ComputerWeekly.com TechTarget E-guide

CW+ Content

Application security: More important than ever

In this e-guide

In this e-guide:

The security of business applications is often overlooked, despite the fact that exploitation of vulnerabilities in software is one of the key attack methods of cyber criminals and that application breaches account for the majority of reported security incidents. But as traditional software and cloud-based, web and mobile applications play an increasingly important role in business and with applications associated with devices making of the internet of things set to explode, application security has never been more important than it is now.

While much of the responsibility lies with application developers to avoid common, exploitable coding practices and design secure code in the absence of any legislation in this area, there is much that businesses can and should do to mitigate the application security risk, including security testing all applications used by the business, reviewing source code, and layering a broad range of security controls to enhance visibility, alerting and real-time blocking.

Warwick Ashford, security editor

Application security more important than ever

Nicholas Fearn, guest contributor

Software applications play an important role in our lives. Whether it is in the home or workplace, we use them for communicating with people, staying up-to-date with the things happening in the world, keeping entertained, doing work and much more.

As an industry, apps are big money makers. Research from Statistica claims the apps market will be worth $189bn by 2020. In 2017, there were 2.8 million apps available in the Google Play Store and 2.2 million in the Apple App Store, but as smartphones, tablets and other connected devices continue to advance and more people buy them, the number of apps will only increase. At the same time, thousands of web applications and sites are created daily.

But while apps continue to deliver benefits, there are also challenges. In particular, connected devices and the software they power have become lucrative targets for hackers.

They are constantly using new and existing tactics to access, steal, change and delete personal and business data. According to research from Akamai, the number of application attacks grew by 63% in 2017, while 73% of security incidents flagged by Alert Logic were application breaches.

To protect users and data, application security has become an important consideration for businesses globally. When it comes to creating and releasing an app, developers must continually monitor, fix and prevent security vulnerabilities.

The most successful techniques work across the entire lifecycle of an application, including design, development, release and upgrade. However, as the internet of things (IoT) ecosystem grows, it is likely these attacks will only grow. Computer Weekly looks at what companies can and should be doing to prevent them.

A security epidemic

Scott Crawford, a security analyst at 451 Research, believes security threats arise because companies are using a diverse range of applications. Often, IT and security teams just do not have the resources or time to identify and respond to attacks.

“Software vulnerabilities remain one of the primary exposures of IT to threats – and are among the most challenging problems for organisations to tackle successfully, at multiple levels. Part of the issue is that the challenge is spread across multiple domains including Cots [commercial off-the-shelf] products, bespoke applications and systems, and embedded software,” he says.

Software in each such domain may have multiple moving parts, notes Crawford. “Endpoint components as well as server-side and back-end functionality, and not infrequently logic in between, as with edge computing in IoT, for example.

“Cots suppliers have come a long way in addressing vulnerabilities in their products, but the deployment of updates often faces many hurdles in organisations. To say ‘just patch it’ reveals a profound lack of understanding of those challenges,” he says.

“Bespoke applications, meanwhile, are undergoing radical change as organisations move from traditional ‘waterfall’ approaches to development toward the more agile techniques of DevOps.”

"Software embedded in other technologies now includes multiple systems encountered in the physical realm, from sophisticated healthcare systems and industrial controls to cars, homes and everyday objects. The logic embedded in these systems may be difficult to protect, let alone update – and technology turnover may take years. Businesses must plan today for the security not just of tomorrow, but for several years that may be – quite literally – down the road."

Fostering security by design

Lev Lesokhin, senior vice-president of strategy and analytics at software intelligence firm CAST, says that many application vulnerabilities are caused by architectural design flaws and that developers need to weave security techniques into the code of their apps.

"As automation and smart software become an increasingly large part of IT systems across several industries, the way these machines are programmed needs to be carefully analysed," he says.

"If developers do not pay attention to the code they write for these systems, the effects could be far-reaching. A chatbot that is not responsive and intuitive is useless. Code added to artificial Intelligence may have the prejudice of its developers. The risks are endless."

Security is an essential requirement for any type of digital business, says Lesokhin. "In a sense, it's a hygiene factor. It must be present, although it seldom contributes directly to the primary business functionality of the system. However, poorly-chosen security approaches can certainly impede usability and efficiency.

