# **10 Best** Practices for Application Security Developers **Need to Know**

## What is your development team doing to ensure the applications that you are working on are secure when it comes to threats and vulnerabilities?

Every development team needs to be working on improving how they implement the best practices for application security. Whether you're designing with an experienced DevSecOps team or building your first application, your team should put security first. Security risks can not only set your project back, they can represent serious financial and legal liabilities that can tank even the most well-established projects.

In this **whitepaper**, we will share the **10 best practices** for application security and why your team should **start implementing them today**.

ProtectOnce

# THE **ATTACKS** AND **VULNERABILITIES** YOU ARE **PROTECTING** AGAINST

Understanding the security risks that your applications face is the core of coming up with a better security practice. Before we can implement better security, we need to know what we're trying to secure against. The landscape of vulnerabilities and attack vectors is constantly changing. As technology and design strategies change, the vulnerabilities and threats change with them.

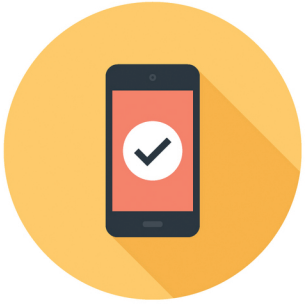
## THESE ARE JUST A FEW OF THE MORE **COMMON THREATS** AND **VULNERABILITIES** FACING APPLICATIONS TODAY.

- Cross-Site Scripting
- SQL Attacks
- DDoS
- Malware
- Cross-Site Request Forgery
- Broken Authentication
- Data Vulnerabilities

ProtectOnce

# UNDERSTANDING **WEB APPLICATION SECURITY**

Here's where it all starts. Web application security is a complicated topic that involves everything from having team members implement better password practices to using the most up-to-date software in your development cycle. This means that there are countless opportunities to improve your application security, but just as many areas where your current strategies fall behind.



One of the biggest challenges with web application security comes with the fact that we often only identify holes in our security after they've become a problem. It takes a lot of effort to proactively identify security flaws and implement fixes before they become an issue. However, implementing the best application security practice allows your team to stay few steps ahead the rapidly evolving landscape of threats and vulnerabilities.

The following are **10 areas** where your team can start building a **better security** culture today.

1. Build a security culture
2. DevSecOps design
3. Security audits
4. Strong logging practices
5. Encryption where possible
6. Threat modeling and risk assessment
7. Penetration testing
8. Staying informed
9. Staying up-to-date
10. Fix your flaws

# 1. **Good Security** is a Team Effort

Good cybersecurity is a skill. This isn't something that is instantly acquired, but it's something that you can learn and build over time. It's just like learning a new language. At first, it might be a little difficult and a little intimidating, but, with time, anyone can build proficiency.

There's an old way of looking at cybersecurity. This outdated methodology suggests that you should have a specialized team of cybersecurity professionals that handle all of your organization's security. However, this ends up creating a system where every single employee who isn't on that security team can accidentally open up new vulnerabilities.

This doesn't mean that every team member should become a master of contemporary cybersecurity, but it does mean that everyone needs to understand their role in the overallsecurity effort. Even the most entry level team members are going to have access to passwords,databases, and other technologies that could become vulnerabilities if mismanaged.

Make sure that every member of your team is updated on the latest security practices based on their roles. This counts everyone from the greenest new hire to executive-level staff. Everyone has access to some systems in today's workplace which means that everyone is a member of your security team.

ProtectOnce

# 2. The **DevSecOps Upgrade** to Your **Design Approach**

DevSecOps upgrades the importance of security in your design and development cycles. It's also a vital approach to prioritizing security in your application development. Here's the basic idea behind DevSecOps

Traditionally, security is outsourced to third-party professionals or a dedicated security team. However, this leaves security outside of the development process. DevSecOps integrates security into your development process.

This makes designing for security a core part of your development cycle. DevSecOps means that security isn't an afterthought or an addition, it's built into the core of your application. There is another approach that takes security one step further by making security the top priority rather than just a core design principle.

SecDevOps is an even more secure way to layer cybersecurity and development. This model makes security not just the core of development, but the number one priority of development. SecDevOps can be a great choice for projects that handle the most vital information like medical records, finances, and other highly sensitive data.

# 3. Third-Party **Security Audits**

We talked a lot about how your internal team members play a vital role in your overall cybersecurity plans, but that doesn't mean you should get rid of third-party security analysts all together. These are experts who know cybersecurity inside and out, and they still play a vital role in your overall application security.

