Inside Track Research Note

In association with

Managing Application Security Risk

Continuous application security matters

March 2017

In a nutshell

In an era when both your systems and the enemy's attack vectors are constantly evolving, the old paradigm of periodic application security testing is no longer valid.

A major unprotected security risk in most enterprises today concerns application software flaws or bugs, especially those in web-facing platforms. The big challenge is bringing 'continuous protection' to bear, because with constant change being the new normal in both software development and threat evolution, the old process of periodic tests and checks is no longer adequate. A key part of that challenge is risk analysis, using human insight and experience as well as automated tools, to ensure that high priority and important bugs and vulnerabilities are dealt with first.

The agile attackers

You might think your systems are safe – after all, they passed a penetration test last week, your firewalls are up to date and you patch weekly. But when your developers – entirely innocently! – roll out an application update tomorrow that adds calls to a third-party API and a commercial component library, they are going to introduce three new vulnerabilities, and by the day after, criminals will have exploited one of them and stolen your customer database.

The danger for the organisation is over-confidence in your armour.

That is the new world of 21st century cyber security, where the threat environment is constantly evolving as criminals combine technical and social attacks to find new chinks in an organisation's IT armour. The danger for the organisation is over-confidence in your armour and under-estimating your opponents.

The challenge of change

The fastest growing risk for many IT or cyber security departments is that, while attacks can happen at any time or even continuously, many of our defences have a significant sporadic or episodic element to them. That might for example be:

Many of our defences have a significant sporadic or episodic element to them.

- periodic updates to a network scanner's set of malware signatures or detection algorithms
- the weekly or monthly deployment of the latest security patches for our operating systems and applications
- refresher training for staff in how to recognise and resist social engineering attacks
- or hiring security professionals from outside to carry out occasional penetration tests (pen-tests, as they are often called).

New security vulnerabilities can take weeks or even months to be fixed.

Yet new security vulnerabilities are being discovered all the time. Throughout the days, weeks or even months between the criminal underworld discovering a vulnerability and the vendor responsible becoming aware and then issuing a fix, you are at risk. This is what is known as a zero-day attack.

At the same time, we are continually changing things ourselves, most especially at the application level. Developers deploy new versions of code and re-use vulnerable code, perpetuating the same holes from one part of an application to another. There are security patches, and of course the business needs are continually changing, as new products go on sale, old ones are withdrawn, and business processes evolve.

We need security tools that support our developers and administrators.

We therefore need security tools that both support our developers and administrators throughout the development cycle, and protect our systems continuously. One example is heuristics or behavioural analysis supplementing or even replacing signature-based detection of malware, on the grounds that even a piece of malicious code that has never been seen before must behave in specific ways in order to exfiltrate or damage data.

Another example is continuous security assessment tools, typically hosted remotely, that dynamically scan your websites and web applications for changes. They then simulate the actions of attackers, in particular those aimed at exploiting run-time vulnerabilities.

We also need access to security skills, both to understand the risks and to advise on how best to remediate the security problems that our tools have detected.

DevOps and constant change are focusing the enemy's attention on the applications layer.

More focus needed on application attacks

As we close off other avenues of attack around the infrastructure, more and more of the enemy's attention is turning to the application layer. There are many reasons for this. One is the process of continuous change discussed above, especially as many organisations adopt agile and DevOps methodologies in order to become more flexible and responsive.

With these methodologies accelerating change, and with many different system elements to work on, your developers could potentially be doing tens or hundreds of builds a day, and frequent releases into the production environment. Add the development-speeding use of ready-made components such as commercial or open source function libraries and SDKs, or remote API calls to third-party services, and your web applications could have vulnerabilities lurking anywhere.

You need to know which application vulnerabilities could cause your organisation the most harm if breached.

Managing web application security is therefore a complex task, and it becomes even more complicated the more websites you have. Simply spotting which applications and which sites have problems and need remedial action is a challenge, and then you need to know which of the applications is the most vulnerable, and which could cause your organisation the greatest harm if it were breached. This requires advanced security knowledge and risk analysis skills, allied to scanning tools.

Built-in, not bolted-on

As part of the above, an organisation must help its developers make sure that application security is built-in right from the start, not bolted-on as an afterthought. Just as manufacturing quality managers know that it is far cheaper to detect and fix problems early in the production process, rather than have to repair finished products, so it is with application security.

In programming terms, built-in security means:

- Training your developers in security skills and best-practices
- Application code scanning, both dynamic and static
- Pre-deployment testing
- Aligning your security testing with your quality control.

While code may not be insecure today, it could become vulnerable tomorrow.

However, even skilled and trained developers can sometimes be outflanked by change. Code may reach production that, while it is not insecure today, could become vulnerable tomorrow as the threat community develops new attack vectors, hence the need for continuous security assessment and dynamic code scanning.

Security software that scans for flaws can therefore save costs, not only by spotting new risks before the software goes live, but also by taking some of the load off your developers. They still need to be security-aware, but automated tools can offload much of the heavy lifting in terms of keeping up to date with the latest vulnerabilities and advising on remediation.

Regardless of how it happens, if a security flaw does get through into production, then fixing it will very probably need to take a higher priority than fixing a user-interface bug, say. However, it is clear that the latter still needs to be fixed, as do any other security bugs that might be found – the problem then becomes one of prioritisation and triage by an experienced analyst, or according to rules and policies they have set.

Effective monitoring and protection requires a whole range of measures.

