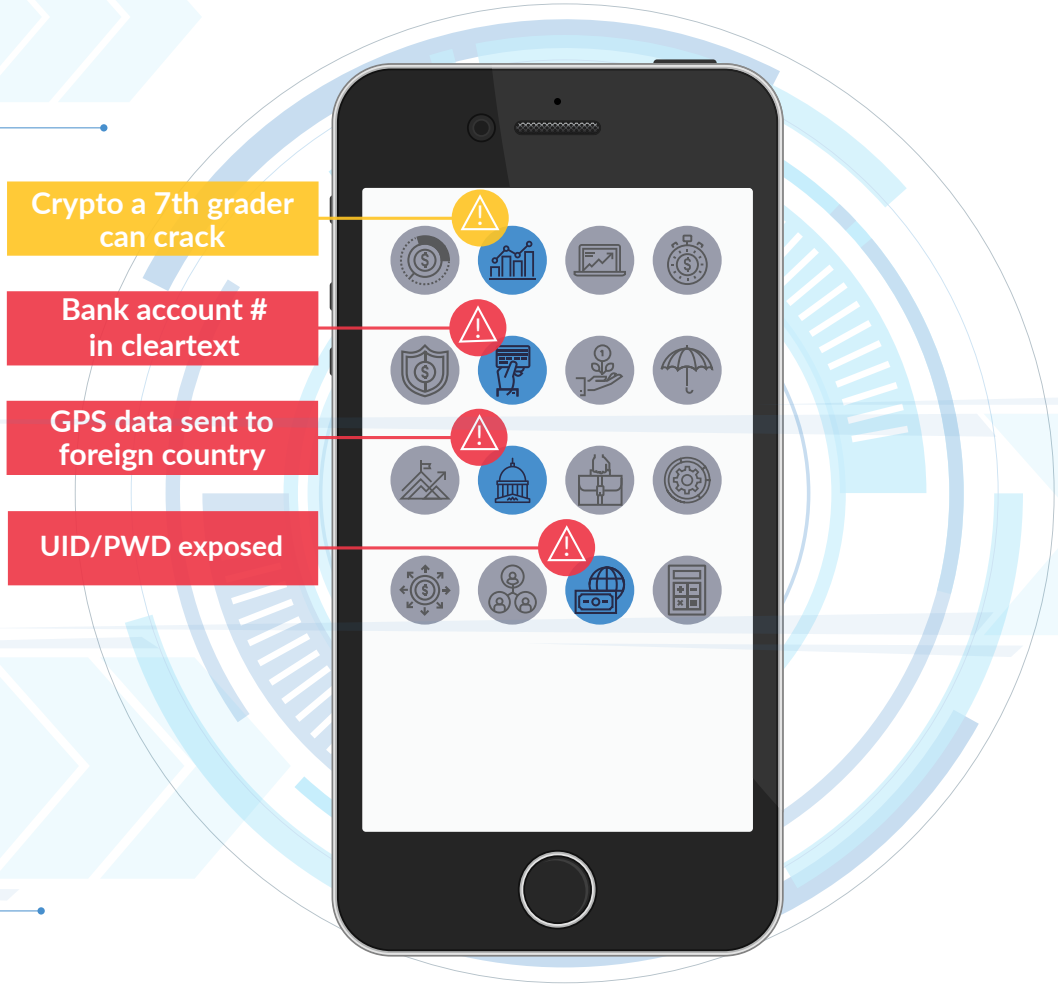
Security assessment offers a perfect opportunity, but don't forget it's just one of many types of security work that isn't as effective today. Simply shoving legacy scanning tools into the CI/CD pipeline is a perfect example of optimizing upstream from the constraint. Because it takes experts to triage false positives, running more scans actually exacerbates the bottleneck.

The only way to improve is to radically optimize for the security expert. Make security visible with a real-time security "bus" that allows threat and vulnerability telemetry to be gathered and analyzed across the lifecycle. The bus also enables *all* stakeholders (developers, testers, operations, audit, executives, etc.) to receive authorized security analytics and notifications through the tools they are already using. In addition, the bus can empower teams to respond to incidents and manage policy across the application portfolio.

NowSecure™

Run Faster with Mobile DevSecOps

Automated Mobile App Security Testing - 8x Faster, 3x Deeper, Most Trusted



Learn how to make mobile DevSecOps a reality:

www.nowsecure.com/go/devsecops

Operationalize Your Mobile Appsec Factory Through Automated Testing and DevSecOps

Traditionally, developers don't want app security to get in the way. Hitting release dates with functional requirements is developers' number one priority, but shipping quality code is a close second. And security is fundamental to quality and success.

Business forces and the rapid pace of mobile app development makes mobile dev teams more likely to embrace DevOps to deliver more features to market at a much faster pace. And while some have experienced appsec as a gatekeeper, the reality is that appsec and development can partner for real speed through DevSecOps.

Only the NowSecure Platform automates 360° coverage of mobile app security testing (MAST) to identify the broadest array of security, compliance, and privacy issues in custom mobile apps — and faster, deeper, and with more accuracy than any alternative. NowSecure Automated cloud software easily integrates with CI/CD/DevOps (e.g., Jenkins) and issue-tracking tools (e.g., JIRA) via open RESTful APIs so as to embrace rather than impede mobile app developers' existing workflows.

NowSecure Automated powers DevSecOps with automated 360° mobile app security testing for DevOps toolchains.

What's more, traditional mobile static source code analysis alone is too slow, prone to false-positives, and doesn't provide enough coverage. NowSecure Automated goes 3X deeper with powerful static binary analysis, dynamic analysis, and behavioral analysis on physical devices — with highly accurate results and recommended resolutions. Only NowSecure delivers post-build, in-line automated assessments in less than 15 minutes (more than 8X faster — each build every day).

To learn how NowSecure can help you make DevSecOps a reality for your mobile app development team, visit www.nowsecure.com/go/devsecops or contact us at info@nowsecure.com.

WRITTEN BY SAM BAKKEN
MOBILE APP SECURITY TESTING EVANGELIST, **NOWSECURE**

PARTNER SPOTLIGHT

NowSecure Platform

NowSecure™

Only the NowSecure Platform delivers automated 360-degree coverage of mobile app security testing (MAST) 8X faster and 3X deeper than any alternative.

CATEGORY
Mobile App Security Testing (MAST)

NEW RELEASES
Multiple releases per month

OPEN SOURCE
No

CASE STUDY

Certifying security, compliance, and privacy in mobile apps is critical to Concur®—the leading provider of expense management. Concur AppSec Manager Rory McEntee needed a rigorous security program for 10 high-risk iOS/Android apps. Third-party testing services lacked transparency, and open source tools were too slow.

McEntee chose NowSecure for enterprise-class penetration testing software and fully automated MAST for the CI pipeline. "Developers bought into NowSecure because of the automation, and the high-quality remediation guidance was key."

McEntee and team scaled their MAST program using the NowSecure Platform to be 8X more productive, ensure customer confidence, and provide consistent reporting and results to developers and executives.

STRENGTHS

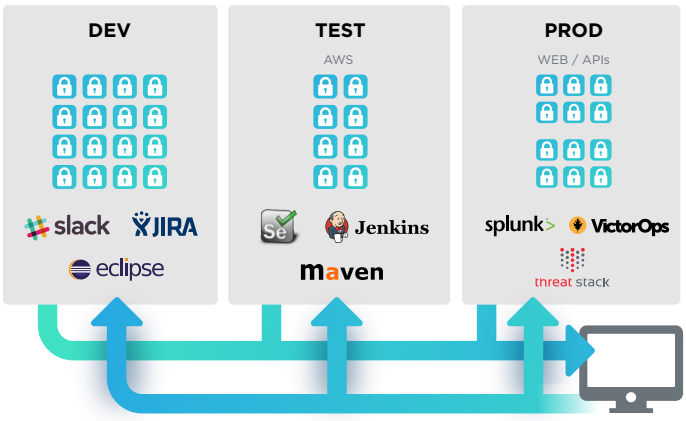
- **8X Faster**: Cut testing time from weeks/days to hours/minutes for 8X productivity
- **3X Deeper**: Only static, dynamic, and behavioral app analysis for both iOS and Android
- **Most Trusted**: By the world's most demanding orgs and most advanced security teams
- **Most Accurate Tests**: Identify latest security, compliance, and privacy issues
- **CI/CD Integration**: Fully automated MAST makes mobile DevSecOps a reality

NOTABLE CUSTOMERS

- Synchrony Financial
- Kaiser Permanente
- FIS
- TD Bank
- Capital One
- Citigroup
- The Home Depot
- Ernst & Young

WEBSITE nowsecure.com | **TWITTER** @NowSecureMobile | **BLOG** nowsecure.com/blog

A Real-Time Security Visibility and Control "Bus"



This bus enables tight feedback loops so that vulnerabilities can be fixed quickly. It also encourages downstream security stakeholders, (like developers) to work with upstream providers (like security testers) to ensure that the work is optimized for them.

Vulnerability assessment is just one of many examples of long feedback cycles in security. Breaking these processes down and reinventing them as small tasks with rapid feedback prevents wasteful rework and saves both time and money.

"THE THIRD SECURITY WAY" — ENCOURAGE A SECURITY CULTURE

Security, even more than most specialties, requires strategy and adaptation. One way of thinking is that security is an emergent property – the result of the constant tension between attackers and defenders. It's an organization's rate of evolution that matters. How fast can you adapt to new threats and vulnerabilities? Unfortunately, most organizations target compliance standards that are several years behind the threat. We *should* be targeting the level of threat that we believe will exist several years in the *future*!

How do we create a culture that constantly advances security with the threat? From *The Phoenix Project*: "The Third Way shows us how to create a culture that simultaneously fosters experimentation, learning from failure, and understanding that repetition and practice are the prerequisites to mastery."

Every organization should have an advanced security team to continuously experiment with techniques to improve security by:

- Looking for ways to enhance the organization's threat model.
- Enhancing and simplifying security architecture.
- Making security defenses easier to use (both by programmers and users).
- Using security to differentiate your product in the market.

This process requires a dedication to science. You'll need to design experiments, measure carefully, and tie your recommendations to the business outcomes you desire. Every time you experience vulnerabilities, attacks, or breaches, take advantage of the opportunity and experiment with ideas to prevent these problems from ever reoccurring.

> "One way of thinking is that security is an emergent property – the result of the constant tension between attackers and defenders."

GET STARTED TODAY

If you're in the process of transforming your business with software, and everyone is, then application security is critical to your future. You need to take it seriously and learn how to actually deliver application security work efficiently – as part of your normal engineering process. If you follow the Three Ways of Security, you'll be able to eliminate a ton of wasted work, redefine what you need to deliver, and deliver results that actually matter to the business. The best part is that the vast majority of application security work can be delivered without security experts and without slowing your software development efforts.

Jeff Williams is the founder and CTO of Contrast Security, revolutionizing application protection with the power of instrumentation. Previously, Jeff was a founder and CEO of Aspect Security and served as Global Chairman of the OWASP Foundation where he created many open-source standards, tools, libraries, and guidelines – including the OWASP Top Ten, WebGoat, ESAPI, XSS CheatSheet, ASVS, and more.

Diving Deeper

INTO SECURITY

TOP #SECURITY TWITTER ACCOUNTS

 @briankrebs

 @WeldPond

 @troyhunt

 @Fisher85M

 @adamcaudill

 @malwareunicorn

 @jeremiahg

 @McGrewSecurity

 @garyjdavis

 @caleb_fenton

SECURITY-RELATED ZONES

Security dzone.com/security

The Security Zone covers topics related to securing applications and the machines that run them. The goal of the Zone is to help prepare the people who make and support those applications stay up to date on industry best practices, remain aware of new and omnipresent threats, and help them to think "security-first."

Performance dzone.com/performance

Scalability and optimization are constant concerns for the developer and operations manager. The Performance Zone focuses on all things performance, covering everything from database optimization to garbage collection, tool and technique comparisons, and tweaks to keep your code as efficient as possible.

DevOps dzone.com/devops

DevOps is a cultural movement, supported by exciting new tools, that is aimed at encouraging close cooperation within cross-disciplinary teams of developers and IT operations. The DevOps Zone is your hot spot for news and resources about Continuous Delivery, Puppet, Chef, Jenkins, and much more.

TOP SECURITY REFCARDZ

Java EE Security Essentials: Specification-Level Security

The Java EE security specification supports a set of required security functionalities including authentication, authorization, data integrity, and transport security. This newly updated Refcard begins by introducing some common terms and concepts related to Java EE security such as identity stores and authentication mechanisms.

