Carnegie Mellon University
Software Engineering Institute

THREAT MODELING FOR CYBER-PHYSICAL SYSTEM-OF-SYSTEMS: METHODS EVALUATION

Nataliya Shevchenko, Brent R. Frye, and Carol Woody, PhD

September 2018

Introduction

Addressing cybersecurity for a complex system, especially for a cyber-physical system-of-systems (CPSoS), requires the strategic view of and planning for the whole lifecycle of the system. For the purpose of this paper, *cyber-physical system-of-systems* is defined as a system, components of which operate and are managed independently [46]. Thus, components of a system-of-systems (i.e., systems by themselves) should be able to function fully and independently even when the system-of-systems is disassembled. These components are typically acquired separately and integrated later. Components of a system-of-systems may have a physical, cyber, or mixed nature. For the sake of simplicity, we will use the term *cyber-physical system* instead of *cyber-physical system-of-systems*.

The nature of a cyber-physical system (CPS) implies a diversity of potential threats that can compromise the integrity of the system, targeting different aspects ranging from purely cyber-related vulnerabilities to the safety of the system as a whole. The traditional approach used to tackle this matter is to employ one or more threat modeling methods (TMMs) early in the development cycle. Choosing a TMM can be a challenging process by itself. The TMM you choose should be applicable to your system and to the needs of your organization. Therefore, when preparing for the task, it makes sense to answer two questions. First, what kind of TMMs exist and what are they? And second, what criteria should a good TMM satisfy? We explored answers to the first question in *Threat Modeling: A Summary of Available Methods* [47]. This paper addresses the second question and will evaluate TMMs against the chosen criteria.

Evaluation Criteria

Before evaluating the list of potential TMM candidates, let’s discuss the criteria that they need to satisfy.

The first criteria is *Strengths and Weaknesses.* Although a number of threat modeling methods are used in the field, there is no one perfect method. Each method was developed with different points of view in mind and addresses different priorities. Some are focused on assets, others on attackers, and still others on risks. Thus, each method has strengths and weaknesses relating to which types of threats it is best suited to discover, which threats it might miss (false negatives) or mistakenly identify (false positives), and how thorough it is. Each method has its own level of maturity and time to implement. Each deals with its own mitigation strategies. These aspects determine how useful a method may be for a given situation.

The second criteria is *Adoptability*. Implementing a comprehensive methodology in an organization will impose some level of burden on everyone involved, so choosing an easy-to-use solution can be important. When this is added to a learning curve for the implementers of the methodology and associated changes to already existing processes, the overall cost of adopting the method may become prohibitive. Availability or absence of good documentation and support can be critical for success of the method adaptation.

The third criteria is *Tailorability*. No two organizations have identical development processes, no matter how similar they may be otherwise. Therefore, a good TMM candidate should be flexible enough to be tailored to the type of system, the organization's priorities, and the system development lifecycle without compromising the quality of the method. This should include whether the candidate method may be integrated into a development process, and, based on growing usage, specifically into the Agile development process. Methods to be used on CPS must be scalable and able to answer the needs of a very large and distributed system.

The fourth criteria is *Applicability to CPS.* Since this paper concentrates on CPS, our evaluation should address those aspects specifically related to CPS. One of the main characteristics of CPS is complexity. Applying the methods recursively is critical. This will account for the relationships between sub-systems, address hardware-software dependencies, and address safety-security interdependencies.

The fifth and final criteria is *Automation.* Does any automation of the method or supporting tools exist? What is its utility? Many cyber-physical systems belong to publicly- or privately-owned critical infrastructure or to government-developed weapon systems. To an organization that requires secrecy, portability of the tools to a stand-alone mode is another important feature.

In summary, we have identified a list of criteria for evaluating TMM candidates.

1. Strengths and weaknesses
    a. Maturity and usage
    b. Focus/Perspective
    c. Time to implement
    d. Effectiveness
    e. Mitigation strategies
2. Adoptability
    a. Easy to use
    b. Easy to learn
    c. Documentation and support
3. Tailorability
    a. Integration with SDLC
    b. Compatibility with agile development process
    c. Scalability
4. Applicability to cyber-physical systems

a. Coverage of safety-security interdependency
b. Integration of hardware and software threats

5. Automation
    a. Availability of tools
    b. Integration options for tools into an SDLC
    c. Portability of tools

Evaluation

In this chapter we will evaluate the following TMMs:

1. STRIDE
2. PASTA
3. LINDDUN
4. CVSS
5. Attack Trees
6. Persona Non Grata (PnG)
7. Security Cards
8. hTMM
9. Quantitive TMM
10. Trike
11. VAST Modeling
12. OCTAVE

For our evaluation, we will use the definitions and findings from *Threat Modeling: A Summary of Available Methods* [47].

