**Threat Modeling Protocol for Non-Hierarchical Organizations**

**Objectives**

Non-hierarchical (horizontal) organizations lack a formal chain of command, which creates unique security challenges and requires a tailored threat modeling approach. The absence of centralized authority can be exploited by novel attack vectors – for example, adversaries may create fake identities (Sybil attacks) or manipulate consensus processes (quorum manipulation) to influence group decisions. The primary objective of this protocol is to strengthen the cyber resilience of decentralized organizations by treating *horizontality* as a strategic asset in security planning. In practice, this means leveraging the distributed nature of decision-making to eliminate single points of failure and equitably spread security responsibilities among members.

Equally important is aligning the security process with the organization’s democratic principles. The protocol is designed so that implementing security measures does not undermine participatory governance or slow down collective decision-making. Instead, it seeks to **balance operational efficiency with inclusive participation**, ensuring that all members can partake in identifying and addressing threats without compromising agility. By filling this gap in existing security practices, the protocol provides practical guidelines for identifying, mitigating, and preventing threats in contexts where distributed trust and collaboration are fundamental. In summary, the objective is a *security framework for horizontal organizations* that preserves their core values of transparency and shared power while proactively defending against threats.

**Methodology**

The protocol adopts a **participatory threat modeling methodology** that involves a broad range of stakeholders in identifying and analyzing threats. Security is approached not as a top-down mandate from a central authority, but as a collaborative process where members collectively map out potential vulnerabilities and attack vectors. To achieve this, the protocol adapts several established threat modeling techniques to the horizontal context. It blends traditional frameworks (such as Microsoft’s STRIDE categorization of threats and attack tree modeling) with inclusive, democratized practices. This hybrid approach ensures that classic security principles (e.g. covering spoofing, tampering, denial-of-service, etc.) are considered, while also capturing insights from the people most familiar with the organization’s unique workflows and assets.

Crucially, the methodology integrates **security analysis with the organization’s governance processes**. Rather than treating threat modeling as a purely technical exercise, the protocol infuses it with the organization’s democratic decision-making mechanisms. For example, risk analysis and mitigation proposals are subject to transparent, auditable consensus among members. Key security policies – such as access control rules or incident response plans – are validated through collective agreement, using consensus-based procedures inspired by distributed trust models. This ensures that security controls have legitimacy and buy-in from the group, and it helps prevent threats that specifically target participatory systems (such as the insertion of malicious proposals or misuse of voting rights).

The methodology operates on both **technical and social levels** to cover the full spectrum of threats. On the technical side, robust cryptographic techniques are employed to safeguard data and communications from external attackers (for instance, end-to-end encryption, digital signatures, and distributed ledgers for tamper-proof record-keeping). On the social side, the protocol enforces radical transparency and peer oversight to deter and detect internal threats or misconfigurations. For example, decisions and security-related actions are logged on immutable records accessible to all members, and rotating review committees of members periodically audit these logs and configurations. This dual-layer approach ensures that no critical security decision is made in the dark; every member has the opportunity to scrutinize and contribute, thereby harnessing collective intelligence to spot vulnerabilities early. The combination of cryptographic safeguards with participatory oversight aligns with research showing that traceability and shared accountability help identify and correct weaknesses in distributed systems.

Attack modeling tools are also leveraged to facilitate collaborative analysis. In particular, **attack trees** are used to visually map out complex threat scenarios, which is especially useful in horizontal teams where threat knowledge may be distributed across members. Attack trees represent a potential attack goal as the root node and break down the various sub-goals or steps (children nodes) that an attacker could take to achieve that goal. This graphical technique enables the group to decompose a complicated attack into smaller components and consider multiple paths an adversary might follow. The figure below shows an example attack tree diagram, illustrating how different attack paths (e.g. network intrusions or credential theft methods) branch out towards an ultimate breach objective. Such visual aids allow members to collectively brainstorm “what-if” scenarios in a structured way, ensuring no major attack vector is overlooked in a decentralized environment. ([Blog - Analysing vulnerabilities with threat modelling using draw.io](https://www.drawio.com/blog/threat-modelling))

By combining these methodologies – stakeholder workshops, consensus-based analysis, and visual threat modeling – the protocol creates a comprehensive and inclusive picture of the threat landscape. This approach inherently embodies the principles of distributed governance: every phase of threat modeling, from identification to mitigation, encourages **transparency, participation, and shared responsibility**. The end result is a living security process that evolves with the organization, continuously informed by its members and resilient against both conventional cyber threats and those exploiting the unique nature of horizontal structures.

**Steps**

The following steps outline a structured process for applying the threat modeling protocol in a non-hierarchical organization. Each step emphasizes collective participation and aligns with the organization’s governance mechanisms:

1. **Threat Identification** – The organization collaboratively identifies potential threats to its assets, processes, and community. This begins with inclusive brainstorming sessions or workshops where members share concerns and past experiences (e.g. attempted breaches, social engineering incidents). By involving a diverse range of participants (technical and non-technical), the process uncovers vulnerabilities that a top-down approach might miss. Established frameworks can guide this stage; for example, using the STRIDE model as a checklist ensures the team considers threats across all categories (spoofing, tampering, repudiation, information disclosure, denial of service, and privilege escalation). The output of this step is a comprehensive list of threat scenarios and attack vectors that concern the group.
2. **Risk Assessment** – Next, the identified threats are collectively analyzed and prioritized based on risk. Members discuss the *impact* of each threat (how severely it could harm the organization’s operations, data, or trust) and the *likelihood* of its occurrence. The protocol encourages a consensus-driven risk rating, where the group agrees on which threats are most critical to address. Adopting a risk-centric approach (inspired by frameworks like PASTA) helps ensure that mitigation efforts align with the organization’s core objectives and values. In practice, this may involve scoring threats or categorizing them (e.g. high/medium/low risk) through open debate or voting. The key is that everyone understands which threats matter most to the collective mission, so resources and attention can be focused accordingly.
3. **Scenario Analysis** – For the top-priority threats, the team conducts an in-depth scenario analysis to understand how those threats could materialize. This involves mapping out potential attack paths and failure modes in the context of the organization’s structure. Techniques like attack trees or “what-if” storyboarding are used to explore both external and internal dimensions of each threat. Importantly, the analysis considers **combined scenarios** that are particularly relevant to decentralized environments – for example, an external attacker colluding with a disgruntled insider, or a flaw in a consensus algorithm that an adversary might exploit. By examining these complex scenarios, the group can anticipate cascading effects and interdependent weaknesses (e.g. how a compromised communication channel could undermine a membership vote). This step results in a clear understanding of *how* attacks might unfold, which informs targeted defenses.
4. **Governance-Based Strategy Design** – In the final step, the organization formulates and implements security strategies for the threats, fully integrating these measures into its governance model. Rather than imposing solutions unilaterally, the protocol requires that mitigations be developed and approved through the same democratic processes that govern the organization. For each major threat or scenario, the team discusses and agrees on controls or policies to put in place – for instance, introducing a verification step for member identities to counteract Sybil attacks, or establishing rules for validating transactions/communications to prevent fraud. These countermeasures are codified as collective decisions. The protocol uses transparent consensus mechanisms to formally ratify critical security actions or policies, ensuring broad agreement and understanding. Once adopted, the measures are implemented in a participatory way: technical safeguards might be enacted (such as requiring multiple members’ digital signatures for approving a sensitive change), and all security-relevant actions are recorded in tamper-evident logs that the community can audit. In essence, security becomes an integral part of organizational governance – the same forums and procedures used to make ordinary decisions are utilized to enforce security, creating a continuous feedback loop. Members remain engaged through regular review meetings or assemblies where the effectiveness of security strategies is evaluated and adjusted as needed. This iterative, governance-grounded approach ensures the security program evolves with the organization and maintains alignment with its horizontal ethos.