Java Application Vulnerabilities: What They Are and How to Fix Them

Java Applications, like any other, are susceptible to gaps in security. This Refcard focuses on the top vulnerabilities that can affect Java applications and how to combat them.

Practical DNS: Managing Domains for Safety, Reliability, and Speed

Introduces DNS and explains how DNS works, DNS configuration, security, and common problems, in order to get you up and running quickly, safely, and reliably.

TOP SECURITY RESOURCES

OWASP Top 10 Project
owasp.org

The OWASP Top 10 is a list of the top ten most critical web application security risks. These are the most important things for security experts to focus on.

Awesome Infosec
github.com/onlurking/awesome-infosec

This GitHub page contains an awesome variety of information security resources, providing you with everything from training material to online courses to books.

The Uber List of Cyber Security Resources
cyberdegrees.org/resources/the-big-list

These cyber security resources can help you learn a lot about attacks and breaches, application security, risk management, and more.

TOP SECURITY PODCASTS

Crypto-Gram Security Podcast
crypto-gram.libsyn.com

This monthly security podcast has been going strong since 2005. Recent podcasts have been about the future of ransomware, measuring vulnerability rediscovery, and separating the paranoid from the hacked.

Risky Business
risky.biz/netcasts/risky-business

This weekly information security podcast features interviews with security aficionados, along with recent security-related news. It's self-described as "a security podcast without the waffle."

Southern Fried Security
southernfriedsecurity.com

All your information security needs are addressed in these podcasts, which address topics ranging from phising to building a security strategy to evaluating security product vendors.

3 iOS App Attack Vectors and How to Strengthen Your Defenses

BY **KATIE STRZEMPKA**
VP OF CUSTOMER SUCCESS AND SERVICES, **NOWSECURE**

QUICK VIEW

01 iOS app security is an ongoing process and effective mobile app security begins in development.

02 Developers can significantly strengthen the security of their iOS apps by preventing insecure data storage, ensuring the security of network communications, and implementing strong authentication.

03 The best way to validate/certify the security of an iOS app is to subject it to regular mobile app security testing as part of the SDLC.

I hate to break it to you, but a set-it-and-forget-it, silver-bullet solution to iOS app security doesn't exist. As I've seen over the past many years, adversaries continuously develop new ways to compromise mobile devices and apps — so developers and security teams also need to work continuously to improve the security of their iOS apps. Effective mobile app defense begins where vulnerabilities and risks originate: in development.

In my experience, iOS apps encounter security gaps in three areas:

1. Insecure data storage on the device.
2. Insecure network communications.
3. Weak authentication.

In this article, I'll explain how developers can avoid or fix these issues. Comprehensive mobile app security extends beyond just these areas, but developers who follow these best practices will significantly strengthen the security of their mobile apps.

INSECURE DATA STORAGE ON THE DEVICE

An iOS app might store data in any number of locations on a device, many times without a developer's knowledge. Common storage locations to watch out for include:

- Device logs
- Private app directories
- RAM
- Keychain

Storing app data insecurely exposes sensitive user information if a device is lost or stolen. Or, in the event of device compromise, malware may harvest data stored in clear text and exfiltrate it to an unauthorized third party.

- Avoid storing any sensitive data in the first place. Unfortunately, that's not always possible. If sensitive data must be stored, encrypt it using a randomly generated master key in addition to a passphrase supplied by the user whenever data is accessed (i.e. a password-based key derivation function). This will make it more difficult for an attacker to recover data even if the master key is extracted from the device.

- Explicitly disable "newCachedResponse" when implementing an NSURLConnection delegate. This will stop an app from caching web data, particularly HTTPS traffic. Many developers don't realize that iOS caches HTTPS requests by default, potentially storing sensitive app data in the "Cache.db" file that resides in the private app directory. The code snippet below illustrates one possible method for disabling this option:

```
NSURLSessionConfiguration *config =
[NSURLSessionConfiguration

      defaultSessionConfiguration];

config.URLCache = nil;
NSURLSession *session = [NSURLSession
sessionWithConfiguration: defaultConfigObject

      delegate:self delegateQueue:[NSOperationQueue
      mainQueue]];
```

- Finally, to validate that the app isn't caching sensitive data, run the app on a test device and **perform testing to ensure that data isn't left behind on the device**.

INSECURE NETWORK COMMUNICATIONS

Developers need to secure an iOS app's network communications or they risk exposing credentials and other sensitive data. A number of vectors put a iOS app's network traffic at risk, including man-in-the-middle attacks leveraging insecure Wi-Fi, a compromised wireless carrier network, or malware on the device that intercepts traffic. Developers can do three things to significantly reduce the risk of exposing sensitive data in transit.

- **Implement App Transport Security (ATS).** ATS enforces a minimum network security policy that forces apps to connect to back-end servers using HTTPS instead of HTTP to encrypt data in transit. SDKs for iOS 9.0 and later enable ATS by default, but some developers opt out because their app relies on certain back-end services that can't support TLS 1.2. This is no reason to completely opt out of all the security benefits of ATS. Use exceptions to opt out on a domain-by-domain basis.

  The following code snippet from Apple shows the structure of ATS configurations:

```
The following code snippet from Apple shows the structure
of ATS configurations:
NSAppTransportSecurity : Dictionary {
    NSAllowsArbitraryLoads : Boolean
    NSAllowsArbitraryLoadsForMedia : Boolean
    NSAllowsArbitraryLoadsInWebContent : Boolean
    NSAllowsLocalNetworking : Boolean
    NSExceptionDomains : Dictionary {
        <domain-name-string> : Dictionary {
            NSIncludesSubdomains : Boolean
            NSExceptionAllowsInsecureHTTPLoads : Boolean
            NSExceptionMinimumTLSVersion : String
            NSExceptionRequiresForwardSecrecy : Boolean
// Default value is YES
            NSRequiresCertificateTransparency : Boolean
        }
    }
}
```

  ATS should be enabled globally by linking to the iOS 9.0 SDK or later. Some developers opt out of ATS by setting the "NSAllowsArbitraryLoads" key to "Yes" or "True." For security's sake, opting out should be a last resort. For any existing apps which communicate to servers over HTTP, an exception must be set using either the "NSExceptionAllowsInsecureHTTPLoads" or "NSThirdPartyExceptionAllowsInsecureHTTPLoads" key.

- **Encrypt or encode sensitive values before transmitting them over the network** for an additional layer of protection in the event TLS/SSL is compromised. Encrypt highly sensitive values with AES (also known as Rijndael) using a key size of 256. For hashing purposes, use an algorithm such as SHA-256 or higher.

- **Implement certificate pinning** to protect against man-in-the-middle attacks resulting from an adversary using a fraudulent certificate to impersonate a back-end service used by an app. When an app "pins" the public key of a service it trusts, it will reject all certificates that do not match the hash of the pinned certificate. Rejecting a certificate will stop any connection to the endpoint and prevent the transmission of data over a potentially insecure channel.

WEAK AUTHENTICATION

Strong authentication practices verify a user's identity. Attackers can exploit weak authentication measures to impersonate a user, bypass authentication to submit requests to an app's back-end services, or hijack a session to perform actions on behalf of the user within the app itself (for example, initiate transactions). We regularly identify one of three problems within iOS app authentication schemes, and recommend that developers do the following to strengthen authentication within their mobile apps.

- **Implement two-factor authentication (2FA)** to mitigate against attackers using stolen credentials. At the very least, 2FA should be used prior to the initiation of sensitive transactions. SMS is falling out of favor as a channel for 2FA, but it's still better than no 2FA at all.

- **Configure session-management cookies securely** which includes making sure they're of sufficient length, of sufficient entropy (i.e. unpredictable), and marked secure. If cookies are not of a sufficient length or sufficiently random, an attacker can execute a brute force attack by guessing or predicting the ID of a valid session or exercise all possible values. Marking a cookie as secure tells the browser to only send the cookie if the request is being sent securely, such as over HTTPS. If the cookie were instead sent over HTTP, the cookie could be sent in cleartext, potentially allowing an attacker to capture the session information and hijack an active session.

- **Perform authentication requests on the server-side** and not locally, because relying on client-side security alone is a bad security model. Never trust the client application when making authentication decisions. Implement server-side session timeouts, as well, because relying on the client to initiate a timeout may leave a session open long enough for an attacker to leverage.

FOLLOW SECURE MOBILE DEVELOPMENT BEST PRACTICES AND VALIDATE IOS APP SECURITY WITH TESTING

Insecure data storage, insecure network communications, and weak authentication are certainly not the only attack vectors for iOS apps. They are simply common areas for improvement that we identify in our testing. Following the guidance above alone will not guarantee the security of your entire mobile app, but it will set a good security baseline. In addition to establishing security standards and following secure mobile development best practices, make sure to validate and certify the security, privacy, and compliance of your app with regular or even automated mobile app security testing after each build, as well as deep dive penetration testing prior to deployment and after each app update.

Katie Strzempka is the VP of Customer Success and Services at NowSecure, oversees the NowSecure mobile app security testing methodology, manages a team of expert mobile app penetration testers, co-authored the book iPhone and iOS Forensics, and is a recognized expert in mobile security. Katie holds a Master's in Cyber Forensics and a Bachelor of Science in Computer Technology from Purdue University.

Securing the Cloud-First Organization

Twenty years ago, when applications were housed in the data center, and users were on the central network, companies began investing in security appliances to build a secure perimeter around the network. It was the best protection against would-be hackers. But today, applications live in the cloud, and most users access them remotely, so that stack of security appliances that has continued to grow is protecting a network perimeter that no longer exists.

In the mobile and cloud-first world, the Internet is the corporate network, and companies need a new approach to security.

Zscaler moves security and access controls off the network and into the cloud, so you can securely connect the right user to the right app or service, regardless of location. Because Zscaler delivers the entire security stack as a cloud service, it eliminates the need for security appliances that are slow, expensive, and difficult to maintain.

The Zscaler platform includes two security pillars:

Zscaler Internet Access securely connects users to the internet and cloud applications, regardless of device, location, or network. It ensures that nothing bad gets in and no data leaks out.

Zscaler Private Access offers authorized users secure and fast access to internal applications hosted in the data center or public cloud, without the need for slow, cumbersome VPNs.

Look to Zscaler if you're:

- Driving toward a cloud-first strategy
- Moving to Office 365
- Securing a distributed and mobile workforce
- Moving apps to Azure or AWS
- Securing an SD-WAN transformation

WRITTEN BY CHRISTOPHER HODSON
SENIOR DIRECTOR, OFFICE OF THE CISO **ZSCALER**

PARTNER SPOTLIGHT

Zscaler Cloud Security Platform

Cloud services that securely connect users to applications in the data center or cloud, no matter where users connect or what devices they're using

CATEGORY
Internet security-as-a-service

NEW RELEASES
Continuous

OPEN SOURCE
No

CASE STUDY

AutoNation, America's largest auto retailer, has 360 franchises and over 26,000 employees. With limited IT resources, AutoNation chose Zscaler to bring consistent security across the organization and centralized controls for managing policies, monitoring threats, and ensuring fast access to the apps and services employees need.

AutoNation forwards all its internet-bound traffic to the Zscaler security cloud, and has added protection against advanced threats with Zscaler Cloud Sandbox. Having established internet-only offices, AutoNation then deployed Office 365. The company acknowledges that it is moving to the cloud quickly, getting away from physical data centers. Zscaler provides a robust platform for enabling this transformation.

STRENGTHS

- Better user experience
- Improved security
- Reduced costs
- Decreased complexity

NOTABLE CUSTOMERS

- National Health Services
- FIS
- General Electric
- Nestlé
- United Airlines

WEBSITE zscaler.com

TWITTER @Zscaler

BLOG zscaler.com/blogs

Baking Security Into the Development Lifecycle

QUICK VIEW

01 Application Security is not new, It has been around for two decades

02 Handling security issues early on in the development cycle will cost less money

03 Security can be applied to either agile or waterfall processes

04 Management support is key to the success of this initiative

BY **BOAZ SHUNAMI**
CEO, **KOMODO CONSULTING**

Back in May 2017, we were shocked to see how the WannaCry ransomware attack spread like fire in a haystack, followed by the devastating Petya and NotPeya ransomware attacks (also dubbed wipers due to their nature).

Application security is not new. In fact it has been around since the early 2000s and in a similar environment, where code-red, nimda, and other viruses were causing global havoc. Securing the software development lifecycle was initially a Microsoft initiative developed as a result of the growing number and impact of vulnerabilities in its products and code.

