10.5 Things That Undermine A Web Application Security Program

i

Organizations know securing their web applications is a critical responsibility, and they normally undertake a variety of measures intended to uncover vulnerabilities and prevent attacks. Unfortunately, some of those measures can undermine rather than strengthen their security posture. Based on our extensive experience in providing end-to-end solutions for web security, we've compiled this list of actions that are intended to bolster security but may in fact be having the opposite effect.

We describe the intended effect of the action and how it undermines your security, and we offer a recommended solution, alternative or change. Note that none of our recommendations taken alone is a silver bullet—they're all pieces of a larger puzzle. Instead of depending on any one of them as your sole defense, you should evaluate and prioritize the list and proceed with all the activities that will help protect your web applications and data.

1. Publishing Best Practices

Intended Effect

"We published a developer's list of best practices. We should be secure now."

How It Undermines Your Security

Best practices are not uniformly followed by developers. Typically the lists organizations publish are too general, and what developers end up doing is different from what the rest of the security organization is doing.

Developers often ask us where they can find a list of the best practices for building web applications. The answer is: It doesn't exist. Web application development and security is a rapidly evolving field, and what was a best practice yesterday is unsafe today. Secure development depends on the development language you're using, the platform, the framework, the libraries, and so on. Keeping such a best-practices document current would be next to impossible.

Recommended Activity

You should trust your developers, but verify their work through testing. Verify that the best practices you publish are being adhered to. For example, you can tell developers that all input must be valid, all outbound user data must be encoded contextually to block cross-site scripting, and all sensitive forms that submit data must have secure cookies and transport protection. But don't assume those rules are being followed—verify that they are.

2. Compliance

Intended Effect

"Following a compliance standard should make us un-hackable, right?"

Organizations think that if, for example, they follow all the PCI standard compliance guidelines, none of the cardholder data they manage will be at risk. Or they think that if they follow all the HIPAA guidelines, no personal health information will be accessible. It's not true. Those guidelines are a baseline level for organizations to meet in order to promote security. But every organization is different and things are constantly changing, so you need to go beyond simple compliance. Not taking actions simply because compliance didn't call for them is a good way to introduce risk into your environment.

How It Undermines Your Security
Checkbox security ≠ Security!

Checklists have negative associations in the security community, mainly because groups undergoing an audit often do the bare minimum to receive a checkmark from an auditor. In other words, mere compliance to checkboxes, rather than security, becomes the primary goal. The result is that organizations focus only on the "letter of the law" and ignore the spirit of the entire exercise. In addition, if groups are self-assessing against a checklist, there is no verification counterweight.

Recommended Activity
Establish a security program or set of activities that maps to the established risk posture of your organization. That, in addition to making sure you're in compliance, is the way to protect the assets and information under your control.

3. Secure Frameworks

Intended Effect

"We've given the developers a list of secure frameworks and tools to use—we should be secure now, right?"

Organizations often believe they're secure because they've given their developers a list of approved frameworks or security libraries. To a lot of people, especially if they're not familiar with the development process, that belief seems reasonable.

How It Undermines Your Security
Frameworks and security controls can be misused, underused, or not used at all.

The problem is that even if developers are given secure frameworks—or a framework that has security controls in it—they don't always wind up using those security controls. So you can't depend on this as 100 percent of your security program; it's just one piece.

Research we did recently showed that in environments where organizations were using secure frameworks, their data tended to be less secure than that of organizations who weren't using them. That's because there was an over-reliance on the secure frameworks, where developers weren't checking input and output validation because the framework already had controls in place. They felt they didn't need to be doing the checking themselves.

This is often the case with ASP.NET applications where organizations are using .NET frameworks out of the box. It has a lot of security controls already baked in, but if you have to turn these off, you're vulnerable.

Recommended Activity
Verify!

Even if you give developers security controls and a "secure" framework to work within, you still need to verify that those controls are being used properly and consistently. You can take a range of steps to put in place a security program, including building robust documentation and writing up security controls, but if you don't verify that developers are following procedures, you can get a false sense of security.

Developers definitely need to have frameworks that have been vetted for security. Be willing to put in the necessary effort and energy—including training and supervising—to make sure developers are enabled to succeed. Just don't assume that giving developers a secure framework has taken care of all your security needs.

4. SSL

Intended Effect

"We can solve all our security problems with SSL."

How It Undermines Your Security

SSL is vital, but it's not a sole security defense. It solves a handful of issues, but there are many it doesn't solve.

