

# Cyberattack threat modeling and risk mitigation strategies for your devices



# Table of contents

<b>Introduction</b>	<b>3</b>
<b>What are we working on?</b>	<b>3</b>
<b>What can go wrong?</b>	<b>4</b>
STRIDE	5
Attack tree	6
Cyberattack lifecycle	7
<b>What are we going to do about it?</b>	<b>9</b>
Rank the threats	9
Microsoft DREAD	9
CVSS by Mitre	10
<b>Suggest countermeasures and mitigation</b>	<b>10</b>
Mitigation strategies	11
<b>Did we do a good job?</b>	<b>11</b>
Document completeness	11
Vulnerability assessment	12
<b>HCLTech's offerings</b>	<b>13</b>
<b>Conclusion</b>	<b>13</b>
<b>Bibliography</b>	<b>14</b>

# Introduction

Embedded IoT devices and their associated ecosystem are complex. They cannot be secured by the traditional one-size-fits-all security blocks approach. The basic security building blocks, as described in the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53 and ANSI/AAMI/IEC 80001, are just the baseline security element. Actual security requirements are derived by understanding the system's interaction with external entities and identifying possible threats and attacks. Now, to mitigate the identified attacks, the system designer should work in a proactive mode and devise a strategy to override them. They shouldn't wait for the hackers to exploit the system and then provide a fix. Instead, they should think like an attacker and identify attack goals and techniques that could be used to compromise the system. They should be able to fix those security loopholes even before they can be exploited.

One cannot envisage all possible threats, but most of the attacks follow a known pattern and can be identified. Also, threat identification is an iterative procedure that starts from the system requirement phase and continues until the system's end of life. The system designers keep looking for the threats and mitigation strategy, which gets converted to security requirements or additional security goals along with the traditional security stack. The basic idea is to think like an attacker, look for possible attack vectors and, accordingly, define security goals for the system.

Threat modeling is all about analyzing the representations of a system to understand the system vulnerabilities and security loopholes and providing a mitigation strategy.

At the highest levels, or in layman's terms, it is about answering the below four questions:

- What are we working on?
- What can go wrong?
- What are we going to do about it?
- Did we do a good enough job?

While threat modeling a system, if we start answering these questions one by one, we will have a well-defined threat model and mechanism to test its effectiveness. Let's begin the journey of threat modeling by analyzing the security of a wristband that sends patient health information to doctors via mobile applications and the cloud. Once the doctor gets all the information and vital statistics, they suggest appropriate health measures to the patient. Let's start the journey of threat modeling by answering these questions one by one.

## What are we working on?

The first step of the threat modeling process is to gather information about the system, find the entities involved and their interactions within the system and/or with the external entities, data flows, etc. Again, it is impossible to get all the details in one go, but somewhere we must get started and this is how we start.

- Get the list of entities involved in a wristband that keeps sending the data to the cloud. The entities involved are the hardware device, mobile application connecting to the device and the cloud and the clinicians that suggest the measures. Get the list of use cases and misuse cases to understand system usage pattern (e.g., the wristband is used to send the patient's data as and when it gets connected to the mobile application and then the data is sent to the cloud). A basic misuse case could be allowing the mobile application to send additional data that is not required (e.g., location).
- Then get the entry points to be used by the attackers to gain access to the system. For example, remote login to the system is an entry point that can be used to attack the system. The level of access required at the entry point should also be considered while documenting the entry points.
- The assets that need to be protected, i.e., sensitive data, proprietary code that could be of some value to the attacker (e.g., the patient's health record).
- Trust levels represent the access controls and access rights granted to the external entities interacting with the system. In this case, the app store is an external entity that provides the application and its upgrade and then within the system, there are other boundaries.
- Third party or standards and technology used to implement the solution. This helps in focusing on technology-specific threats and determining the most appropriate mitigation techniques.
- Data flow diagrams to understand the flow of data and identify the critical paths where the data needs to be secured and identify the trust boundary of normal and privileged access to data.
- Sequence Diagrams and state machines to understand the system state at any point in time, which can help in keeping the system in a known state always.

The purpose of gathering all this information is to be able to identify the core function of the system and define the critical data flow. For example, the process of sharing patient's data with external servers, configuration and OTA updates, along with the trust boundaries. This is an iterative process as the diagrams and details keep evolving as we progress with the different phases of Software Development Lifecycle (SDLC).