The model has significantly evolved since, though the principles remain the same: catch security bugs early on in the development cycle to cut your costs of repairing them. The damage that may be caused to customers and vendors alike will be reduced.

What is the SDLC, SSDLC, or SDL? It stands for Secure Development Lifecycle, which is the way you apply security to the lifecycle of developing a product or a system. In order to apply or "bake" security into the development lifecycle, you will need to consider the type of methodology you want to use for development. For example, whether it is waterfall or scrum, you need to apply the same principles, but in different methods. It is also very dependent upon the risk appetite of the organization as a whole, particularly as it applies to the specific system. The risk appetite will define how many resources should be invested into this process, as it can be quite cumbersome in terms of development and security professional resources for any R&D effort.

A Secure Development Lifecycle can be applied to each stage of development. Let's start with a waterfall example. For every stage of the cycle, you need to apply a specific security activity.

PROJECT INCEPTION

At the initial stage, when the project is in its inception, perform security threat modeling, which will be done at the stage when the product is merely a series of boxes and arrows.

DESIGN PHASE

Following that, you have the design phase, where you can apply security design templates that will allow you to make sure that no security controls have been "mistakenly" forgotten in the process. You can find a link to free secure design templates here.

CODE REVIEW

Once the code has been developed, or while it is in the process of being developed, you can perform code reviews.

PENETRATION TESTING

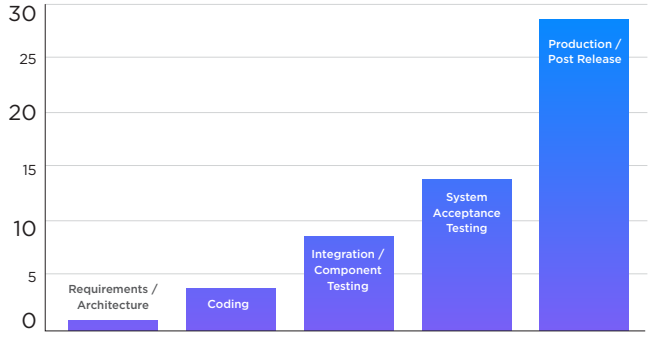
When the system is complete, after it has been tested and is stable and ready to go live, you can perform penetration testing on the application level.

SECURE DEPLOYMENT

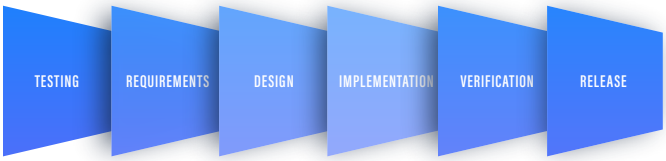
Before going live to production, you need to validate secure configuration and maintenance to make sure that while the system is online, when it encounters a new vulnerability or threat, it will be handled appropriately and the risks of exploitation will be reduced.

Some of these steps can save the company a lot of money when performed properly at the correct time, and their

benefits are huge to improving the overall security of the system as well as properly estimating and handling the timelines of the project delivery.



One of the simplest examples to demonstrate the actual value of incorporating security will be in the design phase. A security design issue that you only discover later in the deployment phase may be very difficult to fix. It may require a complete makeover of the system, or it means that the system will go live with a known issue that is only partially addressed or completely unaddressed.



Secure design review, when done appropriately, should only take several hours to several days depending on the complexity of the system and the risks associated with it, but its benefits are huge.

This activity cannot be performed by just anyone. You will need a professional security consultant to do that, someone with an understanding of the technology side and the requirements from architecture, infrastructure, code, and the associated threats.

The next step will be performing secure source code reviews on the code being developed. This can be done by an automated tool or a security consultant, though hiring an expert using an automated tool will yield the best results for the time spent. The output of code review tools for security usually requires an expert to interpret and decide on the best approach to the solution. It may not be easy for just any developer to understand the output, as many of the attack vectors and vulnerabilities associated are outside the day-to-day operations of coding, so you may require an expert to help resolve false positives, understand their impacts, and understand where false negatives can exist, e.g. issues an automated tool is unable to identify. You should consider the best value for the time invested.

Next, you will have penetration testing, when the perfect scenario will be on a system running on pre-production or staging environments and is ready to be deployed. Penetration testing should be done by a dedicated team of people with the skillset, understanding, and experience to perform it well.

When the system is ready to be deployed, you need to perform what is called secure deployment. That means removing all redundant and unnecessary interfaces, configuration files, backups, and hardening the server and system configurations to make sure that the production environment is robust for this stage.

You can also apply SDLC to scrum, for faster-moving development cycles. In this case, you will need to have a security person as part of the sprint team and decide on the key activities they will need to perform for every sprint, depending on the scope of the specific sprint. In any case, and independently of your chosen development methodology, the steps are the same. You will need to review the design and code, test, and securely deploy the application.

Alternatively, when working in scrum methodology, you can have a security sprint every few sprints, where security is the primary focus. Again, you need to make sure that the significant issues such as design and threat modeling issues have been understood and handled early on to reduce total cost of the project and significant architectural changes in later stages of development.

The key point to understand here is that "baking" security into the development lifecycle cannot be a developer decision or even a security decision (though it can be your initiative). This must come from management, or R&D management, because these processes take effort off development and put them into security. What happens is when you take these resources out of R&D for developing new functionalities and into security? You may run into a problem where you're facing a strict timeline for a project and security is holding it back (as is the case when something critical pops up). In these cases, you run into the problem of not meeting the deadline, and if management does not support the process, then things will be left behind, unhandled.

Having the management onboard with this process is key in whatever development methodology you use. Another thing to consider is that you will have to have a dedicated security consultant (or internal employee) who understands development, code, security, and the business.

Boaz Shunami has two decades of experience in the information technology field, specializing in application security, cyber security, and threat intelligence. Boaz is the CEO of Komodo Consulting, a cyber security consulting company, and has worked and advised on thousands of projects worldwide, ranging from Fortune 500 companies to local enterprises and cutting-edge start-ups. He has a great technological understanding and an ability to explain tough technological challenges to management.

SECURE STRATEGIES

AT CLUB CODE

Software development can feel like a big party sometimes. Everyone can have a lot of fun and celebrate over fixing that latest bug or finally getting that feature to work. However, there's always the potential for things to get messy, or for one bad egg to ruin everything. So what can developers do now to help secure their applications? Here at "Club Code," you can see what the management has to deal with on a daily basis, and how they keep their applications secure. We learned some of the most popular security practices from over 540 developers. Pull up a chair and take a look.

MAKE PERMISSION EXPLICIT AND DENIAL DEFAULT

50% of respondents make permission explicit and denial default. This is sometimes referred to as the "principle of least privilege." This means that each application component or user should only have access to what they need to perform a task. For example, if an API only needs access to a database in order to serve its purpose, then that's the only software it should have access to.

ARCHITECTING AND DESIGNING FOR SECURITY

59% of respondents build apps with security in mind. This includes adding requirements throughout the SDLC, determining how data will be stored, thinking like an attacker to expose vulnerabilities before they're made public, and how many logical tiers group the application's components.

SANITIZING DATA BEFORE SENDING IT TO OTHER SYSTEMS

44% of developers sanitize data before sending it to other systems. Data sanitization in this context refers to the process of changing a user's input to an acceptable format by eliminating special characters after the fact. For example, if someone tried to inject JavaScript through an input field, we could sanitize the data by removing characters like < or /.

DON'T IGNORE COMPILER WARNINGS

48% of developers don't ignore compiler warnings. This is one of the easier tactics developers can implement into their workflows to make security testing easier and applications more secure. Perhaps the most concerning fact is that a majority of respondents ignore compiler warnings!

VALIDATE INPUTS

83% of developers validate their inputs . In order to protect apps against cross-site scripting and SQL injection, developers can ensure that inputs are validated by only allowing users to enter a particular input into a field, for example, making sure users can't add any special characters to a simple "name" field on a web form.

Application Security for Modern Operations Teams

QUICK VIEW

01 Operations teams have seen major benefits from DevOps; now those benefits are extending to security.

02 Modern application security happens in four ways: instrumenting business logic, focusing on authenticated traffic, monitoring account actions, and completing the loop from ops to dev to security.

03 Developers, operations, and security can collaborate to increase application security defense capabilities.

BY **JAMES WICKETT**
HEAD OF RESEARCH, **SIGNAL SCIENCES**

You know the drill: XSS, SQLi, command execution. It's 2017, yet for application security it might as well be 2003. That's not just rhetoric — the same recommendations that the Open Web Application Security Project (OWASP) delivered as the OWASP Top Ten in 2003 has largely remained unchanged over the last 14 years. Along with that, the same defensive practices of input sanitization and parameterized queries are largely still used as the best remedy to application security woes. While these are certainly good practices, they are incomplete in today's modern application ecosystem — they focus solely on developer remediation and offer little to no insight into how operations teams should approach application security.

During the last decade, operations teams have seen a dramatic shift that is still unfolding across the industry. The most forward-thinking teams have already adopted DevOps practices, and the rest are in the process of doing so. Services are being decoupled from their once-monolithic applications into microservice architectures composed of APIs and third-party providers. Deploy rates are moving from once-a-quarter to on-demand, usually multiple times a day. Developers and operations staff now work together to rotate on-call duties and promote availability and reliability of systems.

These changes paint two opposing pictures: security is still largely stuck relying on true-but-tired defensive measures from the days of slow-moving IT, and operations that has radically transformed into a fast-moving IT delivery pipeline. Security practices such as static code analysis and long, drawn-out approval processes continue to slow down the software delivery pipeline with more and more questionable returns. However, there is a better way: operations teams can help security go faster and reduce threats at the same time.

MODERN OPERATIONS' IMPACT ON APPLICATION SECURITY

There are four ways that operations teams can strengthen application security while simultaneously helping to speed development and response to threats as they happen. The four ways to achieve these goals: Instrument Business Logic, Focus on Authenticated Traffic, Monitor Account Actions, and lastly, Complete the Loop from Ops to Dev to Security. Let's look at each of these areas in turn. Addressing each of these four areas should help answer the two questions that application owners and security engineers are asking: Am I being attacked right now? Are the attackers having success?

INSTRUMENT BUSINESS LOGIC

Operations has always been good at monitoring CPUs and memory. However, this is becoming more and more irrelevant in the modern architecture patterns of cloud, microservices, and serverless. Instead, they must focus on instrumenting business logic inside the application. This often gets overlooked by operations and security teams. What does "business logic" mean? Some examples of business logic are: transfer of funds, viewing an account balance, creating a new account, adding a credit card, or changing a customer address. For each application, the

specific business logic requirements are unique, requiring an effort to gather the application owners and determine what to instrument.

Try not to go overboard here and attempt to instrument everything. Instead, work to span across development, operations, and security to identify the highest-risk business logic. If someone had full access and were able to take advantage of our application, what would they do? Of course, if you have the expertise in-house, doing a threat model is a great exercise to help arrive at these requirements. Even if you start with just a handful of application flows, the instrumentation of business logic helps establish a baseline for the application. It can then be monitored for variation that might suggest the application is under attack.

FOCUS ON AUTHENTICATED TRAFFIC

Every single day, websites are under attack. Most likely, a day doesn't go by without your site being targeted by some vulnerability scanner or attack tools. This is the norm in today's internet, and alerting on every XSS attempt you detect will overwhelm your team. SQLi, XSS, and command execution are a lot more interesting in the context of a logged-in user.

Focus on application instrumentation and monitoring that can tell the difference between authenticated users and unauthenticated users. Are any of my customers trying to attack me? Has an account been breached? Looping this instrumentation back to customer support specialists and security teams is key for finding users or partners that have had account credentials stolen or hacked.

MONITOR ACCOUNT ACTIONS

Instrumentation comes easy at the server level. But the threat landscape has shifted. Today's targets, more often than not, are user accounts and personal identification information. A whole class of attacks known as Account Takeovers (ATO) are on the rise. The ATO storyline is a familiar one — millions of accounts are compromised, and data is leaked to the internet and sold to the highest bidder or to some nation-state hacking group. These accounts are valuable because they include working credit cards and user data.

Instrumentation for ATO must include: login success, login failure, password reset request, successful password reset, email address change, and username change. Just like in business logic instrumentation, determining what to instrument will require collaboration across teams. One other leading indicator of ATO is traffic source. Knowing your users' normal geography and network address means substantial differences should set off alerts. "Do I know the normal rate for login failures and success across my entire user base? What IP address blocks do my users normally come from?" Knowing the answers to these questions can help you defend your users' accounts.