**Mitigation**

To reduce and manage threats in a decentralized setting, the protocol emphasizes a mix of technical and governance-based mitigation strategies. These strategies leverage collaborative technologies and participatory practices so that security is maintained **by the group, for the group**, in line with distributed governance principles. Key mitigation approaches include:

* **Collaborative Cryptography**: Employ cryptographic schemes that distribute trust among multiple members rather than concentrating it in a single person or server. This can include multi-signature authentication (requiring several designated members to jointly sign off on critical transactions or configuration changes) and threshold encryption (splitting sensitive keys or secrets across the group so no single individual holds the entire secret). By using such *collaborative cryptography*, the organization mitigates the risk of a single point of failure – an attacker would need to compromise a quorum of members to subvert the system. This approach was inspired by emerging security frameworks which combine cryptographic tools with consensus mechanisms to protect decentralized systems. It ensures that control over assets (like encryption keys, servers, or funds) is fundamentally *shared*, preventing scenarios where one compromised account or insider could breach the whole organization. In practice, collaborative cryptographic measures provide both security and transparency, as every authorized action leaves a verifiable trail (e.g. multiple digital signatures) that members can independently confirm.
* **Decentralized Access Control**: Replace traditional role-based or hierarchical access controls with a democratic, collective model of managing permissions. A prominent example is the Collective Based Access Control (COLBAC) model, which this protocol adapts for general use. In a COLBAC-style scheme, access to critical resources or data is not given unilaterally by a single admin; instead, permissions are granted through group consensus or membership votes. For instance, if a member needs access to a secure data repository, a proposal is submitted and other members approve it according to predefined rules (possibly using tokens or votes). This participatory approach to access control aligns with horizontal governance by ensuring **no single person has unchecked access** to sensitive systems. However, because purely democratic processes can be slow or subject to manipulation, the protocol includes safeguards to strengthen decentralized access control. Independent audits and monitoring can be put in place to track how often and why access is granted or revoked, adding oversight. Additionally, technical limits and rules help prevent abuse: for example, **dynamic quorum thresholds** can be required (a higher percentage of member approval for more sensitive actions) and the use of emergency override credentials can be strictly limited or subject to after-the-fact review. These measures protect against “insider threats” to the collective process, such as a subgroup trying to game the voting system or a rush to approve an unsafe change. Overall, decentralized access control, when combined with audit and policy enforcement, eliminates the vulnerabilities of admin-centric models while preserving the group’s autonomy in managing its resources.
* **Participatory Security Mechanisms**: Embed security responsibilities into the fabric of the organization’s day-to-day collaborative practices. One important mechanism is **radical transparency** in security operations: all security-relevant actions (login attempts, configuration changes, policy updates, etc.) are recorded on immutable, tamper-evident logs that are visible to the membership. Using append-only ledgers or blockchain-based record systems for these logs can ensure that any member can audit the history of changes, which greatly increases accountability. Coupled with this, the protocol advocates **verifiable digital signatures** for member actions – when members perform critical operations or vote on security decisions, their actions are cryptographically signed and attributed. This creates a clear accountability trail without relying on a central supervisor. Another participatory mechanism is the establishment of rotating **security committees or working groups** drawn from the membership. These committees have the mandate to continuously monitor systems, review alerts or anomalies, and conduct periodic security audits from a citizen-expert perspective. Because committee membership rotates (and is often volunteer-based or randomly assigned), many members build security literacy over time, and the oversight remains unbiased and distributed. This helps catch internal fraud or misconfigurations early by leveraging the collective eyes of the community. Finally, the protocol remains adaptable by allowing the organization to **temporarily delegate authority** in emergency situations with full transparency and subsequent accountability. In a crisis (such as an ongoing cyber-attack), a small incident response team may be empowered to act quickly – for example, shutting down a server or blocking an account – but this delegation is done within predefined limits and must be approved or ratified by the group soon after. This ensures rapid responses do not permanently compromise the horizontal structure; as soon as the emergency passes, normal participatory control resumes, and all interim actions are reviewed by the membership. By combining transparency, shared oversight, and flexible yet accountable crisis response, these participatory security mechanisms turn the organization’s governance model into a security asset. Every member becomes a stakeholder in defense, creating a powerful deterrent to attackers and a robust capacity for the group to learn and improve its security posture collectively.

In conclusion, this formal threat modeling protocol provides a blueprint for decentralized organizations to proactively tackle security threats without abandoning their core principles of distributed governance and collaboration. Through clearly defined objectives, a participatory methodology, a step-by-step application process, and innovative mitigation strategies, horizontal organizations can achieve security through solidarity – **securing everyone by involving everyone**. By integrating democratic decision-making with cybersecurity practices, the protocol ensures that the organization’s security measures are not only effective against adversaries, but are also legitimate, transparent, and resilient in the face of evolving threats.

Great. I’ll draft a flexible technical guide for a new threat modeling protocol tailored to non-hierarchical organizations like collectives and co-ops. The guide will merge key features from PASTA, Personae Non Grata, and Security Cards, while incorporating principles from Democratic Centralism to balance participation and operational efficiency.

I’ll organize the protocol as a clear step-by-step framework with embedded guidance for practical application, and ensure it remains adaptable across various levels of horizontality.

I’ll let you know as soon as the draft is ready for your review.

**Flexible Threat Modeling Protocol for Collectives and Co-ops (v1.0)**

This guide presents a **flexible threat modeling protocol** tailored for non-hierarchical organizations (collectives, cooperatives, etc.). It integrates elements from **PASTA (Process for Attack Simulation and Threat Analysis)**, the **Persona Non Grata** approach, and **Security Cards**, adapting them to decentralized governance structures. It also embeds principles of **Democratic Centralism** to balance collective deliberation with efficient decision-making in high-stakes situations. The protocol is organized into clear stages, with notes on adapting each stage for organizations of different sizes and complexities. Emphasis is placed on collaboration, transparency, distributed authority, and auditability throughout the process.

**Methodological Foundations and Principles**

* **Risk-Centric & Collaborative Approach (PASTA):** The core process follows PASTA’s risk-focused stages, which elevate threat modeling to a strategic level with input from all stakeholders – not just IT or security teams ([What Is Threat Modeling? - Cisco](https://www.cisco.com/c/en/us/products/security/what-is-threat-modeling.html#:~:text=PASTA%20,level%2C%20with%20input%20from%20all)). PASTA’s seven-step framework aligns technical risks with business objectives and encourages cross-functional stakeholder collaboration ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=PASTA%20%E2%80%A2%20Helps%20identify%20relevant,in%20prioritization%20of%20threat%20mitigation)). This ensures security discussions involve members from operations, governance, development, etc., reflecting a cooperative’s broad participation.
* **Adversary Perspective (Persona Non Grata):** We incorporate *Persona Non Grata (PnG)* techniques to focus on the motivations and skills of potential attackers. PnG characterizes malicious actors as archetypal personas, forcing the team to view the system from an attacker’s point of view ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=Persona%20non%20Grata%20As%20a,expert%20to%20a%20potential%20attacker)). This helps visualize threats from “the other side” early in the process ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=view%20the%20system%20from%20an,15)), improving our adversary modeling by understanding what drives both external and internal threats.
* **Creative Threat Brainstorming (Security Cards):** The protocol leverages *Security Cards* as a practical brainstorming tool to identify non-obvious and complex threats. Security Cards use a deck of 42 cards across four dimensions (Human Impact, Adversary’s Motivations, Adversary’s Resources, Adversary’s Methods) to spur creative thinking about security ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=Security%20Cards%20Security%20Cards%20is,analysts%20can)). This method is informal and collaborative ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=Security%20Cards%20%E2%80%A2%20Encourages%20collaboration,Leads%20to%20many%20false%20positives)), ensuring that members at all levels can contribute ideas. It helps surface out-of-the-ordinary threat scenarios, which is valuable in a cooperative context where threats might be unconventional (e.g. ideological attacks, insider mishandling, etc.).
* **Collective Governance & Democratic Centralism:** In line with cooperative values, the protocol emphasizes open participation and transparency in decision-making. All members (or their representatives in larger co-ops) are encouraged to partake in deliberating threats and defenses. We follow the principle that decisions are made through free discussion and then unified execution ([democratic centralism](https://universalium.en-academic.com/265131/democratic_centralism#:~:text=Democratic%20centralism%20purported%20to%20combine,Russian%20Communist%20Party%20%281921%29%2C%20the)) – once the group agrees on a security measure, everyone supports it. The organization remains **cohesive enough to act promptly in crises**, meaning that during emergencies or high-stakes situations, a temporarily centralized decision mechanism can take charge ([Literally me fr fr :3 : r/Polcompballanarchy](https://www.reddit.com/r/Polcompballanarchy/comments/18bnbde/literally_me_fr_fr_3/#:~:text=The%20right%20to%20be%20LGBTQ,of%20Rights%2C%20so%20no%20worries)). This could be a designated incident response team or an emergency committee, empowered to make quick decisions on behalf of the collective. Such central action is always bounded (time-limited and issue-limited) and accountable to the group after the fact. This blend of democratic deliberation with the ability to centralize when needed ensures both broad input and effective response.
* **Distributed Authority and Auditability:** No single person unilaterally controls the threat modeling process. Responsibilities for analysis and implementation are distributed, and each decision or assumption is documented. This creates an audit trail so that any member can review why a threat was ranked a certain way or why a mitigation was chosen. Visual tools (like diagrams and cards) and written records (threat lists, meeting notes, risk scores) should be stored in a shared repository. This transparency builds trust in the process and aligns with cooperative principles.

