Threat modeling - A comparison between PASTA and STRIDE

Praneet Kala and Ivan Khudur

April 24, 2023

I/We certify that generative AI, incl. ChatGPT, has not been used to write this essay. Using generative AI without permission is considered academic misconduct.

1 Introduction

The concept of Development Operations (DevOps) is being able to automate development, deployment and infrastructure monitoring and is a culture shift towards collaboration between development, quality assurance and operations [5]. DevOps makes it possible to rapidly deliver services and software. However, the positive effects may easily be disrupted unless one of the most important parts of development is taken into consideration: security [11]. The amount of cyber attacks are increasing every year, and this threat should not be taken lightly [1]. Failing to take measures when it comes to safety-critical systems may and have resulted in major financial losses, misuse of sensitive information and in worst case, loss of life [12].

By inserting the security aspect into DevOps, you get the coined term DevSecOps. It is a mindset that integrates the security aspect into the start of the project, making it a continuous part throughout development [11]. Threat modeling is one security method that can be used to mitigate security risks earlier in the development process [4]. In this essay, we will go through 2 popular threat modeling frameworks and compare them: PASTA (**P**rocess for **A**ttack **S**imulation and **T**hreat **A**nalysis) and STRIDE (**S**poofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service, **E**levation of Privilege).

2 Threat modeling methodologies: PASTA and STRIDE

Many, if not all, organisations these days want to ensure they are safe against cyber attacks and security threats. Architectural weaknesses can also be revealed if the threat modeling process is started earlier in the development life cycle [18]. On the downside, threat modeling is still a growing industry and methods can be time consuming and resource intensive so organisations sometimes avoid it, hence the need for more accessible tools and frameworks [20].

2.1 What is PASTA?

PASTA is a flexible framework that can be adopted to various scenarios. It was founded in 2015 by VerSprite, and can be used to identify threats regardless of the stage of the development process. It is a linear process that allows collaboration between different parties in an organization, and focuses on risks related to the business context [19]. Furthermore, it is risk-centric, i.e. the focus lies on the security threats that are most relevant to the business and pose the highest risk or has the largest impact [8].

The PASTA framework consists of 7 steps, as illustrated in Figure 1. Each step outputs information that the next step uses [19].