COMPLETE THE LOOP FROM OPS TO DEV TO SECURITY

Monitoring multiple dashboards to view pertinent security information is not a valid option in the modern IT organization. More and more teams have adopted the DevOps pattern of ChatOps, wherein tools are integrated into team chat systems like Slack or HipChat. This puts tools where developers and operations engineers already are. When the login portion of your site is seeing suspicious traffic, it makes sense to get that information in front of the team of developers and operations staff that support login functions. Closing the feedback loop doesn't just mean chat. Integration with other systems like ticketing systems, monitoring systems, and even SIEM or logging systems helps enhance the communication between teams.

The focus here is providing tactical security feedback to the teams that are responsible for the application. This creates a security self-service model rather than siloing security information with just the security team. When developers, operations, and security teams distribute security feedback loops throughout the system, security becomes a team sport.

Every possible chance to create a feedback loop between development, operations, and security creates a new opportunity to get more eyes from different disciplines to participate together in defending the application. Feedback loops provide a virtuous cycle. They enable security to no longer be seen as the organization that slows things or inhibits agility, but instead integrates application security feedback and acts as a true business enabler.

MOVING FORWARD

Through these four modern approaches to application security, it is possible to go faster and reduce threats at the same time. A new day is on the horizon. These improvements move application security out of the silo of security and into the DevOps world. Security becomes a team sport, and the goal becomes achieving better security together, which ultimately leads to more defendable systems and applications.

There are two questions that application owners must ask: "Am I being attacked right now? Are the attackers having success?" Through implementing these four approaches, you will feel more confident in your ability to answer these questions.

James Wickett (@wickett) is the Head of Research at Signal Sciences, a web protection platform that high-performing DevOps teams love. He is the author of the most popular courses on DevOps topics in the Lynda.com and LinkedIn Learning platforms. James lives in Austin, Texas and has helped run DevOps Days Austin for the last six years. In his spare time he is trying to make a perfect BBQ brisket.

Software Integrity is a Journey.

Let Us Be Your Guide.

Synopsys offers the most comprehensive solution for integrating security and quality into your SDLC and supply chain.

Explore what Synopsys can do for you at
www.synopsys.com/software

SYNOPSYS®

There's No Finish Line to Building Software Security and Quality Into the SDLC

The sensitive nature of information that many applications consume has made them a prime target for attack. While an application or a piece of software may be considered secure at a certain point, a new vulnerability could emerge at any time. For this reason, it's critical to be proactive about security.

At the same time, organizations are quickly responding to the evolving needs of the market. As these needs evolve, software quality remains a constant consideration in meeting software development goals. Since security vulnerabilities are quality defects that can be compromised, software security should be addressed with the same level of rigor within the software development life cycle (SDLC). After all, software security is an operation that protects critical business practices.

Organizations strive to operate quickly, effectively, and efficiently, and they can do so while infusing both security and quality into their operations. Only when security and quality are treated with equal importance can threat handling become a proactive strategy rather than a reactive response.

In software, this amalgamation of security and quality is known as software integrity.

So how can organizations embrace a proactive strategy? Disregard the emphasis on moving left, and adopt an approach to build security in. There are steps within each SDLC phase an organization can take to ensure that the software it produces is secure and of the highest quality. From training, tooling, and automation to managed and professional services, there are a variety of solutions to infuse integrity throughout the SDLC without negatively impacting velocity.

To stay ahead of threats, your software integrity program must be persistent, comprehensive, and dynamic. Start your journey.

WRITTEN BY DR. ANDREAS KUEHLMANN
GENERAL MANAGER, **SOFTWARE INTEGRITY GROUP**

PARTNER SPOTLIGHT

Software Integrity Platform

SYNOPSYS

The most comprehensive solution for integrating security and quality into your software development life cycle (SDLC) and supply chain.

CATEGORY
Software Security

NEW RELEASES
Quarterly

OPEN SOURCE
No

CASE STUDY

Alcatel-Lucent Improves Code Quality and Security with Synopsys Software Integrity Platform

Alcatel-Lucent (acquired by Nokia) is a leading provider of telecom solutions, serving virtually all major wireless providers worldwide. The integrity of their products is a top priority as downtime for their customers can have devastating consequences. Alcatel-Lucent uses Synopsys Static Analysis and Test Optimization to automate software quality and security testing within their agile development environment. Synopsys solutions are used to ensure Alcatel-Lucent's code is of the highest possible standard, providing a more predictable development process, improving reporting capabilities, and increasing operational efficiency.

STRENGTHS

- **Comprehensive:** Our combination of products, managed services, professional services, and training is unique in the industry.
- **Developer-friendly:** The industry's most advanced automated tools with high developer acceptance.
- **Scalable:** Rapidly respond to changing testing requirements and evolving threats with access to the testing capacity you need, at the depth you need, exactly when you need it.
- **Industry leader:** With 20 years as a leader in software security and quality, we are uniquely positioned to adapt and apply best practices, tools, and strategies to new technologies such as IoT, DevOps, CI/CD, and the Cloud.

NOTABLE CUSTOMERS

- Bally Technologies
- AirFrance
- MStar
- Direct Edge
- Dahua Technology

WEBSITE synopsys.com/software

TWITTER @SW_Integrity

BLOG synopsys.com/blogs/software-security

Packers: How They Work, Featuring UPX

QUICK VIEW

01 Malware packers are in common use today to hide malware signatures and obfuscate intent.

02 UPX, though not designed with malware in mind, was and is commonly used in the malware community today.

03 By examining UPX, we can learn how packers work, how malware authors use them, and how you might want to use them in your own code.

BY **CHRIS LAMB**
CYBER-SECURITY RESEARCH SCIENTIST, **SANDIA NATIONAL LABORATORIES**

UPX, the Ultimate Packer for Executables, is a software packer. A really good one, in fact, and it's currently under active development to boot. The source code for it is available , and it's available on a variety of operating systems. It compresses executables for a variety of operating systems too.

As the name suggests, UPX packs executables. And by "pack," I mean it compresses and compartmentalizes programs. It will take an executable, compress it, and pack the compressed code into another section of the executable. At runtime, it will uncompress the previously compressed code and execute it.

Along the way, it happens to obfuscate the intent and actual code of the program.

Don't get me wrong here—UPX is a remarkable feat of engineering, and I don't begrudge the use of the program. It has real applications in areas where storage space or bandwidth is at a premium. And honestly, it's not hard to detect or to work around when used, and most anti-malware solutions don't have a problem detecting it today. But the idea behind packers in general, and UPX specifically, is very useful to malware authors.

To show how these tools are used to hide malware payloads, I'm going to show you how packers work, using UPX, and how you can detect them. Now I'm using MacOS, but you can follow along with Linux if you'd like. Most of the points I make will carry over. UPX is available on Windows too, and works great, but it won't be as easy to follow the examples in Windows.

Installing UPX. UPX is a breeze to install. You can compile from source if you'd like, but I didn't, so I used Homebrew. You could use MacPorts too, or if you're on Linux, it's likely available via the package manager of your choice. To install via Homebrew, typing brew install upx from the command line, should work just fine.

Creating a test program. Now that you have UPX installed, we're going to create two different versions of `/bin/ls`. One packed, one not, and we'll start to compare the two. I suggest you create a directory you can start to work in. I called mine `upx-test`, but you go with whatever name works for you. Copy `/bin/ls` into the directory.

Now, let's create a packed version: `upx ls -o ls-upx` should do the trick. Now, just so we are able to maintain both versions, copy `ls` to `ls-orig`. Just for fun, compare the output from running `./ls-orig` to `./ls-upx`. Notice they look exactly the same. Just for fun, compare the hashes of `./ls`, `./ls-orig`, and `./ls-upx`. You'll notice that `./ls` and `./ls-orig` have the same hash, but that the hash of `./ls-upx` is different even after executing `./ls-upx`. This is interesting in that it implies that the UPX packed executable extracts the packed code, runs it, and terminates its process without serializing anything to disk. The program representation on disk never changes.

LOOKING THROUGH THE BINARIES

See upx.github.io for more information, source code, and documentation.

Now, let's start to look through the binaries. We'll use Jonathan Levin's fabulous JTool for this. Sorry, there's not an equivalent on Linux, but later I'll use Hopper for some reverse engineering. You can see similar things via the Hopper interface, and Hopper runs on both Linux and MacOS.

Anyway, if you're on a Mac, install JTool (brew install jtool should work. I believe MacPorts has this too). If you're on Linux, you'll need to use gnu binutils for this.

Anyhow, let's take a look at the program formats, shall we?

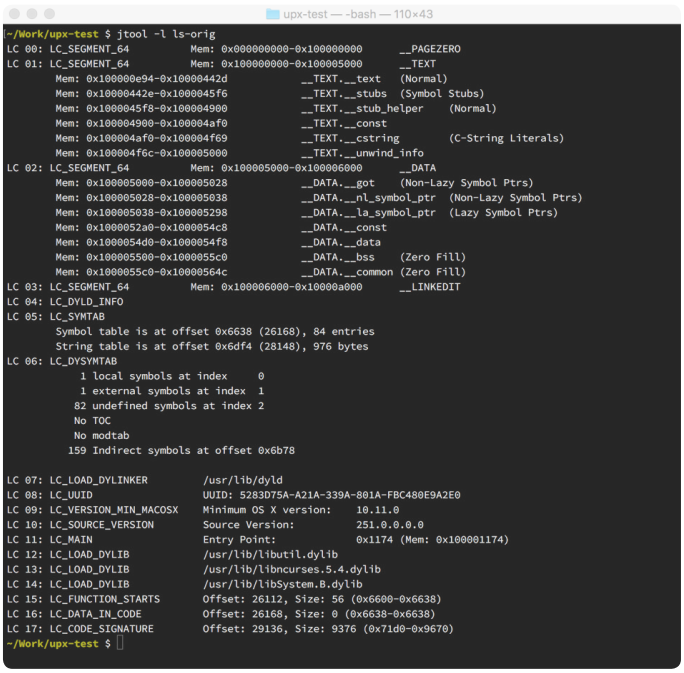


Figure 1: *Segments and sections of the original ls command*

In figure 1, we see the output from `jtool -l` over the original ls executable. This has a large amount of information, including the program entry point and OS versions. The text segment is fairly large as well, and contains constant values, strings, and other information. Now, let's take a look at the version we just packed:

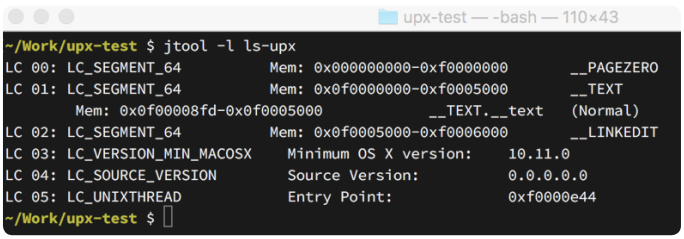


Figure 2: *Segments and sections from a UPX-packed ls command*

Well, it looks like we have a little less information here, and most of the program consists of the text segment too. So what are these sections?

Well, LC_SEGMENT_64 is a standard 64-bit segment to be loaded into an address space (this is MacOS specific). LC_VERSION_MIN_MACOSX lets us know the minimum OS version for this executable, and LC_UNIXTHREAD gives us information about how the program is to be started, including the program entry point. __TEXT contains program code usually, while __LINKEDIT contains information used by the linker—things like strings, function names, and so on. __PAGEZERO is a chunk of memory allocated and protected to throw exceptions on access. This helps protect the program against null pointer or integer dereference errors in code.

It seems that UPX pretty radically changes the program. Let's look a little deeper.

REVERSE ENGINEERING

Now, we're going to start looking into the program. We can do this from the command line using tools like JTool, or using high-end disassemblers like IDA. In this case, we'll use a reverse engineering tool called Hopper. Hopper is a fully-featured reverse engineering tool, and it offers a free trial to boot. If you're following along, go ahead and download it. I'll wait.

Ok, great—let's get started.

Open `./ls-upx` and `./ls-orig` in Hopper, and take a look at the left pane. Select the Proc. tab. The first thing you see is that there are very few functions defined in `./ls-upx` when compared to `./ls-orig`.

You can download Hopper from hopperapp.com. It's something like an order of magnitude less expensive than IDA Pro, so if it fits your needs, I highly recommend it. It doesn't support all the processor types IDA does, nor does it run on Windows, but it works very well for MacOS reverse engineering.