With these foundations in mind, the protocol is structured into sequential steps. Each **step** provides guidance on what to accomplish, tools or techniques to use, and how to adapt the approach based on organizational size or complexity.

**Step 1: Establish Context and Objectives**

**Purpose:** Define what the organization is protecting and set the stage for a collective threat modeling effort. In this step, the group clarifies its key assets, business/community objectives, and risk tolerance. This corresponds to PASTA’s initial stage of defining business objectives and requirements ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=1,Boundaries%20of%20the%20Technical%20Environment)), but with cooperative input.

**Activities:**

* **Identify Critical Assets and Mission Objectives:** Through a group discussion or survey, list the assets vital to your collective. These can include digital assets (member data, communication platforms), physical assets (community space, servers), and intangible assets (reputation, member trust). Also identify the organization’s main objectives and obligations – e.g. providing a service reliably, protecting member privacy, meeting legal compliance. PASTA frames this as business impact analysis ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=1,Boundaries%20of%20the%20Technical%20Environment)); in a co-op, ensure these objectives reflect member priorities and values.
* **Security & Compliance Requirements:** Discuss any compliance needs or policies the organization adheres to (for example, GDPR for data privacy if applicable, or internal bylaws about data handling). Identifying these early ensures the threat model aligns with required standards.
* **Define Scope and Boundaries:** Agree on the scope of this threat modeling effort. Are you modeling threats to a particular IT system, to the co-op’s overall operations, or to a specific project? Clearly delineate what is “in scope.” For instance, you might include your website, cloud services, and internal network, but exclude third-party platforms that are out of your control (except where they interface with you). Setting scope prevents the analysis from drifting and ensures focus on relevant areas ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=%E2%80%A2%20Business%20Impact%20Analysis%202,Data%20Sources)).

*Adaptation:* For a **small collective (e.g. <10 members)**, Step 1 can be done in an all-hands meeting where everyone voices concerns about what needs protection. Use a whiteboard or shared document to record assets and objectives, so everyone sees and agrees on the scope. In a **larger cooperative**, you might first collect input via a questionnaire or breakout sessions (to ensure quieter voices are heard), then have a representative committee consolidate the results. Ensure the draft list of assets/objectives is circulated to all members for feedback (transparency from the start). This collaborative goal-setting sets a democratic tone for the process.

**Step 2: Map Systems, Assets, and Trust Relationships**

**Purpose:** Build a shared understanding of how information and processes flow in your organization, and where the trust boundaries lie. This is analogous to PASTA’s technical scoping and application decomposition stages (defining the technical environment and identifying assets, actors, and trust boundaries) ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=Scope%20%E2%80%A2%20Capture%20the%20Boundaries,Vulnerability)). In a coop, this step should be participatory and visual, creating artifacts that everyone can interpret.

**Activities:**

* **Inventory Assets and Processes:** Enumerate the components that make up your technical and operational environment. This includes hardware (e.g. laptops, servers), software applications, data stores, communication channels, and also human elements (roles and teams). For each, note what data it handles and its importance. A *Data Flow Diagram (DFD)* can be useful to illustrate how data moves between components, which aligns with PASTA’s use of DFDs to map applications ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=Decomposition%20%E2%80%A2%20Identify%20Use%20Cases,Using%20Use%20%26%20Abuse%20Cases)). If not all members are technical, keep diagrams high-level and explanatory.
* **Identify Actors and Roles:** List who interacts with the systems (e.g. members, admins, third-party service providers, customers). In a collective, roles might overlap, but it’s important to note different permission levels or responsibilities. For example, an “IT volunteer” might have server access, whereas a regular member might only use a chat platform. This actor list will help in later adversary modeling and in spotting insider threat possibilities.
* **Map Trust Relationships and Boundaries:** Determine where **trust** exists in your system – i.e., which components or people are assumed to be trustworthy by others. Mark trust boundaries on your diagrams: a trust boundary is a line where data or control passes from a trusted zone to an untrusted one (for instance, from your internal network to the public internet, or from a member’s account to an admin account) ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=Decomposition%20%E2%80%A2%20Identify%20Use%20Cases,Using%20Use%20%26%20Abuse%20Cases)). Pay special attention to places where your co-op implicitly trusts something or someone. For example, do you blindly trust a cloud service to be secure? Do members with certain privileges get unrestricted access to data? Document these assumptions. *Threat modelers should question assumptions about network boundaries and trust relationships between different components* ([Zero trust and threat modeling: Is it time for AppSec to get on board?](https://www.reversinglabs.com/blog/zero-trust-and-threat-modeling-is-it-time-for-appsec-to-get-on-board#:~:text=,within%20the%20network%2C%20allowing%20for)) to uncover hidden vulnerabilities. In cooperative contexts, trust relationships may also be social (e.g. trusting a member not to misuse data); make those explicit too.
* **Infrastructure and External Dependencies:** Note any external services or partners on which you rely (payment processors, web hosting, etc.). Draw them in your diagrams and indicate the connections. These represent external trust relationships – the cooperative trusts those third parties to some extent. Understanding these links will help identify external threat vectors (like supply chain attacks or service outages).
* **Visual Documentation:** Create a visual map (could be a diagram or even a collaborative mural) of the system and its assets. Ensure it’s accessible to all stakeholders (post it in a shared space or online). The goal is a common reference that makes the system layout and trust points clear to everyone (). Keep the representation as simple as possible (avoid over-technical schematics) so that even non-experts can follow. Label key assets and trust boundaries clearly.

*Adaptation:* In a **small co-op**, this mapping might be done by the whole group together, building the diagram step-by-step in a workshop. In a **larger co-op**, it could start with a smaller technical working group drafting the diagrams, then presenting to the wider membership for input and validation. Consider using collaborative diagram tools or large paper sheets on a wall for everyone to annotate. The mapping should be iterative – as members point out missed assets or relationships, update the diagram. The final map should be approved by consensus (or majority) to ensure collective buy-in on “this is how we see our system.”

**Step 3: Identify Threats Collaboratively**

**Purpose:** Brainstorm and catalog potential threats to the organization’s assets and operations, engaging participants from across the collective. This step corresponds to threat identification phases (PASTA’s threat analysis stage) and leverages the **Security Cards** and other creative techniques to draw out diverse perspectives. The aim is to surface both **external threats** (e.g. hackers, malware, service disruptions) and **internal threats** (e.g. insider misuse, human errors) in a structured yet inclusive manner.

