Establishing a Software Ecosystem that Produces Security

By Jeff Williams
October 8, 2010

Abstract: What if the key to efficiently and reliably producing secure code is not better tools or processes, but our software development culture? In this paper, we examine the reasons why software ecosystems systematically discourage security, and what organizations can do about them. We suggest that the most important thing an organization can do is to influence their software development ecosystems to ensure that security is visible, collaborative, and measured. A healthy software ecosystem will enable builders and breakers to iterate quickly, improving security and building history. To give the ecosystem direction, we suggest creating selective pressure for code with both strength and simplicity. Anyone interested in exploring this idea is encouraged to join us at OWASP.

Application Security Is Not Going to Just Happen

Over the last 30 years, software has advanced dramatically, yet the lack of progress in security is stunning. Multiple studies have shown that the vast majority of applications have serious vulnerabilities. Many of these flaws have been well understood for ten to twenty years and yet we are still making the same design and programming mistakes over and over. As a society, we have not been able to drive any of these weaknesses to extinction – even in new software, much less our legacy code.

The future of software is more complexity, more connections, and more critical information. All of these factors make security more difficult. At the same time, attacks like RBS WorldPay, Aurora, and Stuxnet demonstrate that the sophistication of the attackers is clearly rising. We have a perfect storm coming.

Virtually every business on the planet trusts their business to code. These businesses have had to accept the risk that this software is flawed or malicious because they feel that they must in order to remain competitive. Most organizations do not fully appreciate the risk that they are taking.

Nevertheless, Microsoft and a few others have shown that secure software development techniques can cut costs and schedule while also reducing risk. Creating a secure platform also creates a competitive advantage and enables innovation. Despite these advantages, many organizations are unable or unwilling to change the way they create software.

With roughly one trillion lines of code already in existence and over fifteen million developers writing more code every day, our challenge is clear. We need to help every organization that writes code to create and nourish a software ecosystem that produces security along with its code.

Doomed to Repeat

The evidence is pretty clear that we are not making progress. Every generation of technology seems to experience the same problems with injection, spoofing, tampering, etc... For example, when web applications emerged, nobody took advantage of the advances in access control from the mainframe era. Later, when web services arrived, most projects charged forward without considering the lessons learned from fielding web applications. The same thing happened more recently when projects started to add Ajax. Why are we incapable of carrying forward what we have learned about authentication, authorization, input validation, encryption, logging, etc… to new technologies? How many times do we have to repeat this cycle before we learn that mixing up code and data is too dangerous to allow?

To understand the difficulty of the problem, consider just a few of the things we have tried. In the late 1980's, the NSA led the National Computer Security Center. They published the Rainbow Series, held huge national conferences, funded trusted operating systems and databases, and created a system of evaluation labs. Later, in the mid-1990's, almost 50 companies came together and created a maturity model for security engineering called the SSE-CMM which later became an ISO standard.

Since that time, thousands of security researchers have published vulnerabilities in an effort to get software producers to improve security. Even today, OWASP is publishing free and open tools and materials, hosting conferences, and creating local chapters worldwide to help organizations produce more secure code.

Nevertheless, without minimizing any of the accomplishments of these efforts, we have to recognize that none of them has substantially affected the way the world's software gets created. With vulnerability rates rising at alarming rates, we cannot afford to simply study what companies are doing today. There is a clear need to do something different, yet recent "game-changing" proposals amount to nothing more than pursuing some of these failed strategies again.

Today's Security Ecosystem

Unfortunately, our current software ecosystem discourages security. This is true within small development teams, entire corporations, countries, and the entire world. It applies equally to ecosystems focused on a business, specific technology, or even an industry.

One key reason is that people currently tend to trust software without any evidence that it is secure, including test results, design documentation, details of who built it, or anything else. Virtually all the websites on the Internet have a privacy page, but vanishingly few have a security page that might help you determine whether or not the application is safe enough to use.

There is something about software that drives people to blindly trust it until someone discovers a vulnerability that proves it is insecure. Once the vulnerability is patched, people return to blind trust again. Even in the face of massive evidence that new technologies are very likely to have flaws, we instinctively move to new technologies as soon as they are available. The compiled nature of software makes it difficult to investigate and explore, and in the face of this uncertainty, unfortunately, we seem to assume security.

Paradoxically, this trust leads to a culture of minimal security verification. Why look for vulnerabilities when you fundamentally believe something is secure? There are plenty of ways to get more assurance, but our ecosystem seems fairly comfortable relying on sketchiest of evidence. Blind trust also puts security professionals in a difficult position when they discover vulnerabilities. The risk associated with these vulnerabilities is often discounted, forcing many researchers into hyperbolic claims to get noticed.

Despite the fact that the entire software ecosystem encourages insecure software, developers are frequently blamed for these mistakes. Have you noticed the distrust and skepticism between developers and security teams? By and large, developers are extremely motivated to write secure code and even accept help, but blame in the ecosystem makes it very difficult for developers and security experts to collaborate.

