Rolling out Threat Modeling:

A guide to getting started ebook

IriusRisk

Table Of Contents:

Threat Modeling values and principles

This guide is to share tips and tricks for rolling out Threat Modeling in your organization. This collaborative document aims to structure and provide guidance on recommended approaches to rolling out Threat Modeling beyond the security team and enable secure design at scale within engineering teams.

Before introducing the Rollout Plan let's stop to check what Threat Modeling Manifesto is proposing and define:

- Values - what provides worth to your initiative
- Principles - fundamentals, "or general truths that enable successful threat modeling, patterns that are highly recommended, and anti-patterns that should be avoided".

Values

Let's summarize what we have come to value over a more classical approach:

Principles

Under Threat Modeling Manifesto there are some principles to be recommended: Improve the security and privacy of a system through early and frequent analysis. Align with an organization's development practices and follow design changes in iterations that are each scoped to manageable portions of the system.

The outcomes of threat modeling are meaningful when they are of value to stakeholders. Dialog is key to establishing the common understandings that lead to value, while documents record those understandings, and enable measurement.

Let's start with the guide.

What does a rollout plan look like?

A rollout plan can vary greatly in terms of the time it takes to complete. However, regardless of the size of your organization or the length of time required to roll out your threat modeling capability, the one common denominator between successful rollouts is a phased approach.

Phase 1 | Deliver value first

Think of rogue malware infiltrating your systems. Data preprocessing scans for missing values, inconsistencies, and typos – the glitches that could compromise your AI's analysis.

Phase 2 | Recruit your virtual team of Security Champions

One of the best methods of collaboration is to recruit Security Champions within Developer teams. These can be individuals who are friendly to your team and eager to collaborate, or Developers looking to increase their capability in the security space.

They should also be able to speak to each other freely, so it is beneficial to treat them as one single, virtual team, integrated with and acting as part of your own security team. You're aiming for collaboration, so having individuals from security and development in the same team is a good idea.

Phase 3 | Test small

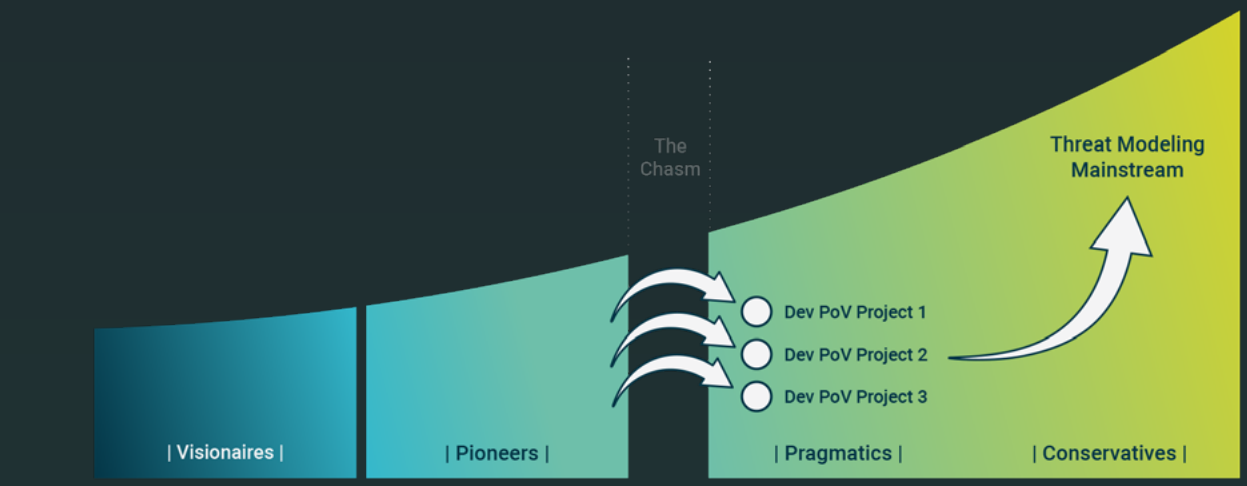
Running pilot programs is a great idea. Ideally, you will want to run one with a small realworld project, leveraging a team you've already demonstrated some value to and who are open to extending collaboration. It's important to collect the value from these teams as this pilot should be re-usable with other teams, gathering together the value that has been created as the ultimate output.

