the elevation of privilege may be as a result of spoofing due to information disclosure or simply the result of the lack of repudiation controls. In such cases, it is recommended to use your best judgment when categorizing threats. One can select the most relevant category or document all of the applicable threat categories and rank them according to the likelihood of the threat being materialized.

Identify, Prioritize and Implement Controls

Merely cataloging a list of threats provides little assistance to a design team that needs to decide how to address the threat.

Risks arising from identified threats that need to be mitigated. Mitigation is accomplished by implementing controls. It is advisable to use standard controls instead of inventing your own. When mitigation is not possible, the risk can be accepted if the level of risk is below what is acceptable for the business or the software can be re-architected to eliminate the threat.

Knowledge of threats and vulnerabilities is worthless unless appropriate controls are identified to mitigate the threats that can exploit the vulnerabilities. The identification of controls needs to be specific to each threat. A threat may be completely mitigated by a single control, or a combination of controls may be necessary. In instances where more than one control is needed to mitigate a threat, the defense in depth measures should ensure that the controls complement rather than contradict one another. It is also important to recognize that the controls (safeguards and countermeasures) don't eliminate the threat, but only reduce the overall risk that is associated with the threat.

Since addressing all the identified threats is unlikely to be economically feasible, it is important to address the threats that pose the greatest risk first, before addressing those that have minimal impact to business operations. The risk ranks derived from the security risk assessment activity (SRA) of the threat modeling exercise are used to prioritize the controls that need to be implemented. Quantitative risk ranks are usually classified into qualitative bands such as High, Medium, or Low, or, based on the severity of the threat, into Severity 1, Severity 2, and Severity 3. These bands are also known as *bug bars* or *bug bands* and they are not just limited to security issues. There are bug bars for privacy, as well. Bug bars help with prioritizing the controls to be implemented post design.

There are several ways to quantitatively or qualitatively determine the risk ranking for a threat. These range from the simple, non-scientific, Delphi heuristic methodology to more statistically sound risk ranking using the probability of impact and the business impact. The three common ways to rank threats are

- Delphi ranking
- Average ranking
- Probability x Impact (P x I) ranking

Delphi Ranking

The Delphi technique of risk ranking is one in which each member of the threat modeling team makes his or her best guesstimate on the level of risk for a particular threat. During a Delphi risk ranking exercise, individual opinions on the level of risk for a particular threat are stated and the stated opinions are not questioned but accepted as stated. The individuals who are identified for this exercise include both members with skills at an expert level and those who are not skilled, but the participating members only communicate their opinions to a facilitator. This is to avoid dominance by strong personalities who can potentially influence the risk rank of the threat. The facilitator must provide, in advance, predefined ranking criteria (1 – Critical, 2 – Severe, 3 – Minimal) along with the list of identified threats, to ensure that the same ranking criteria are used by all members. The criteria are often based merely on the potential impact of the threat materializing and the ranking process is performed until there is consensus or confidence in the way the threats are ranked. While this may be a quick method to determine the consensus of the risk potential of a threat, it may not provide a complete picture of the risk and should be used sparingly and only in conjunction with other risk ranking methodologies. Furthermore, ambiguous or undefined risk ranking criteria and differing viewpoints and backgrounds of the participants can lead to the results' being diverse and the process itself, inefficient.

Average Ranking

Another methodology to rank the risk of the threat is to calculate the average of numeric values assigned to risk ranking categories. One such risk ranking categorization framework is DREAD, which is an acronym for Damage Potential, Reproducibility, Exploitability, Affected Users, and Discoverability. Each category is assigned a numerical range and it is preferred to use a smaller range (such as 1 to 3 instead of 1 to 10) to make the ranking more defined, the vulnerabilities less ambiguous, and the categories more meaningful.

- ***Damage Potential*** – ranks the damage that will be caused when a threat is materialized or vulnerability exploited.
  - 1 = Nothing
  - 2 = Individual user data is compromised or affected
  - 3 = Complete system or data destruction

- ***Reproducibility*** – ranks the ease of being able to recreate the threat and the frequency of the threat exploiting the underlying vulnerability successfully.

  1 = Very hard or impossible, even for administrators of the application

  2 = One or two steps required; may need to be an authorized user

  3 = Just the address bar in a web browser is sufficient, without authentication

- ***Exploitability*** – ranks the effort that is necessary for the threat to be manifested and the preconditions, if any, that are needed to materialize the threat.

  1 = Advanced programming and networking knowledge, with custom or advanced attack tools

  2 = Malware exists on the Internet, or an exploit is easily performed using available attack tools

  3 = Just a web browser

- ***Affected Users*** – ranks the number of users or installed instances of the software that will be impacted if the threat materializes.

  1 = None

  2 = Some users or systems, but not all

  3 = All users

- ***Discoverability*** – ranks how easy it is for external researchers and attackers to discover the threat, if left unaddressed.

  1 = Very hard-to-impossible; requires source code or administrative access

  2 = Can figure it out by guessing or by monitoring network traces

  3 = Information is visible in the web browser address bar or in a form

Once values have been assigned to each category, then the average of those values is computed to give a risk ranking number. Mathematically, this can be expressed as shown below.

