IMMUNIO

You've Been Hacked:

Why Web Application Security Programs Should Start with RASP

Authored by Goran Begic
for **IMMUNIO**

IMMUNIO /eBooks

You've Been Hacked:

Why Web Application Security Programs Should Start with RASP

Authored by Goran Begic for **IMMUNIO**

VP of Product
@gbegicw

Introduction

At most organizations, instituting defensive measures against hacks follows a familiar, well-worn pattern, as does the chain of events that unfolds when a hack actually does take place. In this paper, I make the case that the commonly-accepted best practices for avoiding hacks in the first place, as well as for responding after they occur, are wrong. In fact, at many companies, the accepted approaches are exactly backwards.

Why? Because at many organizations, mitigating weaknesses in insecure and outdated applications is considered only *after* a severe security breach takes place. In this paper, I argue that protecting applications is an integral part of a proactive approach to cybersecurity.

Read on to learn more about the current status quo and how to turn it on its head at your company: safeguarding your web applications and critical customer data *before* attackers have the chance to strike.

It takes at least 90 days to fix major vulnerabilities once identified; your system is still vulnerable while you repair the damage.

Picture the scene: You are a CISO at a mid-sized company. You're about to deliver bad news to the CEO and the board. The disaster you've had nightmares about has come to pass: the company has suffered a data breach. An untold number of customer accounts, and a wealth of other sensitive information, has been offered up for sale on the dark web.

The big question is: how did you get here?

In theory you did everything right. You had incident response plans in place, along with a cyber-risk insurance policy. You performed rigorous security testing on your applications (or so you thought). But the breach happened anyway. What went wrong? And what can you do differently now to minimize the risk of it happening again?

Security experts agree that the number of security incidents, such as data breaches, is expected to rise in the near future. And one of the most vulnerable areas of information security for any organization is web applications.

As has become abundantly clear, web application security is a complex, multifaceted endeavor. Hiring top-notch developers and security engineers and investing in their training, formulating incident response plans, and investigating cyber-risk insurance policies are all elements of this process, but they're not the most important ones.

First, you need to carefully consider your applications' vulnerability to common attack vectors such as SQL injection, cross-site scripting, and account takeover. Your first step should be mitigating common weaknesses in applications through proactive security technology that protects apps from the inside out, such as Runtime Application Self-Protection (RASP).

The average cost of a hack to an organization is $4 million.

PROTECT THE APPS, NOT JUST THE PERIMETER

Half to three-quarters of all attack vectors target web applications because they offer such a direct entrée into a system. According to Verizon, web application attacks were the top attack vector in 2015, representing 40% of all confirmed breaches[1]. This is a significant increase in web application attacks from the previous year, in which this vector represented fewer than 10% of confirmed breaches.

The average cost of a hack to an organization is $4 million. It takes at least 90 days to fix any major vulnerability once it is identified; meaning your system continues to be vulnerable while you repair the damage. Then there is the fallout from an incident that becomes known to the public—lack of trust and loss of reputation.

Most systems in most organizations span several applications or technologies, many of which were developed over the course of the last 50 years. These older technologies were not developed with application security in mind. Hackers know this and they look for vulnerabilities in the outdated parts of a system, taking advantage of outdated security solutions that leave gaps in perimeter protections, leaving web apps exposed. The more layers you have in your system, the better chance that someone is trying to sneak in. It's difficult to protect against all the possibilities and vulnerabilities, so hackers will most likely succeed in breaching your system. If you are a cybersecurity professional: Get ready!

[1] *2016 Data Breach Investigations Report, Verizon, http://www.verizonenterprise.com/verizon-insights-lab/dbir/2016/*

OWASP has developed a comprehensive incident response plan to help organizations be proactive.

EMERGENCY RESPONSE

The board meeting is about to start. You try to focus on the good news: you had a solid incident response plan in place and your team is executing well-practiced routines. You've implemented best-practice tools and guidelines to help you avoid many attacks altogether, and minimize the damage from those that do succeed.

