Proposal for the Use of Quality Characteristics in Security Design Methodologies

Daiju Kato[1] , Daisuke Iwasaki [1]

[1] *Nihon Knowledge Co,, Ltd., JS Building 9F 3-9-15, Kotobuki, Taito-ku, Tokyo, 111-0042, Japan*

Abstract

Security implementation is essential to provide safe and secure products. Security development life cycle must be considered from the early stages of development through requirement to evaluation with traceability. By using SQuaRE to classify and evaluate security requirements, it is possible to assess the adequacy of product development to meet the requirements of safety and security. This paper proposes a method for implementing the requirements using SQuaRE in SDL.

Keywords

SDL, threats analysis, secure by design, risk management, vulnerability, SQuaRE

1. Introduction

With new threats appearing every day, the responsibility for application security is only increasing.

According to the Ministry of Internal Affairs and Communications' 2021 White Paper on Information and Communications, the spread of the new coronavirus infection has led to rapid and forceful digitization of society which has been advancing in areas where it was not, telework and online classes. As the use of digital technology increases, the number of devices and applications connected to the Internet is increasing, and their system configurations and usage patterns are diversifying.

With the demand for safe and secure applications, developers are faced with the challenge of how to incorporate security measures into the software development lifecycle. However, security vulnerabilities revealed in the testing process may increase the number of man-hours to deal with them and may even delay the release of the software. Therefore, it is considered necessary to improve the upstream process.

Security implementation requires developers to have appropriate skills to do even when they follow the SDL (Security Development Lifecycle). To solve this problem, we believe that the use of tools would provide a way that is less dependent on the skills of individual developers and eliminate the tendency toward personalization. ISO/IEC27034-1 [1] explains SDL, other processes and techniques for building security into product development.

This paper proposes a method of security design using a tool that supports security threat analysis with quality characteristics, realizing a detailed design that includes security measures, and introducing security activities into coding and testing processes in order to reduce return work due to the discovery of vulnerabilities in the late development stage.

2. SECURITY AND RISK MANAGEMENT

The Japanese government's promotion of DX (Digital Transformation) has increased the dependence of businesses on information technology. The number and areas of information

4th International Workshop on Experience with SQuaRE Series and its Future Direction, December 06, 2022, Tokyo, Japan

EMAIL: d-kato@know-net.co.jp (A. 1); iwasaki@know-net.co.jp (A. 2)
ORCID: 0000-0001-9904-8554 (A. 1); none (A. 2)

assets to be protected continue to increase due to the dispersion of information assets to cloud services and the dispersion of locations due to the spread of teleworking, etc. Information system-related accidents are becoming a risk that would threaten even the survival of businesses. For example, business operations may be suspended due to a virus such as ransomware, sales may decrease due to service suspension until a vulnerability is discovered and countermeasures are taken, or social trust may be lost in the case of a data breach.

The requirement of security demanded by users have also changed. For example, 20 years ago, HTTPS encryption was used only for pages that handled personal information on websites, but now it is a function that must be implemented. Security has become a "must be quality" in e-mail and EDI (Electronic Data Interchange) communications as well.

Ensuring security is also required for safety because safety design is to protect human life and property from being threatened using a product or device.

This security has such a great impact on human lives, property, and business continuity. Therefore, it is a social responsibility of companies to properly manage information and prevent its leakage and loss in their business activities.

3. THREAT PREVENTION IN SDL

However, the difficulty of security development is that it requires appropriate skills to do and the cost for expensive countermeasures.

For example, pre-release vulnerability checks using security inspection tools can prevent a product from being released with hidden vulnerabilities. However, if many vulnerabilities are detected in the testing process just before release, a large amount of time will be spent to deal with the vulnerability. This will lead to delivery delay and cost overrun. Even if there is no vulnerability at the time of release, there is a risk that it may become a vulnerability from embedded OSS modules due to the emergence of new unknown attack methods.