The natural reaction of developers who are getting blamed for security problems is to hide details. They naturally do not want to participate in the security process or share the information that is needed to make security decisions. In some cases, the blame goes as far as calling for liability for security mistakes. This is probably the most divisive thing we could possibly do and the death knell for application security.

So blind trust leads to blame, blame leads to hiding, and hiding necessitates blind trust. This cycle is toxic to generating assurance in our software development ecosystems.

No Simple Fix

The culture of insecure software appears to be a very strong attractor, resisting our most impressive efforts to change the way things work. However, that does not necessarily mean that the system cannot change, just that we have not figured out how to do it.

Many organizations try get in front of application security using penetration testing. Unfortunately, because testing happens well after an application is created, there are usually no lasting effects on the ecosystem. Organizations also try using static analysis tools, dynamic scanners, compliance, education and training, process models, threat modeling, and other techniques. None of these techniques focus on resolving the fundamental problems in our software ecosystems.

Some organizations do succeed at taking control of application security, but far more programs have no substantial long lasting effect. What if the culture change *itself* is responsible for these results, and not any of the individual techniques applied? Perhaps we should focus on changing the culture and not blindly applying tools and activities.

To change the system, we are going to have to be more thoughtful about how our software ecosystems work and what leverage points we can manipulate so that they produce more secure software. Readers who work in security might think of this as a strange kind of penetration test with the goal of finding weaknesses that we can use to establish a new dynamic that produces security.

Mother Nature Knows Security

In security, as in many areas, nature has evolved some of the most successful techniques. From turtle shells to bird warnings to detachable lizard tails, security is a natural byproduct of a functional

ecosystem. In fact, many species develop defenses specifically to defend against attack techniques of other species. This battle is carried out on an evolutionary timescale. Attacks and defenses are engaged in a never-ending co-evolutionary spiral, always in flux yet frequently in an almost beautiful but strange balance.

Notice that nature does not just jump to a solution. She sets up the whole evolutionary process in order to continually optimize based on the current set of selective pressures. Security works the same way. When your builders design and implement a security control, that is just the first step. Then it is time for your breakers to start exploring and finding weaknesses. When they do, your builders can respond with improved protections. This is the pattern that produces security.

So security is not a product *or* a process. Security can be thought of as an emergent property, an artifact, created by the operation of an entire ecosystem of people, processes, and technologies. Can we hijack nature's approach and use it to force the evolution of security more quickly and permanently than today's model?

Let a Thousand Security Ecosystems Bloom

The way to get security is to influence our software ecosystems so that they will produce it. This idea is scale-free and applies whether we are dealing with a huge ecosystem like our entire software market or to smaller more focused ecosystems. One might be focused on a particular technology, like Java or mobile. Others might focus on a certain type of security control, like input validation or access control. Still others might focus on the risks in a particular line of business.

Every time a new technology is created, we need to bootstrap a security ecosystem as soon as we can and influence its growth with the lessons from the past. If you imagine that there ought to be a single place to go to if you want to find out about some aspect of security – whether it is inside your company or on the Internet – that is an ecosystem that ought to exist.

The Open Web Application Security project is an ecosystem focused primarily on the risks to web applications. Founded in 2000, OWASP brings together thousands of builders and breakers to advance the state of the art in application security and get it to the developers that need it. In the past few years, there has been an increasing amount of specialization within the OWASP ecosystem. Whole communities focused on specific areas of application security like cloud, mobile, web services, etc…

It is important to realize that software and security ecosystems are not walled gardens. They are nested, interlinked, and overlapping in a million complex ways. This is actually a good sign – it happens as an initial ecosystem evolves and moves beyond the core concerns that caused the ecosystem to be started in the first place. The core concerns do not go away, but much of the knowledgebase solidifies and becomes a part of the foundation. Understanding this process is how we can measure the progress in an ecosystem. The current effort to align all of the OWASP guides is evidence that this is occurring.

A Working Security Ecosystem

One great example of a focused security ecosystem is the cryptography community. While it is not perfect, this loose confederation of professionals, researchers, and analysts has created steady security

progress over the years. There is a thriving community of academic and professionals that all work towards advancing the state of the art. Some of these researchers perform cryptanalysis – trying to find flaws in the existing body of knowledge. These are the breakers. Others are builders that focus on creating new algorithms and techniques to improve security.

Right now, NIST is sponsoring a competition to see who can come up with the best new hash algorithm. It is a rigorous process and will eventually produce a new standard. This is an amazing way to force progress in an ecosystem and generate security as an artifact. When you look at areas where we have actually made progress in security, you will find a thriving ecosystem around it. When there is no history, there is no security.