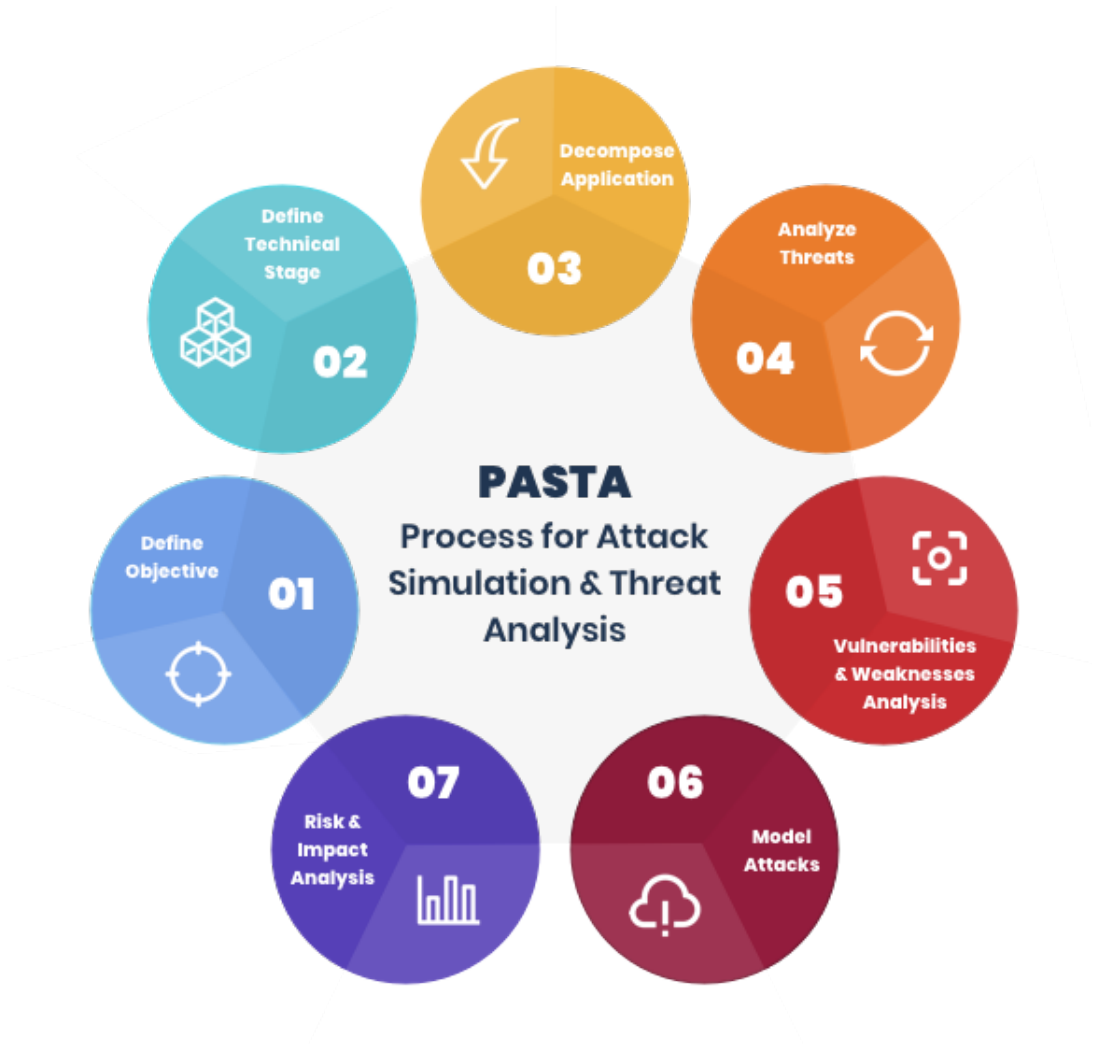

Figure 1: 7 steps of PASTA [21]

2.1.1 The 7 steps of PASTA

The steps can be summarized as follows:

1. **Define objectives** - Identify and define objectives of creating the software. This requires collection of various documents, such as specifying business- and functionality requirements, security policies, standards and guidelines [19].

2. **Define technical scope** - Define how the application should run, i.e. the technology that is used. What is used and how is it setup? What dependencies exists? Should anything be in the cloud, and what third party solutions will be used? [8]

3. **Decomposition of application** - Focuses on how everything is connected and communicates with each other. A data flow diagram (DFD) is created, and any implicit trust models are mapped [19]. This is because implicit trust models may be exploited, as written in [10].

4. **Analyse threats** - This step focuses on studying relevant threats to the application based on the gathered information, and build attack scenarios. It is done by gathering threat information from various resources, to fully understand them and map actual threats (i.e. threats that are likely to occur) [8] [19].

5. **Analyse vulnerabilities** - Vulnerabilities in the code and design are analysed, to better understand potential security risks that will affect the objectives (that were defined in step 1 of PASTA). The severity of the vulnerabilities can be evaluated in this step using different standards, such as CVSS (Common Vulnerability Scoring System) [8] [19].

6. **Attack analysis** - The vulnerabilities and threats that have been documented are mapped together to determine how likely the weaknesses are to be exploited. This is done using an attack tree of either the entire application, or some part of it, where vulnerabilities are mapped to the different nodes. The nodes of the attack tree corresponds to a component of the application (or part of the application) [19].

7. **Risk and impact analysis** - The final step is to combine all the information that has been gathered so far to create a risk profile. This is then used to create strategies to mitigate the risks [19].

2.2 What is STRIDE?

STRIDE is used to build a secure system and identifies security requirements by considering security aspects such as potential threats to target systems and services [7]. STRIDE is a mnemonic of 6 types of security threats. Each letter of the word is related to a security property and can be associated with specific security attributes [7], as shown in Table 1:

Threat	Security property	Threat definition
Spoofing	Authentication	Impersonate something or someone else
Tampering	Integrity	Modify data or code
Repudiation	Non-repudiation	Claim to have not performed an action
Information disclosure	Confidentiality	Expose information to someone not authorized see it
Denial of service	Availability	Deny or degrade service to users
Elevation of privilege	Authorization	Gain capabilities without proper authorization

Table 1: Correlation between six threats in STRIDE and security properties [7]

Microsoft's STRIDE methodology aims to ensure that an application meets the security requirements of Confidentiality, Integrity, and Availability (CIA) [2]. It is a relatively lightweight approach and consists of five high-level steps as proposed by [6]. Another structured approach that has been presented by Open Web Application Security Project (OWASP) discusses three high-level steps as described by [3]. In this essay, we will focus on the OWASP process.