"An organisation's overall approach to security (involving culture, technology and processes) can have a major impact on its agility and, hence, ability to innovate in digital business. The best security organisations see their job as making innovation safe – an attitude that is very supportive of digital business initiatives."

David Smith, chief information security officer at forensic data software firm Nuix, says many hackers are tapping into existing techniques to compromise badly designed applications.

"We are still seeing a lot of the same techniques to hack applications as we have previously seen," he says. "For example, buffer overflows, along with poor coding still remain two of the biggest application security issues. In addition, we have found many organisations are still not applying encryption very well.

"The US has a governmental standard for encryption, which ensures the encryption is being put in place correctly," says Smith. "Many are not applying this properly, which results in applications with flawed security."

Another technique commonly used by hackers trying to get around application security is called fuzzing. "Its power is in its simplicity," says Smith.

"Any time there is an opportunity to enter information, hackers will try entering non-expected information to find a loophole. For example, if a field is asking you for a UK postal code, hackers may try entering more characters than what is expected, and if the program is coded poorly, it may let them in. This works more often than you may think."

New application security threats

Alex Ayers, head of application security at information services firm Wolters Kluwer UK, says the rise of technologies such as artificial intelligence (AI) and the internet of things (IoT) will introduce new application security threats.

"While many application security failures are due to well-known threats, there are always new attack vectors or new ways of using known exploits," he says. "The commoditisation of AI, machine learning and hacking as a service [HaaS] greatly increases the risk to running applications with easily discoverable and well-known vulnerabilities. The ongoing inability to bring cyber criminals to justice does nothing to deter increasingly sophisticated attacks, potentially bolstered by the availability of sophisticated tools developed by government agencies."

However, while there are plenty of ways companies and IT managers can prevent such threats, Ayers believes that they need to be underpinned by strong industry standards and investment.

"Technology is evolving along with the threat landscape," he says. "Firewalling technology is advancing and can provide the ability to rapidly mitigate new threats. Tools to protect applications at runtime are becoming more popular and development frameworks, such as Angular, are building in protection against common vulnerability classes."

But Ayers cautions that tooling is of limited use without the standards, policy and practices to ensure that it is being used effectively. "Projects such as such as OWASP SAMM and BSIMM enable organisations to consistently and reliably measure and improve the security of their software," he says. "It is not for want of information, tools or techniques that applications are vulnerable to long-term and upcoming threats."

Combating threats is less a technical challenge than a management one, according to Ayers. "It is easy for organisations to talk good security, but the resilience of applications will always significantly lag threats unless there is investment in implementing processes for measuring, improving and embedding security within the development pipeline."

Recovery plans

Omid Shiraji, chief information officer of Camden Council, agrees that application security threats are always evolving. He believes that companies and technology teams should develop and implement sophisticated recovery plans to respond to hack attempts.

“We see the current threat vectors evolving so fast and malware becoming increasingly complex,” he says. “In a world where your fridge or lightbulb may be a method of attack, you’ll never be able to build a wall high enough to keep the bad guys out. The investment for UK organisations should be on recovery rather than protection.”

Educating UK boardrooms about the complexity of modern day threats, pragmatically and without scaremongering, is a key role for the technology and data leader, says Shiraji. “More chief information and chief data officers sitting on boards will help keep cyber awareness as a key organisational priority and protect and strengthen an organisation's reputation.”

Cyber criminals are continually coming up with new ways to compromise application infrastructure. Mark Hill, chief information officer at Frank Recruitment Group, says firms should be aware of fraud and hack attempts targeting employees.

“The threat from cyber crime is growing and the perpetrators are no longer using basic ‘spray and prey’ techniques. The lack of inter-government cooperation, and the belief that cyber crime is without risk, is simply fuelling this trend,” he says.

“Frank Recruitment Group takes the emerging security threats very seriously. We strive to continuously improve all of the defences and application defences to protect our both our company and our customers’ data. The key threat we see today is in the extensive and targeted use of confidence scams, which tries to tricks unwitting employees into handing over some form of data or access to systems.”

Digital transformation

To identify and respond to threats, the recruitment firm has undergone a digital transformation drive. “We work closely with our staff to continually raise the awareness of security threats,” he says.