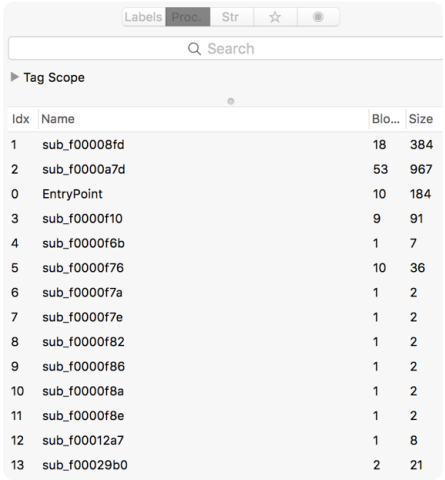
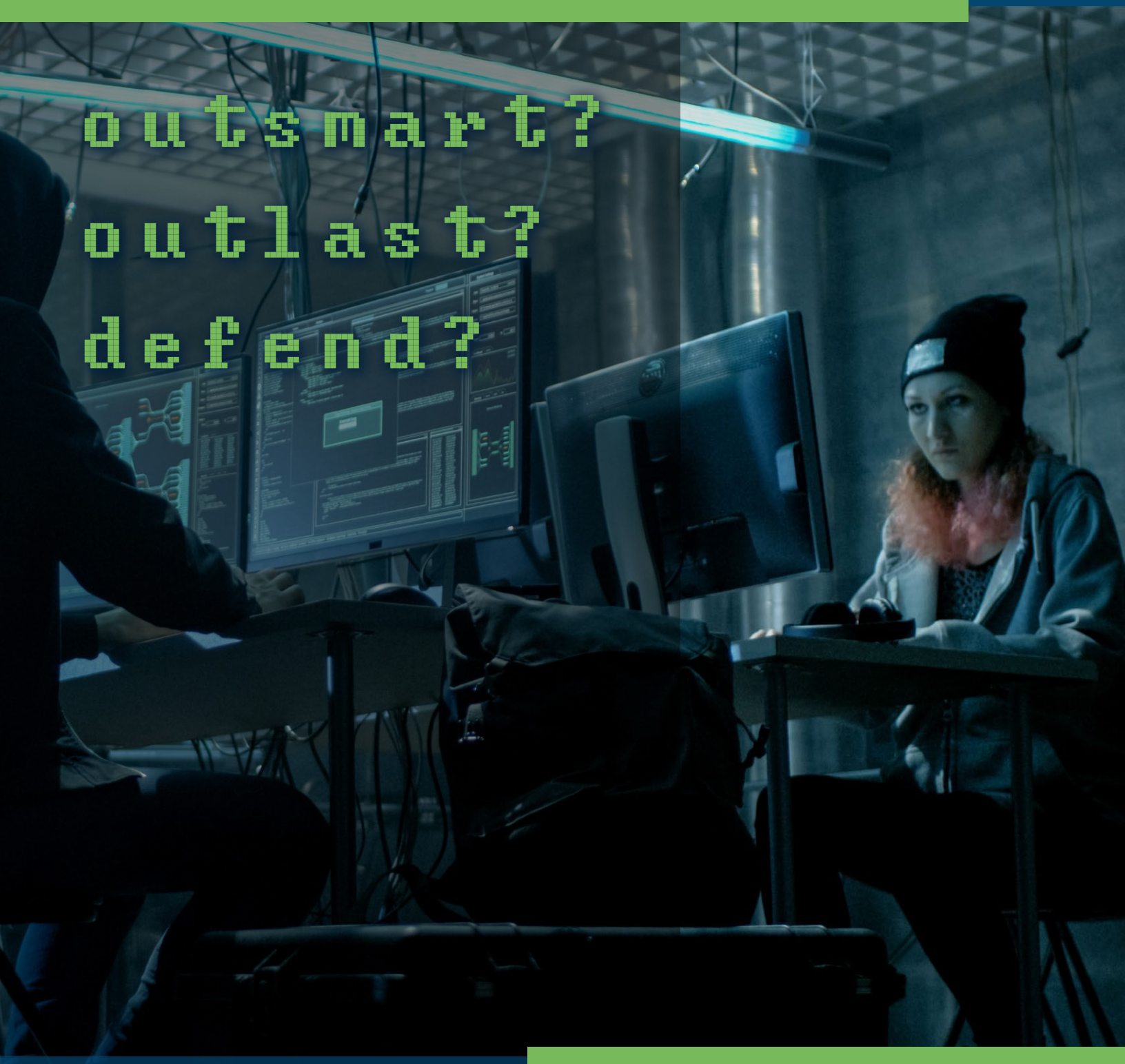


Figure 3: *UPX makes executables not-very-functional!*

Can You...



cmd+ctrl
A SECURITY INNOVATION HACKATHON

www.securityinnovation.com/CMDCTRL

Batten Down the Hatches to Secure Your Software

In today's ever-changing threatscape, developers need to assume their software will be attacked and build applications that can withstand hostile environments, just as buildings are designed to withstand hurricanes, earthquakes, and other natural disasters.

The biggest industry challenges, as I see them, are as follows:

OVER-RELIANCE ON AUTOMATION

Automated security scanning tools are good at finding common vulnerabilities quickly and systematically. However, they are prone to false positives and can't detect certain vulnerability classes. Because of this, scanning tools need to be complemented with manual efforts to find vulnerabilities that elude them. A human can understand the business logic and alter test cases accordingly – logic that is difficult to program into a general purpose, linear-driven automated tool.

GOOD DEVELOPERS AREN'T NECESSARILY SECURE CODERS

Developers genuinely want to write secure code, but don't know how. In fact, many don't understand common attacks on the platform, security shortcomings of the development language they use, how certain frameworks can mitigate security problems, etc.

The notion that security "takes longer" or "costs more" is a fallacy. It takes just as long to write a line of secure code vs. insecure code. However, secure code actually costs much less than insecure code when it comes to maintenance, defect management, patching, and re-work. According to NIST, the cost of addressing security vulnerabilities during coding/implementation is approximately 15% the cost of removing such a defect post-release. Including security in the requirements and design phases results in fewer vulnerabilities to find, validate, and fix, leaving more time to develop features.

This is why organizations need to establish defensive coding best practices. This includes coding standards, code reviews, automated static analysis, unit testing, and defect management.

To learn more, check out our blog post: blog.securityinnovation.com/defensive-coding-best-practices.

WRITTEN BY ED ADAMS
CEO, **SECURITY INNOVATION**

PARTNER SPOTLIGHT

CMD+CTRL Hackathons

SECURITY INNOVATION

"[Playing CMD+CTRL] We got to be creative, competitive, and even a little bit evil...I learned a lot about web app hacking that will make me better at my job."

CATEGORY
Immersive/Gamified Learning

NEW RELEASES
Quarterly

OPEN SOURCE
No

DESCRIPTION

Test your team's security skills in a whole new way as they reverse roles and hack their way through hundreds of vulnerabilities that plague applications today. Without knowing how applications are attacked, your team are flying blind as they will fail to implement countermeasures to thwart them.

Our CMD+CTRL hackathons provide a real-world sandbox to get your hands dirty as you learn and apply creative attacks in authentic web and mobile applications that include:

- Shadow Bank – banking website
- Gold Standard Bank – banking website with advanced challenges
- Shred Retail – eCommerce website
- Account All – HR website
- InstaFriends – social media website
- Runstoppable – Android fitness tracking application

*Hackathons can be delivered as a staffed event or stand-alone practice range

STRENGTHS

- Interactive and engaging
- Utilizes proven Capture-the-Flag (CTF) techniques in a real-world setting
- Websites are fully functional, allowing users to exploit common features (i.e. adding items to a cart, purchasing, transferring money, applying for a loan, submitting time sheets, and more)
- Messages, sounds, and "Easter Eggs" throughout make it fun and engaging
- Real-time scoring creates friendly competition and motivation
- Largest repository of intentionally vulnerable web sites
- Includes 300+ vulnerabilities that cover 20 vulnerability classes including the OWASP Top Ten and CWE Top 25
- Each challenge has a point value based on complexity

WEBSITE securityinnovation.com/CMDCTRL **TWITTER** @SecInnovation **BLOG** blog.securityinnovation.com

Not only are there very few functions defined, they are all local. None of these are imported from external libraries (go to Navigate > Imported Symbols to see what I mean). And if you click on the str tab you'll see that the UPX version has no strings defined in the program, while the non-UPX version has plenty.

Now, select the pseudo-code button at the top of the display area:

Figure 4: *Pseudo-code mode activated!*

If we do this for both the `./ls-orig` and `./ls-upx`, you'll see something interesting. First, the original ls command has a large initial entry point, while the UPX'd version doesn't. The original makes system calls fairly immediately (i.e. getenv(.), ioctl(.), and so on). The UPX'd version is self-contained (we'd expect this, as it doesn't import anything).

```
int EntryPoint() {
    r13 = ret_addr;
    r12 = &arg_0;
    rax = EntryPoint & 0xfffffffffff0000;
    rcx = *(int32_t *)(rax + 0x10);
    if (rcx == 0x0) goto loc_f0000ebc;

loc_f0000e74:
    rax = rax | 0x20;
    rdx = 0x0;
    goto loc_f0000e84;

loc_f0000e84:
    if ((*(int32_t *)rax != 0x19) || (*(rax + 0xa) != 0x544944454b4e494c)) goto loc_f0000e8f;

loc_f0000e9d:
    rbx = *(rax + 0x18);
    r8 = *(int16_t *)rbx & 0xffff;
    r8 = r8 + rbx;
    r9 = rbx + 0x2;
    do {
            r14 = *(int32_t *)(rbx + 0xfffffffffffffffc);
            rbx = rbx + 0xfffffffffffffffc;
    } while (r14 == 0x0);
    rbx = rbx - r14;
    goto loc_f0000ebc;

loc_f0000ebc:
    r15 = sub_f00008fd(rbx, r14, &var_4030, 0x4000, r8, r9, r12 + 0xfffffffffffffff0);
    sub_f0000f7e();
    *(r12 + 0xfffffffffffffff8) = sign_extend_64(r13);
    rax = (r15)();
    return rax;

loc_f0000e8f:
    rdx = rdx + 0x1;
    rax = rax + *(int32_t *)(rax + 0x4);
    if (rdx < rcx) goto loc_f0000e84;
}
```

Figure 5: *Decompiled entry point from UPX'd ls*

Interesting things here! Look at the do/while loop in the center of the function; here, we're traversing the program __TEXT segment from the top down, until we get to a non-null value. Then, we have two additional function calls, and we exit (notice that once we get to the call to sub_f0000f7e(.) we have no more jumps—although, honestly, the called functions could cheat and jump, but we'll take a look at that shortly).

Of these functions, sub_f00008fd(.) is the most interesting. If we look into that function (which does not decompile well, to be honest), we can see that this is the primary function dispatcher for the UPX code. If you look at the assembly block structure of the function (press the button just to the left of the pseudo-code button in Hopper), you can see that it's much more complex than the EntryPoint(.) function, and it delegates to many more functions as well.

You'll also notice that it returns a function pointer which the program them calls (e.g. rax = (r15)();). Guess what this is? This is the original program entry point after the code has been decompressed and is ready for execution.

> "UPX does lots of interesting things that we're not covering, like loading all of the dependent libraries, for example"

UPX does lots of interesting things that we're not covering, like loading all of the dependent libraries, for example—important on MacOS, as static linking isn't supported these days. But this gives you an idea of how these packers work. UPX doesn't encrypt either, which is certainly something you could do, though it shifts a bit into malware specific functionality. But at the end of the day, you can pretty clearly tell if a file's been packed. It has no (or very few) strings. It has no (or very few) library dependencies. And the defined functions in the executable is very small.

With UPX, there are other clues too—if you look through the binary, you'll find sequences of ASCII codes that spell "UPX!" used as tokens. You can see one of them just prior to where the booting function stopped scrolling through the __TEXT segment, for example.

Packers aren't magic, but they are well designed pieces of software engineering. Now you know how they work, and who knows, maybe you'll use UPX someday too.

Dr. Lamb currently serves as a cyber-security research scientist with Sandia National Laboratories. He is also a Research Assistant Professor affiliated with the Electrical and Computer Engineering department at the University of New Mexico. Currently, his research interests center around industrial control system cybersecurity, machine learning, artificial intelligence, and their intersections. He is a TOGAF 9 Certified Enterprise Architect and a Certified Information Systems Security Professional (CISSP) through the International Information Systems Security Certification Consortium.

