IriusRisk

« Modeling threats in an imperfect world

AN UNCOMFORTABLE QUESTION

The internet is a hostile environment. Any application is under threat and despite the growth in investment in cybersecurity, the threats haven't stopped growing. The way an attacker can exploit a vulnerability hasn't changed significantly over the past decade, so now we are forced to ask ourselves an uncomfortable question: What are we doing wrong?

When faced with the task of defining a strategy that can help us resolve this problem, we should start with something more manageable, such as building a map. In it we will place the different elements that we want to protect, such as the flows that relate to them and the threats that endanger them. This map is sometimes referred to as a "threat model". One way to formalise this approach is by leveraging Adam Shostack's four-question scheme[1]:

BUILDING A MAP

"The mere formulation of a problem is far more often essential than its solution"
Albert Einstein

Figure 1. Adam Shostack's four-question scheme1 for threat modeling

Based on this model, we can detect security deficiencies during the design phase of the application. This is the natural place to identify and plan security enhancements since no code has yet been written. The NIST (National Institute of Standards and Technology) has estimated[2] that correcting code once an application is in production can take thirty times the time required for remediation, by implementing it in the design phase, the cost of remediation is significantly reduced. Incorporating security from the early stages of development is not only cheaper and more efficient, it also allows a communication model that can address the quality of the product as well as in erms of security. This approach eases the adoption of security enhancements as a shared challenge between the different departments involved in the development lifecycle.

SCALING THE MAP IN THE REAL WORLD

Threat Modeling has historically been a manual process, an exercise based on meetings between the security team and experts from different areas that are involved in development. This approach presents multiple problems if it needs to scale in an enterprise environment. The manual threat model generated is static and difficult to share between teams. Worse still, it doesn't integrate with development tools, which renders it difficult to include within an automation-based DevOps strategy. In order to deal with these challenges, a modern Threat Modeling tool must focus its efforts on:

- Being a single point of management for the security team. This way, they leverage an updated view of the risks within their portfolio, enabling metrics that help guide efforts within the security programme.

- Having an automated way of generating security requirements based on the application architecture model and its applicable standards (PCI, HIPAA, GDPR, etc.).

- Flexibility to adopt industry-specific risk models or customised security policies based on a pre-regulatory triage.

- Establishing a two-way communication with Application Lifecycle Management (ALM) tools used by development teams (JIRA, Redmine, etc.).

- API access that allows automation, like any other tool within a DevOps environment.

- Ability to dynamically update th risk model and requirements of the implementation strategy.

- Integrate with the main security tools used throughout the development cycle and thus feed the threat model with the vulnerabilities detected.

- Generating a visual diagram of the architecture with data flows and zones of trust, so that it serves as an active document among the different stakeholders involved in the development lifecycle.

- IriusRisk has been designed with these concepts in mind, pursuing one main objective: to provide a collaborative environment that simplifies the creation of actionable and dynamic threat models.

CONCLUSION

A Threat Modeling tool increases the productivity of business processes. Time to market is significantly enhanced with the added benefit of security being built since the design phase. In turn, this allows for better defences against risk scenarios that were previously modelled. It also makes it easier to think about risks in a structured way, creating a communication channel that helps to transfer the security culture all the way to the organisational level.

Although Threat Modeling is not a simple task, the alternative is to remain reactive. After all, the main problem is not having an incomplete threat model, the greatest risk is not knowing what threats are out there and not having a plan to mitigate them.

IriusRisk is a platform that simplifies the creation of threat models, allowing development teams to have security requirements adapted from the design phases so any security team can have a dynamic view of risk that is adapted to agile environments.

The platform uses a hybrid system to define the architecture based on interactive diagrams and forms. It uses a forward and backward chaining inference based rules engine rules which is fed to generate an adapted threat model, taking into account the state of implementation of the controls defined in task managers.

REFERENCES

1. «Threat Modeling, Design for Security», Adam Shostack, 2014. John Wiley & Sons Inc.

2. «NIST (2002): The Economic Impacts of Inadequate Infrastructure for Software Testing», https://www.nist.gov/sites/ default/files/documents/director/ planning/report02-3.pdf.

IriusRisk

www.iriusrisk.com
getintouch@iriusrisk.com