“We only work with Tier-1 cloud applications providers with multi-factor authentication mechanisms, who themselves invest very heavily in security, to a level much more than we could ever realistically afford to do, much the same as any sensible mid-sized business.”

Clearly, the number of applications used by the general public and business world is not going to stop growing anytime soon. But the same thing could be said for hack attempts.

Although hackers are still using existing methods to gain access to applications and steal data, they are also generating new ways to stay under the radar and bypass security systems. Because of this, firms and IT managers need to take proactive and consistent measures to prevent and minimise damage.

▼ Next Article

Application and device security under the spotlight

Peter Allison, guest contributor

Until recently, device security has been of minimal concern. However, recent events – such as hundreds of thousands of IoT devices being co-opted into a botnet and the American casino that had data leaked through a smart fish tank – have highlighted the necessity for robust device security measures for this digitally-connected world that organisations now operate in.

It has been reported that by 2020 there will be 40 billion devices connected to the internet. Every device is a potential threat vector for organisations.

To date, there has been little focus on the security of devices and the applications or software they use. Most of the development has been focused on features and battery life. There has also been little collaboration between manufacturers.

“There are thousands of IoT devices and they are all built differently, they are not using common standards,” says Darron Gibbard, chief technical officer of Qualys. “It is almost like the computer industry twenty years ago, where they are not thinking about security.”

Despite the increasing number of hacks and leaks that have been caused by vulnerable applications and smart devices, security is still not as much of a priority as it should be. Part of this down to suppliers – some believe there are no incentives for them to invest in application security.

However, poor app security leads to reputational damage. For example, some people are still refusing to use the Sony Playstation Network since the 2011 cyber attack, despite the security measures Sony have since put in place. With new UK data protection legislation due to come into force soon, companies will also become more liable for looking after their customer data.

Currently, there is no certification for encouraging suppliers to provide adequate levels of security in their devices and their applications. ISO/IEC 27001 (Information Security Management) is the closest, but it only provides information about how good the supplier/manufacturer is at protecting themselves, not about the security of their products.

The EU's General Data Protection Regulation (GDPR), which will soon be enshrined in UK law, goes part of the way to address this. While it does not define any standards of expected security, it does focus on privacy by design and the use of personal data.

Because suppliers do not generally advertise the security features of their devices, it is up to purchasing organisations to vet and research potential new devices. "CTOs should look to partner with CISOs to build in security by design into their platform and their vendor choices, rather than seeing the CISOs as gatekeepers or even blockers," says Uri Sarid, chief technical officer of MuleSoft.

"Instead of trying to place stringent requirements on the business, which they will likely circumvent to avoid project delays, IT leaders must look to structure their application networks in a way where they can create defence-in-depth."

Organisations may wish to consider employing independent parties to assess new devices and their software before installing them on their network. "There are lots of specialist organisations that will run a penetration test, which will divulge any possible threats," says Gibbard. "That would be something I would always recommend."

The UK government recently published the Secure by Design review, which forms part of the government's Digital Charter and the National Cyber Security Strategy (2016-2021). The risks posed by poorly-secured IT products and services threaten an individual's security and privacy, and can also form parts of large-scale cyber attacks. Such attacks reverberate across the UK and the global economy.

While none of the recommendations in the Secure by Design review are compulsory, the government is strongly advocating that all manufacturers should comply with these best practices. The code of practice was constructed with the hope that a proposed trustmark scheme would align with the recommendations.

The government's review also considers a product-labelling scheme to make buyers aware of a product's security features at the point of purchase. As part of achieving this, the government proposes a voluntary labelling scheme for consumer internet of things (IoT) products to aid purchasing decisions and to facilitate consumer trust in companies. This product labelling would identify if the product is internet connected, the product's minimum support period and privacy-related information.

These recommendations could eventually form part of a possible certification scheme, similar to the BSI Kitemark quality certification for products meeting stringent safety requirements. When internet-connected devices are found to have adhered to the Secure by Design best practice guidelines, they would be awarded a mark of recognition. For this mark to be recognised, such products would need to be independently tested to ensure they meet the requirements.