**Activities:**

* **Threat Brainstorming Session:** Convene a session with representatives from various roles (or all members, if feasible) to brainstorm threats. Use the visual map from Step 2 as a reference and walk through each part asking “What could go wrong here? Who might try to attack or abuse this, and how?” Encourage participants to think broadly: consider cyber attacks, physical security issues, social engineering, insider threats, accidents, and even natural disasters or political/regulatory threats, depending on context. At this stage, **quantity over quality** – list everything that comes to mind without debating likelihood or feasibility yet.
* **Use Security Cards or Prompts:** Introduce the *Security Cards* methodology to spark ideas. For example, divide the brainstorming into the card deck’s four dimensions: **Human Impact, Adversary’s Motivations, Adversary’s Resources, Adversary’s Methods** ([Home | The Security Cards: A Security Threat Brainstorming Kit](http://securitycards.cs.washington.edu/index.html#:~:text=The%20Security%20Cards%20encourage%20you,suits)). Ask questions for each: “Who might be motivated to target us and why? What resources or capabilities would they have? What methods might they use? And what would the impact be on people if they succeed?” Using these prompts helps the group consider threats beyond the obvious. The Security Cards technique often uncovers out-of-the-ordinary threats and encourages creative thinking ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=Security%20Cards%20%E2%80%A2%20Encourages%20collaboration,Leads%20to%20many%20false%20positives)). (If you have the actual card deck, you can deal cards to small groups and have each group come up with threats related to their cards as a fun exercise.)
* **Categorize Internal vs External Threats:** As the brainstorm generates ideas, categorize them roughly into *internal threats* (originating from within the organization or trusted insiders) and *external threats* (originating from outside adversaries or events). This ensures that issues like a disgruntled member leaking information or an accounting error are considered alongside external cyber-attacks or vandalism. It’s important in a cooperative to acknowledge that not all threats are external hackers; sometimes misuse or mistakes by insiders (even if unintentional) can pose risks.
* **Document Threat Scenarios:** For each raw threat idea, write down a brief scenario or description. For example: “An external hacker defaces the website via a plugin vulnerability,” or “A member accidentally shares a sensitive document publicly,” or “Payment provider service outage disrupts our sales.” Detailing scenarios helps clarify the threat and will be useful in later analysis. Make sure this list is accessible and visible (e.g., on sticky notes on a board, or in a shared online doc) during the session.
* **Inclusivity and Safe Environment:** Foster an environment where everyone feels comfortable contributing. Make it clear that all ideas are welcome and there are no bad suggestions in brainstorming. If some members are less technically versed, pair them with others or specifically ask them about scenarios in their domain (“What worries you could go wrong in our finance process? In our member sign-up process?”). Often, non-technical participants will identify different types of threats (like process failures or community issues) that technical experts might overlook.
* **Consolidate and Refine:** By the end of the brainstorming, you might have a long list of threats. Group similar items together and remove obvious duplicates. However, do **not** discard a threat simply because it seems unlikely – that judgment will come later in risk analysis. The goal here is a comprehensive threat inventory. Security Card activities are known to produce a high number of potential threats (including false positives) ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=Security%20Cards%20activities%20help%20identify,this%20method%20is%20rarely%20used)), so expect a broad list.

*Adaptation:* For a **small cooperative**, this might be a single meeting with all members. You can go around in a round-robin to ensure everyone speaks. In a **larger organization**, consider breaking into smaller groups (each group focusing on one category or one part of the system) and then reconvening to share findings. Using facilitation tools like digital whiteboards or issue trackers can help manage a large volume of input. It may also be useful to run an asynchronous brainstorming (e.g. a shared spreadsheet where members can submit threats over a week) for those who can’t attend the live session. The key is to gather input widely so that the threat list reflects the collective’s wisdom and diverse viewpoints.

**Step 4: Adversary Modeling (Develop Threat Personas)**

**Purpose:** Enrich the threat analysis by developing concrete *adversary personas* – fictional profiles that represent different types of threat actors who might target your organization. This step borrows from the Persona Non Grata (PnG) method to put a human face (or motivation) on threats, which helps the team to reason about attacker goals, skills, and tactics ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=Persona%20non%20Grata%20As%20a,expert%20to%20a%20potential%20attacker)). By the end of this step, the collective will have a set of “Personae Non Gratae” that embody both external and internal adversaries, making subsequent analysis more attacker-centric.

**Activities:**

* **Identify Key Adversary Types:** Review the threat scenarios from Step 3 and identify who the adversaries are in those stories. You’ll likely find patterns or recurring types. For example, you might have scenarios involving “opportunistic hacker,” “state-sponsored actor,” “disgruntled ex-member,” “competitor business,” or “malicious insider.” Make a shortlist of distinct adversary categories that cover most of your scenarios. Aim for a diverse set that captures different motivations and capabilities.
* **Create Persona Profiles:** For each adversary type, create a one-page persona profile. Give the persona a nickname (e.g. “Malicious Marvin the Ex-Engineer” or “Sneaky Sam the Script Kiddie”). Detail the following for each:
  + **Motivations/Goals:** What does this adversary want? (e.g. financial gain, ideological disruption, revenge, curiosity, etc.)
  + **Skills/Resources:** What skills, knowledge or resources do they have? (e.g. a well-funded group with sophisticated tools, or a lone insider with limited tech skill but deep inside knowledge)
  + **Typical Targets & Methods:** Which assets would they focus on and how might they attack? (e.g. Marvin might exploit his knowledge of the internal network; Sam might run automated attacks on the website).
  + **Profile Summary:** A short narrative of the persona’s background can make it more vivid (for instance, “Marvin is a bitter former IT administrator who was expelled from the co-op and seeks revenge by leaking data…”). Keep it realistic and grounded in the threats you’ve identified.
* **Use Templates or Cards:** If available, use a template for persona creation so that all profiles have the same format. Some teams treat this as a creative exercise – printing a “persona card” with a photo illustration, and bulleted details. The idea is to **introduce the team to a potential attacker** in a tangible way ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=view%20the%20system%20from%20an,15)). This can even be gamified: small groups each draft a persona and then present “their adversary” to the others.
* **Internal vs External Personas:** Ensure that you include *internal adversaries* if relevant. In a cooperative, an “insider threat persona” (like a rogue member or a careless employee) is very important. This might feel uncomfortable, but modeling a potential malicious insider (even if hypothetical) helps address internal risks openly. Emphasize that this is not accusatory but a standard practice to improve security.
* **Leverage Persona Non Grata Guidance:** Persona non Grata focuses on attacker mindset and is noted for producing high-consistency insights with low false positives ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=Persona%20non%20Grata%20As%20a,expert%20to%20a%20potential%20attacker)) ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=expert%20to%20a%20potential%20attacker,15)). However, it often detects only certain subsets of threats (the human-driven ones). Use it in combination with your broad threat list. For each persona, go back to the threat list and mark which scenarios that persona would undertake. You may discover new scenarios as you think from the persona’s perspective (“What else could Marvin do?”). This method forces the team to view the system through an attacker’s eyes, revealing points of vulnerability that might otherwise be overlooked.
* **Document and Share Personas:** Publish these personas where everyone in the co-op can see them (on an internal wiki or report). They become a communication tool – when discussing security, members can refer to “what if Marvin tries X?” which makes discussions concrete. It also builds a shared understanding of adversaries among non-security-focused members.

*Adaptation:* In a **small organization**, you might create 2-3 personas that cover the main threat types. The whole group can work on each together, or individuals draft them and then get feedback. In a **larger org**, different teams can each take one persona to develop (e.g. tech team designs the external hacker persona, HR team designs the disgruntled insider persona, etc.), then rotate for review. Ensure that even non-technical groups are involved in persona creation – they often bring insight into motivations and methods (for example, the finance team might best articulate how a fraudster would abuse financial processes). The final persona set should be reviewed collectively to check if they resonate with everyone’s sense of plausible threats.

**Step 5: Attack Scenario Analysis and Simulations**

**Purpose:** Using the threat list and adversary personas, the team now conducts a deeper analysis of how attacks could unfold. This step corresponds to PASTA’s attack modeling and analysis stages ([PASTA Threat Modelling - The Complete Cyber Security Meal](https://www.cynance.co/pasta-threat-modelling/#:~:text=Stage%206%3A%20Analyse%20and%20model,to%20exploits%2C%20including%20attack%20vectors)). It involves developing attack scenarios or **attack trees**, possibly running simulations or tabletop exercises, to understand the sequence of events in an attack and evaluate our system’s weaknesses. The focus is on translating static threats into dynamic scenarios: “If adversary X attempts attack Y, how would it happen and what would be the impact?”

