wallarm

A CISO's GUIDE TO CLOUD APPLICATION SECURITY

Protect your enterprise from the everyday risks in applications introduced by the rapid pace of DevOps

The following guidelines will help senior stakeholders set strategy to secure modern applications, learning:

- The new challenges enterprise security faces;
- How to assess risks in your organization and determine exposure;
- Technology changes that introduce application insecurity;
- Organizational responses for addressing application security.

CONTENTS AND SECTION BRIEFS

EXECUTIVE SUMMARY

Applications are the operational mechanism for how a modern enterprise conducts transactions and uses data. Whether internal or customer-facing, apps are critical for your successful business operations. That means securing apps should be a business priority.

It was simpler to ensure the security and availability of applications when IT physically resided on premise. The rise in use of cloud, mobile, and wireless and the rapid pace of application DevOps has complicated app security. Application security touches many aspects of the enterprise. Strategic planning is fundamental to aligning the organization's application security architecture with its new generation cloud strategy — and its risk tolerance.

This guide provides strategy considerations for senior stakeholders securing modern applications. It provides related advice for risk assessment, technologies that affect the app security equation, and organizational issues that will provide a path to effectively help your enterprise protect apps fueling the lifeblood of IT operations.

NEW CHALLENGES FOR ENTERPRISE SECURITY

There are two fundamental issues spawning new challenges for enterprise security. First, there is an imperative for speed in developing new applications. Alongside it isr is the necessary transformation in how apps are developed.

Speed stems from business urgency. There is competitive need to offer new digital capabilities by pushing out location application releases as quickly as possible. These newly arising developments include:

- Launching innovative apps that address new business initiatives;
- Adding new features to existing apps that improve user experience;
- Enhancing internal application features;
- Enhancing technology and support;
- Supporting “behind-the-curtain” requirements towards scalability, performance, and compliance.

Development speed is powerful and potentially essential to:

- Quickly meet customer needs and expectations;
- Reduce time-to-market;
- Provide a first-mover advantage over competitors;
- Capture new or augmented revenues;
- Help a business improve the perception of being a leading-edge brand intensely focused on customer satisfaction.

The term DevOps refers to the new hyper-short development and deployment cycle (CI/CD). What used to require months now takes only weeks, days, or a few hours. DevOps promises to boost the productivity of development teams and reduce app delivery costs. The downside of this increased speed is it leaves less time to ensure applications are secure. Implementing app security traditionally presumes there are security professionals involved in the development process who have deep knowledge of coding best practices and mastery of testing regimes and tools for static, dynamic, mobile, and interactive application security and penetration testing.

The new DevOps world has crowded out the traditional application security. Tasks in securing an application have lost both the allotted time for security and the security team in charge. Instead, it increasingly falls onto developer responsibility to ensure the application is secure before it is deployed. But developers are unequipped to handle formerly dedicated security methods and tasks.

Hardly any developer is master of all security requirements — it is hard enough to hire developers who are good coders, let alone possessing the skills of a security expert. In addition to a lack of time for security implementation, this knowledge gap and lack of tried experience is a major vulnerability for ensuring the creation and maintenance of secure apps.

What is the solution? Security needs to be addressed from top-down. Security executives can make security an enabler of DevOps and boost the speed of development through the right security strategy combined with incorporating the right processes and technologies for the new environment.

The following sections address strategic considerations for these issues from three perspectives: risk assessment, technology, and organization.

RISK ASSESSMENT

Setting a baseline: Catalog applications

The first step in reviewing an organization's strategy for application security is to assess problem scope. Your IT asset management program mostly likely tracks all physical assets and software. Cataloging is essential to create a baseline for vulnerability assessment and management. However, legacy practices may miss two new aspects of assessing internet and internet-facing app security — risks caused by distributed apps and containerized apps.

Distributed apps are vulnerable when a shared service, such as an enterprise database, is hosted on multiple physical or virtual servers, each with its own IP address. Security mappings between servers and apps used to be easy. Now, the single database is accessed by a variety of trusted and untrusted users. For example, some tables may be used by employees and contractors. Suppliers and third-party contractors may also use those, or other, tables. Customers, prospective customers, and web shoppers may access additional tables in the database.

Your organization's degree of exposure depends on the various levels of risk posed by these types of scenarios. Assessing that risk entails cataloging the applications, determining how they are used, and identifying their points of access as well as possible vulnerabilities.

