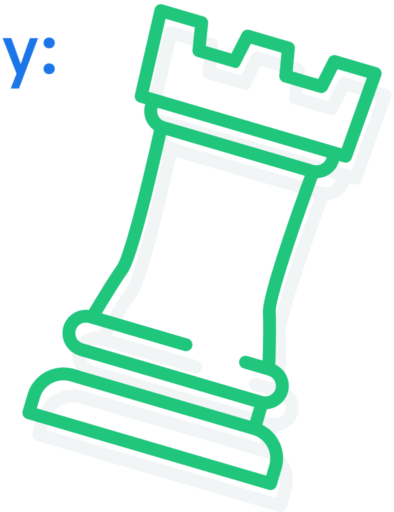
Cobalt
Pen Testing as a Service

Guide to Gamifying Security: It Takes Security + Dev



By Ante Gulam

Whenever I discuss trends with industry peers, the general impression is that this continuous security adaptation journey feels like a technological rollercoaster that is only picking up speed. We've all been witnessing the significant increase of delivery speed from the engineering perspective and the ever-increasing number of adopters of lean and agile/flow principles, fail-fast and fail-forward methodologies etc.

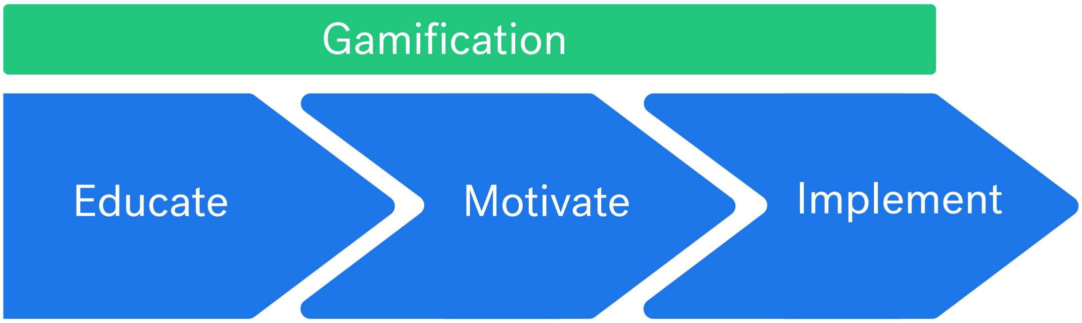
Funny enough, this doesn't just apply to only small and mid-size, dynamic tech companies. Even large enterprises, including some of the largest financial institution, are realizing that they will have to adapt to this new technical landscape and start shipping faster if they intend to preserve and improve their competitiveness in the market. Most of us within the security industry often get that 'we're gonna need a bigger boat' feeling every time we hear about a new layer of abstraction, M&A strategy, platform expansion, deployment pipeline re-design, etc.

The challenge here is not that we need one but rather it's the fact that most of the time we don't know what size boat or even what type of boat will work best (more resources, more tooling, stricter governance and controls, etc.). So it comes down to how can we actually scale security across engineering teams while keeping the TTR (time-to-remediate) metrics under control?

AppSec: Engineering Matters

Looking at various stats across the industry relating to security breaches its quite easy to conclude that almost all of the security issues (~95%) are caused by human error. From the application security perspective everything comes down to engineers and their ability to write code in a secure way and in accordance to industry best practices. In order for them to do that, two key ingredients are necessary.

First being awareness, engineers need to be continuously educated on security and taught how to write safe code and design secure solutions. The second ingredient is more intangible and is based on their motivation to actually do security. As everything else in life, for an individual to do their best to perform above and beyond their regular duties one has to leverage the top of Maslow's pyramid of needs. We need to understand that not all engineers have to be interested in security and feel passionate about it, and not all organizational cultures or regular engineering duties will be involved. This means we need to explore other ways to bring security into their remit. There needs to be a solution that will raise the business importance, potentially through clear and continuous visibility, as well as make it fun. As the definition of gamification states: "gamification is exciting because it promises to make the hard stuff in life fun".



If you educate and motivate your engineering teams then they will implement what they've learned.

Key Ingredients for Gamifying Security

Obviously, gamifying security across the board might not be right for every single organization and it might, at times, present an additional level of disruption for engineering teams. That is why its implementation model and understanding the fundamental components of the program are critical. The role of security should be to educate teams and enable continuous visibility to support them on their mission.

