myredfort community

Bright

Application Development

and its contribution to Security Debt

a myredfort eBook for Bright Security

With applications driving the global economy, developers are under pressure to deliver software and more features at an unprecedented scale and speed.

While no developer wants to create insecure products, most software products are pushed into production with vulnerabilities that stay unremediated causing a spiralling technical and security debt and significant risk for the organisation.

Application security scanning frequency is key, with development teams that scan for security issues early and often mean they substantially reduce their security debt.

Here we look at Security Debt and discuss how early and how often you should be scanning.

What is Security Debt?

Security debt is the continuing accumulation of security vulnerabilities in your software that compound to make it harder (read: impossible) to catch up with remediation to secure your applications and data from attacks.

Unlike technical debt, which may get in the way of releasing new features for the needs of the business, the growing pile of security vulnerabilities puts your organisation at an increased risk of cyber attack.

According to Forrester, the average time to resolve a high vulnerability in production is 4 months

Indeed, in many cases, medium to high severity vulnerabilities are being deferred, including issues like XSS, SQLi and others in the OWASP Top 10.

If you use Forrester's stats as a benchmark, you could be placing your entire business at risk for 4 months which is unthinkable!

How is Security Debt caused?

Security debt is caused when security testing is not baked across the software development life cycle (SDLC), accumulating when development releases software without testing for or fixing vulnerabilities.

With most organisations carrying out periodic (monthly, quarterly, annually) automated, or manual security testing, they decide to release now and fix vulnerabilities later. This results in an increased risk of the exposure until the issues are remediated.

> Two in three CISOs believe technical debt - the difference between what's needed in a project and what's finally deployed - to be a significant cause of security vulnerability, according to the **2021 Voice of the CISO Report**

The main issue is that 'later' keeps on getting pushed back and in many cases, 'later' becomes 'never', making security debt even worse.

When and how often should you be scanning?

Your Security Debt should be treated just like your Credit Card debt – if you keep spending and don't pay off your monthly balance, eventually it will lead to bankruptcy.

With the sheer volume of iterations to applications and APIs produced daily, security testing needs to mirror this cadence, to prevent a security breach and potential bankruptcy too!

Heavy, periodic scanning and quick remediation over a defined limited period to meet a release deadline, forces you to defer issues and add to your security debt.

DevOps and DevSecOps focus on enabling organisations to detect and fix security vulnerabilities as early and as often as possible in the software development life cycle (SDLC).

This mindset, where everyone is responsible for security, has broken down the barriers between developers, QA and security, facilitated by security champions who know what good looks like in terms of security.

With the increased velocity of development, comes an accelerated introduction of vulnerabilities. Security testing and remediation need to become a habitual process and part of your accelerated pipelines. Automation of daily security testing is critical to establishing a cadence of secure software.

The advantages of daily scanning

Here we take a look at what a difference daily scanning will make in **reducing security debt** :

PERIODIC SCANNING:	DAILY SCANNING:
Typically carried out manually	Integrated across the CICD with automation
Reactive – security handed off by developers. 'Tick box' compliance-based scanning by siloed teams	Proactive – Culture of security where Dev, QA and Sec work together enhancing DevOps/DevSecOps
Carried out in bursts – monthly, quarterly, annually	Frequent, regular testing on every build/commit or master merge
Finds large numbers of vulnerabilities very late, often in production	Finds vulnerabilities early to be fixed at 'source'
Too many accumulated issues are difficult to prioritise	Reduced, bite-size load makes prioritisation of vulnerability fixes easier
Increased deferral of remediation	Reduced deferral of remediation
Slow fix rate	10 times faster fix rate than periodic
Risky security posture	Secure by design approach reduces cyber risk
Drain on resources and expensive to remediate issues	Cheapest and most efficient time to remediate issues
Heavy reliance on costly penetration testing	Reduces reliance on and cost of manual penetration testing
5 x increase in security debt	Reduces Security Debt

With regular testing on every build/commit, or at least daily, everyone can be focused on making better security decisions as part of a unified strategy to deliver software with speed, efficiency and security.

Relying on manual testing simply cannot keep up with accelerated development timelines. The success of this strategy relies on development teams having easy to use, accurate and seamlessly integrated automated testing technology.

Traditional legacy Dynamic Application Security Testing (DAST) tools are not built for this regular cadence of security testing that demands speed.

Bright Security's innovative security scanner, Nexploit, is built for organisations of all sizes, whether a start-up, scale-up or enterprise organisation. With no false positives, it has a developer-first approach, to enable you to effectively integrate security scanning on every build/commit or to enable even immature teams to run effective security testing without the need to be a cyber security expert, to reduce security debt and be more secure. This removes reliance on and the cost of manual testing, too.

Automatically test every aspect of an App

Here are the key advantages of using Nexploit from Bright Security:

Automatically tests every aspect of your Apps

Scans any target, whether Web Apps, APIs (REST. & SOAP, GraphQL & more), Web sockets or mobile, providing actionable reports

Seamlessly integrates with the tools and workflows already in use

Works with existing CI/CD pipelines – triggers scans on every commit, pull request or build with unit testing

Super-fast scans

Interacts with applications and APIs instead of just crawling them and guessing.

Scans are fast because the AI-powered engine understands application architecture and generate sophisticated and targeted attacks

Spin-up, configure and control scans with code

- One file
- One command
- One scan
- No UI needed

Bright Security: Our story

Gadi Bashvitz, COO and President of Bright Security tells us how it all started and why he thinks it's a game-changer.

"Traditional Application Security Testing isn't keeping up and focuses on detecting known vulnerabilities. Legacy tools rely on a heuristics-based approach and lengthy and costly manual testing for finding new issues. This doesn't scale and results in substantial delays to remediation, putting your business at risk."

Bar Hofesh and Art Linkov decided to do something about it. They combined their experience in cyber security and biologically-inspired machine learning, creating Bright Security's AIAST technology, which automates a human's critical thinking process when detecting vulnerabilities.

"We think the results speak for themselves with a Dynamic Dynamic Application Security Testing (DAST) solution that fully automates AppSec testing at scale, allowing organisations of all sizes to stay ahead of even the most ruthless of hackers. It lets them comprehensively test, assess and improve their cybersecurity posture regardless of industry, including software, blockchain, FinTech, IoT, automotive, healthcare, and more."