**Activities:**

* **Develop Attack Trees or Flows:** An **attack tree** is a graphical representation of an attack, starting from the attacker’s goal as the root and various paths (branches) to achieve it ([What Is Threat Modeling? - Cisco](https://www.cisco.com/c/en/us/products/security/what-is-threat-modeling.html#:~:text=,assets%2C%20or%20threats%20that%20matter)). Choose a high-priority threat scenario (from the list) and map out step-by-step how that attack could occur. Involve the team: for each step an attacker takes, discuss “What would they do next? What could fail or succeed?” For example, if the threat is “steal member data,” the tree might branch into methods like phishing a credential, exploiting a software flaw, or social engineering an insider. Attack trees help reveal all possible avenues an adversary might attempt, and they highlight required conditions for success. This method was recommended in PASTA Stage 6 for modeling exploits ([PASTA Threat Modelling - The Complete Cyber Security Meal](https://www.cynance.co/pasta-threat-modelling/#:~:text=Stage%206%3A%20Analyse%20and%20model,to%20exploits%2C%20including%20attack%20vectors)). **Visualize** these trees on paper or using a tool, and make sure they are understandable to the group.
* **Simulate with Tabletop Exercises:** For critical scenarios, perform a **tabletop simulation**. This is essentially a role-play of an attack: team members walk through the scenario in a narrative form. For instance, simulate how “Marvin” (insider persona) would try to exfiltrate data – step 1: he attempts to use an old account, step 2: finds it disabled, step 3: he contacts a current member under false pretenses, etc. As you simulate, identify at each step how the organization’s controls would respond. Are there alarms or detections? Would the attack succeed or where might it be stopped? This exercise can be eye-opening, revealing gaps in incident response plans or unclear responsibilities.
* **Assess Vulnerabilities and Controls:** While analyzing scenarios, list the vulnerabilities or weaknesses that make the attack possible. This aligns with PASTA’s vulnerability analysis stage ([PASTA Threat Modelling - The Complete Cyber Security Meal](https://www.cynance.co/pasta-threat-modelling/#:~:text=Stage%205%3A%20Vulnerability%20detection)). For each attack step, ask “What weakness is being exploited here?” It could be a technical flaw (unpatched software), a procedural gap (lack of verification for a request), or an assumption (trusting a user’s identity without second factor). Also note existing security controls that would mitigate each step (firewalls, approval processes, etc.) and whether they are adequate. This process effectively connects threats to specific vulnerabilities and controls, creating a traceability that will feed into risk assessment.
* **Probability and Impact Discussion:** For each scenario, discuss as a group how likely it is to occur and what the impact would be if it did. This doesn’t have to be a precise quantification yet, but getting qualitative input (e.g. “very unlikely but high impact” vs “likely but low impact”) will set the stage for formal risk analysis. Different members may have different views – capture these nuances. Sometimes those closest to a system know of hidden vulnerabilities that raise likelihood; others might know of compensating controls that reduce it. Ensure the adversary personas’ perspective is considered: e.g., “How likely is it that a person like Sneaky Sam would target us, given his motivations and our profile?”
* **Leverage Past Incidents and Intelligence:** Incorporate any historical data or threat intelligence. If the co-op or similar organizations have experienced certain attacks before, factor those into the scenarios (real examples increase the credibility of a scenario). PASTA encourages using threat intelligence to focus on credible threats ([PASTA Threat Modelling - The Complete Cyber Security Meal](https://www.cynance.co/pasta-threat-modelling/#:~:text=and%20build%20a%20threat%20library,that%20existing%20protections%20have%20mitigated)). Likewise, research if your industry (or collective space) has known adversaries or common attack patterns – adjust your scenarios accordingly.
* **Record Scenario Outcomes:** Document the results of each scenario analysis: what are the key vulnerabilities found, what potential damages were identified, and how well current controls fared. This documentation should be accessible for later reference, especially when prioritizing risks. It also serves as an audit trail of why certain risks will be ranked high or low.

*Adaptation:* In **smaller groups**, you might analyze only a handful of the most concerning scenarios in detail due to time and resource constraints. Pick the top threats and dive deep into those. In **larger organizations**, you can assign different scenarios to different teams to simulate in parallel (e.g. an IT team might simulate a network attack, a HR/operations team might simulate an insider fraud scenario). Then share findings across teams. If expertise is limited internally, consider inviting an external facilitator or using open-source attack libraries to guide the scenario building. Regardless of size, try to include at least one scenario that tests your emergency decision-making: this will exercise the democratic centralism principle by seeing how the group handles a simulated crisis (who takes charge, how information is shared, etc.). Learn from these exercises to refine your actual emergency roles and procedures.

**Step 6: Participatory Risk Analysis and Prioritization**

**Purpose:** Now that threats and attack scenarios have been mapped out, the collective evaluates which risks are most important and decides on priorities for mitigation. This step is critical for aligning security efforts with the co-op’s values and risk appetite. It reflects PASTA’s risk and impact analysis stage ([PASTA Threat Modelling - The Complete Cyber Security Meal](https://www.cynance.co/pasta-threat-modelling/#:~:text=Stage%207%3A%20Risk%2F%20impact%20analysis,and%20development%20of%20countermeasures)), but here we perform it *participatorily*, meaning input from the group is central in assessing impact and likelihood. The outcome is a ranked list of risks with a rationale, achieved through collective deliberation.

**Activities:**

* **Define Risk Criteria:** As a group, agree on how you will assess *risk*. Commonly, risk is a function of **Likelihood** (probability of the threat occurring) and **Impact** (severity of consequences if it does) ([PASTA Threat Modelling - The Complete Cyber Security Meal](https://www.cynance.co/pasta-threat-modelling/#:~:text=Stage%201%3A%20Define%20your%20business,objectives)). You might choose qualitative ratings (High/Medium/Low) or numeric scales. Define what “High impact” means for your co-op (e.g. loss of >10% revenue, or any loss of member personal data, or permanent damage to trust could be criteria for high impact). Similarly, define what makes something likely (e.g. has occurred in the past, or easy to execute given current state). Establishing these criteria openly ensures everyone is on the same page.
* **Collective Scoring or Ranking:** Take each major threat scenario (or threat category) and discuss its likelihood and impact. It often works to do this in a workshop setting: present the scenario, let members discuss, then have a voting or scoring mechanism. For instance, you can use dot voting on a chart where one axis is impact and the other is likelihood – people place a dot where they feel it belongs. Alternatively, for a systematic approach, have each participant anonymously rate each threat on a 1-5 scale for impact and likelihood, then aggregate the results. **Stakeholder input is essential to this step** of risk prioritization ([What Is Threat Modeling? - Cisco](https://www.cisco.com/c/en/us/products/security/what-is-threat-modeling.html#:~:text=,steps%20to%20address%20a%20threat)); it brings in operational knowledge on how a threat would play out and communal judgment on what the co-op cares most about. The goal is not exact precision but a consensus (or averaged view) on which risks are the most serious.
* **Identify Top Risks:** Based on the scoring, identify the top tier of risks – the ones that are high likelihood or high impact (or both). Typically, a Pareto principle might apply (20% of the threats account for 80% of the total risk). Highlight these as the issues that require action first. Also note any “quick wins” – threats that might be medium risk but are very easy to mitigate quickly; these can be prioritized as well, given cooperative resources.
* **Document Risk Decisions:** For each risk rated, document the justification. For example: “Data breach via cloud provider – Impact High (would expose member data violating privacy, harm reputation); Likelihood Medium (requires bypassing provider security which is strong, but not impossible). Ranked #2.” Writing down the reasoning provides auditability. Later, if a member questions “Why are we so worried about X?”, you have a recorded rationale. This transparency is important in a cooperative setting to maintain trust in the process: everyone sees that decisions were made logically, not arbitrarily.
* **Cross-Check with Objectives:** Ensure that the highest risks align with what the group agreed are critical assets and objectives (from Step 1). This is a sanity check to validate that the process stayed focused on what matters to the co-op. If a high-ranked risk is tied to a less critical asset, consider whether it should be lower priority or if that asset’s importance was underestimated initially. Likewise, make sure no high-importance asset is left with only low-ranked threats; if so, re-examine if some threat to it was missed or underrated.
* **Plan Transition to Action:** Begin thinking about mitigation broadly for the top risks (though detailed planning is Step 7). Sometimes the understanding of risk severity will naturally lead to discussing possible fixes (“This is our #1 risk; we really need to fix X quickly”). Capture these thoughts but keep the group on track – first finalize what is most important, then move to solutions.