Using the latest 128-bit encryption over SSL doesn't make you hacker-proof. In fact, your security controls will lack the visibility into the traffic steams to observe attacks because they are encrypted. The attackers and their traffic are traveling across that cell, so your IDS or IPS may not be picking them up if they're not in the proper place within your environment, namely after the decryption but before its destination.

Too often, organizations think they don't need to worry about application security because they're using SSL. But that approach actually undermines your security. SSL addresses just one or two issues out of a huge pool of potential problems that could be lurking within your application, and by relying solely on SSL, you're in fact adding more risk.

Recommended Activity

The smart approach is defense in depth. It's often preached throughout the industry, whether the subject is network security or application security, but it's worth reinforcing yet again. Don't rely on a single measure, even one as useful as SSL.

5. Web Application Firewall

Intended Effect

"We've solved all our security problems by installing a web application firewall."

How It Undermines Your Security

Web application firewalls (WAFs) are only one line of defense; they're vital but insufficient as a sole measure. Like all other security practices, WAFs can do a lot of good and offer valuable protection. We recommend them frequently, especially if organizations have a large number of legacy websites they need to protect. However, you can't solve all your security problems simply by installing a WAF. If you depend on one exclusively, you get a false sense of security.

Recommended Activity

When we're doing security assessments of websites, we automatically generate rules for a large number of vulnerability categories, then feed that into the WAF. It gives the WAF the ability to block particular, specific attacks.

The issue here is also related to compliance. PCI 3.0 makes some clarifications to the PCI specification, but not necessarily changes. Note that the PCI Security Council clarified that purchasing a WAF is no longer enough for you to be compliant. Ideally, they will further update the standard to include proper usage and configuration of WAFs. Simply having a WAF on the wire either in inverting mode or, even worse, sitting on a SPAN port or a tap port in collection and aggregation mode, is not enough to ensure you're safe.

If your resources are not robust enough to enable you to have your WAF in a broad protection mode, then it's almost a sure thing you don't have enough resources to simply detect. The amount and the speed at which you have to be able to detect and respond on your own with human effort is simply not going to be good enough to stop an attacker.

For example, a Verizon data breach report from last year showed that the time between discovery and exploitation of a web application vulnerability is only seconds or minutes.

To re-emphasize what we've been saying, a WAF is a great tool and will reduce the likelihood of a data breach happening within those seconds or minutes, but it has to be tuned and used properly.

6. Asset Management

Intended Effect

"We've cataloged most of our assets. That should be good enough to protect us."

How It Undermines Your Security
Poor asset management, where critical assets and associated assets can go undiscovered and/or untested, introduces security risk. Assets with out-of-date dependencies introduce additional risk.

Recommended Activity
You should regularly perform asset discovery activities. Catalog the information along with metadata about the asset.

No one wants poor asset management, including lacking a configuration management database, but it's quite common and not just in the realm of web applications—it occurs with all the assets in an organization, from desktops to BYODs. It's important to know what your assets are if you're going to test them and reduce the risk and the likelihood of exploitation against them.

How can a CSO get a complete inventory of an organization's exposed assets? Including web applications, mobile applications, web services, marketing sites and more?

First, perform asset discovery duties on a regular basis, even if that means simply scanning all your known internal/external partners and things of that nature to collect data. That's one of the easiest activities.

Next, ensure you're getting the widest possible knowledge collection. For example, marketing campaigns can start and end very quickly, and it's important to make sure that you know what the associated assets are and to have a direct line of communication to your marketing department and the individuals in charge of those campaigns. That way, you can make sure that part of their process includes alerting the security organization of changes.

One additional step: Your legal department will have a list of intellectual property that includes domains and associated domains they have to protect. Get that list and make sure all those domains are protected from domain squatters, snipers and more.

Those three activities are a good starting point for collecting your asset management data.

7. Testing Throughout the Phases of SDLC

Intended Effect

"We don't want to constrain our development teams, so we're doing minimal security testing."

How It Undermines Your Security

The SDLC is the software development lifecycle and basically it involves everything from architecture to development to QA to deployment. Testing in only one phase of the SDLC gives you a limited view of security. It's risky not to undertake security activities in design, development and production phases.

Organizations often say they're doing architecture reviews and that therefore they feel safe. Or they did threat modeling and are therefore safe. Or they ran static code analysis during development. The problem is, none of those alone makes an organization safe. Security is not a one-time event; it's something you do throughout the entire process. Organizations need to make sure they don't have blinders on and have only a limited view of security.