Strengths and weaknesses

All of the methods in question are designed to detect potential threats, except for CVSS (which is a scoring method). The number and types of threats will vary considerably, as will the quality and consistency of the methods. Since there is no comprehensive study involving all of these methods, we can only speculate how effective and efficient they are, based on a few sources listing studies that used them [12, 15, 28] and a number of sources that employed these methods for case studies [14, 15, 20, 28, 40],.

The STRIDE method has a moderately low rate of false positives and a moderately high rate of false negatives [28]. Persona Non Grata produces few false positives and has high consistency but tends to detect only a certain subset of threat types [15]. Security Cards can help identify almost all of the threat types but produces a high number of false positives; this method is better used to address non-standard situations [15]. The study on hTMM [24] gave inconclusive results.

Since STRIDE, PASTA, LINDDUN, Trike, VAST Modeling and OCTAVE each provide well-structured and guided frameworks, they can potentially lead to more threats discovered. However, there are some disadvantages. In particular, STRIDE and LINDDUN suffer from so-called "threat explosion," when the number of threats can grow rapidly [12, 28]. Quantitive TMM combines Attack Trees, STRIDE, and CVSS, which allows it to mitigate potential threat explosions from STRIDE by applying the other two methods. The effectiveness of Attack Trees depends on how well the analysts understand both the system and its security concerns. It requires a high level of cybersecurity expertise from analysts who apply it [2].

One study introduced a formal method of timed automata in addition to Attack Trees method for modeling socio-technical attacks [1]. Timed automata is a formal method for modeling and analyzing of computer systems' behavior. It includes language and State Machine-like diagrams to describe possible states of the system. Timed automata was implemented as a tool known as UPPAAL, an integrated modeling environment [48]. The study used these two methods on cyber-physical system, and showed how to generate and validate possible attacks on a system. Even though this method combination was applied only in an academic setting, it has a potential for safety-critical cases.

Table 1 displays a summary of other relevant attributes. *Maturity* is assessed based on how well each method is defined, how often it has been used in case studies, how often it has been combined with other methods, and whether it will be maintained by the owner or a community. *Focus/Perspective* lists the point of view from which the method was designed. *Time/Effort* indicates how time-consuming and laborious the method is. *Mitigation* lists whether any mitigation strategies are provided by the method. Finally, *Consistent results* notes whether the method produces consistent results if it is repeated. (This may depend on the knowledge of those applying the method.)

	Maturity	Focus/Perspective	Time/Effort	Mitigation	Consistent results
STRIDE	High	Defender	High	Yes	No
PASTA	High	Risk	High	Yes	No
LINDDUN	High	Assets/Data	High	Yes	No
CVSS	High	Scoring	High	No	Yes
Attack Trees	High	Attacker	High	No	Yes
PnG	Medium	Attacker	Medium	No	Yes
Security Cards	Medium	Attacker	Medium	No	No
hTMM	Low	Attacker/Defender	High	No	Yes
Quantitive TMM	Low	Attacker/Defender	High	No	Yes
Trike	Low	Risk	High	Yes	No
VAST	High	Attacker	High	Yes	Yes
OCTAVE	Medium	Risk/Organization	High	Yes	Yes

Table 1. Strengths and weaknesses

Adoptability

One cannot overstate the importance of the adoptability of a method. There are very few, if any, easy TMMs. The successful implementation of a TMM requires a deep understanding of the system and extensive knowledge of cybersecurity. However, the intuitiveness of the method can ease the effort needed to learn and use it. If the method employs techniques that are already well understood and used in the field (such as architecture diagrams or brainstorming), that can help during the adoption process.

STRIDE, PASTA, LINDDUN, hTMM, Quantitive TMM, Trike, and VAST Modeling use data flow diagrams (DFDs), which are usually part of the design phase of the system's development cycle. Security Cards and PnG are types of brainstorming, which is also a widely used design technique. STRIDE, LINDDUN, and their combinations use their names as mnemonics, which naturally guides the process of threat discovery. On the other hand, complicated and vague formulas and instructions (such as those used in CVSS) or excessive or laborious steps within a method) such as those found in PASTA) can negatively impact the adoption of that method.