Phase 4 | Gather evidence and plan your next step

At each phase of your rollout, evidence of value should be a key focus. By this phase, you should be able to demonstrate the value of threat modeling and how it enables better collaboration, better security, and lower costs.

At this point, if they haven't been involved already, it may be a great idea to demonstrate the evidence of value to senior stakeholders, and further (or start) to build support from the top-down. This will be important in rolling out to the rest of the organization.

- Refine the Communication Plan and set the interval to remind threat modeling
- Create the threat modeling culture inside your organization's Dev community, and set the update based on a blog, news, etc.
- Update Security Champions continuously as part of your virtual team
- Refine goals, indicators based on threat modeling stakeholders and users' feedback
- Adopt new standards if they are needed
- Refine your Dashboard metrics
- Schedule regular sessions with threat modeling stakeholders



	Phase I	Phase II	Phase III	Phase IV
What	**Deliver value first**	**Recruit your virtual team of Security Champions**	**Start small, gain credibility**	**Gather evidence and plan**
Who	**Security Team**	**Security Team & Security Champions**	**Developers, Security Champions, Security Team**	**Developers, Architects, Compliance**
How	• Define roles, BU's, permissions • Define integrations • Select/Define Standards and Compliance • Define Customized Risk Patterns and Components • Define Workflows • Define value, principles & KPIs • Set Risk, Dashboard & Reports • Session with Stakeholders	• Define Issue Tracker level (component, threats) integrations • Select Security Champions • Check former & architect formats • Define customized components • Select Dev Teams for PoV • Refine goals • Dashboard: Set a Tool for Governance (Analytics module) • Session with Stakeholders	• Track Security Champion progress • Refine integrations • Refine Standards • Refine customized components • Refine Workflows • Set a Communication Plan (Notifications) • Refine goals, indicators • Refine Dashboard metrics • Session with Stakeholders (internal PoV results)	• Refine the Communication Plan • Update Security Champions • Refine goals, indicators • Adopt new standards • Refine Dashboard metrics • Regular sessions with stakeholders

What resources do I need to create?

It's important to create somewhere for all of your stakeholders that allows them to (at their convenience), learn about the goals of the rollout and how to use the tools that they will need to use to perform threat modeling, as well as how to collaborate with all of your other stakeholders.

Training for your team

First and foremost, your team will need to be able to enforce how your team uses your threat modeling tool, and ensure it is configured to make it easy for all of the wider user base to use effectively. So, your first focus should be on ensuring that your team knows where to go when they need to make changes.

Types of resources to consider:

- Threat modeling overview/purpose, strategic aims
- Process training
- Tooling training (some made in-house, some from IriusRisk materials).

Training for other teams

It's important to also use some of these resources with the other teams that will be involved, even if they won't be involved until a later stage. These should leverage the resources you've already created, but also some more team-specific (or at least area of business-specific) resources to help them understand the context.

Different approaches

Confluence/Wiki's are useful here as a repository to be shared. The type of resources to create for these are:

- Articles covering training, the project, the benefits etc.
- How-to guides to get people started quickly
- Screenshots of the tool to help guide people around how they should be using it
- Short videos of certain functions
- Slack channels/Microsoft Teams "teams" to reach out internally for guidance.

Gaining initial buy-in

It's important to bear in mind that opening threat modeling up to everyone in your organization is often an approach that is unsuccessful. There can be a number of reasons for this, including but not limited to:

- Teams who are not currently threat modeling might not realize the value in spending time doing it
- It might sound like extra work
- It's "someone else's responsibility" So, what can we do to get buy-in from other stakeholders?

Start with "why"

You're looking to roll out threat modeling in your organization for a very good reason. So, start with "why"!

