**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.