Comprehensive Review of Threat Analysis in Software Systems Using Stride Methodology

Swagato Chatterjee , Prajwal Ray, Sanskar Tyagi, Amarinder Kaur
Lovely Professional University

***Abstract* —** In the domain of cybersecurity, one of the most important facets is recognizing and diminishing possible hazards in order to secure information systems. The STRIDE methodology is a widely accepted approach to threat analysis that divides threats into six categories: denial of service, spoofing, tampering, repudiation, information disclosure, and privilege escalation. This review article offers a detailed study of the performance of the STRIDE approach as it is applied in the field of cloud computing, network security, software development and other areas. To enlarge and enhance the scope and precision of risk analysis, the appraisal also checks how the STRIDE technique cooperates with other threat modelling approaches such as attack tree and misuse instance. It also deals with the problems and limitations associated with the implementation of STRIDE, namely the lack of scalability and necessity of customised approaches in specific areas.

***Index Terms* —** Threat Analysis, Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of privilege, Cyber Security.

1. Introduction

In most cases, the software is the code that develops the interface to the potential breaches of computer and information security systems. Information systems security is limited in formalizing practices like depending heavily on the "Penetrate-and-Patch" approach. Developing an independent and formal method for defining security needs and secure systems design is currently impossible, which is the most critical obstacle. Threat analysis, threat modelling, and threat assessment have a common phrase when we discuss related to possible threats. At first sight, threat modelling appears to be mostly connected with computer security, but threat analysis is the concept which is used in different kinds of security areas, such as computer security and physical security. Security in modern cyber-physical SoSs grow more and more significant, and it demands new solutions. This article will take threat analysis, which is usually performed as a step in the risk assessment process to detect and evaluate threats, size up their effects and propose precautionary measures before system implementation. The security approaches which are time tested and are being supported by governmental authorities or standardization groups normally involve the examination of the systems that are expected to be static i.e. not to change much and, therefore, functional as well as non-functional requirements do not alter substantially. Thus, system properties like security prevail for all the lifetime round, but gaining them is made possible by risk identification, risk management, and installation of security measures that are not frequently changed. We are reviewing one of the methodologies of threat analysis, namely STRIDE, in our paper. Through the use of such approaches to the analysis of potential threats, frameworks and methodologies give a way to identify, assess and eliminate potential avenues to malicious acts in a disciplined manner. This includes proper development, testing, and documentation of the security policies and procedures.

2. Literature review

A. STRIDE

The STRIDE-based threat model distinguishes threats into different groups, which comprise of Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, and Elevation of privilege. Each of these categories comprises a particular attack vector carrying a different level of security threat. Threats and vulnerabilities are scrutinized in order to evaluate how they directly affect owned assets while implementing security rules including non-repudiation, secure lifecycle, confidentiality, integrity, and availability. In the process of assigning risk scores, it is essential to consider the severity of each threat.

A separation of the seven-step process designated to run STRIDE threat modelling on Distributed Control System (DCS) is opined. Firstly, the asset and the security goals with their heights are defined. To begin with, the purpose of the system, who the users and what they are provided with, the data to be utilized together with the relationships among all the elements making the system explicated. The following operation is to list on the Data Flow Diagram (DFD), which will show the system components, their function and relationship. Outside, Process, Data Flow, Data Storage, and Trust Barriers supposes the most of them is presented with DFD Symbols. The following part is multiple attack methods to impose on STRIDE. In the next on the agenda is (STRIDE) which detects threats using functionalities and components of the system. Finally, recorded accidents are classified as fatal or non-fatal incidents depending on the extent of injury. Lastly, such strange action is achieved only when the risks are appraised with the help of the model introduced and called DREAD method. At last, risk assessment outcomes are applied to steer mending actions.

After one learns the features of each type of threat within the STRIDE model, it is apparent that the Aristotelian or binary logic methods are insufficient for analysis. A critical approach that can cope with the complexities of the problem and be specific to the situation is necessary at this time. The fuzzy logic problem can be solved if the blend of human knowledge into technical decision-making process procedures is enabled. Through Fuzzy Logic, sharp inputs like the number of threats are taken and are given out as the intensity of the attack which is outputted.