2.2.1 Step 1 - Decompose the application

The first step is to decompose the application. This means that you have to understand the system and the components involved (excluding physical components), and how it is

all connected and communicates with each other (similarly to step 3 of PASTA). An important part of this step is to gather information, such as:

- Architectural diagrams
- Project documentation and requirements

This makes it possible to identify and evaluate different items and/or areas of interest that an attacker would be interested in. This sets up the basis for creating a DFD in step 2. [9].

2.2.2 Step 2 - Determine and Rank Threats

In the second step, a DFD is created. It is an architectural representation that visually shows the flow of data between various key components. In this part it is important to stick with the key components, to reduce the difficulty in understanding the system. To make this work effectively, it is necessary to engage with various teams in the project, including non-technical team members. A good example of how a DFD is constructed for a Hospital Information System (HIS) is shown in Figure 2. Being able to capture the necessary elements such as external entities, data flows, processes, and data stores in a lightweight manner is a good start among stakeholders [15].

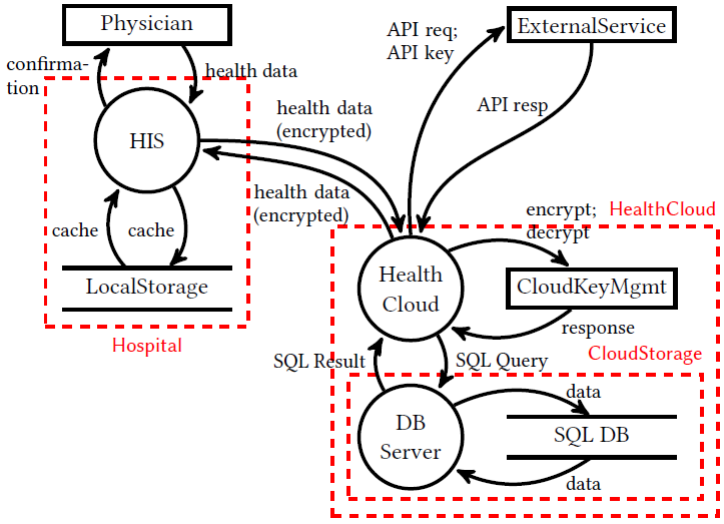

Figure 2: Data Flow Diagram of a Hospital Information System (HIS) [15]

The threats are then categorised and STRIDE is applied to systematically identify threats. This is an iterative process after all the possible threats have been drawn up and evaluated. A good way to view and analyse specific situations, for example a *information disclosure*, is to use a threat tree diagram as show in Figure 3.

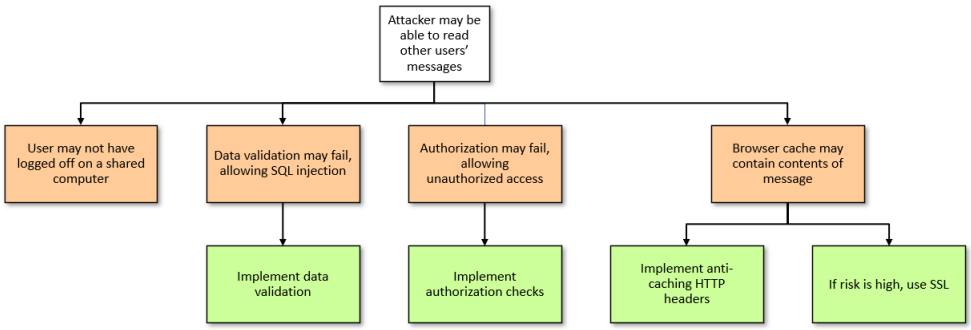

Figure 3: Threat Tree Diagram [3]

Lastly, this step requires the threats to be ranked according to highest priority and using risk factors for creating a strategy for countermeasures.