Security mapping is more difficult with the containerized apps central to the modern DevOps process. Containers are fully functioning code modules that spin up and down as virtual systems and apps specify the unpredictable processes that sometimes occur in only a few seconds and may be required to process workload increases.

Risk is Real: What to Learn from Big Breaches

Major data breaches are becoming more and more common — and application exploits are the greatest risk area for breaches. Depending on who gets hacked, compromised records number in the tens or hundreds of millions, or more. We will skip reviewing the much-described regulatory penalties and civil damages resulting from the disclosure of personally identifiable data, financial records, or personal health information. Equally disastrous is loss of corporate intellectual property, business interruption, reputation damage, or targeted attacks by hacker groups or rogue governments. The focus of this paper is on vulnerable apps.

New, distributed and containerized apps are not the only threat vectors. Some of the biggest breaches on record have occurred through exploits of old app vulnerabilities that are well known to security professionals — and often given short shrift by enterprise policies that miss the strategic importance of application security. Web applications, in general, are the biggest vector of reported breaches, according to the *Verizon 2019 Data Breach Investigation Report* (p. 10).

In the infamous Equifax breach of 2017, 143 million U.S. consumer records were lost, including financial history and social security numbers. A web application or code execution vulnerability gave hackers access to the data — access that remained undetected for two months. This breach and evasion suggests the

Equifax cybersecurity team was not using proactive security filters like a sufficiently advanced Web Application Firewall, or WAF. Nor were they patching application frameworks in a timely manner. Proper security measures and

solutions would have stopped attackers from maintaining an undetected intrusion for any length of time.

Your application security strategy should address the integrated effectiveness of your network, host, and data security architecture by having:

- Host layer protected by HIPS / HIDS and AV;
- Network layer (OSI L4) with the following controls: firewalls, network-based IDS / IPS;
- Application layer (OSI L7) protected by an adaptive WAF, behavioral attack protection tools, and a combination of a vulnerability scanner and a bug bounty program;
- Data protection layer includes data partitioning, encryption, and access controls;
- Operational processes (DevOps and SecOps) should be in place and tested regularly for the team to act on the information received from all security tools.

The Right Tools to Help Compliance

Compliance is a major area of executive responsibility, particularly in highly regulated industries such as payments, finance, and healthcare. Regarding app security, many of the industry compliance standards — such as HIPAA, SOX, PCI, NIST, GDPR, and others — call for best practices in securing the application stack, including transmitted data and the application logic. Run-time application security is key to complying with such requirements. Non-compliance can trigger costly fines, penalties, and other complications. In the payments industry, for example, the associated loss of a certification like as “PCI compliant” could elevate exposure to civil lawsuits if an entity is breached due to faulty application security controls.

Some of the app security compliance requirements are more general and require interpretation by security auditors. Others are quite specific. For example, the PCI Data Security Standard requirement 3.2.6.6 specifies a scenario which is addressed by a specific technology — a web application firewall. It mandates you “Inspect any protocol (proprietary or standardized) or data construct (proprietary or standardized) that is used to transmit data to or from a web application, when such protocols or data are not otherwise inspected at another point in the message flow.”

Since the Wallarm WAF can inspect nested protocols within SPAs and APIs and understand the underlying data structure, it’s a natural choice to satisfy this requirement in payment, eCommerce, and financial applications without additional compensating controls.

TECHNOLOGY

Application Security in Cloud Migration

Cloud migration is a common enterprise initiative. Some have gone all-in with a 100% adoption of cloud infrastructure for apps and services. Others are incrementally converting pieces of enterprise operations into the cloud. Methods of implementing cloud migration vary from 100% use of a public cloud to hybrid cloud combinations of public and private to only using a private cloud. Many organizations elect to outsource by using a SaaS service(s) for certain app functions. Whatever your approach, your enterprise application security strategy needs to address how apps and data are secured in the new environment.