(Dvalue + Rvalue + Evalue + Avalue + DIvalue) / 5

Figure 3.21 - **Use of an Average Ranking to rank various web application threats.**

Threat	D	R	E	A	DI	Average Rank (D+R+E+A+DI)/ 5
SQL Injection	3	3	2	3	2	2.6 (High)
XSS	3	3	3	3	3	3.0 (High)
Cookie Replay	3	2	2	1	2	2.0 (Med)
Session Hijacking	2	2	2	1	3	2.0 (Med)
CSRF	3	1	1	1	1	1.4 (Med)
Audit Log Deletion	1	0	0	1	3	1.0 (Low)
High: 2.1 to 3.0; Medium: 1.1 to 2.0; Low: 0.0 to 1.0						

Figure 3.22 – **Average Ranking**

The average rank and categorization into buckets such as High, Medium, or Low can then be used to prioritize mitigation efforts.

Probability x Impact (P x I) Ranking

Conventional risk management calculation of the risk to a threat materializing or to exploiting a vulnerability is performed by using the product of the probability (likelihood) of occurrence and the impact the threat will have on business operations, if it materializes. Companies that use risk management principles for their governance use the formula shown below to assign a risk ranking to the threats and vulnerabilities.

Risk = Probability of Occurrence X Business Impact

This methodology is relatively more scientific than the Delphi or the average ranking methodology. For the probability-times-impact (P x I) ranking methodology, we will once again take into account the DREAD framework. The risk rank will be computed using the formula given below.

Risk = Probability of Occurrence X Business Impact

Risk = (Rvalue + Evalue + DIvalue) X (Dvalue + Avalue)

Figure 3.23 is an example illustrating the use of the P x I ranking methodology to rank various web application threats.

From this example, we can see that the Cross-Site Scripting (XSS) threat and SQL injection threats are high risks, which need to be mitigated immediately, while the cookie replay and session hijacking threats are of medium risk. There should be a plan in place to mitigate those as soon as possible. CSRF and audit log deletion threats have a low risk rank and may be acceptable. To prioritize the

	Probability of Occurrence (P)			Impact (I)		P	I	Risk (P x I)
Threat	R	E	DI	D	A	(R+E+DI)	(D + A)	P X I
SQL Injection	3	2	2	3	3	7	6	42
XSS	3	3	3	3	3	9	6	54
Cookie Replay	2	2	2	3	1	6	4	24
Session Hijacking	2	2	3	2	1	7	3	21
CSRF	1	1	1	3	1	3	4	12
Audit Log Deletion	0	0	3	1	1	3	2	6
High: 41 to 60; Medium: 21 to 40; Low: 0 to 20								

Figure 3.23 – **Probability x Impact (P x I) ranking**

efforts of these two, high risk items (SQL injection and XSS), we can use the computed risk rank (P x I) or we can use either the probability of occurrence (P) or business impact (I) value. Since both SQL injection and XSS have the same business impact value of 6, we can use the probability-of-occurrence value to prioritize mitigation efforts, choosing to mitigate XSS first and then SQL injection, because the probability-of-occurrence value for XSS is 9, while the probability-of-occurrence value for SQL injection is 7.

While the Delphi methodology usually focuses on risk from a business impact vantage point, the average ranking methodology, when using the DREAD framework, takes into account both business impact (Damage potential, Affected users) and the probability of occurrence (Reproducibility, Exploitability, and Discoverability); however, because of averaging the business impact and probability-of-occurrence values uniformly, the derived risk rank value does not give insight into the deviation (lower and upper limits) from the average. This can lead to uniform application of mitigation efforts to all threats, thereby potentially applying too much mitigation control effort on threats that are not really certain or too little mitigation control effort on threats that are serious. The P x I ranking methodology gives insight into risk as a measure of both probability of occurrence and the business impact independently, as well as when considered together. This allows the design team the flexibility to reduce the probability of occurrence or alleviate the business impact independently or together, once it has used the P x I risk rank to prioritize where to focus its mitigation efforts. Additionally, the P x I methodology gives a more accurate picture of the risk. Notice that in the average ranking methodology, both cookie replay and session hijacking threats had been assigned a medium risk of 2.0. This poses a challenge to the design team: which threat must one consider mitigating

first? However, in the P x I ranking of the same threats, you notice that the cookie replay threat has a risk score of 24, while the session hijacking threat has a risk score of 21, based on probability of occurrence and business impact. This facilitates the design team's consideration of mitigating the cookie replay threat before addressing the session hijacking threat.

Document and Validate

The importance of documenting the threat model cannot be underestimated because threat modeling is iterative and, through the life cycle of the project, the protection controls to address the identified threats in the threat model need to be appropriately implemented, validated, and the threat model itself updated.

Threats and controls can be documented diagrammatically or in textual format. Diagrammatic documentation provides a context for the threats. Textual documentation allows for more detailed documentation of each threat. It is best advised to do both. Document each threat diagrammatically and expand on the details of the threat using textual description.

When documenting the threats, it is recommended to use a template to maintain the consistency of documenting and communicating the threats. Some of a threat's attributes that need to be documented include the type of threat with a unique identifier, the description, the threat target, attack techniques, security impact, the likelihood or risk of the threat's materializing, and, if available, the possible controls to implement. *Figure 3.24* depicts the textual documentation of an injection attack.

Threat Identifier	T#0001
Threat Description	Injection of SQL commands
Threat targets	Data access component. Backend database.
Attack techniques	Attacker appends SQL commands to user name, which is used to form an SQL query.
Security Impact	Information Disclosure. Alteration. Destruction (Drop table, procedures, delete data etc.). Authentication bypass.
Risk	High

Figure 3.24 – **Threat documentation**