Recommended Activity

We recommend that organizations do architecture reviews, threat modeling, static code analysis, and threat development. You might also bring in a pen tester or code reviewers in QA and during deployment, where you need constant monitoring, which people often forget to do.

It can be mere hours between having a very safe website and having a site with known vulnerabilities that are causing you pain. For example, last year, the Ruby on Rails framework was relatively safe in January, but in February it turned out that under certain conditions, Ruby on Rails websites had an SQL injection within the framework itself. If you don't have monitoring on your website, if you're not evaluating your security frequently, if you don't have an inventory of applications—not only of the frameworks you're using but the libraries—you could be at risk.

Other activities that don't get performed often enough are architectural reviews. They're not always seen as being part of the security testing methodology, but they are, and they shouldn't be excluded. Performing those activities will make the job of performing source code analysis easier, because you will have already identified a number of vulnerabilities and risks in the environment well before you can get to the dynamic stage. If you already have applications under production, start dynamically—start testing the ones already out there. But as you get a handle on that and firm up your security program, you're going to want to expand out to all SDLC phases.

8. Security Reports (false positives / negatives)

Intended Effect

"We cast a wide net, report all potential vulnerabilities, and trust our developers to sort which vulnerabilities are real and which aren't."

How It Undermines Your Security

This approach wastes time, productivity and resources. It also burdens developers with making key security decisions.

Sometimes organizations decide to throw out a big net, gather as much information as possible, and hand that all over to the development community. The next step, as they see it, is for developers to figure out what's actually an issue and what's not.

There's also the inverse of this, where an organization has a limited amount of time to perform this activity, so the time is spent on learning how to properly use desktop application testing tools, configure them, and understand how they work. That can introduce false negatives, which tends to consume a lot of time and resources.

Giving developers a report listing all potential vulnerabilities isn't fair to them, because when you tell them, for example, that 50 percent of it is false, and they need to figure out which 50 percent that is, it's very frustrating for them. It's like getting a home inspection and being handed a report listing all the things that need to be fixed. Imagine if the inspector then says that 50 percent of the issues cited in the report are wrong—you figure it out. But you can't because you're not an electrician and a plumber. Developers are in the same position. They're not security experts. How can they determine what really is posing a risk and what isn't?

False positives include such things as an application returning a positive 2xx response, regardless of whether the data is filtered or not. Many tools will interpret that as a successful attack because the application responded positively, when in fact the data could have been properly filtered and the system simply returned the user to their referring page to maintain the user experience.

Recommended Activity
Always work from verified vulnerability lists.

Make sure you're working from a completely verified list, whether you have to perform that activity yourself or have someone perform it for you. It will save time and resources, whether you're a $20 billion organization or a $1 million organization. Not wasting resources will also ensure you get the buy-in you need from partners within your organization to support your activities.

9. Static Reports (delivered only every X months)

Intended Effect

"We do a static report as a security assessment to give us a snapshot of our security posture. We think that's sufficient."

How It Undermines Your Security
The intended effect of static reports is to get a snapshot of where your organization is today with its security program. What really happens is that such reports offer a very limited view because applications change continuously and are being updated and deployed continuously.

Reporting on your web applications is like trying to hit a moving target because the code base changes so often that vulnerabilities identified one day are no longer there the next, and new ones have been introduced. Static reports might work if you have a very linear process, for instance, a waterfall-based application that has a well-padded timespan where you can do evaluations throughout. But even then, a security consultant would have to do evaluations on a continuous basis.

Recommended Activity
Continuous assessment is the gold standard. At a minimum, testing should keep even with or exceed the rate of change in the application.

We recognize that a lot of organizations can't perform assessments on a continuous basis. Maybe they have blackout windows, maybe they have development windows. There are valid reasons for not doing 100 percent continuous assessments.

However, at an absolute minimum, testing should match or exceed the rate of change. If you know an application is being updated weekly, then your testing should be done no less frequently than every six days. If an application is updated quarterly, then you should be updating every 89 days. This is also a good strategy if you have a large number of applications or if you simply have more applications than resources. Sit down with your development teams, understand the rate of change for your applications, and map your testing to that.

You can also decide which applications are critical and test those more frequently. You may have a critical application that changes only once every quarter and you're testing it weekly. That may not be necessary. Or you may have a not-so-critical application that changes every week, but testing it quarterly is acceptable. The key is to map your activities to the change rate, not simply to the criticality of the risk that an asset may introduce.