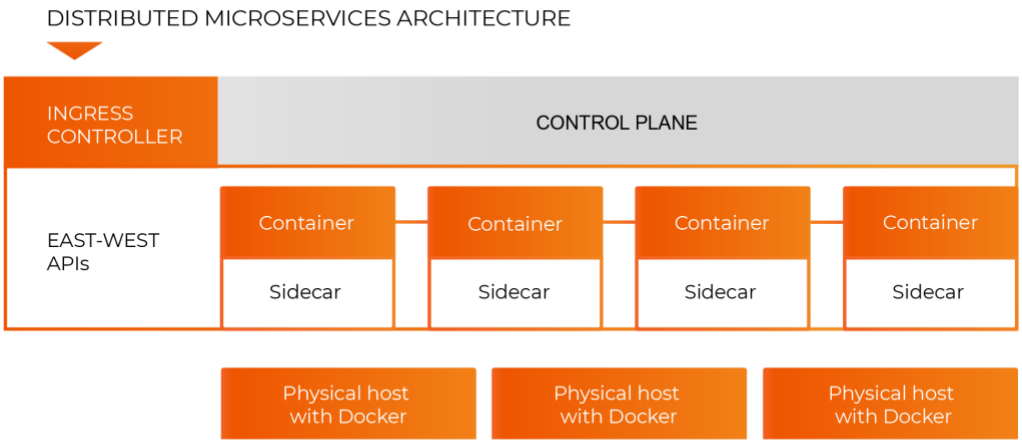
The essence of app security in the cloud hinges on a shared responsibility model. Cloud service providers are responsible only for securing the infrastructure hosting their cloud and services. Securing the apps and data are still the responsibility of the enterprise using that cloud infrastructure. Never assume apps and data are safe just because they're "in the cloud!".

Beware a false sense of app security in using an outsourced SaaS service or a Cloud Access Security Broker (CASB) to access cloud services. As your organization catalogs apps used with these technologies, the security team should also note associated weak points that must be addressed by your enterprise strategy. Reliance on third-party APIs is a major vulnerability; security controls for the APIs must match the risk profile of each app. API and other cloud risks are described in the next section, *Application Architecture*.

Application Architecture: Protection from the Ground Up

As noted, many new issues with application security arise when changing from monolithic or even a client-server to a distributed microservices-driven application architecture. Within this new framework, several new elements come to life. How they interact may cause new attack vectors and security issues.

In the legacy architecture, the application was only accessible by a client or via client APIs — also known as "North-South." By protecting these APIs on the network and the application layer, the IT organization could be fairly sure of its security.



The landscape changed when applications became distributed — whether or not they use a public cloud. Let us consider a generic distribute distributed architecture in the diagram below:

DISTRIBUTED MICROSERVICES ARCHITECTURE

Exploring issues from the physical layer up — physical host and operating system security — remains important unless it has been taken care of by your public cloud provider.

However, in the pervasive Docker-based architecture, another system layer that can potentially become a target of an attack is the Docker system itself. There are many known vulnerabilities in this layer, found just last year, that would make a savvy executive pause and take stock of the versioning and mitigating controls.

Docker also poses “escape” issues when problematic code within a container is able to affect neighboring containers or the host. It’s worth noting that to use a container as a microservice, it needs to be so-defined. Such definition typically comes in the form of an API and a configuration policy. API is how system modules and services are accessed. Visibility, security, and integrity of the APIs are what makes distributed applications effective and reliable.

Moving up the stack, in microservices, containers are able to discover other containers in the pod and communicate to them directly. Since the premise of the architecture is container reuse, without thorough security testing of each container in the system and each communication channel within the East-West mesh, critical data and services may be vulnerable. Similarly, limiting access points and privileges of each of the microservices in this configuration is paramount.

In distributed architectures like Kubernetes, there is a control plane that decides if additional instances of specific services need to be spinned up. Attacks on this layer can not only cause a denial of service, but also can potentially pull malicious containers out of repositories or distort the operation of the valid containers with faulty configuration parameters.

Last, but probably most important, is the information that enters the cluster, as governed by the Ingress controller. This is the module that decides what APIs are allowed and which services within the pod may process this information. Real-time security controls that are compatible with the API protocols in use in this layer are extremely important — so is the speed of processing. Scalability and fast processing are why distributed architectures have become popular.

Updating Real Attack Protection

Legacy solutions are likely insufficient for new kinds of attacks. The app security strategy should address how effective existing controls are against new modes of cyberattacking. Often, a solution that provides parity with an older security technology is still required, but with new capabilities added to address current exploits.