At the moment, the guidelines set out in the Secure by Design review are purely voluntary, but there could be possible legislative and regulatory requirements in the future. Paragraph 5.19 of the Secure by Design review states that:

"The government has begun exploring where we can further leverage existing legislative measures to place selected guidelines from the Code of Practice on a regulatory footing. Parts of the Code of Practice are already legally enforceable based on the legal requirements set out in new UK data protection legislation. The government will continue this work throughout 2018 in consultation with stakeholders, such as industry and consumer organisations."

The government is also apparently monitoring the regulatory action taken by other countries, such as Germany, where the German government recently banned children's smart watches. "The government's preference is for the market to regulate itself, and follow the steps set out in the code of practice," said a spokesperson for the Home Office.

"However, due to the ongoing spread of insecure devices, the government will be exploring options throughout 2018 to examine where key guidelines from the Code of Practice, which are not already legally enforceable, can be placed on a regulatory footing."

If developers were prepared to invest in their product's security features, this would become a marketable asset that organisations would be prepared to pay for. "It is giving you a degree of assurance that the device has been tested and built to a standard," says Gibbard. "As a percentage, I would be willing to pay an extra 10% to 20% for that assurance."

With the UK government considering legislating best practice guidelines as a statutory requirement, developers need to consider making security an important part of the development process. This could also become, with appropriate marketing, a selling point for new devices and services.

Likewise, purchasing organisations need to proactively investigate the security of prospective purchases, as well as considering pen testing on any new purchases before committing, to ensure they are reasonably secure.

"Organisations should look for products and vendors that have security by design baked in," says Sarid. "As a CTO, when I look at incorporating a technology or product in something important, especially if it's foundational to where I'm going, I want to understand in what sense is security baked into the supplier."

Nothing is ever 100% secure, but with careful network management and device access rights, the risk can be mitigated. "I want my suppliers to add security," he says. "Not to add vulnerabilities."

▼ Next Article

Application security vulnerabilities are often known exploits

Cameron McKenzie, guest contributor

All software developers are trained in secure coding. Making sure deployed software is safe from various nefarious threats is always a top priority. Or, at least, it should be a top priority.

Many IT teams only pay lip service to security. The fact remains that the most common application security vulnerabilities tend to be well-known exploits that developers can easily code around.

Microservices security

It's hard to believe that basic access control continues to be an issue, but in this new age of containers and microservices, it's not uncommon at all for developers to package and deploy server-side software components without any access control settings at all. Annotation-based security, a fine feature that should make applying access control easier, not harder, is often to blame.

With annotations, Java classes and their contained methods need only be decorated with a small piece of text, which subsequently applies security. When this is done correctly, access control problems disappear. But the simplicity of annotation-based access control can have devastating consequences. Developers can get into trouble if they delete an annotation while troubleshooting and then forget to code it back in before deployment.

"Just leave out an annotation, and all of a sudden, you've got a microservice that's not protected," said Jeff Williams, CTO and co-founder of Contrast Security.

It's a rookie mistake to make, but it's a problem that arises in the most professional of environments. Another well-known exploit that should never appear on an assessment of application security vulnerabilities is the cross-site request forgery (CSRF) issue.

"Basically, the solution to CSRF is to add a token to your forms and links in order to prevent them from being forged," Williams said. "And most people either don't do it or they don't do it right."

CSRF attacks will succeed despite the fact that precautions to prevent and thwart such attacks are well-known and well-documented.

New scripts, same security vulnerabilities

It's almost embarrassing to mention it, but injection techniques -- be it SQL injection, Lightweight Directory Access Protocol injection or command-line injection -- are all regularly flagged when Angular front ends and microservices back ends get scanned for application security vulnerabilities. All it takes to prevent most of these injection problems is to perform simple validations on any text your applications might consume -- whether that text is garnered from the client through an online submission form or information embedded within a JSON string. The need to validate incoming text for escape sequences and executable scripts is not a new revelation to the world of software development, yet it's a task that even experienced developers routinely skip.

So, why are all of these easily addressable application security vulnerabilities still regularly appearing in production deployments? Williams said we can blame much of it on the shift toward new JavaScript frameworks based on the model-view-controller methodology, like

Angular and Ember. Not that these frameworks have any inherent security flaws, but instead, the simple shift from one programming paradigm to another has made developers forget about the basics.