2.2.3 Step 3 - Determine Countermeasures and Mitigation

The final step is defining countermeasures to mitigate potential threats that could occur. In most cases, there are protective measures available (policies, security controls, authentication, authorisation protocols) to prevent threats and once these are identified, they are linked to each of the threats that were ranked. If there are no countermeasures for threats listed, they are considered vulnerabilities. The final list is then categorised into three different criteria (non-mitigated, partially mitigated and fully mitigated) and added to the documentation.

3 Comparing STRIDE and PASTA

STRIDE and PASTA are 2 popular threat modeling methodologies that we have explained in the previous sections [14]. STRIDE comes from Microsoft and has been highly influenced by them, making it a more developer focused threat modeling process. PASTA on the other hand comes from a cybersecurity consulting organisation that have a focus from an attackers perspective [16]. PASTA also incorporates business impact analysis and the strategic objectives of an organisation when going through the process, and it is a collaborative process that involves all IT departments in the process [16]. STRIDE has been around for a longer time than PASTA and has the benefit of being rather easy to use and understand. However, the simplicity of STRIDE also make it less reliable [19].

Following the methodology it is evident that STRIDE focuses on identifying vulnerabilities with regards to the 6 different threats that compose its name. It is therefore rather easy to miss out on other cybersecurity threats [13]. PASTA on the other hand is more encompassing when it comes to identifying threats. This is because part of the PASTA process is to build a threat library, in which relevant threats to the application are added as mentioned in section 2.1.1. STRIDE is very traditional and more categorical, and creates an association of risks. PASTA on the other hand considers relevant threats, simulates attacks on assets, and leverages other processes for security and can find vulnerabilities such as static code scanning, vulnerability management, risk management and maps threats and attacks [17].

Though there are differences between PASTA and STRIDE, both methodologies make use of decomposing the application/software that you use your threat model on. In both methodologies DFD's are used to understand how data flows between different parts of the application and if there are any implicit trust model implemented that could be exploited [8].

4 Conclusion and Reflection

It is clear that both methodologies follow a step-by-step process to reach an end goal, which is to create a risk profile that is then used to formulate a strategy to mitigate risks. PASTA is a more elaborate methodology that involves many different parties in the business and is able to provide a more comprehensive list of relevant threats. This makes it suitable to identify gaps in security that should be fixed, however due to its scope it becomes rather time consuming. STRIDE on the other hand is more focused on a developer-driven approach and can be implemented rather quick. The quickness comes however with a trade-off becomes apparent as STRIDE is limited in identifying relevant threats. That said, both methodologies fulfill a purpose: to identify threats and help make software more secure.

The research that was performed for threat modeling methodologies and processes is very relevant in DevSecOps. It is a vital component to consider for risk management in DevOps teams. Throughout the research of the 2 chosen methodologies we encountered many other threat model methodologies. We chose STRIDE because it is old but still relevant, while PASTA is rather new and becoming ubiquitous across organizations and businesses. This makes PASTA relevant DevSecOps considering it built upon collaboration and you get strategic input from a variety of stakeholders.

To wrap it up, here is a quote to chew on: *"If PASTA can't fix it, it's a serious problem".*

References

[1] Moatsum Alawida et al. *A deeper look into cybersecurity issues in the wake of Covid-19: A survey*. Vol. 34. 10. 2022. DOI: 10.1016/j.jksuci.2022.08.003. URL: https://www.sciencedirect.com/science/article/pii/S1319157822002762.

[2] Archer Charles. *What is STRIDE Methodology in Threat Modelling?* May 2022. URL: https://www.koenig-solutions.com/blog/stride-methodology-in-threat-modelling.

[3] Larry Conklin, Victoria Drake, and Sven strittmatter. *Threat Modeling Process*. Sept. 2022. URL: https://owasp.org/www-community/Threat_Modeling_Process.

[4] Simone Curzi et al. *Integrating threat modeling with DevOps*. July 2022. URL: https://learn.microsoft.com/en-us/security/engineering/threat-modeling-with-dev-ops.

[5] Christof Ebert et al. "DevOps". In: *IEEE Software* 33.3 (2016). ISSN: 19374194. DOI: 10.1109/MS.2016.68. URL: https://ieeexplore.ieee.org/document/7458761.