## Identify threats or what can go wrong?

Once the data is gathered, the next step is to identify the threats and attacks that could affect the system. Many methodologies are used, including STRIDE and Attack Trees. There are many other approaches – a comparison of them is given here. With practice, we have realized that a combination of multiple approaches works well. One needs to practice and try multiple techniques and then figure out what works well for the specific use case. The idea is to start with the gathered information, identify the possible threats using STRIDE, identify the attack trees,

identify the steps that can be used by the attacker to get into the system by cyber life cycle and then finally apply ATT&CK Framework and get the list of what can go wrong.

## STRIDE

STRIDE is very useful when it comes to categorizing and identifying potential threats. It is an acronym for six categories of threats against a system.

- **Spoofing** – Impersonating to obtain access to the system
- **Tampering** – Altering the system or data, making it less useful to the intended users
- **Repudiation** – Plausible deniability of actions taken under a given user or process
- **Information disclosure** – Release of information to unauthorized parties (e.g., a data breach)
- **Denial of service** – Making the system unavailable to the intended users
- **Elevation of privilege** – Granting a user or process additional access to the system without authorization.

STRIDE is the starting point where the system designers identify the threats and map them to the specific security controls, as shown in the table below:

Type	Examples	Security control
Spoofing	Threat action aimed to illegally access and use another user's credentials, such as username and password. This could be achieved by guessing the password, dictionary attack or tricking the user to reveal their credentials.	Authentication
Tampering	Threat action aimed to maliciously modify data at rest or data in transit. This can be achieved by getting illegal access to the system or the network.	Integrity
Repudiation	Exchange information without leaving an audit trail behind. This can be achieved by forging origin of information or proof of delivery.	Non-Repudiation
Information disclosure	Illegal access to information by reading the data in transit or by accessing the system's files illegally.	Confidentiality
Denial of service	Valid users are denied access to execute valid operations. This can be achieved by bombarding webserver with too many requests and making it unavailable temporarily.	Availability
Elevation of privilege	Attain privilege access to the system data and controls which can be used to compromise the security of the system.	Authorization

**Table 1** STRIDE definition

The basic approach that we follow for applying STRIDE is STRIDE per element. The table below shows the applicable threats on entities:

Element	Spoof	Tampering	Repudiation	Info disclosure	DoS	EoP
External entity	x		x			
Process	x	x	x	x	x	x
Data store		x		x	x	
Dataflow		x	?	x	x	

**Table 2** STRIDE per entity

Note: '?' indicates that it will be applicable during the condition where the data store contains logs

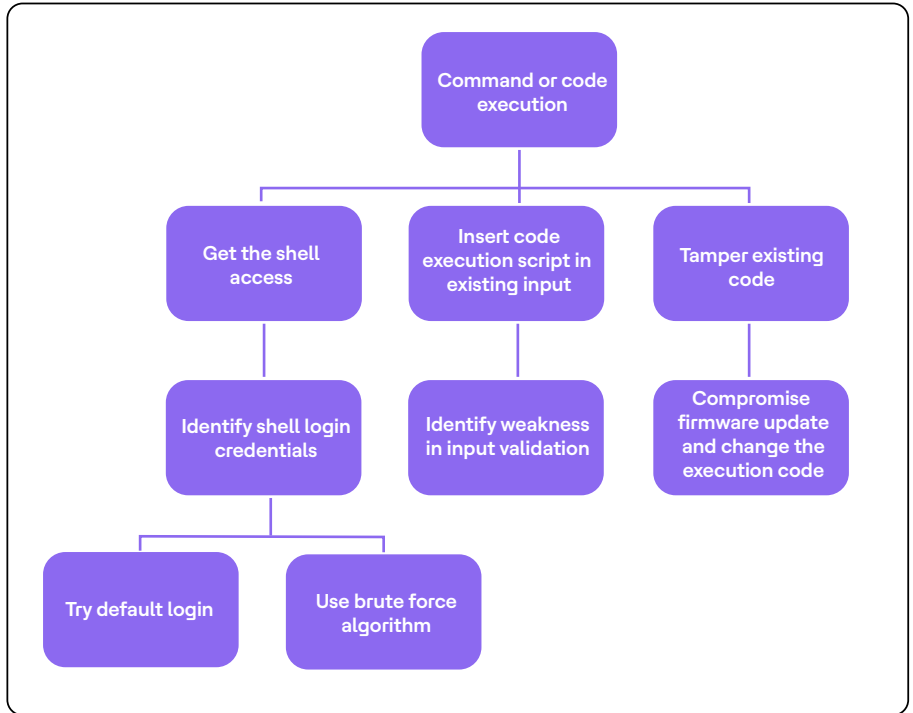
In our example, external entities are the patient's cellphone, Appstore, cloud infrastructure and doctor's computer, whereas the device control plane, mobile application, update application and cloud application are processes and patient data and service-related data are the Datastore. Hence, we get the list of controls to be applied to the specific entity so that the threats can be mitigated.

