


# Application Security 101

Security issues often arise as a result of applications being rushed for deployment without adequate checks and protections. What are the top security risks to applications and what can organizations do to secure their DevOps pipeline?

---

---

July 27, 2020

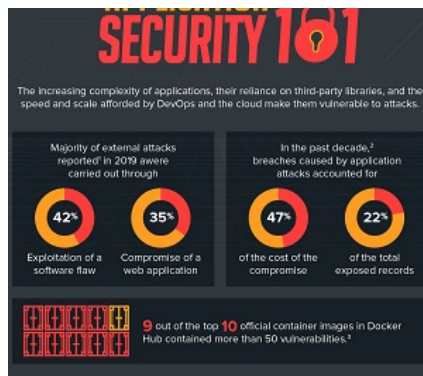
Digital transformation is an

APPLICATION

1 1 1 1

important step that organizations need to take to keep up with evolving industry landscapes. As the world currently grapples with the disruption brought about by the coronavirus pandemic, the need for such a transformation has become not only more apparent but also more urgent. With businesses pivoting their digital footprints and modernizing processes for their employees to work from anywhere, organizations are also having to reconsider how they meet customer demands and streamline change. This digital transformation has already become evident in the last few months, with the use of applications experiencing a **notable surge** across various sectors.

Applications now play an integral role, with many businesses and users relying on a wide range of applications for work, education, entertainment, retail, and other uses. In this current reality, development teams play a key role in ensuring that applications can provide users great usability and performance as well as security from threat actors who are always on the lookout for weaknesses, vulnerabilities, misconfigurations, and other security gaps that they can abuse to conduct malicious activities. Security risks have become even more pronounced as organizations have had to rush applications to market in order to maintain business and revenue-generating processes. The privacy risks posed by recently rolled out contact tracing applications best exemplify the perils of rushing application development and deployment. In May, The Washington Post **reported** that contact tracing applications, while useful for governments and researchers in their efforts to contain the pandemic



[View Infographic: Application Security 101](#)

## Related Posts

[How Residential Proxies and CAPTCHA-Solving Services Become Agents of Abuse](#)

[Rethinking Tactics: Annual Cybersecurity Roundup 2022](#)

[Trend Micro Security Predictions for 2023: Future/Tense](#)

[Uncovering Security Weak Spots in Industry 4.0 CNC Machines](#)

[Bridging Security Gaps in WFH and Hybrid Setups](#)

## Recent Posts

[How Cybercriminals Can Perform Virtual Kidnapping Scams Using AI Voice Cloning Tools and ChatGPT](#)

[How Residential Proxies and CAPTCHA-Solving Services](#)

outbreak, can inadvertently provide hackers sensitive details of people who tested positive for Covid-19.

The serious risks posed by unsecure applications highlight the need for **application security** or the process of finding, fixing, and enhancing the security of applications in the design, development, and post-deployment phase. This article discusses the security risks and threats that applications could be susceptible to, how organizations can integrate adequate cybersecurity protections in their **DevOps** pipeline, and more.

## Top security risks to applications

The increasing complexity of applications and their reliance on third-party libraries, among other concerns, make them vulnerable to security risks and threats. Security professionals revealed that majority of external attacks are carried out through exploiting a software vulnerability or a web application, as stated in a 2020 Forrester report. The same report describes open-source software as a main concern in the security of applications, citing the 50% increase of open-source security vulnerabilities since last year.

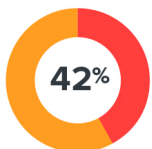
Become Agents of Abuse

Unmasking Pig-Butchering Scams and Protecting Your Financial Future

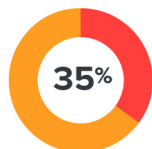
Ransomware Spotlight: TargetCompany

Email Threat Landscape Report: Cybercriminal Tactics, Techniques That Organizations Need to Know

Majority of external attacks reported in 2019 were carried out through

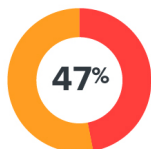


Exploitation of a software flaw

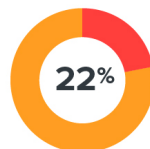


Compromise of a web application

In the past decade, breaches caused by application attacks accounted for



of the cost of the compromise



of the total exposed records

The increased adoption of containers and necessity of

APIs have also introduced new risks to applications. A **2020 Snyk report** reveals that nine out of the top 10 official container images in Docker Hub contained more than 50 vulnerabilities. Meanwhile, a **2019 F5 report** found API breaches that emerged from large platforms that offer many third-party integrations, mobile applications, and application misconfigurations.