Covering the bases

It becomes clear then that effective application monitoring and protection requires a wide range of measures to be put into place. The following table isn't exhaustive, but it illustrates the main areas of security that need to be covered, and where the main types of cyber defences fit. Most often, it's the elements on the right-hand side of the coverage matrix that are found to be the weakest when application security measures and processes are reviewed objectively, i.e. the big challenge is the pieces to do with the continuous and ever-changing nature of the threat.

APPLICATION COVERAGE MATRIX	PRE-RELEASE PREVENTION AND REMEDIAL WORK			POINT-IN-TIME LIVE TEST	CONTINUOUS PROTECTION AND ASSESSMENT OF LIVE ENVIRONMENT		
	Prevention of security related defects in code	Detection of possible vulnerabilities in code	Assessment and triage of possible vulnerabilities	Detection of possible vulnerabilities in live website	Blocking of malicious network-level traffic	Detection of possible vulnerabilities in code	Assessment and triage of vulnerabilities in live apps
Staff knowledge, time and effort	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
Coding standards and best practices	✓	(✓)	(✓)	✗	✗	✗	✗
On-demand code scanning solutions	(✓)	✓	(✓)	✗	✗	✗	✗
Software build and test automation tools	(✓)	(✓)	✗	✗	✗	✗	✗
Integrated application testing suites	(✓)	✓	(✓)	(✓)	✗	(✓)	(✓)
Web application firewall	✗	✗	✗	(✓)	✓	✗	✗
Classic full pen testing services	✗	✗	✗	✓	(✓)	(✓)	(✓)
Continuous security assessment services	(✓)	✓	✓	(✓)	✗	✓	✓

✓ Suitable as a primary enabler

(✓) Adds value or supports, but generally inefficient or falls short as a primary enabler

✗ Little or no useful contribution

While these tools – and indeed cyber security tools in general – might well appear to overlap, this is both deliberate and necessary. No single defence will be perfect, so a degree of overlap and redundancy is good. If an attack manages to evade one layer or type of defence, there is a better chance that it will be picked up by the next one.

Many of the solutions we see here are typically hosted remotely and bought in the form of a managed service. Managed services have a number of advantages, most notably that they give shared access to scarce and expensive security skills – a security services provider could have dozens or even hundreds of security engineers and analysts on staff, for example.

Managed services bring an external perspective, untainted by assumptions and politics.

Such services also bring an external perspective, untainted by the assumptions and politics inherent in most organisations. A key element of that outside perspective is application risk analysis, derived from a wider industry knowledge and awareness, to help determine which vulnerabilities are the most dangerous and urgent.

Automated vs manual – or can you have both?

Most security tools rely upon automated processes. Both attackers and pen-testers will also use automatic probes to look for common network vulnerabilities or application bugs, before deciding upon their course of action. All this automation is only to be expected – there are just too many possibilities that must be checked, and there is too much traffic for anyone to find meaningful patterns in it without technological assistance.

Few organisations will be able to afford to find and hire sufficiently skilled and experienced analysts themselves.

But as we hinted above, at some point manual activity must join in. Just as an attacker will go first for the opportunities that present the greatest chance of success or reward, so must the defender fix the biggest or most dangerous faults first. In an environment where an automated continuous security assessment scanner could turn up dozens or even hundreds of potential bugs, but the resources to fix them are limited, the prioritisation of fixes is essential.

That means you need an analyst to make an informed judgement on how important a new vulnerability or bug is, based on their experience and human insight. Once this has been done, there is the potential to automate subsequent responses. However, as mentioned above, few organisations will be able to afford to find and hire sufficiently skilled and experienced analysts themselves, which is a good reason for taking the managed service route. Security service providers are also able to aggregate data across their customer base, for example to provide metrics on which applications are the most vulnerable, how long a fault might take to fix, and so on. But local context still needs to be applied.

Security service providers are able to aggregate data across their customer base to provide metrics on which applications are the most vulnerable.

The bottom line

In an era of agility and change, application-level defences based on periodic security checks and tests are fundamentally inadequate. Nor is application security entirely covered by the more general layers of infrastructure cyber security, which are designed to protect the network and its systems – software flaws are something else altogether.

At the same time, your application developers and testers are busy people, doing their best. With the right training and tools they can catch logic errors and many other security flaws. What they cannot do is predict all the interactions that can take place

With constant change and limited resources, well-informed triage is key to determine which code fixes are the most urgent.

in a complex system, determine what order to fix bugs in, or anticipate the actions of cyber criminals in a threat environment that is in constant flux.

So you need application security that combines dynamic scans with a practical and experience-based approach to remediation. When constant change is normal but resources are limited, the order of the day must be well-informed prioritisation – triage, to borrow a medical term – to determine which of the many code fixes needed are the most urgent.

It's no longer enough to secure the perimeter.

About Freeform Dynamics

Freeform Dynamics is an IT industry analyst firm. Through our research and insights, we aim to help busy IT and business professionals get up to speed on the latest technology developments, and make better-informed investment decisions.

For more information, and access to our library of free research, please visit www.freeformdynamics.com.

About WhiteHat Security

WhiteHat Security has been in the business of securing web applications for 15 years. Combining advanced technology with the expertise of its global Threat Research Center (TRC) team, WhiteHat delivers application security solutions that reduce risk, reduce cost and accelerate the deployment of secure applications and web sites. The company's flagship product, WhiteHat Sentinel, is a software-as-a-service platform providing dynamic application security testing (DAST), static application security testing (SAST), and mobile application security assessments. The company is headquartered in Santa Clara, Calif., with regional offices across the U.S. and Europe.

Terms of Use