### Attack tree

This is a goal-oriented approach where the attacker's goal is identified and then the path to attack is defined using the identified risks. Common threats grouped by people, process and technology that includes network, host and application categories are used to identify the threats specific to the system.

When the threat list is applied to the system architecture, prepare an attack tree and attack patterns, which will help in identifying the potential threats to the system.

An attack tree represents the attack methodologies used by the attacker to attack the system. Here root node represents the ultimate goal, and the leaf nodes define the attack techniques used to reach the goal. A simple example is shown below:



**Figure 1** Attack tree

Attack patterns are generic representations of commonly occurring attacks that can occur in a variety of different contexts. An example of an attack pattern, the code-injection attack pattern, is shown in the table below:

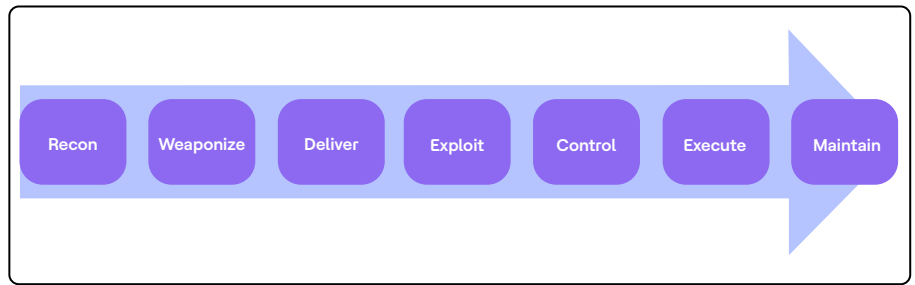
Pattern	Code injection attacks
Attack goals	Command or code execution
Required conditions	<ol style="list-style-type: none"> <li>I. Weak input validation</li> <li>II. Code from the attacker has sufficient privileges on the server</li> </ol>
Attack technique	<ol style="list-style-type: none"> <li>I. Identify the program on the target system with an input validation vulnerability.</li> <li>II. Create code to inject and run using the security context of the target application.</li> <li>III. Construct input value to insert code into the address space of the target application and force a stack corruption that causes application execution to jump to the injected code.</li> </ol>
Attack results	Code from the attacker runs and performs malicious action.

**Table 3** Code injection attack

The sequence of attacks can help in identifying the required security controls.

### Cyberattack lifecycle

This methodology is based on modeling an attack as a lifecycle and trying to break the cycle as early as possible. The below figure shows a basic attack lifecycle:



**Figure 2** Attack lifecycle

It is useful since attacks are live events and if an attack is mitigated at any stage, the attacker may try some other technique to break into the system. Though the framework is depicted as an arrow, there might be loopbacks where an attacker changes their technique to overcome the mitigation. Following are the seven stages of the lifecycle:

- **Recon** – Identify, select and investigate a target
- **Weaponize** – Identify the tools that can be executed on the target system to identify vulnerability
- **Deliver** – Vulnerability is ready to be exploited and all the required tools are in place
- **Exploit** – Attack the target
- **Control** – Take control of the target
- **Execute** – Execute the plan and achieve objectives
- **Maintain** – Maintain access to the target computer or network

A major challenge in applying the cyberattack lifecycle is that it expects security professionals to have detailed knowledge of what an attacker may do. For example, an attacker could exploit the wristband by getting the command line interface and then executing its own code, but anticipating such attacks need expertise or real-world data where such attacks have already occurred.

To reduce the amount of expertise needed, there are several public repositories and frameworks that describe attackers' actions on the basis of real-world data. One such repository is the MITRE ATT&CK framework. The ATT&CK framework is a knowledge base of adversary tactics and techniques based on real-world observations. There are several variations of the ATT&CK framework, including ones for enterprise computing, mobile and industrial control systems (ICS).