Table 2 summarizes the evaluation of the main attributes that contribute to the adoptability of the method.

	Easy to Use	Easy to Learn	Documentation
STRIDE	Medium	Medium	Very Good
PASTA	No	No	Very Good
LINDDUN	No	Medium	Good
CVSS	No	No	Good
Attack Trees	Yes	Medium	Good
PnG	Yes	Yes	Some
Security Cards	Yes	Yes	Very Good
hTMM	Medium	Medium	Good
Quantitive TMM	No	No	Some
Trike	Medium	Medium	Good for v1
VAST	Medium	Medium	Very Good
OCTAVE	No	No	Good

Table 2. Adoptability

Tailorability

All methods except OCTAVE are designed to be applied in the beginning of the system development life cycle (SDLC), during the requirements and design phases. This allows them to be integrated into any development lifecycle that includes these phases. Some, like PnG, Trike and VAST, integrate with agile development process better than others. PASTA and Trike explicitly map their activities not only to the requirements and design stages of SDLC, but to the implementation and test stages as well. OCTAVE is an evaluation process oriented to the organization rather than a specific system. It will not integrate well with any development cycle.

Since none of these methods were designed with a specific type of system in mind, all may be applied to any kind of system. Case studies illustrate specific tailoring of STRIDE, PASTA, CVSS, Attack Trees, hTMM, Quantitive TMM, LINDDUN, and PnG to cyber systems [13, 28, 31] and cyber-physical systems [1, 2, 3, 15, 19, 20].

OCTAVE was initially designed for large organizations; PASTA and VAST modeling were designed for large systems. The remainder of the methods may be scaled to accommodate large systems or system-of-systems relatively easily.

Applicability to CPS

As the literature review in *Threat Modeling: A Summary of Available Methods* shows, most of the methods under evaluation were used to model threats against cyber-physical systems: railway communication networks [3], drone systems [15], and the automotive industry for connected cars [20]. However, none were used as the sole modeling method. Combinations of two or more methods seemed to perform better. In many cases other techniques were added to the mix, such as the National Institute of Standards and Technology (NIST) guidelines and standards (Special Publications 800-30 [49], 800-82 [50], and 800-53 [51]), Failure Modes and Effects Analysis (FMEA), the Risk Priority Number (RPN), and Threat Agent Risk Assessment (TARA) [3, 8, 20, 39]. The methods used most often in these studies were STRIDE and CVSS. Combining methods and adding domain-specific techniques allows for deeper analysis of the system, and thus, better threat discovery.

Only one study [39] specifically talked about the importance of integrating safety analysis with cybersecurity analysis. It suggested using FMEA in addition to STRIDE, and stated that there is no conflict between these two types of analysis. In fact, combining these methods helps to identify more possible threats as well as specific points of failure. Another study mentioned that Attack Trees was developed as an adaptation of the Fault Trees technique from safety engineering [1].

Specifics of CPS requires focused attention not only on the application or system software related threats, but also on hardware and physical threats. Malware installed on a hardware or physical tampering with a component can cause cyber or cyber-physical impact and put a system into an undesirable state. Studies show that Attack Trees or frameworks like PASTA, where building Attack Trees is one of the steps, are capable of identifying physical and hardware threats including their impact on the system as a whole [1, 3, 19].

All methods that start with modeling the system, for example, data flow diagrams (STRIDE, PASTA, LINDDUN, hTMM, Quantitive TMM, Trike, and VAST), can be used recursively with some modifications. In *Software and attack centric integrated threat modeling for quantitative risk assessment* [3] a technique was described to account for the risk propagation between components called attack ports.

Automation

Very few of the methods examined were automated. In fact, most of them exist only as a framework of instructions, questionnaires and checklists. Those methods that were automated include STRIDE, implemented as the Threat Modeling Tool as a part of the Microsoft Secure Development Lifecycle (SDL) [29], the CVSS online calculator [52] (which cannot be installed as a stand-alone tool), and VAST Modeling - implemented as ThreatModeler [53], which can be installed as an on-premises solution.