To reduce such risks, security measures should be implemented prior to testing and should have process of continuous management of vulnerabilities. By implementing security measures at all stages of SDL, we can prevent vulnerabilities from being discovered just before the release and threats after the release.

Security by Design [2] is defined by the Cabinet Cyber Security Center (NISC) in Japan as "a measure to ensure information security from the planning and design stages", which is a concept to ensure cyber security by incorporating security measures at the planning and design stages, rather than after system installation and operation. SDL is one of the ways to realize this concept. By implementing security measures from the upstream process, it is expected to improve the security and reduce the cost of security measures.

SDL implementations security related activities or practices to meet into V-model or agile process. In the case of the agile process, security requirements are also managed in the backlog, and secure-by-design is achieved through security practices such as secure-by-design, static analysis, and vulnerability testing within a sprint. These tasks are generally automated as a pipeline.

Threat analysis is performed for security requirements in beginning of SDL. Threat Modeling Tool [3] provides support functions for a threat analysis, and MITRE ATT&CK [4] provides advice on countermeasures against classified threats. The Thread Modeling Tool applies a threat framework to a data flow diagram (DFD) to find potential security problems and analyze threats to systems and software to be built. The tool classifies threats using the STRIDE model, shown as Table-1.

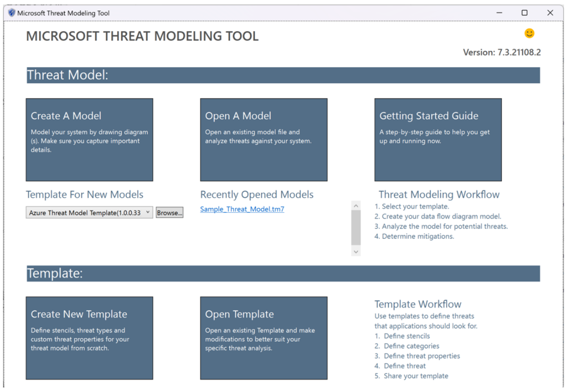
Table-1: STREIDE MODEL

Spoofing
Tampering
Repudiation
Information Disclosure
Denial of Service
Elevation of Privilege

STRIDE is a threat analysis model proposed by Microsoft that can classify various types of threats [5].

Since DFD diagrams can be created after the basic design, the Threat Modeling Tool is suitable for threat analysis at the detailed design stage ().

MITRE ATT&CK is used to improve the accuracy of threat identification by conducting threat analysis using a different approach from the Threat Modeling Tool. MITRE ATT&CK provides a framework that systematically organizes knowledge about attacks to defend against and dealing with attacks. It has a large amount of information on actual examples, mitigation measures, detection methods, and reports from security vendors and white-hat hackers for each tactic's individual attack techniques and methods. It is highly regarded attracting attention by security practitioners.



Microsoft Threat Modeling Tool

MICROSOFT THREAT MODELING TOOL

Version: 7.3.21108.2

Threat Model:

Create A Model

Model your system by drawing diagram (s). Make sure you capture important details.

Open A Model

Open an existing model file and analyze threats against your system.

Getting Started Guide

A step-by-step guide to help you get up and running now.

Template For New Models

Azure Threat Model Template(1.0.0.33 | Browse...

Recently Opened Models

Sample_Threat_Model.tm7

Threat Modeling Workflow

1. Select your template.
2. Create your data flow diagram model.
3. Analyze the model for potential threats.
4. Determine mitigations.

Template:

Create New Template

Define stencils, threat types and custom threat properties for your threat model from scratch.

Open Template

Open an existing Template and make modifications to better suit your specific threat analysis.