At a high level, the ATT&CK framework is broken down into tactics, techniques, sub-techniques and procedures that can be used by attackers to attack the target. Using this data, the security designers too can create a database of threats applicable to the system. Finally, after applying all these techniques, we have a list of threats applicable to the system, threat actors and their roles, along with the data flow stories, though the exercise needs to be repeated at every stage and also later on at any feature change.

# What are we going to do about it?

Now we have the applicable threats and we need to figure out the course of action to be taken. The first step is to prioritize the threats by ranking them and then accordingly, the mitigation strategy can be defined.

## Rank the threats

They can be ranked using either of the following two approaches.

### Microsoft DREAD

DREAD modeling is a technique to calculate the risk and rate them according to the damage potential, reproducibility, exploitability, affected users and discoverability. The following algorithm is used to calculate the risk as follows:

$$\text{Risk} = (\text{Damage} + \text{Reproducibility} + \text{Exploitability} + \text{Affected users} + \text{Discoverability}) / 5$$

The calculation always produces a number between 0 and 10, the higher the number, the more serious the risk. Here are some examples of how to quantify the DREAD categories.

**Damage potential** - This is used to determine the extent of damage caused when the exploitation occurs.

Here, 0 = Nothing, 5 = Individual user is impacted, so only a single user data is compromised and 10 = Impacts the whole system.

**Reproducibility** - This determines the efforts and access controls required to reproduce the exploit.

Here, 0 = Only administrator with highest privilege can exploit the vulnerability, 5 = A valid user can exploit the system using 1-2 commands, 10 = The system can be exploited without user authentication and access control using the address bar of web browser.

**Exploitability** - This is used to determine the level of proficiency required to launch the exploit.

Here, 0 = Only proficient programmer with advanced attack tools can exploit the system, 5 = The system can be exploited using the readymade scripts available on internet and 10 = The system can be exploited using a web browser.

**Affected users** - This criterion determines the number of users that will be impacted if the exploit is launched.

Here, 0 = None, 5 = some users are affected and 10 = All users.

**Discoverability** - This category is used to determine the efforts required to unveil the exploit.

Here, 0 = Privilege access and source code access is required to exploit the vulnerability, 5 = System log access and normal user access is required to exploit the threat, 9 = Details of faults like this are already in

the public domain and can be easily discovered using a search engine and 10 = The exploit can occur by using information visible on address bar or the form.

## CVSS by Mitre

This is the standard approach designed to calculate the severity ratings for software vulnerabilities using a universally accepted standard. The CVSS assessment measures the vulnerability using the following metrics :

- Base metrics are used to calculate the inherent security qualities in terms of access vector, complexity, confidentiality, integrity, authentication and availability of the system
- Temporal metrics are used to calculate the exploitability of the vulnerability over time as and when the vulnerability is exposed and remediation is provided
- Environmental metrics are used to calculate the holistic impact of the vulnerability and is dependent on the implementation and deployment environment

# Suggest countermeasures and mitigation

The first step is to identify the countermeasures of known and applicable threats if they are available. For STRIDE, the mitigation controls can be defined as:

STRIDE threat & mitigation techniques list	
Threat type	Mitigation techniques
Identity spoof	Appropriate authentication, provide secure storage for keys and sensitive data
Data tampering	Appropriate authorization, Hash, digital signature, implement secure protocol to transfer data
Repudiation	Digital signatures, timestamps, audit trails or using logs to trace data path
Information disclosure	Authorization, secure and GDPR aware protocols, encryption
Denial of service	Appropriate authentication, authorization
Elevation of privilege	Execute with least privilege

**Table 4** STRIDE mitigation controls

Once the possible attacks and their mitigation strategies are identified, it's time to classify the threats using the following criteria:

- Known threats: System is delivered along with these threats. These threats are documented in the user manuals and no security controls are available to mitigate these threats. If exploited these threats can affect the system adversely.

- Partial threats: These threats are partially mitigated by applying security control that can limit the impact on the system.
- Mitigated Threats: These threats are fully mitigated with appropriate security controls and cannot be exploited by adversaries.