The two existing portable tools (Threat Modeling Tool (STRIDE) and ThreatModeler (VAST)) can be potentially integrated into SDLC during requirements and design stages. For non-automated methods that utilize DFD or other diagrams, system design tools (e.g., Enterprise Architect, Microsoft Visio, Gliffy, NoMagic, and Cameo EA) can be used to create the diagrams. Those design tools can be integrated into SDLC.

	Automation	Portable	Tool Integration with SDLC
STRIDE	Yes	Yes	Yes
PASTA	No	Yes	No
LINDDUN	No	Yes	No
CVSS	Yes	No	No
Attack Trees	No	Yes	No
PnG	No	Yes	No
Security Cards	No	Yes	No
hTMM	No	Yes	No
Quantitive TMM	No	Yes	No
Trike	No	Yes	No
VAST	Yes	Yes	Yes
OCTAVE	No	Yes	No

Table 3: Automation and Portability

Conclusion

Examples of cyber-physical systems-of-systems (CPSoS) include rail transport systems, power plants, and integrated air defense capability. Each of these systems consist of large physical, cyber-physical and cyber-only sub-systems with complex dynamics They are connected via one or more cyber networks and operated by one or more human operators. The components of those systems are often distributed and are sometimes partially autonomous, with multi-level control and management. They are safety or life critical. Thus, threat modeling for this kind of systems should address the full spectrum of threats: kinetic, physical, cyber-physical, cyber-only, supply chain, and insider threats.

Evaluation of existing TMMs showed that there is no one method that can cover all pieces. Therefore, a framework that employed a combination of methods and techniques should be used.

- Our recommendation is to use the PASTA modeling method as the basis of this framework.
- In addition to PASTA, we recommend using components of STRIDE and LINDDUN.
- We also recommend using other tactics that address threat aspects that are not covered by these three models.

PASTA provides most detailed guidance for the process of threat modeling, including resources that can be easily adapted to different kind of systems. It can be incorporated into the existing SDLC and allows for easy addition or removal of activities from stages as needed. PASTA also mitigates the threat explosion weakness of STRIDE and LINDDUN by utilizing risk and impact analysis. This flexibility makes this combination a good candidate for a comprehensive framework.

Some modification should be done to this combination of methods to accommodate the scope of the problem. To start, PASTA should be implemented for the whole system using high level architecture and treating sub-systems as black boxes. This initial round will not require the user to go through every activity, but it should effectively define all inputs and outputs for each sub-system. Then, PASTA should be implemented recursively for each sub-system—and, in turn, each sub-system of the sub-systems. All discoveries from a higher level should be passed to the next level as an input. Expect to encounter quite a few levels of sub-systems, depending upon the complexity of the system.

In addition to the base PASTA stages, the following activities should be added to address the full spectrum of threats.

Stage 1. Define Objectives

Additional documents:

- safety standards and guidelines from related industries
- data security requirement document
- logistic documents
- identify critical functions and assets

Stage 2. Define Technical Scope

Additional activity:

- identify system critical dependencies from supply chain including dependencies from trusted third-party systems
- identify system critical dependencies from external infrastructure (sources of power and other resources, protection from physical damage and destruction)

Stage 3. Application Decomposition

Additional activity:

- identify physical boundaries (direct and indirect access) to the system's components
- implement corresponding supply chain techniques

Stage 4. Threat Analysis

Additional documents:

- supply chain threat related documents

- physical safety and security related documents

Additional activities:

- build Fault Trees and/or FMEA for hardware [39]
- apply supply chain analysis
- apply internal threat identification methods
- perform step 2 from STRIDE method for cyber threat finding
- perform step 2 and 3 from LINDDUN method to identify data privacy and security threats

Stage 5. Vulnerability and Weakness Analysis

Additional activities:

- analyze vulnerabilities in hardware
- analyze vulnerabilities in supply chain including trusted third-party systems
- analyze vulnerabilities in physical protection of assets

Stage 6. Attack Modeling

Additional activities:

- generate attack ports [3]

Stage 7. Risk and Impact Analysis

Additional activities:

- use mitigation strategies from step 5 of the LINDDON method for data privacy and security threats
- use mitigation strategies from STRIDE method
- calculate risk propagation [3]

The following are a few of the “best practices” that will help with the process of adopting a TMM [14]:

- It is important to recognize that threat modeling works best if applied in early stages of the project—i.e., the requirements and design phase.
- Threat modelling is an ongoing process. It is hard to perfect it on the first run and you cannot refine it indefinitely. You need milestones along the way. It does not stop after your system is delivered. Some steps must be repeated when the system changes.
- In threat modeling, it is dangerous to concentrate exclusively on threats. Modeling users and attackers, and controlling impact on requirements and mitigations are just as important.