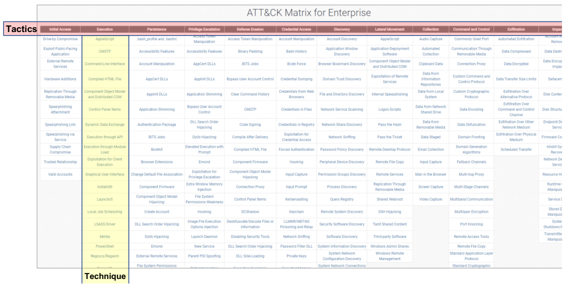
Template Workflow

Use templates to define threats that applications should look for.

1. Define stencils
2. Define categories
3. Define threat properties
4. Define threat
5. Share your template

Figure-1: Thread Modeling Tool

MITRE ATT&CK provides MITRE ATT&CK Navigator [6] as a tool to explore and visualize a vast number of attack methods. We selected MITRE ATT&CK Navigator, shown as Figure-2, because of its intuitive operation and ease of use as a tool.



ATT&CK Matrix for Enterprise

Tactics

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impa
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Account A Remov
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Destr
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Component Object Model and Distributed COM	Clipboard Data	Connection Proxy	Data Encrypted	Data Encryp Impac
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Defacem
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Clear Command History	Credentials from Web Browsers	File and Directory Discovery	Internal Spearphishing	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Conter
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Disk Structu
Spearphishing Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	Code Signing	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Endpoint De Servic
Spearphishing via Service	Execution through API	BITS Jobs	Dylib Hijacking	Compile After Delivery	Exploitation for Credential Access	Network Sniffing	Pass the Ticket	Data Staged	Domain Fronting	Exfiltration Over Physical Medium	Firmware Co
Supply Chain Compromise	Execution through Module Load	Bootkit	Elevated Execution with Prompt	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Inhibit Sy Recove
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Emond	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Input Capture	Fallback Channels		Network De Servic
Valid Accounts	Graphical User Interface	Change Default File Association	Exploitation for Privilege Escalation	Component Object Model Hijacking	Input Capture	Permission Groups Discovery	Remote Services	Man in the Browser	Multi-hop Proxy		Resource Hi
	InstallUtil	Component Firmware	Extra Window Memory Injection	Connection Proxy	Input Prompt	Process Discovery	Replication Through Removable Media	Screen Capture	Multi-Stage Channels		Runtime Manipula
	Launchctl	Component Object Model Hijacking	File System Permissions Weakness	Control Panel Items	Kerberoasting	Query Registry	Shared Webroot	Video Capture	Multiband Communication		Service S
	Local Job Scheduling	Create Account	Hooking	DCShadow	Keychain	Remote System Discovery	SSH Hijacking		Multilayer Encryption		Stored D Manipula
	LSASS Driver	DLL Search Order Hijacking	Image File Execution Options Injection	Deobfuscate/Decode Files or Information	LLMNR/NBT-NS Poisoning and Relay	Security Software Discovery	Taint Shared Content		Port Knocking		Syster Shutdown/
	Mshta	Dylib Hijacking	Launch Daemon	Disabling Security Tools	Network Sniffing	Software Discovery	Third-party Software		Remote Access Tools		Transmitte Manipula
	PowerShell	Emond	New Service	DLL Search Order Hijacking	Password Filter DLL	System Information Discovery	Windows Admin Shares		Remote File Copy		
	Regsvcs/Regasm	External Remote Services	Parent PID Spoofing	DLL Side-Loading	Private Keys	System Network Configuration Discovery	Windows Remote Management		Standard Application Layer Protocol		
	[illegible]	File System Permissions	[illegible]	[illegible]	[illegible]	System Network Connections			Standard Cryptographic		