### Mitigation strategies

The objective of risk management is to reduce the impact that the exploitation of a threat can have on the application. This can be done by responding to a threat with a risk mitigation strategy. In general, there are five options to mitigate threats:

- Do nothing: For example, hoping for the best
- Inform about the risk: For example, warn the user population about the risk
- Mitigate the risk: FPut countermeasures in place
- Accept the risk: After evaluating the impact of the exploitation (business impact)
- Transfer the risk: For example, through contractual agreements and insurance
- Terminate the risk: Shutdown, turn off, unplug or decommission the asset

The decision of which strategy is most appropriate depends on the impact an exploitation of a threat can have, the likelihood of its occurrence and the costs for transferring (i.e., costs for insurance) or avoiding (i.e., costs or losses due to redesign) it. That is, such a decision is based on the risk a threat poses to the system. Therefore, the chosen strategy does not mitigate the threat itself, but the risk it poses to the system.

## Did we do a good job?

This is the final step and we can evaluate it by knowing the quality of documents created during this process and by evaluating the system against the availability vulnerability test. This can be done in two stages: One is the evaluation of the documents created and the other is the vulnerability assessment of the system.

### Document completeness

The document can be evaluated based on coverage of high-value dataflows like:

- Authentication protocols used with external servers
- Commands used for the configuration and programming of the device
- Firmware upgrade
- Sharing critical data with external servers
- Transmitting and silencing the alarm event

- Procedures to restore from backups

Following is the checklist for the evaluation of coverage of threats pertaining to the data flows:

**Completeness** – The information captured by STRIDE analysis or attack trees is complete, or there are gaps in between. Also, the mitigation control should be identified and documented.

**Clarity and specificity** – Are the threats clearly understood and mitigated along with the specific details of the security control (e.g., algorithm, protocol version, etc.)

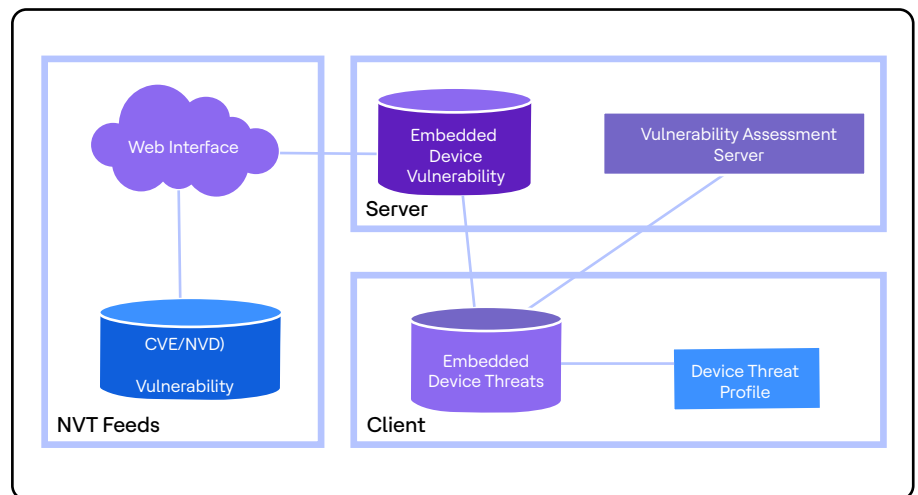
**Traceability and consistency** – Any correlation between multiple threat models that should be tracked to provide consistency in information and avoid duplicity.

**Roles and responsibilities** – Risk identification can be done with well-defined roles and responsibilities of entities involved, so this information should be captured.

**Assumptions and rationales** – All the assumptions and rationales behind any decision taken should be documented.

### Vulnerability assessment

Vulnerability assessment is a process that defines, identifies and classifies the security holes (vulnerabilities) in a computer, embedded device, network or communications infrastructure using automated scanning tools and available threats database.



**Figure 3** Vulnerability assessment

The above example shows an open-source tool, OpenVAS, used to assess the vulnerability of the system using potential vulnerability to the device. The assessment can be performed externally using tools like Nmap for port scanning and OpenVAS for service enumeration and internally using system-specific tools that can assess the security of the system in terms of authentication, authorization, etc.

The final output of the vulnerability assessment is the report containing the actual threats that need to be remediated, removing the false positives.