The Open Web Application Security Project (OWASP) has developed a comprehensive incident response plan[2] to help organizations be proactive—identifying, investigating, and addressing the incident and protecting the system and organization. Here is a compressed version of OWASP's recommendations:

Pre-incident

- Analyze existing processes to understand necessary elements of an incident response plan, and identifying the organization's assets, to know which equipment, materials, and information is most critical.
- Identify experts with the appropriate technical skills and authority from different areas of the organization. This should include a team leader, a triage team with the responsibility to decide whether an incident requires a response from the team, and an incident handler.
- Create a basic response plan to address the roles and responsibilities of team members, and processes for investigation, triage and mitigation, recovery, and documentation of the incident.
- Identify and document how your organization categorizes incidents and the factors that will trigger the incident response plan. For example, loss or theft of equipment or information, unauthorized access or attempted unauthorized access, etc.

[2] *https://www.owasp.org/index.php/OWASP_Incident_Response_Project*

During the Incident

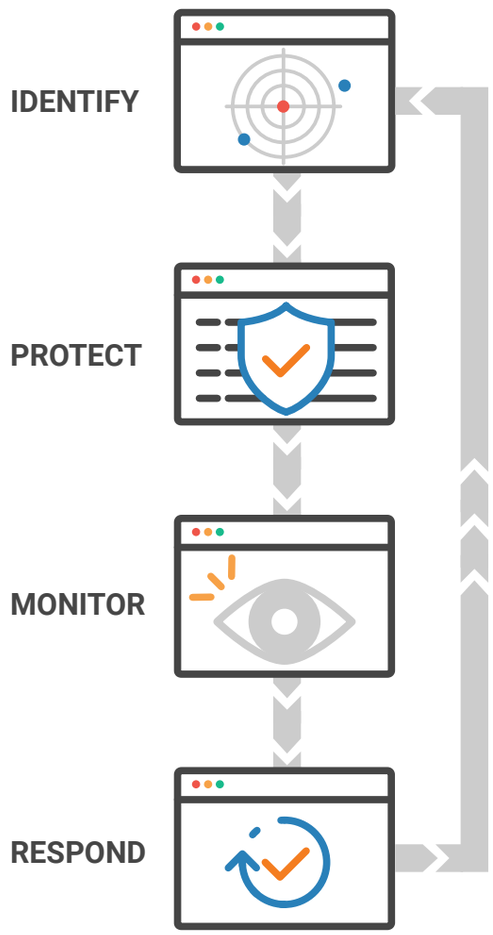
- Determine that there is a problem and the nature of the problem (such as denial of service, session hijack, etc.). Identify the people, processes, or systems that have been affected. Verify the information disclosed. Identify the impact to the business.
- Classify the type of incident, prioritize it, assign tasks to incident response team members.

Post-incident

- Restore services to the system, ensure integrity of business systems and controls, replace compromised records, reset passwords on compromised accounts, install updates or patches, etc.
- Write a comprehensive report covering all aspects of the incident such as data and time, systems or materials affected, mitigation steps taken, etc.
- Analyze the organization's response to the incident and the performance of the team.

At all times

- Run through simulated incidents.



INCIDENT RESPONSE: QF32

What does a great incident response look like and what can we learn from other industries?

Here is a good example from Qantas Airlines, which has one of the best safety records in the history of aviation. The context is vastly different from cybersecurity, yet the path to resolution outlines some of the key points in the OWASP document.

Pre-incident

- Pilots and crews and extensively trained in dealing with in-air emergencies
- Day of incident: Captain and flight crew (including two additional pilots on the flight deck) discussed chain of command and team roles

Incident

- The team trusted its leader (the captain) and performed their roles:
  1. One pilot dealt with the alarms and checklists
  2. One dealt with flying the plane
  3. One dealt with the customer service team and passengers
- They refocused during the crisis on systems that did work instead of listing those that failed

Post-incident

- The captain communicated openly and honestly with passengers and the public about the incident
- Qantas found hotel rooms and met individual passenger needs

When considering this type of coverage, organizations should first learn what risks are already covered by their insurance.