"When we moved from pure web apps with HTML interfaces to web services with Angular front ends, we ended up exposing all of these services," Williams said. "It's as though people just forgot all of the security-related best practices they had previously learned."

It's not a problem unique to the realm of security either. Organizations somehow start to omit compliance rules that were rarely overlooked in the traditional data center when they move to the cloud.

It's as though people just forgot all of the security-related best practices they had previously learned.

CTO, Contrast Security

Jeff Williams

"How do you expire passwords? Do you have your firewall turned on? Do you have mandatory access controls on the system machine?" Julian Dunn, director of product marketing at Chef, said. "These best practices are easy to ignore when you're deploying into that environment."

Don't trust vendors with your security

But Dunn doesn't attribute compliance or security omissions to negligence by software engineers but, instead, attributes it to the way various vendors provision their cloud computing environments to their clients.

"When a cloud vendor provides you its API gateway and they tell you that it has a web allocation firewall in front, you immediately think, 'Oh, somebody else is accountable for that, right?'" Dunn said.

The result? Well-meaning software developers assume the firewall meets all of the compliance requirements or that it will meet all of the organization's security needs. But this type of blind faith in the cloud provider can create serious shortcomings in terms of security and compliance.

Williams also asserted that some of the existing tools intended to detect security holes are failing to deliver and took particular aim at existing static analysis tools. "Static analysis tools take too long to run; they require experts in order to operate them and interpret the results, and they produce a huge number of false positives. They also miss a lot," Williams said. "If it takes a week to get your application scanned and another week to interpret the results, that's just way too long."

Cloud computing and the new, modular, container-based systems used for developing and deploying microservices and serverless applications have significantly changed the way applications are developed. But that doesn't mean security and compliance best practices should be abandoned. In fact, going back to first principles and making sure all of the basics are covered will go a long way to ensure deployed applications meet all of the required compliance standards, while, at the same time, are free of any oblivious application security vulnerabilities.

▼ **Next Article**

In this e-guide

Five critical tests for cloud application performance, security

Kenneth Milberg, guest contributor

When it comes to cloud application performance, you never want to just cross your fingers and hope for the best. Instead, development teams should regularly, and thoroughly, conduct tests to ensure applications meet the expectations of end users and the business.

While app testing can be tedious, it's worth it. Otherwise, poor performance and security vulnerabilities can negatively affect your bottom line. Here's an overview of five types of tests that are crucial to ensure high-performing and secure cloud applications, as well as some tools that can help you conduct them.

Load tests

Stability is a major factor for cloud applications, given that they often need to support hundreds -- or even thousands -- of simultaneous users. A load test is a good way for development teams to determine how cloud applications run under varying loads and user requests. Enterprises should run this test regularly when the load is high to accurately measure application response time.

Public cloud providers offer tools to help perform load tests. The Microsoft Azure portal, for example, integrates with Visual Studio Team Services, which you can use to test cloud application performance. With a few clicks, the service simulates workloads for users who visit your website, and then reports how many requests have failed or are slow to respond. Enterprises can then view these results in a real-time dashboard.

IBM integrates with a third-party tool, called Load Impact, to perform these kinds of tests on its cloud platform. Users create an account and then conduct and track both load and stress tests via the Load Impact Dashboard.

Stress tests

> **Stability is a major factor for cloud applications, given that they often need to support hundreds -- or even thousands -- of simultaneous users.**

Stress tests ensure that your public cloud-hosted applications can continue to be effective, even under excessive load or unfavorable circumstances. For example, retail organizations should perform a stress test before important events, such as Black Friday, to ensure the application can handle large traffic spikes. In most cases, a DevOps team will perform a stress test and handle the process of load simulation.

These types of tests are especially important in public cloud, given that it's a multi-tenant environment, where users share infrastructure. Cloud, in general, also offers a more cost-effective way to perform these kinds of tests, compared to a complex, on-premises testing lab.

Hewlett Packard Enterprise's LoadRunner is a well-known stress testing tool that can simulate thousands of concurrent users. Another example is Apache JMeter, which lets development teams actively simulate, and monitor, huge spikes in traffic.