- Limit discovery of security vulnerabilities late in the software development cycle
- Security is trying to make things easier for developers, and having the security discussion at the beginning means security requirements are understood early
- Increased collaboration makes development easier
- Help developers to understand security requirements better
- Save time by removing long workshops and processes
- Help speed up deployment by removing security bottlenecks

This is not an exhaustive list, but these are examples of benefits as to why we should all care about threat modeling.

Open with value, don't jump in!

Before you get your developers using IriusRisk, the best approach is to first introduce it into your conversations as a tool that the security team currently uses. Bring it into your current conversations and workshops - show the kinds of security requirements you are able to uncover with IriusRisk.

What to avoid: jumping straight in and getting your developers doing threat modeling straight away is likely to get strong pushback from them. The reason "why" they should be threat modeling is unlikely to be obvious to them straight away, so it's best to use threat modeling to have a collaborative conversation. Show them how you reach your conclusions and why the process is valuable.

Getting developers involved

So, we've defined our reasons why everyone should be threat modeling and we're starting to have conversations around threat modeling and secure design. Next up - how do we actively go about getting developers starting to do threat modeling?

Security Champions

Establishing a security champion programme is one of the most effective things you do to start to really roll out threat modeling. Simply speaking, it is a programme for recruiting developers and getting them to take the lead for threat modeling within their own development teams.

So what does the programme look like?

- Developers who have an interest in security
- Developers who are already friendly to your team can be a good choice
- They will become your security representation within their team
- Train them like you would your own security team
- They can help explain the value to their own team
- They can train their devs and act as a local/immediate contact for their team
- We want a representative from each Dev team - 1 from each team is enough

Virtual overlay teams

To effectively engage with your security champions, you should create a virtual overlay team. This will act as a virtual team that brings all of your security champions together. Remember, they are the sole security champion within their own development team. So, it's worth making them part of the team. There are other benefits too:

- Establish a two-way connection for feedback to and from the Dev teams
- Keep your security champions up to date with latest developments in the threat modeling process and the tools they use
- Get messaging out to the Dev team via the security champions
- Track the usage of threat modeling within the design process more effectively via the security champions
- Be able to provide your security champions the support they need. Remember: secure design is still the security team's responsibility, the champions are there to help you, not be responsible for implementation
- A genuine extension of your security team

Increasing buy-in from the top down

In the late phases of rolling out threat modeling, we want to ensure that the benefits that we have realized are being shared. So, it's important to highlight the benefits back up the chain of command.

Getting buy-in from the top down will ensure that the benefits of threat modeling are understood, which will be especially useful for ensuring that threat modeling is adopted as a core part of your design and security process.

Demonstrating collaboration

There are some useful ways of demonstrating the value threat modeling has had on your processes. Focus on contrasting the previous state with the new state of things now threat modeling is being performed. Things like:

- How long was the process of identifying security requirements before?
  - Multiple workshops with teams?
  - Having to manually identify threats from scratch?
  - Defining appropriate countermeasures?
- If you weren't threat modeling at all before, how many security requirements are you able to define at the start of the development process now?
- How much more secure does this make your organization?
- What was the relationship between Development and Security before?
  - Is it closer now?
  - Is it easier to communicate security requirements now?

Proving ROI

Proving aspects of ROI can be a powerful tool in gaining buy-in. Much like the above, it's useful to put figures together such as:

- Time spent before vs time spent now in creating security requirements
  - In the middle is time saved
- If you weren't doing secure design before, then when would we normally identify the threats vs. when we identify them now?
  - This is time and money saved by implementing each security requirement earlier in the SDLC.

If you are looking for specific figures for savings from IriusRisk, **download** the study commissioned by IriusRisk, where Forrester Consulting evaluates the Total Economic Impact™ of our Threat Modeling platform. Some key highlights include:

- Returned 203% ROI in efficiency over manual modeling
- Saved almost $5m in software remediation costs
- Saved $4m in reporting and compliance
- Reduced time to Threat Model from 80 hours to only 8 hours