- Threat modeling is not an innate skill. It is learnable and improves with practice. With each iteration, it become better and deeper.

By combining components of PASTA, STRIDE, and LINDDUN with tactics that address additional aspects of CPSoS, we believe this combination will provide better coverage of threats than any one model by itself. Adoption of the proposed framework will be laborious and time-consuming process, but will allow for creation of flexible and comprehensive structure for modeling of wide range of threats.

Bibliography

URLs are valid as of the publication date of this document.

[1] David, N.; David, A.; Hansen, R. R.; Larsen, K. G.; Legay, A.; Olesen, M. C.; & Probst, C. W. Modelling Social-Technical Attacks with Timed Automata. Pages 21-28. In Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats. Conference on Computer and Communications Security. October 2015. DOI 10.1145/2808783.2808787.

[2] Cheung, C. Y. Threat Modeling Techniques [Master's Thesis]. Delft University of Technology. November 2016. http://www.safety-and-security.nl/uploads/cfsas/attachments/SPM5440%20%26%20WM0804TU%20-%20Threat%20modeling%20techniques%20-%20CY%20Cheung.pdf

[3] Potteiger, B.; Martins, G.; & Koutsoukos, X. Software and attack centric integrated threat modeling for quantitative risk assessment. Pages 99-108. In *Proceedings of the Symposium and Bootcamp on the Science of Security*. April 2016. DOI 10.1145/2898375.2898390.

[4] Agrawal, A.; Ahmed, C. M.; & Chang, E. Poster: Physics-Based Attack Detection for an Insider Threat Model in a Cyber-Physical System. Pages 821-823. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. June 2018. DOI 10.1145/3196494.3201587.

[5] Datta, A. & Rahman, M. A. Cyber Threat Analysis Framework for the Wind Energy Based Power System. Pages 81-92. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy.* Conference on Computer and Communications Security. November 2017. DOI 10.1145/3140241.3140247.

[6] Humayed, A. & Luo, B. Cyber-physical security for smart cars: taxonomy of vulnerabilities, threats, and attacks. Pages 252-253. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems.* April 2015. DOI 10.1145/2735960.2735992.

[7] Hasan, K.; Shetty, S.; Sokolowski, J.; & Tosh, D. K. Security Game for Cyber Physical Systems. Article 12. In *Proceedings of the Communications and Networking Symposium*. April 2018. https://dl.acm.org/citation.cfm?id=3213212

[8] Allodi, L. & Etalle, S. Towards Realistic Threat Modeling: Attack Commodification, Irrelevant Vulnerabilities, and Unrealistic Assumptions. Pages 23-26. In *Proceedings of the 2017 Workshop on Automated Decision Making for Active Cyber Defense*. Conference on Computer and Communications Security. November 2017. DOI 10.1145/3140368.3140372.

[9] Kreimel, P.; Eigner, O.; & Tavolato, P. Anomaly-Based Detection and Classification of Attacks in Cyber-Physical Systems. Article 40. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*. September 2017. DOI 10.1145/3098954.3103155.

[10] Agadakos, I.; Chen, C.; Campanelli, M.; Anantharaman, P.; Hasan, M.; Copos, B.; Lepoint, T.; Locasto, M.; Ciocarlie, G. F.; & Lindqvist, U. Jumping the Air Gap: Modeling Cyber-Physical Attack Paths in the Internet-of-Things. Pages 37-48. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy*. Conference on Computer and Communications Security. November 2017. DOI 10.1145/3140241.3140252.

[11] Ding, J.; Atif, Y.; Andler, S. F.; Lindstrom, B.; & Jesufeld, M. CPS-based Threat Modeling for Critical Infrastructure Protection. *ACM SIGMETRICS Performance Evaluation Review*. Volume 45. Issue 2. September 2017. Pages 129-132. DOI 10.1145/3152042.3152080

[12] Wuyts, K.; Van Landuyt, D.; Hovsepyan, A.; & Joosen, W. Effective and efficient privacy threat modeling through domain refinements. Pages 1175-1178. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. April 2018. DOI 10.1145/3167132.3167414.