Now, there are also problems with the cryptography ecosystem. Currently, much of the focus is on creating stronger and stronger algorithms and very little on practical application of the technology. For years, people have complained that risks come from the management of keys and certificates. This is exactly why it is so important to make sure your ecosystem is designed to produce the results you want.

OWASP's Python Security Ecosystem

OWASP is itself a large ecosystem focused on improving application security. But recently we have started several smaller ecosystems to focus on particular issues. For example, some OWASP contributors noticed that while there were a few articles about Python and security on the web, there was no community…no ecosystem. So OWASP started pythonsecurity.org.

The first step was to gather existing information, tools, and articles from the Internet and start a knowledgebase. Then we created a forum for discussing ideas. The participants are discussing ideas for static analysis, taint tracking, and more. The group already has several contributors.

At some point, the ecosystem will achieve a "sustainable" level where it has enough people, projects, culture, and community to ensure its ongoing existence. Eventually, the ecosystem will achieve a "thriving" level where it is the single driving force in Python security.

Growing Security Ecosystems

When you think about how security works within your organization, try to think what drives the culture and whether there are ways you can encourage your security ecosystem to evolve faster in the right direction. This absolutely does not mean that you should establish Draconian policies and attempt to force them on developers. Rather, your job is to create the right conditions for security to evolve naturally.

One of the reasons that Agile has been so successful is that it recognizes human limitations and explicitly establishes an ecosystem that can evolve useful software. Below are a few of the characteristics that can encourage the development of healthy security ecosystems. Unlike security itself, these conditions are relatively easy to establish within your organization.

Ensure visibility. Remember that blame and the resultant information hiding are toxic to security ecosystems. Try to make sure that all the security-relevant information in your ecosystem is available to

whoever needs it. That way, people will be in a much better position to identify risks and make informed decisions about what to do about them. When information stays hidden, organizations are forced into taking blind risks. Security ecosystems cannot evolve without sunshine.

Create history. Capture the knowledge and decisions that you have made in the past. Otherwise your ecosystem will get stuck in a permanent loop, making the same decisions over and over without progress. There are a number of ways to create this history, including wikis, standards and guidelines, training, and forums.

Enable collaboration. To achieve security, the business, developers, and security folks are going to have to work together. By including both builders and breakers, the evolution required to achieve security can happen quickly. For example, Microsoft's BlueHat briefings are a chance for security researchers and Microsoft's engineers to work together on security. OWASP's Browser Security Working Group is another example of bringing together interested stakeholders to work on a common set of issues.

Select for strength and simplicity. Ecosystems can only evolve if they have some drivers for natural selection. Any security ecosystem should constantly seek out ways to make things stronger and simpler. Stronger means that there should be a constant search for weaknesses and a corresponding constant effort to eliminate weaknesses. Simpler means that it should be as simple as possible for developers to create secure code, and as simple as possible for verifiers to make sure that the developers did not make mistakes. The interaction of these builders and breakers will drive the evolution of your ecosystem and create security as a result.

React quickly. New threats and vulnerabilities emerge constantly. Both builders and breakers in a healthy security ecosystem will encourage investigation into these new developments and determine how to react. Just as natural ecosystems must adapt to changes in weather conditions and invading species, so too must security ecosystems recognize and adapt to new attacks.

Measure everything. This is an aspect of visibility, but it is worth repeating. Even a functional ecosystem will be eliminated if it is not producing demonstrable value. Reduction in risk is difficult to measure, so demonstrating faster software development time, increased sales, increased agility, or improved ability to innovate are all more likely to ensure the future of the ecosystem.

Consider what *your* software ecosystem is designed to optimize. If you are like most organizations, you may be inadvertently encouraging blame, obscurity, and reaction when it comes to security. Recognize that this creates a toxic environment for application security. Instead, if you focus your application security program on creating an ecosystem that encourages the characteristics above and you will see security take root, grow, and thrive.

Call to Action

We are very early in the process of investigating what creates successful security ecosystems, but OWASP has started a project to create ecosystems and study what works. We are interested in finding out why some organizations are successful while others that perform the same practices and use the same tools fail. We are optimistic that we can accelerate application security by focusing on getting the

conditions right for it to evolve on its own. You can join the project by contacting the author or just visiting OWASP.

About the Author

Jeff Williams is the founder and CEO of Aspect Security and serves as the volunteer Chair of the Open Web Application Security Project (OWASP). Aspect has helped dozens of organizations establish a healthy application security ecosystem that encourages the creation of secure code. Jeff has created many popular free and open source projects at OWASP, including the Top Ten, WebGoat, Secure Software Contract Annex, Enterprise Security API, Application Security Verification Standard, OWASP Risk Rating Methodology, and the worldwide local chapters program.

Contact Information:

Jeff Williams
Aspect Security, Inc.
9175 Guilford Road, Suite 300
Columbia, MD 21046
410-707-1487
jeff.williams@aspectsecurity.com
Twitter @planetlevel