[6] Rafiullah Khan et al. "STRIDE-based threat modeling for cyber-physical systems". In: *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT-Europe 2017 - Proceedings*. Vol. 2018-January. 2017. DOI: 10.1109/ISGTEurope.2017.8260283. URL: https://ieeexplore.ieee.org/abstract/document/8260283.

[7] Kyoung Ho Kim, Kyounggon Kim, and Huy Kang Kim. "STRIDE-based threat modeling and DREAD evaluation for the distributed control system in the oil refinery". In: *ETRI Journal* 44.6 (2022). ISSN: 22337326. DOI: 10.4218/etrij.2021-0181. URL: https://onlinelibrary.wiley.com/doi/10.4218/etrij.2021-0181.

[8] Nick Kirtley. *PASTA Threat Modeling*. July 2022. URL: https://threat-modeling.com/pasta-threat-modeling/.

[9] Nick Kirtley. *STRIDE Threat Modeling Example for Better Understanding and Learning*. Aug. 2022. URL: https://threat-modeling.com/stride-threat-modeling-example/.

[10] Biplob Paul and Muzaffar Rao. "Zero-Trust Model for Smart Manufacturing Industry". In: *Applied Sciences (Switzerland)* 13.1 (2023). ISSN: 20763417. DOI: 10.3390/app13010221. URL: https://www.mdpi.com/2076-3417/13/1/221.

[11] Redhat. *What is DevSecOps*. Mar. 2023. URL: https://www.redhat.com/en/topics/devops/what-is-devsecops.

[12] Mary Sánchez-Gordón and Ricardo Colomo-Palacios. "Security as Culture: A Systematic Literature Review of DevSecOps". In: *Proceedings - 2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops, ICSEW 2020*. 2020. DOI: 10.1145/3387940.3392233. URL: https://dl.acm.org/doi/10.1145/3387940.3392233.

[13] Riccardo Scandariato, Kim Wuyts, and Wouter Joosen. "A descriptive study of Microsoft's threat modeling technique". In: *Requirements Engineering* 20.2 (2015). ISSN: 1432010X. DOI: 10.1007/s00766-013-0195-2. URL: https://link.springer.com/article/10.1007/s00766-013-0195-2.

[14] Simplilearn. *What is Threat Modeling: Process and Methodologies*. 2020. URL: https://www.simplilearn.com/what-is-threat-modeling-article.

[15] Laurens Sion et al. "Security Threat Modeling: Are Data Flow Diagrams Enough?" In: *Proceedings - 2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops, ICSEW 2020*. 2020. DOI: 10.1145/3387940.3392221. URL: https://dl.acm.org/doi/10.1145/3387940.3392221.

[16] ThreatModeler. *STRIDE, VAST, TRIKE, & More: Which threat modeling methodology is right for your organisation?* Aug. 2019. URL: https://threatmodeler.com/threat-modeling-methodologies-overview-for-your-business/.

[17] Toreon. *Threat Modeling can be considered as fun as cooking a good PASTA meal*. Aug. 2022. URL: https://www.toreon.com/tmi-newsletter-18-threat-modeling-can-be-considered-as-fun-as-cooking-a-good-pasta-meal-part-2/.

[18] Peter Torr. *Demystifying the threat-modeling process*. Vol. 3. 5. 2005. DOI: 10.1109/MSP.2005.119. URL: https://www.researchgate.net/publication/3437733_Demystifying_the_Threat-Modeling_Process.

[19] Tony UcedaVélez. *What is PASTA Threat Modeling?* Nov. 2021. URL: https://versprite.com/blog/what-is-pasta-threat-modeling/.

[20] Margus Välja et al. "Automating threat modeling using an ontology framework: Validated with data from critical infrastructures". In: *Cybersecurity* 3.1 (2020). ISSN: 25233246. DOI: 10.1186/s42400-020-00060-8. URL: https://link.springer.com/article/10.1186/s42400-020-00060-8.

[21] VerSprite. *The Process for Attack Simulation Threat Analysis*. 2021. URL: https://versprite.com/ebooks/leveraging-risk-centric-threat-models-for-integrated-risk-management/.