*Adaptation:* In a **small co-op**, risk analysis might be done in a single roundtable discussion, ranking perhaps 10-15 threats. It can be informal but ensure everyone voices agreement on the top few risks. In a **large co-op**, a structured approach is helpful: you might use an online survey for members to score each threat scenario after a presentation of findings, then hold a meeting to discuss outliers and finalize rankings. If consensus is hard to reach, use democratic voting mechanisms (e.g. each member or delegate gets to vote on their top 3 concerns, and aggregate the results). Democratic centralism’s unity principle applies here: after open debate, once the group (or a majority) decides on the priorities, the decision is binding ([democratic centralism](https://universalium.en-academic.com/265131/democratic_centralism#:~:text=Democratic%20centralism%20purported%20to%20combine,Russian%20Communist%20Party%20%281921%29%2C%20the)) – the cooperative should commit to addressing those top risks first, without re-litigating the priority order unless new information arises. This ensures focus and collective alignment going forward.

**Step 7: Mitigation Planning and Decision-Making**

**Purpose:** Develop countermeasures for the highest priority threats and integrate these decisions into the cooperative’s action plans. In this step, the group moves from analysis to concrete **security controls and policies** to implement. It is also where the principles of collective governance and democratic centralism come into play to decide on and authorize these mitigations. We design interventions to reduce risk, assign responsibilities for implementation, and set up a structure for emergency response when needed. This corresponds to PASTA’s final stage of risk mitigation strategy identification ([PASTA Threat Modelling - The Complete Cyber Security Meal](https://www.cynance.co/pasta-threat-modelling/#:~:text=Stage%207%3A%20Risk%2F%20impact%20analysis,and%20development%20of%20countermeasures)), with added emphasis on participatory decision-making and accountability.

**Activities:**

* **Identify Mitigation Options:** For each top risk or threat scenario, brainstorm possible **mitigations**. Mitigations can be technical (e.g. apply a security patch, enable 2FA, add encryption), procedural (e.g. change a workflow, add a review step, improve backup routines), or educational (e.g. training members on phishing awareness). Consider multiple solutions and their feasibility. Use the adversary perspective: ask how each mitigation would stop or deter the personas you created (“Would this stop Marvin’s ability to get in?”). Also consider **countermeasure identification and residual risk** as PASTA suggests ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=%E2%80%A2%20Attack%20Surface%20Analysis%20%E2%80%A2,%E2%80%A2%20ID%20Risk%20Mitigation%20Strategies)) – meaning if you apply a certain fix, what risk remains?
* **Evaluate Mitigations Collectively:** Discuss the pros/cons of mitigation strategies with stakeholders. In a cooperative, this might involve considerations of cost, alignment with values, and member impact. For example, a technical fix might be effective but expensive, or a policy change might improve security but inconvenience members. The group must weigh these factors. If needed, do a quick vote or use a decision matrix to help identify which mitigations are acceptable and effective. Strive for consensus on the chosen mitigations, but if consensus fails, use the co-op’s agreed decision rule (majority vote or other) to finalize the plan. The *democratic* part of democratic centralism is critical here: everyone has input into what will be done.
* **Define Roles and Owners:** For each chosen mitigation or security control, assign an **owner** or responsible team. Because we operate in a distributed authority model, responsibility might be spread out: e.g. the IT working group handles software fixes, the finance team updates procedures for transaction checks, the membership committee handles member training, etc. Make sure these assignments are clear and recorded. Each responsible person/team should understand the task and commit to a timeline, reflecting accountability to the collective. This distribution prevents security tasks from all falling on one person (which is important in non-hierarchical orgs to avoid burnout and power concentration).
* **Plan for Democratic Centralism in Execution:** Agree on how decisions will be executed and how to handle urgent situations:
  + *Normal Implementation:* Once the group has decided on a mitigation, it should be carried out uniformly. All members should respect and follow new security policies or changes (for instance, if the group decides to enforce password managers, everyone should comply). This is analogous to “unity around the agreed policy” ([Democratic Centralism - an overview | ScienceDirect Topics](https://www.sciencedirect.com/topics/social-sciences/democratic-centralism#:~:text=Topics%20www,unity%20around%20the%20agreed%20policy)) – after deliberation, the policy is binding for all.
  + *Emergency Protocols:* Establish a mechanism for **temporary centralization** in case a security emergency occurs (e.g. an active cyber attack or a critical vulnerability discovered). Decide in advance who or what team has the authority to make snap decisions in such cases. For example, the co-op might empower a Security Incident Response Team (SIRT) composed of 2-3 trusted members who can make immediate decisions and expenditures to contain an incident without waiting for full group consensus. This follows the idea that the organization must be able to act promptly in crises with a cohesive response ([Literally me fr fr :3 : r/Polcompballanarchy](https://www.reddit.com/r/Polcompballanarchy/comments/18bnbde/literally_me_fr_fr_3/#:~:text=The%20right%20to%20be%20LGBTQ,of%20Rights%2C%20so%20no%20worries)). Define the scope of their authority (e.g. they can take the website offline if under attack, spend up to X amount on emergency contractors, etc.) and the requirement that they report and justify actions to the membership afterward. By planning this, you ensure that when a high-stakes threat arises, the response is swift (centralized execution) but still grounded in the mandate given by the collective.
  + *Decision Logging:* Require that any emergency decisions and actions taken are logged and later reviewed by the whole group. This maintains auditability and trust – the temporary central actors are not above the collective, they are acting on its behalf and will be accountable to it.
* **Develop Security Policies and Include in Governance:** Write down the agreed mitigations as formal or semi-formal **security policies/protocols**. For instance, create a policy on “Access Control” if you decided to implement role-based access restrictions, or a policy on “Member Offboarding” if you added steps to handle departing members’ accounts. Integrate these policies into the co-op’s governance documents or handbook. This ensures security becomes part of the organizational processes and isn’t forgotten. It also allows new members to learn the security norms as part of onboarding, aligning everyone with the security posture.
* **Set Milestones and Review:** Decide when and how to review the implementation of mitigations. For example, schedule a follow-up in 3 months to check progress on all tasks. This could be part of regular co-op meetings (embedding it into governance cycles). Also, decide on a schedule for re-running parts of this threat modeling process (perhaps annually or when major changes occur). Embedding these into the calendar makes security an ongoing practice rather than a one-time project.

*Adaptation:* In **smaller co-ops**, mitigation decisions might be straightforward to implement by a few members, but beware of overloading the same people with all tasks. Spread the work if possible and document help needed from external sources if required (maybe a volunteer with security expertise or a friendly consultant). In **larger co-ops**, you might need a formal vote to approve certain mitigations, especially if they involve budget (e.g. buying new security software) or significant policy changes. Use the cooperative’s governance channels (general assembly, board, etc.) to ratify these decisions as needed, but leverage the fact that the threat modeling team’s recommendations come with broad member input – this should ease the approval. Keep the wider membership informed of what security improvements are being made and why, reinforcing transparency. By the end of Step 7, you should have a clear, collectively sanctioned plan to reduce the organization’s risk, and a structure in place to handle both routine security management and the unexpected.

**Step 8: Continuous Improvement and Governance Integration**

**Purpose:** Ensure that threat modeling and security practices become an integral part of the cooperative’s ongoing operations and governance. Security is not a one-off exercise; threats evolve and the organization changes. This final step establishes how the collective will maintain and improve upon the threat model over time, and how it will align these efforts with its democratic governance.