The list below details the **most common risks** to applications that software developers should be mindful of in order to secure the code they produce. The Open Web Application Security Project (OWASP) Foundation has a comprehensive list of risks for **web applications** and **APIs**. It is important that developers are aware of the most common application security risks – ones that usually result from unsecure code – so they can check the bases they need to cover at each stage of the development pipeline.

- **Using components with known vulnerabilities.** Developers use components such as libraries, frameworks, and other software modules in their applications to avoid redundant work and provide needed functionality. However, threat actors look for known vulnerabilities in these components to erode application defenses and conduct various attacks.
- **Data leaks and exposure.** Web applications that do not properly protect sensitive data could allow threat actors to steal or modify weakly protected data. They could also conduct malicious activities such as credit card fraud and identity theft, among others. Improperly configured or badly coded APIs could also lead to a data breach.
- **Weak backend access controls.** Weak back-end access controls result from improperly enforced restrictions on what authenticated users are allowed to do. Threat actors can exploit these flaws to access unauthorized functionality, which include accessing other user accounts, viewing sensitive files, modifying other user data, and changing access rights.
- **Injection.** Flaws in or improper configuration of SQL.

NoSQL, OS, and LDAP can be abused in injection attacks, for example, when untrusted data is sent to a code interpreter through a form input or other data submission methods to a web application. Threat actors can use hostile data to trick the interpreter into executing malicious commands or providing unauthorized data access.

- **Security misconfiguration.** This is the most common concern for web applications. It occurs due to unsecure default configurations, misconfigured HTTP headers, incomplete or ad hoc configurations, open-cloud storage, and verbose error messages that contain sensitive information. Operating systems, libraries, frameworks, and applications should not only be securely configured to stay protected from threat actors, but also patched in a timely fashion.
- **Broken authentication and authorization.** When application functions concerning authentication and session management are implemented incorrectly, threat actors can abuse them to compromise passwords and keys or session tokens. In turn, threat actors can hijack user or admin accounts that could be used to compromise an entire system.
- **Cross-site scripting (XSS).** Threat actors can abuse XSS flaws to execute scripts in a browser and hijack user sessions, deface websites, or redirect the user to malicious sites. XSS flaws occur if an application includes untrusted data in a new webpage without proper validation or escaping. Such flaws could also occur if an application updates an existing webpage with user-supplied data through an HTML or JavaScript-creating browser API.
- **Unsecure deserialization.** This flaw, which is the improper conversion of serialized data back into objects that the application can use, often leads to remote code execution (RCE). This can also allow threat actors to perform replay, injection, and privilege escalation attacks.
- **Insufficient logging and monitoring.** Lack of capability in detecting threats could allow malicious actors to tamper, extract, or destroy data, as well as further attack systems, maintain persistence, and pivot to more systems.

# Integrating adequate cybersecurity layers in the DevOps pipeline

Security has a tendency to become an afterthought for developers working in traditional development teams because they are too focused on building applications and meeting release dates. Traditional processes result in insufficient security and communication gaps between development and security teams, and, in turn, pose the risk of huge financial losses to businesses due to data breaches. In addition, vulnerabilities uncovered in the implementation phase could cost over six times more to remedy than the ones spotted in the design phase, according to [research by IBM](#).

To build secure applications, development teams should integrate [adequate cybersecurity layers](#) that conduct analysis in the container, source code, and dependencies, among other components. In particular, these are the cybersecurity layers they need to look into:

- **Container scanning.** Container technologies enhance the speed and efficiency of the development process, but their increasing adoption also means there could be [a wide range of potential risks and threats](#) that development teams need to secure the development process from. Tools for analyzing container images can help development teams scan for known vulnerabilities, secrets keys, compliance checklists, and malware variants at all stages of the software development life cycle (SDLC). Such tools can provide visibility and insights into the security concerns within the container before they are pushed to the production environment. To further minimize the risks to containers, development teams can also reduce their use of third-party software and use verifiable ones to ensure that malicious software does not penetrate the container

environment.