How to Go Zero Trust Like Google's BeyondCorp

By Ivan Dwyer
VP of Product Marketing, **ScaleFT**

When a sophisticated attack originating from China targeted a number of large enterprises in 2009, the common response by those affected was to bolster their perimeter security by buying more firewalls and VPNs. Google, one of those targeted, recognized that further protecting the perimeter was the wrong response for the times given the advent of mobile devices, cloud services, and SaaS applications. Instead, they began an internal initiative called BeyondCorp, meant to redesign their corporate architecture with regards to how employees access company resources. The result has been a smashing success, enabling Google employees to work from any location without the need for a VPN.

Google has since released a number of research papers about BeyondCorp, attracting the attention of companies who also wish to improve their corporate security posture. Looking at it purely through the lens of Google may appear too daunting, though. How can companies achieve a similar outcome in a feasible manner? In following the principles of Zero Trust, an enterprise security framework meant for the modern cloud era, a BeyondCorp-inspired architecture is attainable for companies of all kinds.

KEY BENEFITS OF ZERO TRUST

- **Make smarter trust decisions by redefining identity.** Instead of a static record in a database, Zero Trust builds a dynamic identity profile as a user plus their device at a point-in-time, providing the right context to authenticate and authorize a request.
- **Eliminate the common attack vector of static credentials being lost, stolen, or misused.** Zero Trust eliminates this wide attack surface entirely by issuing ephemeral credentials with every request that are tightly limited in scope and time.
- **Improve employee security posture through encouragement.** With a Zero Trust workflow that factors in device state through company-wide policies, employees will naturally stay up-to-date to avoid being blocked from getting their job done.

HOW TO GET STARTED WITH ZERO TRUST

- **Form your access policy framework**, which includes the user and device elements to collect, and the associated rules needed to make smart trust decisions. The easier the policies are to understand, the easier they will be to implement across the company.
- **Implement the access controls** to ensure every request to a protected resource is fully authenticated, authorized, and encrypted. This can be accomplished through a central access gateway, either by building your own distributed system, or by leveraging a commercial platform.
- **Migrate your resources** behind the access controls by pointing traffic to the central access gateway. You can then lock down your resources to only accept traffic from the gateway, significantly lowering the attack surface.

Executive Insights on Proactive Security

QUICK VIEW

01 While awareness of the importance of security has grown with more high-profile hacks, lack of emphasis from C-level executives continues to make it a low-level priority.

02 Companies are slow to adopt a sound security strategy in which they identify their most important assets and allocate security budget to protect them.

03 There is an opportunity and need for a fundamental shift in security constructs and workflows to align with enterprise trends to achieve agility and business goals.

BY **TOM SMITH**
RESEARCH ANALYST, **DZONE**

To gather insights on the state of proactive security today, we spoke to 25 executives from 25 companies who are familiar with the current cybersecurity threat landscape and the actions being taken to mitigate the damage being caused. Here's who we talked to:

KEVIN FEALEY PRINCIPAL CONSULTANT PRACTICE LEAD AUTOMATION AND INVESTIGATION SERVICES, ASPECT SECURITY

CAROLYN CRANDALL CMO, ATTIVO

JOSEPH SALAZAR TECHNICAL MARKETING ENGINEER, ATTIVO

AMIT ASHBEL DIR. OF PRODUCT MARKETING & CYBER SECURITY EVANGELIST, CHECKMARX

ASH WILSON STRATEGIC ENGINEERING SPECIALIST, CLOUDPASSAGE

PAUL KRAUS CEO, EASTWIND NETWORKS

ANDERS WALLGREN CTO, ELECTRIC CLOUD

ALEXANDER POLYAKOV CTO, ERPSCAN

PATRICK DENNIS PRESIDENT AND CEO, GUIDANCE SOFTWARE, INC.

CRAIG LUREY CTO, KEEPER SECURITY

BOAZ SHUMANI CEO, KOMODO CONSULTING

ERIC TRANLE GLOBAL CMO, LOGTRUST

DARRIN BOGUE, SENIOR SOLUTIONS ENGINEER, LOGTRUST

DAVID WAUGH VP SALES, MANAGEDMETHODS

MAT KEEP DIR. OF PRODUCT MARKETING AND ANALYSIS, MONGODB

AARON LANDGRAF SENIOR PRODUCT MARKETING MANAGER, MULESOFT

KEVIN PAIGE HEAD OF SECURITY, MULESOFT

FRED WILMOT CEO, PACKETSLED

GARY MILLEFSKY CEO, SNOOPWALL

WIE LIEN DANG VP PRODUCTS, STACKROX

CODY CORNELL CO-FOUNDER AND CEO, SWIMLANE

TERRY DUNLAP FOUNDER AND CEO, TACTICAL NETWORK SOLUTIONS

CHRIS WYSOPAL CO-FOUNDER AND CTO, VERACODE

YITZHAK VAGER VP CYBER PRODUCT MANAGEMENT AND BUSINESS DEVELOPMENT, VERINT

PRABATH SIRIWARDENA DIR. OF SECURITY ARCHITECTURE, WSO2

KEY FINDINGS

01 The most important elements of security are following the fundamentals of secure coding and putting a high priority on the security of what you are developing and sending to production. Companies need to educate entire development and production teams on security best practices and ensure everyone is following them. There are too many insecure coding practices – especially with regards to new technologies like cloud, mobile, IoT, and even web apps. People have forgotten what they know about security when working on new platforms. We will have a significant reduction in attack vectors if we follow secure coding best practices.

Companies also need a "security-first" mentality, which starts with prioritizing what's most important to secure since you cannot secure everything: security by design. Security is not a plug-in, it must be established early in the SDLC and maintained through production. Finding and fixing security bugs in the development stage is much cheaper than finding bugs during production. The bugs are easier to fix, and this approach results in more profitable code and applications over the long-term.

02 The cybersecurity threat landscape is changing on two fronts: 1) the number of threat vectors; and, 2) the sophistication of hackers. The proliferation of mobile, IoT, cloud, and cloud-based applications have increased the number of endpoints and threat vectors exponentially. All of this presents greater risk. Security best practices have been around for 20 years, but app and IoT developers fail to follow best practices. We see a lot of insecure coding calls that don't track user bounds and lead to buffer overflow – a problem that was solved years ago. It's a mess that's going to get worse. IoT devices are not being patched. These present a challenging operational capability, since we'll have more than 40 billion devices by 2020. We lack the talent and capability to prevent breaches today. The biggest challenge is educating those who are not educated, as well as the people who don't know what they don't know, and are not interested in hearing about the importance of security.

We still have politically motivated attacks; however, new groups with new tools are as sophisticated as nation-states. Threats are multifaceted using well-known infrastructures like Twitter and DropBox that serve as an obfuscation layer. Criminals are making more money from the Internet than any other source. Companies are making six-figure ransomware payments that are unreported because they do not want the negative publicity. Most attacks today are at the application level. Hackers don't need to struggle with firewalls and IPS systems when they have access to applications that provide a direct communications channel to enterprise data.

03 While a lot of security techniques were suggested as being effective, the most frequently mentioned were **continuous threat management, which includes real-time visibility and automatic responses to threats and intrusions**. This enables you to see vulnerabilities and intruder activities as they move around the network so you can determine what portion of your network has already been infiltrated and what information has been compromised, as well as continuous threat management and timely incidence response.

Continuous threat management does not diminish the importance of maintaining security best practices, good data hygiene, a sound security strategy based on the prioritization of protection, security by design, and pursuing holistic hybrid security solutions.

04 Not surprisingly, **financial services is the most frequently mentioned industry that benefits most from security**, followed by healthcare, manufacturing and retail – because that's where the money is. The most frequent work is around securing applications, devices, and APIs. Security companies are also preventing malware, phishing, and ransomware attacks, as well as identifying and visualizing where attackers are in the network. They're helping companies move to the cloud and share data securely across networks through encryption. Security companies are also helping clients understand best practices and inculcate security tests to prevent future damage.

05 Issues affecting security are diverse and seemed to fall into five broad categories: **1) awareness and knowledge, 2) bad practices, 3) velocity and scale, 4) risk assessment; and, 5) visibility**.

Many people in organizations still need to be educated on the importance of security overall, as well as basics like encrypting plain text data, how to set up and use security software that's already been purchased, connecting to malicious websites, and the lack of skills and knowledge to implement a secure network or applications.

Poor practices and lack of security policies are also hurdles to be overcome. Developers are using insecure coding practices and leaving behind crypto keys. There's poor data hygiene and systems not being properly patched in a timely manner, if at all. Security best practices are not followed due to a demand to get to market as quickly as possible, so controls fall through the cracks. There is particularly a lack of policy and procedures for developers. Other departments are procuring and producing their own solutions outside of the IT department's knowledge, which is known as shadow IT.

The cybersecurity threat landscape is changing quickly and companies are not prepared to handle the speed or scale of the change. The volume of incidents can range from 1,000, to 10,000, to 100,000 per day depending on the industry and company size. Automated tools are required to handle this volume of incidents. There are a myriad of tools to choose from, and they typically require a knowledgeable security professional to set up and interpret the results. Multiple tools are required to implement a holistic, hybrid security strategy, which is necessary for a company of any size.

Only the most sophisticated companies are performing risk assessments (risk = threats x vulnerabilities x assets) to inform their security strategy and budget planning. Few know what's on their network or where all their data lies. This is a requirement to make an informed decision about where investments should be made in security. You need to put your dollars into protecting your most important data.

Lastly, there is a lack of visibility into, and control of, cloud applications, as well as a lack of ability to monitor and respond to threats in real time.

06 Concerns regarding the current state of security are diverse as well; however, one prevailing theme is that **attackers are winning the war by becoming more sophisticated and spending more money on hacking tools** than companies are spending on security. Hackers are now offering ransomware-as-a-service, and the adoption of cloud, containers, microservices, and IoT is moving so quickly that it's difficult for companies to keep up with security best practices, let alone stop the bad guys. Companies need to change their state of mind and assume they've already been breached. Every company audited has been breached.

07 **AI (Artificial Intelligence) is the greatest opportunity for security to improve**. AI will automatically learn threat detection without human interaction. AI, machine learning, and deep learning will all enable improvements to security, along with sensors around the world detecting anomalies – differentiating between good and bad behavior.

08 Developers need to **keep security in mind from the beginning of the software development lifecycle**. Security should always be structured in upfront application design. Know the policies and procedures that will make your applications secure. Eliminate SQL injections and cross-site scripting (XSS). Secure code reviews increase your ability to see ways to secure your code over time. Focus on security by design and you will have a lucrative career. Look at the 25 worst security flaws and ensure what you are developing doesn't have any of these. Don't grab code from Stack Overflow, or anyone else, and put it into production without testing it first.

09 **Developers and security professionals need to collaborate for companies to have a stronger security posture**. For this to happen, developers need to keep their egos in check while security professionals need to stop spreading fear, uncertainty, and doubt. The two groups should work together to solve problems, share best practices, scan code for vulnerabilities, and provide correction and virtual patches so everyone can share the information necessary to solve issues while learning from each other. More collaboration will result in stronger, more secure code and networks.

Tom Smith is a Research Analyst at DZone who excels at gathering insights from analytics—both quantitative and qualitative—to drive business results. His passion is sharing information of value to help people succeed. In his spare time, you can find him either eating at Chipotle or working out at the gym.

CLOUD COREO

Integrated Infrastructure Security

The only solution that proactively tests infrastructure at deploy time

APPLICATION LAYER

HOSTS & CONTAINERS

CLOUD INFRASTRUCTURE

Fix AWS infrastructure issues at the source by integrating security insight into the development process

Out-of-the-box security and compliance checks (like CIS and NIST 800-171) ensure correct configuration throughout your application

Retain application context to better assess risks and determine exceptions

CLOUDCOREO.COM | INFO@CLOUDCOREO.COM

Proactive Cloud Security: Bringing Infrastructure Checking into DevOps

We've seen many organizations working to integrate security into DevOps, but the most effective approach we've seen is the idea of Security as enablers. In this model, the focus of security is to give guidance and support to developers and DevOps engineers, rather than try to control what development can or cannot do. This fundamentally shifts the perceived role of the Security team from arbitrary rule maker to a resource to help teams deliver better, more secure applications. This shift also results in a more educated, security-conscious development team, which has the added benefit of scaling security practices and awareness across all the stakeholders responsible for the application.

Moreover, by making application and cloud infrastructure security validation part of the CI/CD toolset, feedback goes directly to developers so they are able to fix the violation as soon as the problem is created. Remediation happens faster because the application and infrastructure code is fresh in the developer's mind.