**Activities:**

* **Incorporate into Governance Cycle:** Add security/threat modeling as a standing item in the co-op’s governance or operational meetings. For example, have a brief security report in each monthly meeting, or a more thorough threat review quarterly. This keeps security on the collective radar. During these updates, review any incidents (were there near-misses or strange events?), track progress on mitigation tasks, and note any changes in the organization (new services, new partnerships, etc.) that might alter the threat landscape.
* **Periodic Threat Model Refresh:** Plan to repeat a scaled-down version of this threat modeling protocol at a regular interval (e.g. annually). New threats emerge (especially in tech); also the co-op might undertake new projects that introduce different risks. By revisiting the threat list and personas periodically, you ensure the model stays up to date. You don’t always have to start from scratch: reuse the artifacts (asset maps, personas, etc.) and just update them. The Security Cards can be brought out again to see if any new wild ideas come up. PASTA emphasizes that threat modeling is not a static one-time assessment ([PASTA Threat Modelling - The Complete Cyber Security Meal](https://www.cynance.co/pasta-threat-modelling/#:~:text=,on%20old%20data%20or%20intelligence)) – treat it as an evolving practice.
* **Monitor and Audit:** Establish logs or metrics to monitor security-related events. For example, keep a log of security incidents or anomalies and review it for patterns. If a particular type of minor incident keeps occurring, that may warrant adding a new threat scenario and mitigation. Conduct internal audits or peer reviews of whether the implemented controls are being followed (auditability in practice). For instance, if you instituted a policy that ex-members’ accounts are deleted within 24 hours of departure, check periodically that this is actually happening. These audits can be done transparently, with results shared among members, to collectively verify that the security posture is as expected.
* **Training and Collective Learning:** Turn the insights from threat modeling into educational material for members. New joiners should learn about the major threats the co-op is concerned about and the security policies in place. You might create a short “security 101 for members” document derived from the threat model (including perhaps the personas to illustrate why the policies matter). Engage members in drills or exercises, such as an annual phishing test or a mock emergency drill, to keep preparedness high. Emphasize a culture of openness where anyone can report a potential security issue without fear – this way the cooperative’s collective intelligence continues to feed the threat modeling process.
* **Community and External Input:** Co-ops often exist in a network of similar organizations. Consider sharing non-sensitive parts of your threat model with other collectives or learning from theirs. This participatory approach can extend beyond your org’s borders – e.g., joining security mailing lists for co-ops, or collaborating on open-source threat scenarios relevant to your domain. Such external engagement can bring in new perspectives (maybe threats you hadn’t considered) and also demonstrate your co-op’s commitment to security in the wider community.
* **Adjust Governance as Needed:** Finally, be willing to adjust the governance structures based on what you’ve learned. If, during an incident, the emergency team approach wasn’t smooth, refine the rules around it. If members felt left out of a decision, improve the communication process. Democratic centralism in practice might need tweaking – perhaps the threshold for when to centralize decision-making needs to be clarified. Use post-mortems after any security incident or annual review to ask: did our system of collective + centralized decision-making work? Make amendments to protocols accordingly, and document them.

*Adaptation:* For **small organizations**, continuous improvement might be as simple as a yearly meeting to go over the threat model and any incidents. For **larger organizations**, formalize these processes more – you might have a dedicated security committee that meets more frequently and reports to the whole membership. The key is not to let the documents gather dust. Even if the co-op changes leadership or members over time, the threat modeling practice should be institutionalized so it carries on. Consider versioning your threat model (v1.0, v1.1, etc.) and keep archives; this helps in auditability to see how risks and decisions evolved, and it signals that this is a living process.

By following these eight steps, a non-hierarchical organization can systematically analyze and address its security threats while staying true to its values of participation and transparency. This protocol merges robust technical methodologies (like PASTA’s structured stages and attacker focus) with tools for creative engagement (Security Cards and personas) and the realities of collective governance. The result is a threat modeling process that not only yields a stronger security posture but also reinforces cooperative principles – everyone has a voice, everyone learns, and everyone takes responsibility for safeguarding the organization.

**The outcome** of this process is a set of prioritized threats, documented attacker models, and agreed-upon security measures that the entire collective understands and supports. It achieves a balance between broad democratic input and effective centralized action when necessary, embodying the idea that a cooperative can be both open and agile in the face of threats. In practice, this means the co-op is better prepared for attacks, more resilient through shared knowledge, and more unified in response – all hallmarks of a mature security culture in a decentralized organization.

**References:** The protocol integrates concepts from established threat modeling approaches and cooperative governance theory to ensure relevance and effectiveness. Key influences include the seven-stage PASTA framework (risk-focused, involving stakeholders across business and tech ([What Is Threat Modeling? - Cisco](https://www.cisco.com/c/en/us/products/security/what-is-threat-modeling.html#:~:text=PASTA%20,level%2C%20with%20input%20from%20all)) ([PASTA Threat Modelling - The Complete Cyber Security Meal](https://www.cynance.co/pasta-threat-modelling/#:~:text=,cybersecurity%20decisions%20the%20organisation%20makes))), the Persona Non Grata method for attacker profiling ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=Persona%20non%20Grata%20As%20a,expert%20to%20a%20potential%20attacker)), the University of Washington’s Security Cards for brainstorming diverse threats ([Threat Modeling: A Summary of Available Methods](https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf#:~:text=Security%20Cards%20Security%20Cards%20is,analysts%20can)), and the principle of democratic centralism, which advocates free discussion followed by unified action ([democratic centralism](https://universalium.en-academic.com/265131/democratic_centralism#:~:text=Democratic%20centralism%20purported%20to%20combine,Russian%20Communist%20Party%20%281921%29%2C%20the)) and the ability to act collectively and swiftly in crises ([Literally me fr fr :3 : r/Polcompballanarchy](https://www.reddit.com/r/Polcompballanarchy/comments/18bnbde/literally_me_fr_fr_3/#:~:text=The%20right%20to%20be%20LGBTQ,of%20Rights%2C%20so%20no%20worries)). These elements have been adapted to fit the collaborative and flat structure of collectives and co-ops, resulting in a first version protocol that can be further refined with experience and feedback.

### Key Points

- Research suggests that creating a threat model protocol for non-hierarchical organizations involves adapting existing methods like STRIDE, PASTA, and Security Cards to fit distributed decision-making structures.

- It seems likely that incorporating Democratic Centralism can balance collective deliberation with unified action, ensuring security aligns with organizational values.

- The evidence leans toward emphasizing collaborative processes, such as workshops and group discussions, to identify assets and threats, given the lack of formal hierarchies.

### Introduction

As a 30-year-old cybersecurity specialist, I’ve designed the first version of a threat model protocol tailored for non-hierarchical organizations. This protocol focuses on distributed governance, mixing features from PASTA, Personae Non Grata, Security Cards, and Democratic Centralism to ensure security without compromising democratic participation. It’s built for organizations like cooperatives or flat startups, where everyone has a say, and aims to be accessible to members without deep cybersecurity expertise.

### Protocol Overview

The protocol is a set of guidelines for identifying, analyzing, and mitigating threats collaboratively. It includes steps like asset identification, process mapping, and risk assessment, all done through group sessions to reflect the organization’s structure. It also incorporates emergency response mechanisms for quick action while maintaining transparency, ensuring it fits non-hierarchical values.

### Unexpected Detail

One unexpected aspect is how blockchain security practices, such as those for consensus manipulation, can inform organizational threat modeling, highlighting risks like decision subversion in non-hierarchical settings.

---

### Survey Note: Detailed Threat Model Protocol for Non-Hierarchical Organizations

#### Background and Context

Threat modeling is a cornerstone of cybersecurity, traditionally used to identify assets, analyze attack vectors, and establish controls. For non-hierarchical organizations—those rejecting traditional hierarchies and prioritizing distributed decision-making, such as worker cooperatives or social movements—it requires a unique approach. These structures lack formal authority lines, making internal processes and trust relationships critical, as noted in studies like [Colbac Protocol Analysis](https://en.wikipedia.org/wiki/Worker\_cooperative). The rapid evolution of threats and the need for flexibility underscore the importance of adaptable methods, as discussed in [Threat Modeling: Designing for Security](https://www.howardposton.com/blog/threat-modeling-for-the-blockchain).