- **Software composition analysis.** Chunks of code, which are potentially sourced outside the organization and generally not checked during the static analysis phase, are embedded and run inside the DevOps environment. To check for outdated or vulnerable libraries in your code, tools like the **OWASP dependency-check** can be used. **Snyk**, a leader in developer-first open security, also provides free third-party verification for open-source projects.
- **Static Application Security Testing (SAST).** Also known as a “**security code review**” or “**code auditing**,” SAST helps developers find vulnerabilities and other security issues in the application source code earlier in the SDLC. Finding security issues in this stage can help companies save money and remediate the code faster.
- **Dynamic Application Security Testing (DAST).** Also called “black box” testing, DAST can find security vulnerabilities and weaknesses in applications by employing fault injection techniques on an application such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). DAST solutions can help test the resilience of applications, containers, and clusters when subjected to malicious techniques that threat actors use to find potential vulnerabilities and weaknesses.
- **Interactive Application Security Testing (IAST).** **IAST** performs runtime testing for web applications to spot security vulnerabilities. Bases that SAST and DAST might not be able to cover can be filled by IAST as it combines elements of both approaches, allowing it to cover more code, provide more accurate results, and verify a wider range of security rules. IAST solutions conduct all their analysis in the application in real time and anywhere in the following: development process IDE, QA, continuous integrated environment, or even in production.

## Application security best practices

With the world increasingly relying on applications for a myriad of purposes, organizations are tasked to build

any kind of purposes, organizations are tasked to build applications that are secure enough to withstand a variety of risks and threats that they could be exposed to. Below are some best practices to follow to ensure that applications are developed securely. While some of these practices focus on the adoption of tools for scanning and testing, other practices also entail the encouragement of a culture that prioritizes data privacy and security.

- **Implement Principle of Least Privilege (PLOP).** PLOP should be implemented within organizations. This policy limits access rights for users to only those permissions that are necessary for them to perform their tasks, thereby mitigating the risk of account abuse or hijacking and sensitive data leak.
- **Adopt automated testing.** Since the number of **software vulnerabilities has been increasing since 2017**, development teams should employ automated testing early in the development stage to catch errors or flaws while they are still easy to manage.
- **Scan for vulnerabilities regularly.** Developers often use external libraries or packages from open-source projects when developing software, but these libraries could be riddled with known vulnerabilities. To spot and subsequently patch vulnerabilities as early as possible, regular scanning should be performed on these dependencies.
- **Employ Runtime Application Self-Protection (RASP) solutions.** Development teams should make use of RASP solutions to monitor traffic to applications, containers, and serverless architecture. RASP solutions are designed to detect attacks in real time. The adoption of RASP solutions allows the interception of all kinds of traffic, including ones that indicate malicious behavior like SQL injection, cross-site scripting (XSS), vulnerabilities, bots, and other web application attacks.
- **Train cross-functional teams for DevSecOps culture development.** A DevSecOps culture should be fostered within organizations. This can be done by creating cross-functional teams that specialize in training developers on



security discipline as well as teaching security professionals about the software development process. These can allow security teams to gain a better understanding of programming languages and learn more about how APIs can be used to automate simple processes. Moreover, such skills that security teams can acquire from training ultimately reduce their workload and allow them to focus on more critical tasks.

- **Include application security in data privacy compliance strategy.** Application security is part of an organization's overall efforts to comply with data privacy regulations and IT standards. Organizations can use an efficient compliance strategy that follows the "privacy by design" approach. For instance, **an adequate compliance strategy for the General Data Protection Regulation (GDPR)** includes performing privacy and security checks through the implementation of identity and authentication controls and appropriate access controls, data protection via PLOP and encryption, and the use of secure frameworks and libraries, among others. The same practices can be followed by organizations to comply with other data privacy and protection laws, such as the **Health Insurance Portability and Accountability Act (HIPAA)**.