For example, an enterprise-grade web application firewall (WAF) remains crucial for instant protection against SQL injection, cross-site scripting, illegal resource access, remote code execution, remote file

inclusion, and other OWASP Top-10 threats. A new and essential refinement is granular blocking on the API level to minimize performance impact on legitimate traffic. Wallarm implements this capability with adaptive security rules defined and refined by machine learning, including techniques like supervised learning, neural networks, and reinforcement learning. Machine learning-generated profiles allow Wallarm to detect anomalies in application requests or payloads and automatically flag them. These rules evolve with app changes to ensure continuous security.

Another new risk vector that cannot be covered by legacy defenses is the tide of bots pervading the internet. An internet bot, or web robot, is a software app that runs automated tasks or scripts over the internet. About half of all internet traffic consists of bots; about half of them are “good” bots such as Google’s indexing websites and the other half are “bad” bots used for malicious probes and attacks. A typical botnet credentials attack can include as many as 25,000 to 100,000 agents or bots. A successful attack can inflict material damage on the affected company or service.

One of the more frequent uses of botnet attacks is credentials stuffing, which attempts to reuse credentials stolen from other services to access the application. While the fundamentals of credentials stuffing are well known to security and operational professionals, stopping it is not easy. Good enterprise app security strategy has to address bad bots, which requires a lightning-fast automatic response to an attack. Legacy defenses are unable to address this scenario.

ORGANIZATION

Security Roles: Shifting Left onto Developers in DevOps

Corporate restructuring occurs frequently to address changing requirements in the enterprise. The dramatically new way in which apps are produced by DevOps is a good reason to look at your organization's role structure regarding application security. Legacy roles like "developer" and "security professional" are too siloed for DevOps and often miss vulnerabilities intrinsic to rapid development cycles where new versions of apps are published on a daily or hourly basis.

The DevOps process is so fast that security is "shifting left," which primarily falls on developers' shoulders. Most developers have little to no expertise in the complexity of application security. Organizations are responding to this dilemma by changing the security role's function from being a gatekeeper approving every code iteration of each app to a "guard rail." The idea is to collaboratively address security throughout the DevOps process as a set of policies, recommendations, and safeguards. A nickname for this approach is DevSecOps, often headed by a "Dev Security Champion" on the DevOps team.

This pit-team approach includes the use of security automation and toolkits by DevOps who need this functionality due to lacking bandwidth or inclination in becoming security experts. From a practical perspective, this approach helps to ensure app security in ways that are incapable by legacy roles.

Automation: A Necessary Security Measure

Automation is an essential part of any comprehensive app security strategy. Automation can increase the pace of DevOps processes and simultaneously boost the safety of the resulting apps via more secure code and stronger defenses that act quickly against attacks. Automation is an evolving theme in the world of DevSecOps for which there are no defined best practices. It is very important for an app security strategy to include automation and use it to gain an edge over new and evolving threats.

Automation has long been used to ensure software quality and functional completeness. For example, many organizations are implementing a form of test automation with Selenium, which is a portable framework for testing web apps. Selenium is open source software so web developers can download and use it without charge. The Wallarm platform enables security test inside DevSecOps by building security testing into the CI/CD pipeline.

Wallarm FAST generates security tests from existing functional tests. Security tests are then executed, alongside the functional tests within CI/CD pipelines without needing to change development processes. Another useful feature of the platform is the Wallarm scanner for use in conjunction with Wallarm Advanced WAF for applications that have already being deployed. Wallarm Scanner automatically discovers application-specific issues and actively verifies threats to find high-risk incidents among a large number of irrelevant attacks.

Penetration Testing & Bug Bounty Programs are Still Important

Penetration testing is an authorized simulated attack on apps aimed to discover vulnerabilities and strengths for overall risk assessment. The "pen test" typically accompanies a security audit, such as for compliance with the PCI DSS. As described by the National Cyber Security Center, penetration testing is: "A method for gaining assurance in the security of an IT system by attempting to breach some or all of that system's security, using the same tools and techniques as an adversary might."

One criticism of pen testing is its static nature. That means the results of a test are only good for the moment it was conducted. As application code and application-specific threats are evolving on a continuous basis, the pen test is somewhat limited. But it's still an important aspect to app security, especially when required for regulatory compliance.

One valuable angle to this requirement is a bug bounty program, which is a crowd-sourced version of pen testing. A bug bounty is an incentivized invitation to expert hackers and security professionals to break your app code. Public platforms for conducting a bug bounty program include bugcrowd and hackerone.

Your app security strategy should consider using bug bounty as an additional layer of best practices providing specific utility.