Technique

Figure-2: MITRE ATT&CK Matrix

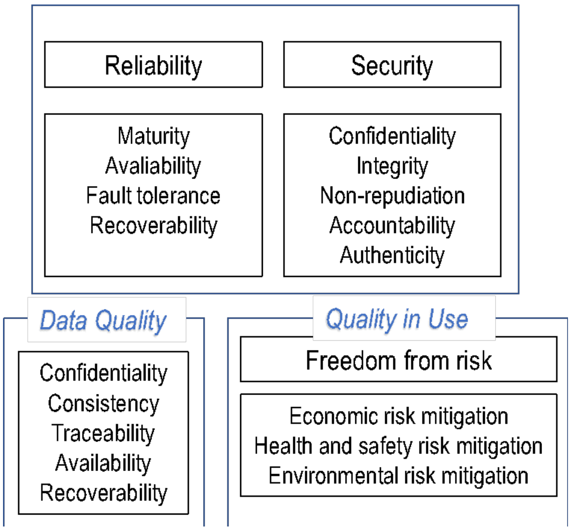
MITRE ATT&CK provides a matrix to show the specific technical elements required for an attack which consist of Tactics and Technique.

ATT&CK has selected the following 12 tactics.

1. Initial Access
2. Execution
3. Persistence
4. Privilege Escalation
5. Defense Evasion
6. Credential Access
7. Discovery
8. Lateral Movement
9. Collection
10. C&C (Command and Control)
11. Exfiltration
12. Impact

The attackers use the techniques and methods in the initial access to conduct the attack, and when that tactic is achieved, the attacker moves on to the next tactic. The attacker proceeds to the last tactic of impact to achieve the final objective. MITRE ATT&CK Navigator is provided as a web application and is used via a web browser. Users can display only the methods of a particular platform, or highlight the methods used by a particular adversary, or search by keywords. The search function allows users to select from techniques, attacker groups, software, mitigations, etc., as well as to extract attack methods by keyword search. Analysis results can be defined as layers, and the importance of each layer can be assigned as a threat score. Multiple layers can be created and overlaid to visualize overlapping threats. Other features include color-coding of the matrix and the addition of comments. The attack methods in the matrix can be linked to a detailed threat page, where you can see the details of the attack and mitigation measures.

Figure-3 shows security and safety related quality characteristics from quality model in SQuaRE. Mapping requirements of security and safety into selected sub-characteristics are easily judged the requirement is validated or not.



Reliability

Maturity
Availability
Fault tolerance
Recoverability

Security

Confidentiality
Integrity
Non-repudiation
Accountability
Authenticity

Data Quality

Confidentiality
Consistency
Traceability
Availability
Recoverability

Quality in Use

Freedom from risk

Economic risk mitigation
Health and safety risk mitigation
Environmental risk mitigation

Figure-3: Quality characteristics related with security and safety

4. IMPLEMENTATION SUMMARY

We have mapped security requirements to security quality characteristics and applied SDL to a client-server web system. In order to reflect these quality requirements in the design, a threat analysis is conducted using Threat Modeling Tool and MITRE ATT&CK Navigator.

From the results of the analysis, “risk management checklist" is created by Kusaka et al. [7]. The risk management checklist is a list of threat risks to information assets and functions, their risk assessment, and mitigation measures. The risk management checklist can be used in the detailed design, coding, and testing processes, and is expected to be effective in preventing the creation of vulnerabilities. It can also be used for test.

Developers follow those steps to identify threat and find tactics.

1. Classified security requirements with quality characteristics
2. Creating a DFD drawing and threat analysis by Threat Modeling Tool
3. Listing of analysis results with MITRE ATT&CK Navigator
4. Risk assessment and risk reduction

The security design here is performed both in the architectural and the detailed design on V-model development. In the basic design, system configuration, servers, databases, and other elements are identified, and threat analysis is conducted on these elements, and the results of the analysis are passed to the next stage of detailed design. The detailed design based on the threat analysis is expected to reflect the security activities in the subsequent coding and testing processes. In case of agile process, those security activities execute for practices.

5. IMPLEMENTATION DETAILS

5.1. Classified security requirements with quality characteristics

Classify the security and safety requirements of a system or application by mapping them to the security quality sub-characteristics of the product quality model [8]. Also map the data to be used to the data quality characteristics at data quality model [9] and them consider which requirements need to be met.