[13] Sion, L.; Yskout, K.; Van Landuyt, D.; & Joosen, W. Solution-aware data flow diagrams for security threat modeling. Pages 1425-1432. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. April 2018. DOI 10.1145/3167132.3167285.

[14] Shostack, A. *Threat Modeling: Designing for Security*. Wiley, 2014. ISBN 978-1118809990.

[15] Mead, N.; Shull, F.; Vemuru, K.; & Villadsen, O. *A Hybrid Threat Modeling Method.* CMU/SEI-2018-TN-002. Software Engineering Institute, Carnegie Mellon University. 2018. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=516617

[16] Shull, F. *Evaluation of Threat Modeling Methodologies*. Software Engineering Institute, Carnegie Mellon University. 2016. http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=474197

[17] Schneier, B. Attack Trees. *Dr. Dobb's Journal.* July 22, 2001. http://web.cs.du.edu/~ramki/papers/attackGraphs/SchneierAttackTrees.pdf

[18] Shostack, A. Threat Modeling Thursdays. July 23, 2018 [accessed]. https://adam.shostack.org/blog/category/threat-modeling/threat-model-thursdays/

[19] UcedaVélez, T. *Threat Modeling w/PASTA: Risk Centric Threat Modeling Case Studies*. Technical Report. Open Web Application Security Project (OWASP). 2017.

[20] Karahasanovic, A.; Kleberger, P.; & Almgren, M. Adapting Threat Modeling Methods for the Automotive Industry. In *Proceedings of the 15th ESCAR Conference.* 2017. https://research.chalmers.se/en/publication/502671

[21] Simeonova, S. Threat Modeling in the Enterprise, Part 2: Understanding the Process. *Security Intelligence.* August 15, 2016. https://securityintelligence.com/threat-modeling-in-the-enterprise-part-2-understanding-the-process/

[22] Beyst, B. Which Threat Modeling Method. *ThreatModeler.* April 15, 2016. https://threatmodeler.com/2016/04/15/threat-modeling-method/

[23] Cleland-Huang, J. How Well Do You Know Your Personae Non Gratae? *IEEE Software*. Volume 31. Number 4. July 2014. Pages 28–31. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6834694

[24] Mead, N. & Shull, F. The Hybrid Threat Modeling Method [blog post]. *SEI Blog.* April 23, 2018. https://insights.sei.cmu.edu/sei_blog/2018/04/the-hybrid-threat-modeling-method.html

[25] Lawson, C. & Pratap, K. Market Guide for Security Threat Intelligence Products and Services. 2017. https://www.gartner.com/doc/3765965/market-guide-security-threat-intelligence

[26] Hernan, S.; Lambert, S.; Shostack, A.; & Ostwald, T. Uncover Security Design Flaws Using the STRIDE Approach. *MSDN Magazine.* November 2006.

[27] Howard, M. & Lipner, S. *The Security Development Lifecycle.* Microsoft Press. 2006.

[28] Scandariato, R.; Wuyts, K.; & Joosen, W. A descriptive study of Microsoft's threat modeling technique. *Requirements Engineering*. Volume 20. Number 2. June 2015. Pages 163–180. DOI 10.1007/s00766-013-0195-2

[29] Microsoft Corporation. SDL Threat Modeling Tool. *Security Development Lifecycle*. July 20, 2018 [accessed]. https://www.microsoft.com/en-us/sdl/adopt/threatmodeling.aspx

[30] Martins, G.; Bhatia, S.; Koutsoukos, X.; Stouffer, K.; Tang, C.; & Candell, R. Towards a systematic threat modeling approach for cyber-physical systems. Pages 1-6. In *Proceedings of the 2015 Resilience Week*. August 2015. DOI 10.1109/RWEEK.2015.7287428.

[31] UcedaVélez, T. *Real World Threat Modeling Using the PASTA Methodology*. Technical report. Open Web Application Security Project (OWASP). 2012. https://www.owasp.org/images/a/aa/AppSecEU2012_PASTA.pdf

[32] UcedaVélez, T. & Morana, M. M. Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis. Wiley. 2015. ISBN 978-0-470-50096-5.