Learning of potential issues early in the process puts policy departure decisions with the DevOps team

Most importantly, learning about vulnerabilities early in the process puts policy departure decisions with the DevOps team. This ensures that any organizational policy exceptions are intentional and grounded in the context of the application, instead of accidental. There are always going to be some policy exceptions that are deemed acceptable if the choice is between achieving implementation goals and not achieving them. As a result, the Security team should still set the standards, but the DevOps teams can work with both the security team and application owners to determine when a need is so urgent that an exception should be allowed.

WRITTEN BY PAUL ALLEN
CTO, **CLOUDCOREO**

PARTNER SPOTLIGHT

CloudCoreo Platform

CLOUD coreo

Ensure cloud infrastructure security from deployment time

CATEGORY
DevOps Security Automation Platform

NEW RELEASES
Weekly

OPEN SOURCE
No

CASE STUDY

Bazaarvoice powers reviews and other consumer generated content for millions of shoppers and products online. Today, they leverage the cloud to run hundreds of microservice applications, managed by dozens of autonomous teams. With millions of cloud objects to manage, Bazaarvoice leverages CloudCoreo to give their DevOps teams the autonomy required to innovate quickly, while still ensuring secure deployments.

CloudCoreo provides both centralized visibility to the security team while automatically distributing insights about infrastructure risks and vulnerabilities to each DevOps team for the infrastructure they own. This proactive model embeds critical security feedback into the development process, ensuring all infrastructure is secure by design.

STRENGTHS

- Find the difficult-to-find risks and vulnerabilities across your cloud stack
- Drive best practices and easily create deployment guardrails
- Automate compliance checks and reports for every build
- Full visibility and history of vulnerabilities across cloud accounts and teams
- Plug infrastructure checks into Jenkins to block jobs on critical violations

WEBSITE cloudcoreo.com | **TWITTER** @cloudcoreo | **BLOG** cloudcoreo.com/blog

Solutions Directory

This directory contains anti-tamper software, authentication, cloud access security, DDoS protection, endpoint security, and penetration testing tools, as well as many other tools to assist your application security. It provides free trial data and product category information gathered from vendor websites and project pages. Solutions are selected for inclusion based on several impartial criteria, including solution maturity, technical innovativeness, relevance, and data availability.

COMPANY	PRODUCT	CATEGORY	FREE TRIAL	WEBSITE
A10 Networks	Thunder TPS	DDoS protection	30 days	a10networks.com/products/thunder-series/ddos-detection-protection-mitigation
Acunetix	Acunetix Web Vulnerability Scanner	DAST, IAST	14 days	acunetix.com/vulnerability-scanner
Ad Novum	Nevis Security and Compliance Suite	WAF, authentication, identity management	Available by request	adnovum.sg/en/sg/solutions/products/nevis.html
Akamai	Akamai	CDN, DDoS protection, WAF	N/A	akamai.com
Akamai	Kona Site Defender	WAF, DDoS protection	N/A	akamai.com/us/en/solutions/products/cloud-security/kona-site-defender.jsp
Alert Logic Inc.	Alert Logic Web Security Manager	WAF	Available by request	alertlogic.com/solutions
AlienVault	Unified Security Management	Security monitoring	Available by request	alienvault.com
Amazon	AWS WAF	WAF	N/A	aws.amazon.com/waf
Amazon	CloudFront by Amazon	CDN, DDoS protection	N/A	aws.amazon.com/cloudfront
Anomali	Threatstream Platform	Security monitoring	Available by request	anomali.com/platform
AppMobi	AppMobi Security Kit	Apache Cordova app encryption and authentication	Available by request	appmobi.com
AppRiver	SecureSurf	WAF	30 days	appriver.com/services/web-protection
Appthroity	Appthority	Mobile AST	Available by request	appthority.com

COMPANY	PRODUCT	CATEGORY	FREE TRIAL	WEBSITE
Arbor Networks	Arbor Networks APS	DDoS protection	N/A	arbornetworks.com/ddos-protection-products/arbor-aps
Armor	Armor Anywhere	Cloud security platform	Demo available by request	armor.com/armor-anywhere-security
Arxan	Arxan Application Protection	Intrusion prevention system, code analysis	Available by request	arxan.com/products/product-overview
Attivo	Attivo Portfolio	Threat detection and monitoring	Demo available by request	attivonetworks.com/product/deception-technology
Auth0	Auth0	SSO and identity management	Free tier available	auth0.com
Avecto	Defendpoint	Privilege management, applicaton control	Demo available by request	avecto.com/defendpoint
Barracuda Networks	Barracuda NextGen Firewall	WAF	N/A	barracuda.com/products/ngfirewall
Bay Dynamics	Risk Fabric	Predictive security analytics	Available by request	baydynamics.com/risk-fabric
Bitdefender	Bitdefender GravityZone	Endpoint security, app security scanning	Available by request	bitdefender.com
BlackDuck	Black Duck Hub	Open source scanning	Demo available by request	blackducksoftware.com/products/hub
BMC	SecOps Response Service	Security management and analytics	Available by request	bmc.com/it-solutions/secops-response-service.html
Bricata	Bricata	Threat detection and analytics	Demo available by request	bricata.com/product
Bromium	Bromium Secure Platform	Virtualization-based endpoint security	Free beta available (as of September 2017)	bromium.com/platform.html
Browser Exploitation Framework Project	BeEF	Web browser penetration testing	Open source	beefproject.com
CA Technologies	Advanced Authentication	Authentication	Available by request	ca.com/us/products/ca-advanced-authentication.html
CA Technologies	Veracode Cloud Platform	SAST, DAST, mobile AST, penetration testing	Demo available by request	veracode.com/products/application-security-platform
Carbon Black	Cb Protection	Enpoint security	Demo available by request	carbonblack.com/products/cb-protection
Cavirin	Cavirin Platform	Cloud security scanning	Demo available by request	cavirin.com/cavirin-platform.html
CDNetworks	DDoS Protection	CDN, WAF, DDoS Protection	N/A	cdnetworks.com/us/en/products/ddos-protection
Check Point Software	vSec for Public and Private Cloud Security	Infrastructure, WAN, PaaS security	N/A	checkpoint.com/products-solutions/vsec-cloud-security

COMPANY	PRODUCT	CATEGORY	FREE TRIAL	WEBSITE
Checkmarx	Checkmarx Static Code Analysis	SAST	Demo available by request	checkmarx.com/technology/static-code-analysis-sca
CipherCloud	CipherCloud	Cloud access security broker	Demo available by request	ciphercloud.com
CIRT.net	Nikto2	Web server scanner	Open source	cirt.net/Nikto2
Cisco	Cisco ACE WAF	WAF	N/A	cisco.com/c/en/us/products/collateral/application-networking-services/ace-web-application-firewall/data_sheet_c78-458627.html
Cisco	CloudLock Security Fabric	Cloud access security broker	Demo available by request	cloudlock.com/platform
Citrix	NetScaler AppFirewall	WAF	90 days	citrix.com/products/netscaler-appfirewall
CloudCoreo	CloudCoreo Platform	DevOps security automation platform	N/A	cloudcoreo.com
Cloudfare, Inc.	CloudFlare	CDN, DDoS protection, WAF	Free tier available	cloudflare.com
Cloudmark	Cloudmark Insight Server	Real-time threat detection	Available by request	cloudmark.com/en/s/products/cloudmark-insight-server
CloudPassage	CloudPassage Halo	Cloud access security broker	Available by request	cloudpassage.com/why-halo
Code42 Software	Code42	Endpoint security	Available by request	code42.com
Contrast Security	Contrast Enterprise	Automated IAST, RASP	Available by request	contrastsecurity.com/overview
Core Security	Core Impact	Penetration testing	N/A	coresecurity.com/core-impact
Core Security	Actionable Insight & Response Platform	Security monitoring	N/A	coresecurity.com/actionable-insight-platform
CounterTack	Endpoint Threat Platform	Malware detection, endpoint security	Demo available by request	countertack.com/endpoint-threat-platform
CounterTack	Threat Scan Pro	Endpoint scanning	Demo available by request	countertack.com/threatscan-pro
Covata	Covata Platform	RBAC, cloud security	Demo available by request	covata.com/solutions/covata-platform
CrowdStrike	Falcon	Endpoint security	Available by request	crowdstrike.com/products
CyberArk Software	CyberArk	Authentication	Demo available by request	cyberark.com
Cybereason	Deep Hunting Platform	Endpoint security	Demo available by request	cybereason.com/hunting-platform
Cylance	CylancePROTECT	AI-based endpoint security	Demo available by request	cylance.com/en_us/products/our-products/protect.html

COMPANY	PRODUCT	CATEGORY	FREE TRIAL	WEBSITE
Darktrace	Enterprise Immune System	AI-based endpoint security	Demo available by request	darktrace.com/technology/#enterprise-immune-system
Datiphy	Datiphy Platform	Real-time threat detection	Demo available by request	datiphy.com/products/enterprise
Deepfield (now Nokia)	Deepfield Defender	DDoS protection	Demo available by request	networks.nokia.com/solutions/deepfield-ip-network-analytics-DDoS-protection
DenyAll	DenyAll WAF	WAF	N/A	denyall.com/products/web-application-firewal
Digital Guardian	Digital Guardian Platform	Enpoint security, threat detection	N/A	digitalguardian.com/products/threat-aware-data-protection-platform
DXC Technology	DXC Security	Endpoint, application, and data security	Demo available by request	dxc.technology/security
Elastic Beam	Elastic Beam API Behavioral Security	API transaction attack detection and compliance reporting	Available by request	elasticbeam.com/product/#api-behavioral
Endgame	Engame Platform	Endpoint security, security automation	Demo available by request	endgame.com/platform
Ergon Informatik AG	Airlock Suite	WAF, authentication, identity management	Demo available by request	airlock.com/en/home
ESET	ESET Endpoint Protection	Endpoint security	Demo available by request	eset.com/us/business/endpoint-protection
Evident.io	Evident Security Platform	Cloud security and compliance automation	Free risk assessment	evident.io/what-is-esp
F5 Networks	Herculon	DDoS protection, SSL orchestration	N/A	f5.com/products/herculon/ddos-hybrid-defender
F5 Networks Inc.	F5 Big-IP ADC platform	WAF, DDoS protection	N/A	f5.com/products/big-ip
Fidelis Cybersecurity	Fidelis Platform	Threat detection and monitoring, endpoint security	Demo available by request	fidelissecurity.com
FireEye	FireEye as a Service	Threat detection and forensics	N/A	fireeye.com/products/fireeye-as-a-service.html
FireEye	FireEye Helix	Endpoint and network security	N/A	fireeye.com/products/helix.html
Forcepoint	Forcepoint Next Generation Firewall	WAF	Demo available by request	forcepoint.com/product/network-security/forcepoint-ngfw
Fortinet	Next-Generation Firewall	WAF	Available by request	fortinet.com/products-services/products/firewall.html
Fortinet	FortiClient Next-Generation Endpoint Security	Endpoint security	Demo available by request	fortinet.com/products/endpoint-security/forticlient.html
Forum Systems	Forum Sentry	Authentication, API Security Gateway	Demo available by request	forumsys.com/en/products/forum-sentry-api-security-gateway
Gemalto	Authentification Management Platforms	Authentication	Demo available by request	gemalto.com/enterprise-security/identity-access-management