Third-party load testing tools, such as Blitz, are also available.

Functional tests

Functional tests are all about user experience. They evaluate whether public cloud-hosted applications meet business requirements and work as intended. Two common types of functional tests include system unit testing and user acceptance testing.

With system tests, developers check that the individual modules of an application work properly -- a process that might include tests on some hardware and software components. User acceptance tests evaluate how the application performs for its intended, real-world audience. Users typically appreciate it when you involve them in this type of cloud application performance test because it shows that you care about their experience.

Enterprises should also ensure proper interoperability during functional tests and check that applications can run efficiently on different platforms, or as they move from one cloud provider to another.

Latency tests

Enterprises -- typically the networking team -- perform this type of test to ensure acceptable latency between an end-user request and cloud application response time. For example, a team could test the latency between your application's IP location and any Azure data center across the globe, using the Azure Latency Test service. Third-party tools, such as CloudHarmony, can also help assess latency.

Security tests

While not directly related to cloud application performance, security tests are also critical to prevent vulnerabilities from negatively affecting users. Penetration tests, for example, simulate activity from a malicious user to identify vulnerabilities, such as cross-site scripting.

You might have to get permission from your public cloud provider to run such tests, as they can sometimes be indistinguishable from real-life events. For example, in AWS, you need to complete the AWS Vulnerability/Penetration Testing Request Form before you begin. Enterprises can also opt to have Amazon perform a security vulnerability assessment on their behalf via the Amazon Inspector service.

Developers should always involve their security team before they begin to perform these types of tests.

▼ Next Article

How to manage application security risks and shortcomings

Kevin Beaver, guest contributor

Application security is arguably the most critical part of an overall security program.

Software is where business transactions happen. It's where sensitive information is captured, processed and stored. It's where a large -- arguably the largest -- number of security threats and flaws exist to create the perfect risk scenarios for criminals.

Still, organizations struggle to find -- and fix -- the big application security risks. This includes things such as SQL injection at the application layer for web and mobile applications, missing patches that facilitate remote access at the server layer, and even default or weak passwords at the database layer.

However, we can't blame the flaws or the hackers for what's happening, as it goes much deeper. First off, there's a serious lack of proper application testing. The three main areas of concern are as follows:

- No security testing at all -- including critical applications, this is the most common issue due in large part to assumptions that someone else is taking care of things;
- Basic vulnerability scans that don't properly address the application layer; and
- Not looking at source code -- an exercise that often reveals flaws that won't turn up with traditional vulnerability and penetration testing.

Secondly, there's often a lack of communication between developers and quality assurance (QA) analysts, security teams and management in general. The main gaps that I see are as follows:

- Assumptions are made: This can involve security standards, threat modeling and overall policy enforcement during the software development lifecycle and well into the application implementation phase.
- Bystander apathy: Not unlike the assumption above that a third-party vendor or cloud service provider is doing the proper testing, in many cases, developers and QA professionals assume that any security flaws will be uncovered downstream from their duties, such as once their code goes into production. At that point, it may be too late to find and fix the flaws.
- Misguided priorities: For example, when technical staff ask for the proper resources to do their security due diligence and they're shot down by management. However, if an independent outside party, such as a security auditor or customer performing its own testing on your environment, mentions anything in their report, it suddenly grabs the attention of management. Executives want to know why these issues weren't uncovered previously without seeing the bigger picture of their actions getting in the way.

Finally, I have found that many developers, QA analysts and security teams lack insight into the software security best practices and frameworks made available to us, such as the OWASP Top 10, the Common Weakness Enumeration and ISO 27034. Or they're unable to take the time to integrate such standards and practices into their application security program because of the time-to-market demands of other parts of the business.
Further compounding this, many of the very people in charge of application security that I speak with have never taken a course on the subject. That said, it's so easy to get caught up seeking out the best standards and having others tell you what you should be doing that you can overlook the gaping flaws right under your nose. Be careful, think for yourself and use common sense.

In the end, application security risks can be lessened via three methods.

