ComplianceCow

ComplianceCow: Re-inventing Security GRC

(Security GRC meets Cloud Policy Engine on Slack/Teams)

"No controls left behind"

Security GRC needs a do-over

Security Governance, Risk and Compliance is fundamentally broken. Existing GRC products suck, and they only focus on "User workflow automation". These traditional, boring GRC products allow users to store files, provide a framework for controls, allow users to assign controls to one another and, most importantly, produce dashboard views of compliance.
This solves "once a year, check the box" compliance problems, however it largely does not address how Security Assurance (they are also called "security and compliance", "IT compliance" and "security trust") teams can manage IT or Security risks continuously.

Do companies manage IT and security risks continuously today?

No. Security GRC is an expression of trust.

On average, a mid-market company spends $3m to $5m in labor costs every year on IT and security compliance and assurance. The purpose of any IT and security framework (SOC2, PCI-DSS, FedRAMP, OWASP, MiTRE ATT&CK, CIS v8, NIST CSF etc.) is to create trust in the operations of the company. The controls specified in these frameworks are a typical reflection of the risks that a company faces. Identifying, protecting, detecting, responding, and correcting these risks in a meaningful and timely manner is critical.
But the reality is quite different. The frameworks and the attestations are a mere formality done once a year with expensive auditors. They are far removed from the continuous risks that a company faces. For the employees it is simply about doing enough not to get fired. No wonder IT and security compliance is no longer about assurance and trust but is about checking the boxes once a year.

So, is Security GRC even relevant?

Software development is a highly creative process. Building high performance, highly distributable software is very hard and is error prone. This truth, coupled with the democratization of software through open source, makes IT systems vulnerable. According to Mandiant, more than 50% of the cybersecurity attacks are neither prevented nor detected and less than 20% of CISOs surveyed were confident of their enterprise security posture. Standardization of deployment architecture on Cloud (particularly on Kubernetes) and adopting: zero trust approaches, SecDevOps tool chains and GitOps reduces risk exposure and improves posture. However, given the nature of software development, it will always be a whack-a-mole game. The recent examples of solorigate and Log4j2 serve as a good reminder. Deploying more security tools without understanding the context can ironically increase the attack surface. It is analogous to having all the x- rays and the MRIs but not knowing where to look and how to interpret the results. Security teams need a balanced approach of prevention and detection. In particularly they need to look at the context holistically. The IT and security compliance frameworks do a great job of providing the "physiology" of the security apparatus and not just its "anatomy". We believe that an independent, timely assessment of security through a holistic framework supports a defense in depth mechanism (and that does not contradict with zero trust). In fact, this is the idea behind the 3 lines of defense. But it doesn't help if the lines of defense do not operate at the speed of DevOps.

Why is automating IT and Security GRC so dang hard?

1. Going from "intent" to "implementation": IT and security assurance is contextual. For example, "limit administrative or console access to critical services" (say, SSH, RDP) can be implemented many different ways; implementing a firewall rule, jumping through an LDAP bastion host, dropping the packets at eBPF filter etc. depending on how the company has designed and architected the solution. Any out-of-the-box product will have to automate all possible paths to fully automate a simple control. Standardized deployments on Cloud, and particularly on Kubernetes, help narrow the number of integrations with cloud native services and third-party tools. However, there still needs to be a customer/vendor partnership; one in which vendors can offer pre-built solutions and allow customers to seamlessly customize or even develop complex rules for their context

2. Lack of policy automation tools: Many of the current tools offer a simplistic view of automating security compliance policies. For example, Palo Alto Networks Prisma Cloud uses a SIEM + Query approach (they use Google Big Query as the data store and a proprietary RQL query language) to define policies. Azure Security Center Policies take a similar approach; Log Analytics workspace + KQL. This approach is very limiting as the policy statement can only be expressed as a SQL statement. This absolutely limits the modularity, readability, and reusability of the policy code. Beyond “marketing”, these tools are pretty much useless in managing IT and security assurance
3. Friction with old GRC tools: Security GRC is a combination of automatable and manual verifications tasks. Only 40-50% of the controls are automatable for evidence collection and control testing. The rest requires the SMEs in security, platform engineering and IT operations to collaborate with the GRC analyst. It is painful. These teams are already stretched thin, and they have no interest nor time to deal with GRC platforms, let alone GRC staff.
4. Where is G&R in C: When IT and security teams operate at the speed of DevOps, the GRC team assesses once a year. The risk assessment is already outdated even before it has begun. Unless the GRC tooling provides consumable APIs for a) the DevOps teams to operate and b) allow data engineers to analyze and create risk signals, it will continue to be meaningless. Consequently, there is no incentive to operate GRC for its intended purpose

How does ComplianceCow redefine Security GRC?

To solve these problems, we will need to have the following capabilities:

1. A robust rules engine that will not only have a pre-built catalog of IT and security rules but also allow you to configure, customize or even develop new ones that are specific to your organization. The reality is that there is no “magic one size fits all” tool because IT implementations, whether they are on cloud or on-premises, are too complex and custom
2. You need an API-first approach that will not only allow you to execute these policies on-demand but also allows you to seamlessly integrate downstream with your DevOps tool chain
3. You need a data-first approach where you can consume evidence and control test results to produce better, intelligent security signals to integrate with the reporting tool of your choice
4. You need to meet the user where s/he is; slack or teams. Who has the time to learn another set of applications just to maintain their compliance posture?

5. You need a zero-vendor lock-in solution/approach to future-proof your investments
6. And finally, you need a smart, programmable platform that allows you to programmatically annotate and generate context specific operational and management reports

It should also seamlessly integrate with enterprise tools.

None of the above is possible with the current GRC tools for Security. We have built an API-first unique middleware platform significantly accelerates customers capabilities to deploy and manage security controls 10x faster, better, and cheaper.