In order to address the points mentioned, a STRIDE model was published by Microsoft. The model set is ready to combat different kinds of attacks on the network. The word 'STRIDE' represents the acronym which was derived from the six different types of threat categories.

1. Spoofing

- Email spoofing: it is a type of fraud which occurs with the presentation of emails which look like to be from a trusted source in order to trick the recipients for sharing their personal information or carrying out malicious activities.
- DNS spoofing: it is the use of spoofing DNS (Domain Name System) responses and route users to malicious sites or servers, which may bring vulnerabilities to malware downloads or phishing attacks.
- Caller ID Spoofing: By hiding their phone number behind the real caller’s name, attackers convince the intended victim to provide credentials or send money by spoofing the Caller ID information.

2. Tampering

- Data Modification: During data transmission within the transaction, attackers may get the data packets modified, and it may end up with transferring money to another bank account.
- Code Injection: By the means of code injection through web applications, hackers find out security holes to tamper with the databases, steal data or take over user accounts.
- Firmware Modification: The hackers replace the firmware (routers and IoT devices) to add backdoors and change the behaviour of those devices.

3. Repudiation

- Transaction Repudiation: May affect the partnership and lead to damages when a user declines to confess their involvement in the financial business beyond a reasonable doubt.
- Digital Signature Forgery: Such deception involves the injection of false digital signatures that can alter and mislead the content of electronic documents and transactions, thus endangering their authenticity and security of information.
- Log manipulation: This means stealthily deleting or altering log entries so that the path of the misconduct becomes undetectable and, hence, hard to track down.

4. Information Disclosure

- Data Breach: Slang when hackers utilize their illegal actions to enter databases holding personal information like login passwords and privacy data they are exposed.

- Social engineering: With regard to such processes, cyber criminals can exploit a victim's mindset and get them involved in activities that will lead them to revealing confidential details, including security codes or passwords.
- Network Sniffing: Often, a causation approach is used. During it, hackers record network traffic, viewing, and intercepting it, which allows them to get valuable packets of information (for instance, passwords or money transactions).

5. Denial of Service (DoS)

- Distributed Denial of Service (DDoS): In turn, hackers create DDoS using botnets so that these attackers flood the websites and networks of the targeted server with excessive traffic, which renders those services inaccessible to legitimate users.
- Resource Exhaustion: This describes the state in which the system performance can get deteriorated or even fail if an attacker is able to take advantage of vulnerabilities to consume everything like CPU, memory, or bandwidth.
- Application Layer Attacks: These are sending evil instructions that are targeted at specific applications or services to overwhelm them and disrupt their services is one of the tactics used.

6. Elevation of Privilege

- Privilege escalation: This happens when an attacker identifies the cracks in the program to have his/her rights enhanced, e.g., to be promoted to admin access, which enables them to go past security controls and conduct unauthorized operations.
- Credential Theft: This is the problem which results in breach of secure user-validated data by hackers in order to access accounts or system having secured rights. NCAs are the credentials where username and password are examples.

Zero-Day Exploits: Such activities refer to the employment of the unreported vulnerabilities (zero-day) in programs or systems and therefore exploiting them to obtain privileged access before patches or updates are available that denies a chance to launch sophisticated attacks easily.

3. Listing Possible Threats in a Software System using STRIDE:

Description	Type	Level
Phishing	Spoofing	High
Altering network traffic after intercepting it	Tampering	High
Gaining access of a system and deleting footprints	Repudiation	Medium
Server not configured properly	Information Disclosure	High
Flooding the network traffic with requests	Denial of Service (DoS)	High
Gaining administrative rights without proper permission	Elevation of Privilege	High

4. Conclusion:

In the cybersecurity industry, ML has been successful in redefining malware detection with more speed, adaptability and efficiency which makes it necessary to tackle the rampant threats. This work has rigorously analysed ML methods, algorithms, and possible future developments, through which a complete knowledge of the essential features of this field can be achieved. By leveraging the ability of ML and confronting the sophisticated nature of its challenges, the path for a more secure digital environment is cleared.