[33] Leblanc, D. DREADful [blog post]. *David LeBlanc's Web Log*. August 14, 2007. https://blogs.msdn.microsoft.com/david_leblanc/2007/08/14/dreadful/

[34] Downloads. *LINDDUN: Privacy Threat Modeling.* July 23, 2018 [accessed]. https://distrinet.cs.kuleuven.be/software/linddun/download.php#

[35] Deng, M.; Wuyts, K.; Scandariato, R.; Preneel, B.; & Joosen, W. A Privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering.* Volume 16. Issue 1. March 2011. Pages 3-32. https://link.springer.com/article/10.1007/s00766-010-0115-7

[36] Alberts, C.; Dorofee, A.; Stevens, J; & Woody, C. *Introduction to the OCTAVE Approach*. Software Engineering Institute, Carnegie Mellon University. August 2003. https://resources.sei.cmu.edu/library/Asset-view.cfm?assetid=51546

[37] Common Vulnerability Scoring System v3.0: Specification Document. *Forum of Incident Response and Security Teams.* July 23, 2018 [accessed]. https://www.first.org/cvss/specification-document

[38] National Vulnerability Database. *National Institute of Standards and Technology*. July 23, 2018 [accessed]. https://nvd.nist.gov/vuln-metrics/cvss#

[39] Hanic, D. & Surkovic, A. *An Attack Model of Autonomous Systems of Systems* [Master's Thesis]. Mälardalen University. 2018. http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1218262&dswid=-7458

[40] Khan, R.; McLaughlin, K.; Laverty, D.; & Sezer, Sakir. STRIDE-based Threat Modeling for Cyber-Physical Systems. In *Proceedings of the 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe.* September 2017. DOI 10.1109/ISGTEurope.2017.8260283.

[41] Denning, T. A.; Friedman, B.; & Kohno, T. Home. *Security Cards: A security threat brainstorming toolkit*. 2013. http://securitycards.cs.washington.edu/index.html

[42] Saitta, P.; Larcom, B.; & Eddington M. Trike v.1 Methodology Document [Draft]. *OctoTrike.* July 13, 2005. http://www.octotrike.org/papers/Trike_v1_Methodology_Document-draft.pdf

[43] Stanganelli, J. Selecting a Threat Risk Model for Your Organization, Part Two. *eSecurity Planet.* September 27, 2016. https://www.esecurityplanet.com/network-security/selecting-a-threat-risk-model-for-your-organization-part-two.html

[44] Common Vulnerability Scoring System SIG. *Forum of Incident Response and Security Teams*. July 30, 2018 [accessed]. https://www.first.org/cvss/

[45] Mead, N.; Hough, E.; & Stehney, T., II. *Security Quality Requirements Engineering Technical Report*. CMU/SEI-2005-TR-009. Software Engineering Institute, Carnegie Mellon University. 2005. https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=7657

[46] Maier, M.W. Architecting Principles for System of Systems. CH1-460, The Aerospace Corporation. November 3, 1998. John Wiley & Sons, Inc. CCC 1098-1241/98/040267-18

[47] Shevchenko, N.; Chick, T.A.; O'Raige. P.; Scanlon, T.P.; & Woody, C. *Threat Modeling: A Summary of Available Methods*. Software Engineering Institute, Carnegie Mellon University. August 2018. https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=524448

[48] *UPPAAL Website*, Department of Information Technology at Uppsala University in Sweden, Department of Computer Science at Aalborg University in Denmark. August 30, 2018 [accessed] http://www.uppaal.org/

[49] Joint Task Force Transformation Initiative. *Guide for Conducting Risk Assessments* (NIST Special Publication 800-30 Rev. 1). https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final (2012).

[50] Stouffer, K.; Lightman, S.; Pillitteri, V.; Abrams, M.; & Hahn, A. *Guide to Industrial Control Systems (ICS) Security* (NIST Special Publication 800-82 Rev. 2). https://csrc.nist.gov/publications/detail/sp/800-82/rev-2/final (2015).

[51] Joint Task Force Transformation Initiative. *Security and Privacy Controls for Federal Information Systems and Organizations* (NIST Special Publication 800-53 Rev. 4). https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final (2015).

[52] Common Vulnerability Scoring System Calculator, Version 3. *NIST Website.* August 30, 2018 [accessed]. https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator

[53] *Threat Modeler Website.* August 30, 2018 [accessed]. https://threatmodeler.com/

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479
Web: www.sei.cmu.edu
Email: info@sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM18-1096