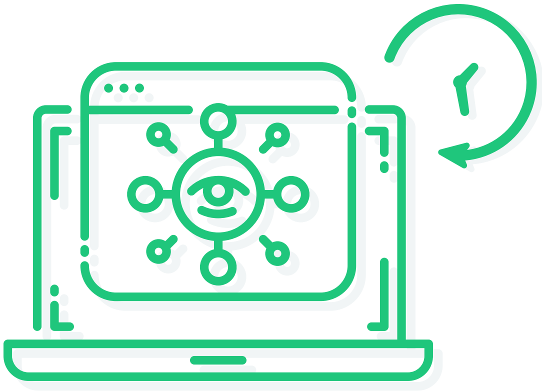
This means that we need to finally do the tool-centric to culture-centric strategic shift and have the ability to establish a real-time feedback loop. Shifting from gatekeeping to giving engineering the ability to become security sufficient. In order to bring security closer to engineers it is crucial to avoid ad-hoc engagements and develop capabilities for a targeted security model. I believe that most of us would agree that generalized approaches are not effective enough unless we want to focus on basic security hygiene. Having a general security education and training program means that every single team across the organization takes the same course that is based upon general security metrics collected across the board. Instead, we need the capability to monitor and track metrics for specific teams, or even individuals, and target a particular program based on data-driven decision making.

Continuous Visibility + Real-time Feedback Loop

For example, when security reaches out to engineering teams once every month, quarter, or year this breaks up the engineering continuum and turns security into an imposter. Raising remediation tickets after your regular security assessment is completed has two negative side effects. It breaks their momentum of shipping software in a frictionless way while also transforming security into a cultural obstacle.

These negative effects became the main drivers for experimenting with security gamification. If the model is designed and implemented properly it can enable the organization to tackle complex challenges involved with scaling security teams while making it fun for the the engineering teams.

Many organizations are combining tools and services in their deployment pipeline in order to ensure adequate security measures. It is important to normalize and standardize the way we monitor risk across teams so they can be presented as a whole and not as tool-specific snowflakes. Secondly, it is important to provide engineering teams with a real-time overview of the risk across different business units and teams as well as the overall business security posture (risk-based maturity model).



Customizing Security Education for Specific Teams

As previously mentioned, I've never seen an efficient security education and awareness program that was made generic across an organization. Every company has engineering teams with a more advanced skill set, for example, teams that are utilizing more advanced or sophisticated technologies etc. Having the ability to collect data and use it to shape a security training program (e.g. content, frequency) that is specific for certain teams or individuals is one of the most effective ways for assuring continuous improvement. If we could also automate this process somehow, then we might just be on the right track to tackling the security first principle challenge.

Ideally, the position that we should try to get ourselves into could be seen through the following scenario: Let's assume that a certain engineering team has made n deployments during the period t. We have performed various reactive checks over the code throughout the SDLC and those deployments contain X true positive security flaws out of which 90% of them are input sanitization vulnerabilities.

This case should trigger two main actions. First being the remediation activity, where we need to mitigate the risk and notify the teams as soon as possible based on criticality (automated through the real-time feedback loop-- ChatOps). While the second action should trigger mandatory training for that team/individuals which is focused on that vulnerability type/s as to make sure they are being educated on that specific topic.

Gamification: Creating Competition and Rewarding Security Practices

Once we have laid out the foundations and enabled continuous visibility of risk scores for our engineering teams we should start consider sharing these scores so that everyone is aware of their own security performance. This can and should involve a reward scheme for teams with tops scores, but should also be designed in a way that does not introduce potential disruptions or distractions from their main workstreams.

In the next installment of this series,
I will go into detail about the execution
of a program as described above.

About the Author

Ante Gulam is an Information Security professional with a strong technical background and over 15 years of progressive experience in the security industry across a wide range of B2B and B2C sectors. Ante has lead infosec teams at leading-edge technology companies where he fused security with innovation and business operations.

He has significant contributions to various public and private bug bounty programs as an independent, application security researcher. In addition, he has independently conducted and presented information security research at security conferences worldwide including OWASP Summit, ISN, CyberEurope, DevSecCon Singapore, PCI London, eCrime Dubai, FIC Lille, FSec, etc.

Visit Cobalt.io to learn more about how Pen Testing as a Service can help gamify your teams