Keep in mind that the rule of thumb is: Don't send unevaluated code into production. Everything has to be assessed from a security perspective, whether that's done automatically or manually, because one line of code improperly written can put your entire organization at risk. One SQL injection vulnerability, which can literally be one line of code, can unfortunately result in a complete compromise of your data.

10. Degree of Senior Management Support

Intended Effect

"Our executives don't believe in devoting resources to security. They see it as a grass-roots effort, and think that's sufficient."

How It Undermines Your Security

Security requires time, resources and commitment. By not having top-level support, security will not be consistent across the organization, nor will it be sufficient.

There are times when senior management simply doesn't understand the value in performing certain activities or doesn't see the investment of resources in terms of dollars and human resources as providing a sufficient return. That can lead to grass-roots security efforts, where an individual believes he or she is the sole person who sees security as an important activity and takes on the whole task. That person may think it's the right thing to do, but in the end it doesn't have a positive effect and tends to waste resources. It requires time and commitment in both financial and human terms to move the needle on security. A security effort has to have top-level support from management, otherwise all those isolated activities undertaken by individuals will be for naught.

Recommended Activity

If security is important to the organization, company leaders must make it a priority.

Just as we need building inspectors to check the work of builders and safety inspectors to check the work of car and airplane manufacturers, we need people to assess the security implications of the work developers do. Developers are responsible for building functional applications that work. Security is a different responsibility altogether. It's essential to have someone figure out all the ways an application could be misused, then decide what security controls need to be in place.

There are a handful of scenarios where a lack of support is a direct by-product of needing to move fast in terms of building products. A classic startup, for example, has to get products out the door. Performing security assessments may not be an activity they have the time or resources to do. In fact, in Silicon Valley, it's something of a badge of honor when you're finally hacked. It's one of those things that says you've made it. But that's an outdated attitude. Attackers and hackers are opportunistic. If your company is in a scenario like this, there are still ways you can protect yourself, such as by using secure frameworks, making sure you're using them properly, and enforcing the security measures you do have the time and resources to put in place.

10.5 Basing Your Security Program on a Top Ten List

Intended Effect

"We know we have comprehensive security coverage because we based our program on a Top Ten list we downloaded from the web."

How It Undermines Your Security

Following someone else's Top Ten list creates blind spots, and doesn't take into account your organization's unique security posture or company-specific threats.

There are widely used Security Top Ten lists, such as the one from OWASP, the Open Web Application Security Project. There's nothing wrong with the OWASP Top Ten. In fact, when there were no definitions for many vulnerabilities and people were describing them using different terms, the OWASP Top Ten was fantastic for codifying and standardizing terminology. But that list shouldn't be used as the only yardstick, because not everybody has the same risk profile. A startup has to worry about different things than a defense contractor or a government agency. The OWASP Top Ten doesn't apply to all organizations and situations in the same way.

In fact, if you collect your own data and have your own methodology, you should be creating your own lists of top offenders or top things you need make sure do not exist in your web or mobile applications. As consultants, even if we start with the OWASP list, we reprioritize the items on it based on the risk needs and thresholds of a specific organization.

Recommended Activity

Generate your own Top Offenders vulnerability list.

A sensible approach is to use the OWASP list as a starting point, then reprioritize it and make that your new baseline. Prioritize vulnerabilities based on either risks you've already identified within applications, and/or, if you're at a more advanced stage, based on expected loss. Look at vulnerabilities, understand the expected loss they could cause your organization, then fix them based on their priority.

It's Not One Thing, It's Everything

The key take-away for organizations in all industries and at all stages of maturity is that no single measure is sufficient to guarantee the security of your web applications and the data you manage. Your applications and security risks are both constantly changing. To stay secure, we recommend you make security a priority; give it the resources it needs in terms of people, time and budgets; and look at security as a job that's never done.

About WhiteHat Security

Founded in 2001 and headquartered in Santa Clara, California, WhiteHat Security is the leader in application security, enabling businesses to protect critical data, ensure compliance, and manage risk. WhiteHat is different because we approach application security through the eyes of the attacker. Through a combination of technology, more than a decade of intelligence metrics, and the judgment of real people, WhiteHat Security provides complete web security at a scale and accuracy unmatched in the industry. WhiteHat Sentinel, the company's flagship product line, currently manages tens of thousands of websites – including sites in highly regulated industries, such as top e-commerce, financial services, and healthcare companies. For more information on WhiteHat Security, please visit www.whitehatsec.com.

WhiteHat Security, Inc. | 3970 Freedom Circle | Santa Clara, CA 95054 | 1.408.343.8300 | www.whitehatsec.com