## HCLTech's Offerings

Threat modeling is not just for new products, but may also be useful when carried out on legacy devices. It identifies threats that could adversely impact the safety and security of a device. Threat modeling is an information-generating process that informs quality process activities. Creating a threat model is not a paperwork exercise to check a compliance box. Instead, the threat model helps in taking decisions about design, development, testing and post-market activities for the system. It serves as a baseline document to make security decisions and identify security goals for internal stakeholders, customers and regulatory reviewers.

Our experts at HCLTech have extensive experience in threat modeling and risk assessment. We provide end-to-end security engineering services for new and legacy products. Some of our offerings include:

- Defining security requirements
- Perform threat modeling
- Identify the mitigation strategy
- Risk assessment

We work in various domains, including medical devices, automotive, aerospace & defense, industrial control systems, consumer electronics, etc. We also have expertise in threat modeling software-based applications that include desktop, mobile and cloud-based applications.

## Conclusion

In this document, threat modeling is described via a four-question framework. A step-by-step approach to threat modeling results in a better understanding of the system and the system designers get an insight into how an attacker can break into the system. They are better prepared to handle failures when the device is deployed in the field. Threat modeling is a continuous process and is performed repetitively throughout the SDLC. It begins at the early design stage because the designers have more flexibility to accept and introduce the changes that can improve the security and safety portfolio of the system. But threat modeling does not stop there. It is executed at every stage and threats are mitigated as and when they are found right from the conception to development and deployment to future upgrades. These practices help organizations update their threat mitigation strategies based on their learnings from design, development, testing and field monitoring and then working on feedback.

# Bibliography

- Medical Device Innovation Consortium, "Playbook-for-Threat-Modeling-Medical-Devices", Nov 2021
- Software Engineering Institute, "Threat Modeling: A Summary of Available Methods," August 2018. Available: <https://resources.sei.cmu.edu/library/assetview.cfm?assetid=524448>  
Automating Threat Modeling through the Software Development Life-Cycle Available
- Privacy Engineering Objectives and Risk Model –Discussion Deck Available: [NIST Privacy Engineering Objectives and Risk Model – Discussion Deck](#)
- The MITRE Corporation, "Overview of How Cyber Resiliency Affects the Cyber Attack Lifecycle," 2015. Available: <http://www2.mitre.org/public/industryerspective/documents/lifecycle-ex.pdf>
- B. Schneier, "Attack Trees," Schneier on Security, 12 1999. Available: [https://www.schneier.com/academic/archives/1999/12/attack\\_trees.html](https://www.schneier.com/academic/archives/1999/12/attack_trees.html)
- L. Martin, "Cyber Kill Chain," Available: <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>
- The MITRE Corporation, "Overview of How Cyber Resiliency Affects the Cyber Attack Lifecycle," 2015. Available: <http://www2.mitre.org/public/industryerspective/documents/lifecycle-ex.pdf>
- MITRE Corporation, "MITRE ATT&CK Framework," Available: <https://attack.mitre.org/>
- Threat Modeling Manifesto Working Group, "The Threat Modeling Manifesto," Available: <https://www.threatmodelingmanifesto.org/> [Accessed 22 August 2021].
- OWASP, "Threat Modeling Cheat Sheet," Available: [https://cheatsheetseries.owasp.org/cheatsheets/Threat\\_Modeling\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html) [24] Software Engineering Institute, "Threat Modeling: A Summary of Available Methods," August 2018. Available: <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=524448>
- MITRE, "Common Weakness Scoring System (CWSS)," September 2014. Available: [https://cwe.mitre.org/cwss/cwss\\_v1.0.1.html](https://cwe.mitre.org/cwss/cwss_v1.0.1.html) [Accessed 21 August 2021].
- Wikipedia, "DREAD (risk assessment model)," Available: [https://en.wikipedia.org/wiki/DREAD\\_%28risk\\_assessment\\_model%29](https://en.wikipedia.org/wiki/DREAD_%28risk_assessment_model%29) [Accessed 21 August 2021].

# Author information



**Shivani Agarwal**

<https://www.linkedin.com/in/shivani-agarwal-solutionarchitect/>

# HCLTech | Supercharging Progress™

HCLTech is a global technology company, home to 222,000+ people across 60 countries, delivering industry-leading capabilities centered around Digital, Engineering and Cloud powered by a broad portfolio of technology services and software. The company generated consolidated revenues of \$12.3 billion over the 12 months ended December 2022. To learn how we can supercharge progress for you, visit [hcltech.com](https://hcltech.com).

[hcltech.com](https://hcltech.com)