OVERVIEW: A Checklist for Modern Application Security

The thrust of DevOps for enterprise applications brings many benefits to a modern business. We've learned the associated switch to a distributed microservices-driven application architecture also brings new risks. Many of these are unaddressed by legacy security solutions and best practices.

Wallarm encourages your organization to use our checklist as a guide for evaluating application security strategy. The checklist provides a pallet of the most urgent areas for assessment. By addressing checklist issues first, your organization can take big strides toward stronger app security in today's challenging environment. To learn more or request help, please contact Wallarm at info@wallarm.com or visit our site at www.wallarm.com.

To sum up the guidelines for best application security practices, here are quick bullet points to checkover.

1. PERFORM A RISK ASSESSMENT

a. Catalog Applications to set a baseline

- ❑ Maintain regular, comprehensive IT risk assessments;
- ❑ Do thorough security mapping to keep up with new development cycles and challenges caused by any changes to infrastructure and processes;
- ❑ Check that risks caused by distributed and containerized apps are included in IT risk assessments.

b. Assess business risk and app security

- ❑ Put continually proactive and advanced security solutions and practices in place, such as sufficiently-advanced, automated WAF and timely patching.
- ❑ Check that application security strategy looks at the integrated effectiveness of network, host, and data security architecture;
- ❑ Plan for targeted protection at each layer:
    - ❑ **Host layer**: HIPS/HIDS and AV;

- ❏ **Network layer (OSI L4)**: firewalls and network-based IDS/IPS;
- ❏ **Application layer (OSI L7)**: adaptive WAF, behavioral-attack protection tools, and a combination of a vulnerability scanner and a bug bounty program;
- ❏ **Data protection layer**: data partitioning, encryption, and access controls;

❏ Ensure operational processes (DevOps and SecOps) are in place and tested regularly for the team to act on the information received from all security tools.

c. Ensure compliance

❏ Check all compliance requirements affecting your sector, which may require specific solutions, like a WAF;

❏ Make sure securing your application stack includes run-time application security;

❏ Limit access points and privileges of each of the microservices;

❏ Ensure real-time security controls are compatible with API protocols and both are being used;

❏ Maintain high processing speed.

2. GET THE RIGHT TECHNOLOGY ALIGNED

a. Protect cloud environments

❏ Plan for cloud migration, assessing what security holes or structural changes will come;

❏ Check that your security strategy is aligned to any changes of your environment, such as cloud or hybrid migrations;

❏ Have safeguards in place for new security challenges instead of assuming cloud providers will provide security.

b. Ensure application architecture is not vulnerable

- ❏ Remember Docker-based architecture is vulnerable to Docker system attacks;
- ❏ Test each container in the system and each communication channel within the East-West mesh to protect critical data and services.

c. Prepare for new attacks with the right security tools

- ❏ Supplement, update, and/or replace legacy solutions that cannot address new types of attacks and vulnerabilities;
- ❏ Keep security solutions updated and look into new technologies that respond to changes to your total environment.

3. UPDATE YOUR ORGANIZATION

a. Update and arm security roles

- ❏ Check to see if DevOps has left your development team with responsibilities they are unequipped to handle;
- ❏ Check to see that security is being implemented and look into the effect of CI/CD-derived pressures or obstacles;
- ❏ Assess whether security solutions work inside your pipeline and are easily used as part of your total DevOps toolchain;
- ❏ Look at adopting a more pit-team approach with new functions and security responsibilities in DevOps and Security roles.

b. Automation

- ❏ Look for automated solutions that do not interrupt with your pipeline and help test software before going into production.

c. Penetration Testing

- ❏ Use pen testing in conjunction with dynamic testing;
- ❏ Consider a bug bounty program to keep testing up-to-date.

ABOUT WALLARM

Wallarm is an innovative AI startup focused on website, applications and API security. Wallarm's integrated adaptive approach provides better security and ease-of-use than the alternatives. The Wallarm platform includes adaptive Next-Gen WAF, a vulnerability scanner, incident verification, and development-time testing modules. Founded in 2013, Wallarm has already helped hundreds of SaaS and enterprise customers discover and fix critical vulnerabilities, automate web applications and API runtime protection, and prioritize security risks.

Wallarm is a privately held company headquartered in S. San Francisco, California, and backed by Y-combinator, Partech Ventures, and other investors.

https://wallarm.com
415 Brannan St.
San Francisco, CA 94107-1703
(415) 940-7077