One of the best services that they offer is a third-party security audit. This audit takes a look at every aspect of your application's security and gives your team a report full of action items that you can improve on. These audits allow your dev team to identify, and fix, errors before your clients and customers spot them

Why should you have a third-party handle this task? They're not looking at your application as it stands. They're looking for security flaws. They're able to see things that your team members are just too close to the application to notice.

# 4. Build **Logging** Into Your **Development**

A good data log doesn't just give you detailed metrics that you can analyze later on, it also gives you important information that you can use in your application security.

One of the best things that good data logging offers is a snapshot when something goes wrong. You'll get to see exactly what went wrong, the factors that led up to the problem, and any other anomalies that precipitated a data breach or another vulnerability. Your team can use this log not only to recreate these problems and vulnerabilities, but to also work them out of your system. Without a log in place, your team is stuck working in the dark as they attempt to resolve complicated issues without a roadmap

# 5. Always **Encrypt**

Another vital step in application security is ensuring that all of your data is encrypted. Data should be encrypted to the strongest encryption standards that are practical as well as viable for the data that you're working to secure. Encryption should happen at all layers of your application design.

The basic levels of encryption should go without saying. HTTPS, HSTS, and using SSL are a few of the basics that meet today's highest encryption standards. These standards are great for protecting data in motion from a Man-in-the-Middle attack, but we also need to look at your data at rest.

This means keeping the data that is on your servers encrypted. Countless people have access to the information inside of your servers. This could be former employees, hackers working a social engineering project, or exploits in your application's code that your derv team overlooked. Making sure that this data is also encrypted ensures that you have another layer of security protecting some of your most vital assets.

# 6. Start **Threat Modeling** Early

What are the biggest threats that your application faces? This is a question that you should be able to quickly and concisely answer. Every application has a unique suite of threats and vulnerabilities that it's going to go up against and your team should be well aware of the challenges on the horizon.

The thing about threat modeling is that it's going to evolve as your project evolves. The threats and vulnerabilities your application faces during alpha and beta testing are going to be different from the threats that it faces during launch. These threats will evolve as your application becomes more successful.

Start threat modeling on your project early and allow those threat models to evolve as your project grows and changes.

# 7. Make the Most of **Penetration Testing**

One of the best ways to find vulnerabilities in your system is to turn a hacker loose on your application and see what kind of damage they can do. This might sound like a dicey proposition, but there are ethical hackers, known as penetration testers, that do this kind of work.

Penetration testing is done by experts that are sometimes known as white hat hackers. These are individuals who know all of the tools, tricks, and exploits used by hackers, but they use these technologies to help your development team identify and fix vulnerabilities.

You can hire penetration testing experts to help your application be more secure. These experts will first go over the scope of their project, then conduct their testing, and finally present a report to your development team that you can use to go back and correct any security flaws that the pentesters identified.

# 8. Stay **Informed**, Stay **Educated**

Here's an easy thing that you can start doing today to stay more prepared for application security.

If you've been developing applications for any length of time, you know that things change rapidly. New security standards, new products to design for end users, and a rapidly changing technology landscape mean that few things stay the same year after year. Staying informed about the latest security advancements and threats is an easy step to building a better security culture.

Remember when we said that every member of your team should be part of your security? This goes for staying informed and educated too. This doesn't mean that members outside the dev team should learn how to write secure code, but everyone should get to learn about the threats they will face when and where they interface with your project.

# 9. **Update** When Available

Let's face it, we've all postponed updates whether it's the web browser we use for work or the laptop we use every day at home. Whether it takes a few hours or a few minutes, people tend to skip updates until it's too late. This is even more important when you're dealing with workplace technology you're using to design your applications.

You need to not only be ready to update when new updates become available, but also build that downtime into your development schedule. As software developers, we know that one of the reasons why companies push updates is because those updates are fixes for security problems. Keeping your software up-to-date closes out those security threats and makes your project all the more secure.

# 10. **Fixes** are **Better** Than **Patches**

All of the other best practices are things that your team can do before there's a problem. They are preventative measures designed to make your applications stronger and more secure before it even hits the ground. However, things will go wrong eventually, and you need to have a plan in place when they do.

An important thing to remember here is that fixes are better than patches. Of course, you should push out a patch that will fix a problem as a stopgap solution. However, you should also be aware that the root cause of that problem needs to be addressed in order for your application security to be as strong as it can be.

# How ProtectOnce Can Help

**ProtectOnce** empowers developers to secure their application with a developer-centric solution that is extremely easy to deploy, manage and scale, offering nearly zero friction.

ProtectOnce's microagent is deployed in your application in minutes, just like any other library, and provides complete protection leveraging next-gen in-app WAF and RASP.

Start protecting your app **for free.**