DEALING WITH THE AFTERMATH

You tell the board about the list of investments in the plan that was approved by this very same board not long ago.

- *Cyber-security insurance*
- *Compliance with standards such as PCI DSS 3.2*

From your new perspective, having cyber-risk insurance certainly looks like a good idea. It will help ease the financial burden that the organization will need to take. Can it save your job? One of the board members is reviewing the latest news article about the attack on your company. Customers are angry that their data has been stolen. The reputation of your organization is on the line. Another board member asks how it is possible for the hack to have happened when you complied with industry standards and best practices.

Cyber-security insurance

The goal of cyber-security, or cyber-risk, insurance, is to protect an organization from financial losses due to hacking and cybercrime including damage to the system, business interruption, and data breaches. What it will not do is protect your organization's reputation if a breach occurs.

In the event of a successful breach, organizations face a variety of potential losses. These include:

- Direct costs from the hack like extortion, data destruction, or repairing physical damage to the system
- Associated costs such as crisis management or legal claims regarding fraud or privacy breaches
- Loss of business due to business interruption or damaged reputation

Compliance with standards and regulations is table stakes for organizations responsible for digital assets.

Compliance

Compliance is top of mind in application security, but compliance is just one of many considerations. It's not something that allows an organization to compete on a higher level in terms of security. Business ethics, the basic principles of dealing with right and wrong in a business environment, are not as sophisticated as they should be for web application security.

The basic level of ethics is compliance with appropriate standards and regulations. This is table stakes for any organization, and especially for those whose digital assets include personal, financial, and medical information for employees, customers, or other stakeholders. Yet compliance is not enough to create a truly secure organization.

Meeting compliance standards is necessary to perform very basic business functions, but this alone will not address security risks, or differentiate you in the marketplace.

APPLICATION SECURITY TESTING

You explain to the board that the attack vector was a web application. More questions come your way: we invested in application security testing. Why didn't all this testing prevent the attack? You explain that the web app in question was known to be a weak link, but that remediation had not yet taken place. Testing allowed you to identify the risk, but the marketing initiative that this app was using had to proceed, even knowing the app was at risk of being exploited by attackers. Engineers were in the midst of building high-priority features that had to be delivered to keep the project on track.

There are several application security methods in use today—techniques to identify vulnerabilities and coding errors, prioritize them, and fix them. Unfortunately, testing alone helps quantify risk in applications, but it does not reduce it. Testing only helps you manage the risk.

By the time you finish counting vulnerabilities and coding flaws, the software will be outdated and new vulnerabilities will be found.

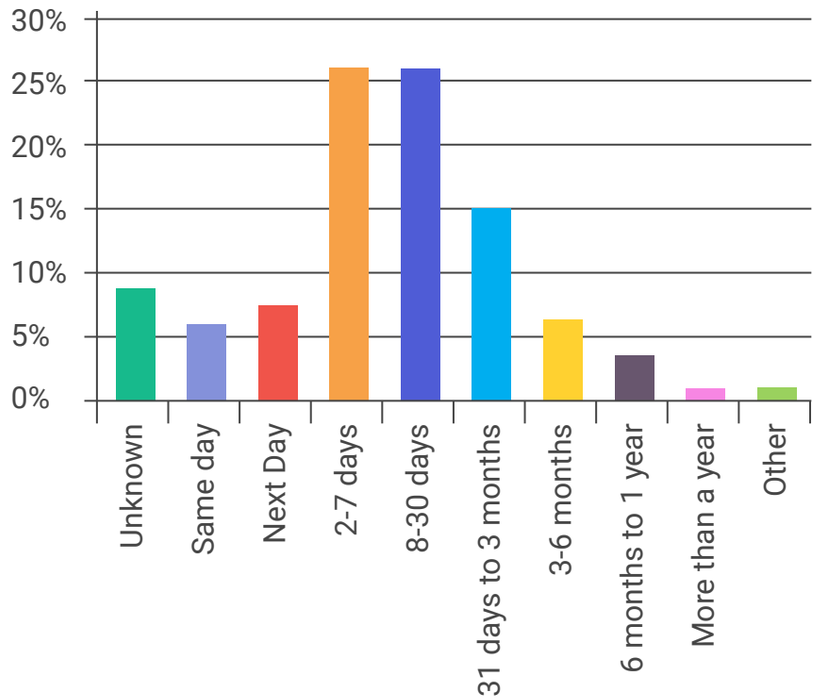
For one thing, these methods all take time to implement, and building an effective testing program that reduces risk across many applications can take months or even years. The SANS Institute's 2016 State of Application Security study reports that many organizations fail to repair critical vulnerabilities in a timely manner:

The study notes: "Respondents unfortunately register a low level of satisfaction with their patching and repair process. Fewer than 30% are achieving a 75%–99% level of satisfaction with the speed it takes to repair their vulnerabilities, while only 11% felt 100% satisfied. The speed at which patches are applied is comparable to last year's survey, with 26% of vulnerabilities being patched within two to seven days, and another 26% within eight to 30 days, as illustrated below:"

By the time you finish counting vulnerabilities and coding flaws, the software will be outdated and new vulnerabilities will be found. Finding issues does not fix vulnerabilities. Knowing the type and nature of vulnerabilities can help you manage risk, but this knowledge will not fix the issues in the code, or the application environment that opened the door for exploitation.

On average, how long does it take your organization to fix and deploy a patch to a critical application security vulnerability for systems already in use?

Time to Patch a Vulnerability



Source: The SANS Institute

The five most commonly-used application security methods today (methods which, unfortunately, do not provide comprehensive protection for web applications) are:

1. **SAST** (Static Analysis)
   Detailed analysis of data and control flow done by processing source code, or binaries without running the application
2. **DAST** (Dynamic Analysis)
   Discovery of application interfaces and injection of vulnerable traffic with the aim of discovering security weaknesses; performed while the application is running
3. **Developer education**
   Secure coding guidelines, online and in-person training courses, capture the flag competitions, etc.
4. **IAST** (Interactive Application Security Testing)
   Analysis of application execution performed during runtime, and utilizing access and insights into application code
5. **Manual Penetration Testing**
   Experts manually running the program. Internal penetration testing identifies how much damage someone can do from inside the trust barrier. External penetration testing identifies access points and how far a hacker can get into a system

All these methods are important steps towards improving the security of your web applications. Yet these are just empty exercises if they are not followed with systematic remediation of the issues they help identify.

WITH RASP TECHNOLOGY, ORGANIZATIONS CAN:

1. Instantly reduce risk of exploitation
   - In vulnerable, or outdated applications
   - In applications for which you don't have remediation resources
   - In all mission critical web applications and web services
2. Prevent account takeover and reduce time to detection of stolen accounts
3. Add security to rapid DevOps iterations
4. Collect application security intelligence
5. Reduce waste in the software development cycle by providing evidence on exploitable vulnerabilities found during security testing, or bug bounty programs.

START HERE: RUNTIME APPLICATION SELF-PROTECTION

The board meeting is over. You've made your case and now things are largely out of your control. The CEO has taken charge of the remediation effort and hired an external consultancy experienced in breaches like this to lead the way. The marketing team is already working on a press release that discusses sophisticated methods used in this attack, saying there was little more that could have been done to change the outcome.

Was this episode really inevitable? Is it possible that with all the advances in software security at our disposal, we must simply shrug our shoulders and admit defeat?

Experience with the traditional web application security technologies mentioned above has led to the development of new approaches to application security such as Runtime Application Self-Protection (RASP). Remediation of code can be a tedious and lengthy process. The right approach is to develop security components that prevent exploitation of inherently insecure or outdated software and to integrate them with the application itself.