5.2. Creating a DFD drawing and threat analysis by Threat Modeling Tool

Using the Threat Modeling Tool, create a diagram that represents the data flow in the system configuration based on the basic design and the elements such as clients, servers, databases, network devices, etc. Draw a Trust Boundary to clarify which elements belong to which boundary. A trust boundary is a line that separates areas with different levels of trust. It is a line of defense against threats that occur when the trustworthy and the untrustworthy cross the boundary.

Figure-4 shows a configuration with one client, one client storage, three servers, and three databases.

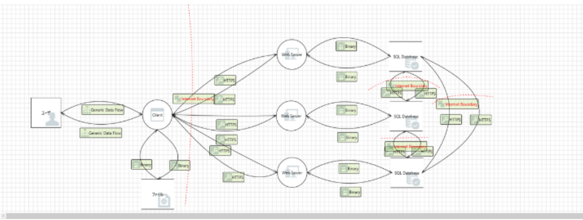
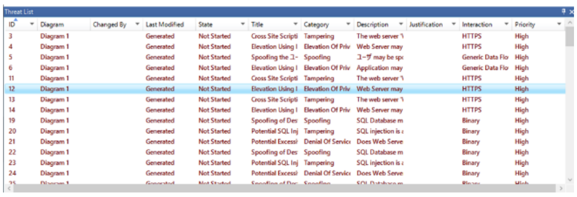


Figure-4: DFD diagram of the Treat Modeling Tool

After running the analysis from the created DFD diagram, a list of detected threats is shown for each element. The tool exacts results of threat analysis, shown as Figure-5.



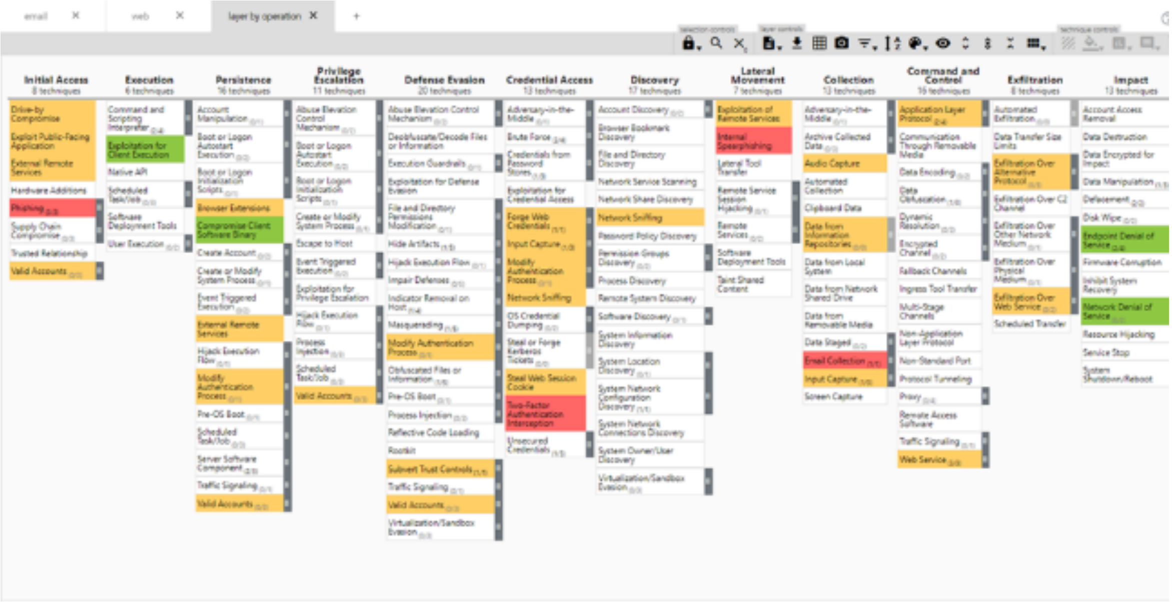
ID	Diagram	Changed By	Last Modified	State	Title	Category	Description	Justification	Interaction	Priority
3	Diagram 1		Generated	Not Started	Cross Site Scripti	Tampering	The web server '		HTTPS	High
4	Diagram 1		Generated	Not Started	Elevation Using I	Elevation Of Priv	Web Server may		HTTPS	High
5	Diagram 1		Generated	Not Started	Spoofing the ユ-	Spoofing	ユーザ may be spo		Generic Data Flo	High
6	Diagram 1		Generated	Not Started	Elevation Using I	Elevation Of Priv	Application may		Generic Data Flo	High
11	Diagram 1		Generated	Not Started	Cross Site Scripti	Tampering	The web server '		HTTPS	High
12	Diagram 1		Generated	Not Started	Elevation Using I	Elevation Of Priv	Web Server may		HTTPS	High
13	Diagram 1		Generated	Not Started	Cross Site Scripti	Tampering	The web server '		HTTPS	High
14	Diagram 1		Generated	Not Started	Elevation Using I	Elevation Of Priv	Web Server may		HTTPS	High
19	Diagram 1		Generated	Not Started	Spoofing of Des	Spoofing	SQL Database m		Binary	High
20	Diagram 1		Generated	Not Started	Potential SQL Inj	Tampering	SQL injection is a		Binary	High
21	Diagram 1		Generated	Not Started	Potential Excessi	Denial Of Servic	Does Web Serve		Binary	High
22	Diagram 1		Generated	Not Started	Spoofing of Des	Spoofing	SQL Database m		Binary	High
23	Diagram 1		Generated	Not Started	Potential SQL Inj	Tampering	SQL injection is a		Binary	High
24	Diagram 1		Generated	Not Started	Potential Excessi	Denial Of Servic	Does Web Serve		Binary	High
25	Diagram 1		Generated	Not Started	Spoofing of Des	Spoofing	SQL Database m		Binary	High