COMPANY	PRODUCT	CATEGORY	FREE TRIAL	WEBSITE
Gigamon	GigaSECURE	Network security, load balancer	N/A	gigamon.com/products/core-offerings/visibility-platform/gigasecure-security-delivery-platform.html
GrammaTech	CodeSonar	SAST for IoT	30 days	grammatech.com/products/codesonar
Hillstone Networks	Intelligent Next-Gen T-Series Firewall	WAF	N/A	hillstonenet.com/our-products/intelligent-next-gen-firewalls-t-series
HP Enterprise (now Micro Focus)	Fortify Static Code Analyzer	SAST, DAST, IAST, RASP	Available by request	software.microfocus.com/ko-kr/software/sca
IBM	Security AppScan by IBM	SAST, DAST, IAST	Available by request	www-03.ibm.com/software/products/en/appscan
IBM Bluemix	Bluebox	Mobile access security broker	Demo available by request	ibm.com/cloud-computing/bluemix/bluemix-private-cloud
iBoss	iBoss	WAF	Demo available by request	iboss.com/why-iboss
Imperva	Imperva Incapsula	WAF, DDoS protection	Available by request	incapsula.com
Infoblox	InfoBlox DNS Firewall	WAF	60 days	infoblox.com/products/dns-firewall
Intel	McAfee Small Business Security	Antivirus and monitoring	30 days	mcafee.com/consumer/en-us/store/m0/index.html
Ixia	Breakingpoint VE	Penetration testing	Demo available by request	ixiacom.com/products/breakingpoint-ve
Janrain	Social Login	Identity management and authentication	Free tier available	janrain.com/product
Juniper Networks	SRX Series Firewall	WAF	N/A	juniper.net/us/en/products-services/security/srx-series
Kali Linux	Kali Linux	Penetration testing	Open source	kali.org
Lastline	Lastline Breach Defender	Network security and monitoring	Demo available by request	lastline.com/products/defender/
Level 3	Level 3 Content Delivery Network	CDN, DDoS protection	N/A	level3.com/en/products/content-delivery-network
Lieberman Software	Lieberman RED Suite	Identity management and authentication	Demo available by request	liebsoft.com/red
LogRhythm	LogRhythm Security Intelligence Platform	Predictive security analytics	Demo available by request	logrhythm.com/products/security-intelligence-platform
Malwarebytes	Malwarebytes Endpoint Security	Endpoint Security	N/A	malwarebytes.com/business/endpointsecurity
Metaflows	Metaflows Malware Detection Software	Cloud security scanning	Available by request	metaflows.com/the-metaflows-security-system/malware-detection-software

COMPANY	PRODUCT	CATEGORY	FREE TRIAL	WEBSITE
Microsoft	Cloud App Security	Cloud access security broker	Demo available by request	microsoft.com/en-us/cloud-platform/cloud-app-security
N-Stalker	N-Stalker Web Application Security Scanner X	SAST, DAST	Free tier available	nstalker.com/products/editions
Neustar, Inc.	Neustar	DDoS Protection	N/A	neustar.biz
NGINX (Sponsor)	NGINX Plus	Authentication, DDoS Protection	30 days	nginx.com/products
NMAP.org	Nmap	Penetration Testing and Network Mapping	Open source	nmap.org
NowSecure	NowSecure	Penetration testing, compliance, security scanning services	N/A	nowsecure.com
NSFOCUS	NSFOCUS ADS	DDoS protection	N/A	nsfocusglobal.com/products-overview
NSS Labs	CAWS Continuous Security Validation Platform	Security monitoring	30 days	nsslabs.com/caws-continuous-security-validation-platform/overview
Okta	Okta Platform	Authentication	Free tier available	okta.com/products/developer
OPSWAT	Metadefender	SAST, endpoint security	Available by request	opswat.com/metadefender-core
OWASP	OWASP Projects	Various open source security projects	Open source	owasp.org/index.php/Category:OWASP_Project
Palo Alto Networks, Inc.	PA-7000 Series Firewall by Palo Alto Networks	WAF	N/A	paloaltonetworks.com/products/secure-the-network/next-generation-firewall/pa-7000-series
Palo Alto Networks, Inc.	Palo Alto Enterprise Security Platform	RASP WAF	Available by request	paloaltonetworks.com/products/designing-for-prevention/security-platform
Peach Technology	Peach Fuzzer	Automated penetration testing	Available by request	peach.tech/products/peach-fuzzer
Peach Technology	Peach API Security	Automated security scanning	Available by request	peach.tech/products/peach-api-security
PhishMe	PhishMe Simulator	Phishing simulation software	Demo available by request	phishme.com/product-services/simulator-2
PhishMe	PhishMe Intelligence	Threat detection and analytics	Demo available by request	phishme.com/product-services/phishing-intelligence
PortSwigger LLC	Burp Suite	SAST, DAST, penetration testing	Free tier available	portswigger.net/burp
Pradeo	AuditMyApps	Mobile AST	Available by request	pradeo.com/en-US/application-security-testing
Prevoty	Prevoty RASP	RASP	Demo available by request	prevoty.com/products

COMPANY	PRODUCT	CATEGORY	FREE TRIAL	WEBSITE
Protectwise	ProtectWise Grid	App security scanning	Demo available by request	protectwise.com/platform.html
Qualys, Inc.	Qualys Cloud Platform	DAST, WAF, monitoring	Available by request	qualys.com/cloud-platform
Radware	AppWall	WAF, DDoS protection	Available by request	radware.com/products/appwall
Radware	Alteon VA for Developers	Load balancing, DDoS protection	Free solution	radware.com/Resources/SoftwareDownloads
Rapid7	AppSpider Pro	DAST	Demo available by request	rapid7.com/products/appspider
Rapid7	Metasploit	Penetration Testing	Open source	rapid7.com/products/metasploit
Rogue Wave Software	Klocwork	Code quality scanning	Available by request	klocwork.com
RSA	RSA SecurID Suite	Identity management	Available by request	rsa.com/en-us/products/rsa-securid-suite
RSA	Threat Detection and Response	DAST	Available by request	rsa.com/en-us/products/threat-detection-and-response
Security Compass	SD Elements	Secure coding platform	Available by request	securitycompass.com/sdelements/how-it-works/#!/step/1
Security Innovation	CMD+CTRL Hackathons	Training hackathons	N/A	securityinnovation.com/training/hackathon
Securonix	Securonix Security Analytics Platform	Real-time threat detection	Demo available by request	securonix.com/security-intelligence
SentinelOne	SentinelOne	Endpoint security, DAST	Demo available by request	sentinelone.com/platform/
ServiceNow	IT Service Management	Security monitoring	Demo available by request	servicenow.com/products/it-service-management.html
Signal Sciences	Signal Sciences Dashboard	RASP, WAF, real-time threat detection	Demo available by request	signalsciences.com/product
SiteLock	SiteLock TrueCode SAST	SAST, DAST	Available by request	sitelock.com/truecode.php
SonicWall	SonicWall Cyber Security Products	WAF	N/A	sonicwall.com/en-us/products
Sophos	Sophos XG Firewall	WAF	30 days	sophos.com/en-us/products/next-gen-firewall.aspx
Splunk	Splunk Enterprise Security	Security monitoring and analytics	Available by request	splunk.com/en_us/products/premium-solutions/splunk-enterprise-security.html
StackPath	StackPath	WAF, DDoS protection	15 days	stackpath.com
Sucuri	Sucuri Website Firewall	WAF, DDoS protection, app security scanning	Available by request	sucuri.net/website-firewall
Symantec	Cloud Access Security Broker	Cloud security testing and scanning	Free risk assessment	symantec.com/products/cloud-application-security-cloudsoc

COMPANY	PRODUCT	CATEGORY	FREE TRIAL	WEBSITE
Symantec Corporation	Symantec Advanced Threat Protection	IAST, RASP	60 days	symantec.com/products/threat-protection/advanced-threat-protection
Synopsys	Synopsys Static Analysis	SAST	On Request	synopsys.com/software-integrity/products/static-code-analysis.html
Tanium	Tanium Endpoint Platform	Endpoint security	Demo available by request	tanium.com/products
Tenable	SecurityCenter	Security monitoring and analytics, DAST	60 days	tenable.com/products
ThinAir	ThinAir	Security monitoring and analytics	Demo available by request	thinair.com/product
Trend Micro	Trend Micro Hybrid Cloud Security	SAST, DAST	N/A	trendmicro.com/en_us/business/products/hybrid-cloud.html
Tripwire, Inc.	Tripwire Enterprise	IAST, RASP	Demo available by request	tripwire.com/it-security-software/scm/tripwire-enterprise
Trustwave	ModSecurity	WAF	Open source	modsecurity.org
Trustwave	Trustwave Secure Web Gateway	CDN, DAST	N/A	trustwave.com/Products/Content-Security/Secure-Web-Gateway
Trustwave	Trustwave Web Application Firewall	WAF, penetration testing	N/A	trustwave.com/Products/Application-Security/Web-Application-Firewall
UpGuard	Discover	Penetration testing, analytics	Demo available by request	upguard.com/discover
Varonis	Datavantage	Insider threat detection/prevention	Available by request	varonis.com/products/datadvantage
Venafi	Venafi Platform	Device security identity management	Available by request	venafi.com/platform
Vera	Vera Platform	Information rights managament	Available by request	vera.com/product/sdk-api
Virtual Forge GmbH	CodeProfiler for ABAP	Code quality scanning	Available by request	virtualforge.com/en/portfolio/codeprofiler.html
Webroot	BrightCloud Threat Intelligence by Webroot	DAST, endpoint security	Available by request	webroot.com/us/en/business/threat-intelligence
WhiteHat Security	WhiteHat Sentinel	DAST	30 days	whitehatsec.com/products/dynamic-application-security-testing
WhiteHat Security	WhiteHat Sentinel	SAST	30 days	whitehatsec.com/products/static-application-security-testing
WhiteSource	Open source Security	Open source scanning	Available by request	whitesourcesoftware.com/open-source-security
Wireshark Foundation	Wireshark	Penetration testing and packet-level Monitoring	Open source	wireshark.org
Ziften	Ziften Zenith	Endpoint security	30 days	ziften.com/product-overview
Zscaler	Zscaler	Cloud security	Demo available by request	zscaler.com/products/zscaler-overview

GLOSSARY

APP TRANSPORT SECURITY (ATS):
A feature in the iOS operating system that enforces a minimum security level for communications (HTTPS vs. HTTP) between a mobile app and web services that support its functionality — enabled by default in SDKs for iOS 9.0 or later.

AUTHENTICATION:
A mechanism that confirms a user's identity when they are requesting access to a resource in a system. This is generally handled by granting users an access token when they confirm their identity through a mechanism such as a password.

BRUTE-FORCE ATTACK:
An attack that uses software to automate the guessing of numerous values very quickly in order to identify the correct value that will grant access to a protected resource.

CROSS-SITE REQUEST FORGERY (CSRF):
A malicious web exploit in which an attacking program forces a user's browser to perform an unwanted action on a site where the user is currently authenticated.

DATA EXFILTRATION:
An unauthorized transfer of data. It can be carried out manually or through a malicious automated program.

DENIAL OF SERVICE ATTACK (DDOS):
A type of attack that uses multiple compromised systems that are forced to visit a website or system and overload its bandwidth in order to cause an outage.

DEVSECOPS:
the integration of security into the DevOps methodology.

DYNAMIC APPLICATIONS SECURITY TESTING (DAST):
An analysis of an application's security that only monitors the runtime environment and the code that is executed in it. It simulates potential attacks and analyzes the results.

INJECTION ATTACK:
A scenario where attackers relay malicious code through an application to another system for malicious manipulation of the application. These attacks can target an operating system via system calls, external programs via shell commands, or databases via query language (SQL) injection.

INTERACTIVE APPLICATION SECURITY TESTING (IAST):
A combination of SAST and DAST that is usually implemented in the form of an agent that monitors attacks and identifies vulnerabilities within the test runtime environment.

KEYCHAIN:
Within the iOS operating system, a keychain encrypts and securely stores data used by apps and services. When the keychain is unlocked via a password, only trusted applications can read that data.

MAN-IN-THE-MIDDLE ATTACK:
An attack whereby an adversary inserts themselves between an application and its backend services to eavesdrop on the communications between them.

PASSWORD-BASED KEY DERIVATION FUNCTION (PBKDF2):
An algorithm that makes passwords stronger by encrypting a password along with a "salt" value (i.e. random data) numerous times to make the password harder to guess.

RUNTIME APPLICATION SELF-PROTECTION (RASP):
A feature that is built into an application in order to detect and halt attacks in real-time, automatically.

RUNTIME PACKER:
A tool that compresses software that will execute once the compressed file is opened. This is frequently used as a way to distribute malware.

SECURE SOCKETS LAYER (SSL):
An encrypted link that serves means to keep information that is passed between the web server and browsers private.

STATIC APPLICATION SECURITY TESTING (SAST):
An analysis of an application's security that looks at an application's source code, bytecode, or binary code to determine if there are parts that could allow security exploits by attackers.

WEB APPLICATION FIREWALL (WAF):
An appliance or application that monitors, filters, and blocks HTTP transmissions to a website based on customizable rules.

INTRODUCING THE

AI Zone

What's new with Predictive Analytics, Natural Language Processing, and Neural Nets? Check out DZone's new AI Zone – a central place for AI resources and tutorials.

Keep a pulse on the industry and go beyond the hype. This Zone gives you access to practical applications and use cases for AI technologies like: Machine Learning, Cognitive Computing, and Chatbots.

Visit the Zone

MACHINE LEARNING

COGNITIVE COMPUTING

CHATBOTS

DEEP LEARNING