RASP is a category of technology solutions that monitors the interaction of the application and its users, and the behavior of the application itself, by monitoring critical junctions inside the application. This approach allows for a better understanding of the inner workings and vulnerabilities of the app. Most importantly, it allows for protective actions to be embedded inside applications so that they can protect themselves.

RASP allows for protective actions to be embedded inside applications so that they can protect themselves.

One of the biggest sources of vulnerabilities stems from tricking software applications into executing actions they were not designed for. Unfortunately, software today is so complex that there are many different ways to abuse functionality and services of software applications in order to:

- Undermine the integrity of the application code or its host.
- Take over user accounts and gain access to private information, some of which may be sensitive.
- Plant malware and use the compromised application as a springboard for other attacks.

Instantly Reduce Risk of Exploitation

RASP technology as developed by IMMUNIO is capable of monitoring not just the HTTP traffic and data sent to the application, but also of monitoring the reaction of your applications to a given input. Any anomalies detected during normal operation can be categorized and appropriate mitigation actions can be applied to prevent data leaks and execution of scripts, system commands, and database commands that can compromise the system. This means that you can add a protection layer to mitigate built-in weaknesses instantly, giving your engineering teams the opportunity to manage their workload.

Prevent Account Takeover

One of the common first steps in successful exploitation of application vulnerabilities is access to as much functionality as allowed by the privileges of stolen accounts. Without valid account information, the attack surface is limited. Using RASP can help you prevent armies of bots deployed by attackers from overtaking user accounts and it can also help you detect compromised accounts, allowing you to slow down fraudulent activities and make it difficult for attackers to run criminal activities through your software.

Add Security to DevOps

Many new technologies rely on cloud resources (private and public) to accelerate

RASP provides a safety net for security weaknesses that may remain undetected during quick iterations and prevent those from being exploited.

and automate software development and delivery. Often, cloud developments demand acquisition of new skills, new technologies, new processes, and a refresh of the whole software development cycle. Rapid development cycles are a big challenge for application security methods that require time for end-to-end testing -- manual, or automated with traditional SAST and DAST solutions. Runtime Application Self-Protection provides a safety net for security weaknesses that may remain undetected during quick iterations and prevent those from being exploited.

Application Security Intelligence

Network security solutions such as Web Application Firewalls have no insight into the correct functioning of applications. Such methods can address standard and well known risks, but they cannot prevent exploitation of weaknesses that are not well understood, or that may have just been introduced in that last code refresh. RASP can augment network-based solutions with application-specific information about users, their accounts, and application functionality. It can also allow you to learn about the system and correlate application-specific data with information from other systems to get a full understanding of intrusions and their impact.

Reduce Waste in the Software Development Lifecycle

Testing on developer machines and in test environments with traditional methods is often riddled with false positives and true positives that may not actually be exploitable when deployed in production. It is absolutely critical to improve quality of the code over time, but it also makes it difficult to identify and address critical issues that are easily exploitable in production environments. RASP, with its ability to prevent exploitation, can provide evidence of weaknesses in production and improve prioritization of vulnerabilities and flaws detected earlier in the development cycle.

CONCLUSION

Web application security is complex. The threat landscape is changing all the time and web applications are increasingly the attack vector of choice for hackers.

You are already implementing big changes for your organization (think cloud, mobile, APIs, etc.) and more open to considering new solutions. What you've been doing for the last 10 years is not enough in today's world. To reduce risk, you would be remiss not taking advantage of RASP to protect your organization and its assets.

ABOUT IMMUNIO

IMMUNIO is a pioneer in real-time web application security (RASP), providing automatic detection and protection against application security vulnerabilities. The company's mission is to make truly effective real-time web protection technology easily available and widely deployed, and by doing so, stop the biggest source of breached data records.

For more information, visit **https://www.immun.io/** *or follow* **@immunio**.