Non-hierarchical organizations can disperse vulnerabilities through decentralized governance, increasing resilience but also creating multiple entry points for threats, as highlighted in [Non-Hierarchical Forms](https://community-wealth.org/content/worker-cooperatives). This contrasts with hierarchical structures, where centralized control can be both an advantage and a vulnerability, as seen in [Microsoft Threat Modeling Technique](https://www.techtarget.com/searchsecurity/definition/threat-modeling).

#### Methodological Foundations

The protocol draws from several established methodologies:

- \*\*PASTA (Process for Attack Simulation and Threat Analysis):\*\* A risk-centric approach with seven steps, including defining objectives and assessing risks, which can be adapted for organizational processes.

- \*\*Personae Non Grata:\*\* Focuses on characterizing threat agents, useful for identifying both external and internal adversaries in non-hierarchical settings.

- \*\*Security Cards:\*\* A tool from the University of Washington that uses cards to prompt discussions on threats, ideal for collaborative sessions.

- \*\*Democratic Centralism:\*\* A governance model balancing collective deliberation with centralized execution, relevant for decision-making in security measures.

These methods are integrated to create a protocol that is both technical and social, considering organizational dynamics as much as technical vulnerabilities, as suggested in [Advanced Threat Modeling](https://www.imperva.com/learn/application-security/threat-modeling/).

#### Protocol Design

The protocol is structured as a formal document, not software, targeting members without specialized cybersecurity expertise. It ensures simplicity and accessibility, supporting stakeholders in decision-making, operations, and coordination. Below is the detailed framework:

##### Steps of the Protocol

1. \*\*Asset Identification\*\*

- Involves collaborative sessions to list and prioritize assets, such as member data, financial information, or reputation. Methods include workshops and surveys, ensuring all voices are heard.

- Example: A cooperative might identify its member database as a critical asset, prioritizing it based on mission impact.

2. \*\*Process Mapping\*\*

- Maps key organizational processes, such as decision-making, communication, and resource allocation, using tools like flowcharts or online whiteboards (e.g., [Miro](https://miro.com)). This visual representation helps identify potential threat points.

- Example: Mapping how decisions are made could reveal vulnerabilities in consensus processes.

3. \*\*Threat Identification\*\*

- Uses adapted STRIDE categories (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) with additional organizational-specific threats like Consensus Manipulation and Trust Exploitation. Facilitated by Security Cards or brainstorming sessions.

- Table: Adapted STRIDE Categories for Non-Hierarchical Organizations

| \*\*Category\*\* | \*\*Description\*\* | \*\*Examples\*\* |

|----------------------------|------------------------------------------------------|--------------------------------------------------|

| Spoofing | Impersonation risks | Fake member applications, email spoofing |

| Tampering | Data or decision alteration | Unauthorized changes to shared documents |

| Repudiation | Denying actions | Member denies agreeing to a decision |

| Information Disclosure | Leaks of sensitive information | Insecure communication channels leaking data |

| Denial of Service | Disruptions to operations | DDoS on websites, overwhelming with spam |

| Elevation of Privilege | Unauthorized access or influence | Exploiting trust for restricted access |

| Consensus Manipulation | Subverting decision-making processes | Vote stuffing, misinformation campaigns |

| Trust Exploitation | Abusing high trust levels | Insider theft, embezzlement |

- This step ensures threats are identified from multiple perspectives, leveraging the collective knowledge of members.

4. \*\*Threat Agent Characterization\*\*

- Identifies potential adversaries, such as disgruntled members, external hackers, or competitors, and their motivations. This aligns with Personae Non Grata, adapted for organizational contexts.

- Example: A competitor might attempt social engineering to gain access to strategic plans.

5. \*\*Vulnerability Analysis\*\*

- Identifies weaknesses, such as weak passwords, lack of encryption, or insufficient training, through group discussions. This step ensures vulnerabilities are not overlooked due to the lack of centralized oversight.

- Example: Lack of logging in decision-making processes could be a vulnerability for repudiation.

6. \*\*Risk Assessment\*\*

- Evaluates likelihood and impact using simple matrices, ensuring accessibility. For instance, a 1-5 scale for both likelihood and impact can help prioritize threats.

- Example: High likelihood, high impact threats like data leaks would be prioritized.

7. \*\*Mitigation Planning\*\*

- Proposes countermeasures, such as implementing two-factor authentication, encrypting communications, or establishing clear logging procedures. Ensures alignment with organizational values, such as preferring open-source tools.

- Example: Using [Mend Renovate](https://www.freecodecamp.org/news/oss-security-best-practices/) for dependency updates in software-related cooperatives.

8. \*\*Decision-Making\*\*

- Uses democratic processes, reflecting Democratic Centralism, to decide on mitigations. This involves open discussion followed by voting, ensuring collective buy-in.

- Example: A vote on whether to implement encryption for all communications, with majority agreement.

9. \*\*Implementation\*\*

- Executes agreed-upon mitigations, assigning responsibilities collectively. This could involve forming working groups or task forces, ensuring distributed effort.

- Example: A group handles setting up two-factor authentication for all members.

10. \*\*Monitoring and Review\*\*

- Involves ongoing monitoring and regular reviews, such as quarterly threat modeling sessions, to adapt to changes. This ensures the protocol remains relevant as the organization evolves.

- Example: Reviewing incident logs to identify new threats.

##### Integration of Democratic Centralism

Democratic Centralism is embedded in the decision-making process (Step 8), where threats and mitigations are discussed collectively, and decisions are made democratically. Once agreed, actions are executed uniformly, ensuring unified implementation. For emergencies, the protocol includes provisions for temporary delegation, such as forming a crisis response team, with actions later audited collectively, as outlined in [Colbac Protocol](https://en.wikipedia.org/wiki/Worker\_cooperative).

##### Tools and Techniques

- Recommends collaborative tools like [Miro](https://miro.com) for process mapping and [Trello](https://trello.com) for task management.

- Suggests using Security Cards for threat identification, drawing from open-source security practices like those in [Open Source Security Best Practices](https://www.wiz.io/academy/open-source-security-best-practices).

- Encourages open-source tools where possible, aligning with non-hierarchical values, such as [Mend Renovate](https://www.freecodecamp.org/news/oss-security-best-practices/) for dependency management.

##### Emergency Response

- Outlines mechanisms for quick action in crises, such as designating a response team with temporary authority, ensuring transparency through immutable records (e.g., blockchain-inspired logging, as seen in [Threat Modeling for Blockchain](https://www.howardposton.com/blog/threat-modeling-for-the-blockchain)).

- Actions are subject to collective audit, maintaining accountability and trust.

##### Case Studies and Examples

While this is the first version, hypothetical scenarios illustrate application:

- \*\*Scenario:\*\* A cooperative identifies its member database as an asset. Mapping shows onboarding processes are vulnerable to spoofing. Using STRIDE, they identify fake applications as a threat, propose verification procedures, and vote to implement them, ensuring collective agreement.

#### Insights from Related Fields

Drawing from blockchain security, such as [Cryptocurrency Security Threat Modeling](https://www.nopsec.com/blog/cryptocurrency-security-threat-modeling-beyond-vulnerabilities/), highlights risks like consensus manipulation, relevant for organizational decision-making. Open-source security practices, from [Open Source Software Security Risks](https://www.sentinelone.com/cybersecurity-101/cybersecurity/open-source-software-security-risks/), suggest continuous monitoring and community involvement, which can be adapted for non-hierarchical organizations to enhance security through collective effort.

#### Conclusion

This protocol provides a comprehensive framework for non-hierarchical organizations to engage in threat modeling, ensuring security aligns with their governance principles. It balances efficiency and democratic participation, adapts to organizational dynamics, and improves resilience against threats while maintaining transparency. Future iterations can refine based on real-world application and feedback.