Following these best practices can help organizations strengthen their approach to application security. As illustrated previously, it is imperative for organizations to perform regular scanning and employ advanced security tools to detect malware, vulnerabilities, and other threats. Furthermore, they should also implement policies that enable a strong security culture – one that empowers development and security teams through training and saves organizations from paying hefty fines by ensuring that applications remain compliant with data protection mandates.

## Trend Micro solutions

The **Trend Micro Cloud One™** security services platform,

which powers **Trend Micro™ Hybrid Cloud Security**, enables software developers to build and run applications their way. It has security controls that work across existing infrastructure or modern code streams, development toolchains, and multiplatform requirements.

**Application Security**, which is offered by Cloud One, provides full diagnostic details about code vulnerabilities and runtime protection against automated attacks and the most common threats like SQL injection and RCE. It also offers complete coverage and reporting of every attack instance, as well as insight into an attacker's identity and attack methodology.

Cloud One also offers the following cloud security technologies to further help developers identify and resolve security issues sooner and improve delivery time for DevOps teams:

- **Workload Security**: runtime protection for workloads
- **Container Security**: automated container image and registry scanning
- **File Storage Security**: security for cloud file and object storage services
- **Network Security**: cloud network layer IPS security
- **Conformity**: real-time security for cloud infrastructure — secure, optimize, comply

Posted in **Virtualization & Cloud, Infographics, DevOps, Vulnerabilities**

## We Recommend

Intern Virtuali Ranso Securi

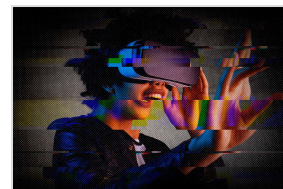
Trend Micro  
Security  
Predictions for  
2023:  
Future/Tense

et of  
Things

ization  
&  
Cloud

mware

ng  
Home  
Router  
S



Uncovering  
Security  
Weak  
Spots in  
Industry 4.0  
CNC

Machines  
Leaked  
Today,  
Exploited  
for Life:  
How Social  
Media

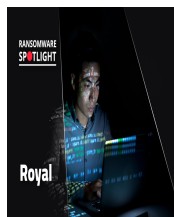
Biometric  
Patterns  
Affect Your  
Look Into  
Future  
Security  
and  
Technology  
Upgrades  
Working in  
Tandem



Analyzing  
the Risks of  
Using  
Environme  
nt Variables  
for

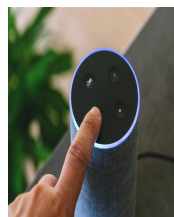
Serverless  
Management  
of Azure  
Managed  
Identities  
Within

Serverless  
Environme  
nts  
Containers  
in  
Serverless  
Environme  
nts for  
Better  
Security



Ransomwa  
re  
Spotlight:  
Royal

Rethinking  
Tactics:  
Annual  
Cybersecur  
ity  
Understand  
ing  
2022  
Ransomwa  
re Using  
Data  
Science

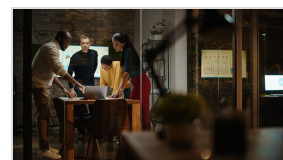


Alexa and  
Google  
Home  
Devices  
can be  
Abused to  
Phish and  
Viruses  
on Users,  
Spotted  
Research  
Using  
Multiple  
Exploits  
Targets  
Various  
Routers  
Network  
Security  
Threats of  
2017

Enterprises and organizations are facing a period of transition and uncertainty – malicious actors will hunker down and reuse tried-and-tested tools and techniques.

[View the 2023  
Trend Micro  
Security  
Predictions](#)

**Annual  
Cybersecurity  
Roundup 2022**



Our annual cybersecurity report sheds light on the major security concerns that surfaced and prevailed in 2022.  
[View the report](#)

T  
r  
y  
o  
u  
r  
s  
e  
r  
v  
i  
c  
e  
s  
f  
r  
e  
e  
f  
o  
r  
3  
0  
d  
a  
y  
s

Resources

Support

About  
Trend

---

Start your free trial today



Select  
a  
country /  
region

United  
States



Privacy

Legal

Accessibility

Site  
map

Copyright  
©2023 Trend  
Micro  
Incorporated.  
All rights  
reserved