> **That said, it's so easy to get caught up seeking out the best standards and having others tell you what you should be doing that you can overlook the gaping flaws right under your nose.**

- Know your environment: Everything from critical applications to low-risk applications, and all the information and processes they tie into, both internally and externally. Overlooking attack surfaces is one of the most dangerous oversights of security management. The golden rule of security always has been and always will be that *you cannot secure what you don't acknowledge*.
- Understand your application security risks: The technical vulnerabilities, process gaps and policy opportunities across the spectrum of threats your systems face. Getting others involved here to help determine priorities is key, as keeping risk analysis within IT circles works well until it doesn't. Once an incident or breach occurs and executive management and lawyers get involved, it's too late. Why not involve them from the get-go?
- Do something about it: Manage urgent issues on your most important systems first and, eventually, manage the issues on all the systems that are accessible via web, mobile or source code. No matter how much I analyze such situations, I can't come up with a single excuse for not doing something about known flaws that can create tangible business risks.
  Fingers can be pointed at management for not providing enough resources or IT/security teams for not having enough time -- or discipline -- to get things done. In the end, no one wants to hear it. Cut the nonsense. Stop having meetings that spawn more meetings and, thus, giving people an excuse to kick the application security can down the road. Do what needs to be done and do it now.

Whether it's in the early design or development phases, production phases, or ongoing maintenance of legacy applications, you have to develop some structure around your application security program with all the right people to help eliminate application security risks. Nothing more, nothing less.

Be it a stand-alone program or part of your overall information security efforts, if you treat application security as the critical business function that it is, you'll get good results. If you don't, well, it'll be more of the same over and over again.

▼ Next Article

How layered security can help and hinder application security

Kevin Beaver, guest contributor

In IT circles, we often hear -- and preach -- about the benefits of using layered security defenses to minimize information risks. After all, having layered security is a core tenet of information security dating back several decades to the origins of IT.

There are layers at the endpoint, layers at the network perimeter and, today, layers out to mobile and into the cloud. That's what's needed for a secure network, right?

Yes, generally speaking. However, in terms of application security, layered security can come up short in the areas of visibility and control -- especially when they're not properly implemented, which is often the case.

I've tested many web and mobile application environments over the years, and it doesn't seem to matter whether strong network security controls are in place. Even the best filtering and blocking technologies can't prevent application vulnerabilities from being exploited. This is largely due to the way SSL/TLS encryption can mask communication streams. Compensating controls often miss the obvious because they have no insight into what's happening in those streams.

Even if attacks can be seen, exploits can be carried out via a good old-fashioned web browser or HTTP proxy to make it look like legitimate traffic. It's easy to assume that application attacks are going to generate a lot of noise, like denial-of-service attacks do, but that's not true. It is one thing to detect and block automated web vulnerability scans, but quite another to prevent web traffic that looks like everything else, especially if it's coming from a trusted user whose login credentials have been compromised.

For example, SQL injection is a very common application security flaw. However, it can be extremely difficult to detect and prevent without the most granular of controls that only a web application or next-generation firewall could offer.

Furthermore, based on my experience, most browser-to-server-based SQL queries run over SSL or TLS. The problem is not that the communication session is encrypted, but the fact that they're even allowed in the first place; there's very little that traditional network security controls can do to detect or mitigate such an exploit. The same goes for webpages that are vulnerable to malware injections because, unless malware protection is running on the server itself, the injection will likely go undetected.

Compensating controls often miss the obvious because they have no insight into what's happening.

There's a myriad of other technical security flaws in both web and mobile platforms that can be carried out in plain sight. These vulnerabilities can remain undetected for weeks or months, if not indefinitely. Looking at the soft side of application security, such as business logic, password policies and login mechanisms, there's little that can be done about these threats in terms of proactive prevention and response.

This highlights the importance of rooting out security flaws by other means, namely secure coding practices combined with proper vulnerability and penetration testing. You can have the best web application firewall, IPS or managed security service in place, but those mean nothing at layer 7 when they are poorly managed, and I'm not sure that will ever change.

A broad range of controls that serves to enhance visibility, alerting and real-time blocking can certainly benefit your application security efforts, but don't let it create a false sense of security, and know that traditional security controls will likely be insufficient for application security as a whole. On the other hand, software flaws need a different kind of attention -- you should know your application environment and where it's vulnerable, as the odds are good that there are some unique gaps that remain unaddressed -- layered security or not.