The interaction among security analysts, ML specialists and cybersecurity practitioners is the most fundamental factor for strengthening the defences against the complex malware. The way to create one's robust cybersecurity mechanisms involves recurrent cycle of learning, innovation and implementation. The integration of the ML-driven analysis and real-time threat intelligence gives an opportunity to anticipate and neutralize emerging threats prior to their occurrence.

In the near future the convergence of ML with other advanced technologies e.g. Blockchain, Edge computing and Quantum computing is bound to create a strong base for cyber resilience. The adoption of multi-disciplinary collaborations and cultivating a culture of inventiveness will hence play a critical role in setting up a dynamic and resilient cybersecurity framework.

"In the era of fast-growing digital ecosystems, we remain committed to using the full potential of ML technology and constantly improving it in order to secure digital data and personal information. Through being alert, proactive, and working together, we shall create an environment where cyber security becomes the bedrock of our digital era."

References:

[1] Abuabed, Z., Alsadeh, A., & Taweel, A. (2023). STRIDE threat model-based framework for assessing the vulnerabilities of modern vehicles. Computers & Security, 133, 103391. https://doi.org/10.1016/J.COSE.2023.103391

[2] Association for Computing Machinery., & SIGAPP. (2012). Proceedings of the 27th annual ACM symposium on applied computing 2012 : Symposium on Applied Computing : Riva del Garda, Trento, Italy, March 26-30, 2012. ACM Press.

[3] Ceccarelli, A., Zoppi, T., Vasenev, A., Mori, M., Ionita, D., Montoya, L., & Bondavalli, A. (2018). Threat analysis in systems-of-systems: An emergence-oriented approach. ACM Transactions on Cyber-Physical Systems, 3(2). https://doi.org/10.1145/3234513

[4] IEEE Power & Energy Society, & Institute of Electrical and Electronics Engineers. (n.d.). 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe) : conference proceedings : Torino, Italy, 26-29 September 2017.

[5] Kaneko, T., Sasaki, R., & Takahashi, Y. (2019). Threat analysis using STRIDE with STAMP/STPA.

[6] Kim, K. H., Kim, K., & Kim, H. K. (2022). STRIDE-based threat modeling and DREAD evaluation for the distributed control system in the oil refinery. ETRI Journal, 44(6), 991–1003. https://doi.org/10.4218/etrij.2021-0181

[7] Rouland, Q., Hamid, B., & Jaskolka, J. (2021). Specification, detection, and treatment of STRIDE threats for software components: Modeling, formal methods, and tool support. Journal of Systems Architecture, 117. https://doi.org/10.1016/j.sysarc.2021.102073

[8] Tuma, K. (n.d.). Thesis for The Degree of Doctor of Philosophy Efficiency and Automation in Threat Analysis of Software Systems.

[9] Tuma, K., Calikli, G., & Scandariato, R. (n.d.). Threat Analysis of Software Systems: A Systematic Literature Review. https://www.bsimm.com

[10] Whitmore, J., Türpe, S., Triller, S., Poller, A., & Carlson, C. (2014). Threat analysis in the software development lifecycle. In IBM Journal of Research and Development (Vol. 58, Issue 1). IBM Corporation. https://doi.org/10.1147/JRD.2013.2288060

[11] Widjajarto, A., Lubis, M., & Ayuningtyas, V. (2021). Vulnerability and risk assessment for operating system (OS) with framework STRIDE: Comparison between VulnOS and Vulnix. In Indonesian Journal of Electrical Engineering and Computer Science (Vol. 23, Issue 3, pp. 1643–1653). Institute of Advanced Engineering and Science. https://doi.org/10.11591/ijeecs.v23.i3.pp1643-1653

[12] Xu, D., & Pauli, J. J. (n.d.). Threat-Driven Design and Analysis of Secure Software Architectures.