Figure-5: Extracted threat analysis results

In some cases, some of the threats are duplicated because the same threat is detected in each of the three redundant servers. This tool classifies threats by STRIDE and can be used for risk analysis and mitigation assessment. This is accomplished by considering risk mitigation measures for classified threats rather than for the elements themselves.

5.3. Listing of analysis results with MITRE ATT&CK Navigator

MITRE ATT&CK Navigator visualizes listed threats. A list of is shown as Figure-6. It supports creating two layers and check for overlapping attack methods. In case of "email" and "web", layer 1 shows a threat score of 30. Layer 2 is set to the threat score is set to 50. By superimposing the search results of layer 1 and layer 2 with these conditions, the merged results are displayed. Layer 1 is displayed in red, Layer 2 in yellow, and the overlapping items (threat score 30+50=80) in green. From these results, only threats to the system are identified. Also, it is possible to investigate what vulnerabilities have been discovered in products with similar architectures, and to identify possible threats by using CWE (Common Weakness Enumeration) to identify



Initial Access 8 techniques	Execution 6 techniques	Persistence 16 techniques	Privilege Escalation 11 techniques	Defense Evasion 20 techniques	Credential Access 13 techniques	Discovery 17 techniques	Lateral Movement 7 techniques	Collection 13 techniques	Command and Control 16 techniques	Exfiltration 8 techniques	Impact 13 techniques
Drive-by Compromise	Command and Scripting Interpreter	Account Manipulation	Abuse Elevation Control Mechanism	Abuse Elevation Control Mechanism	Adversary-in-the-Middle	Account Discovery	Exploitation of Remote Services	Adversary-in-the-Middle	Application Layer Protocol	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	Exploitation for Client Execution	Boot or Logon Autostart Execution	Boot or Logon Autostart Execution	Deobfuscate/Decode Files or Information	Brute Force	Browser Bookmark Discovery	Internal Spearphishing	Archive Collected Data	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
External Remote Services	Native API	Boot or Logon Initialization Scripts	Boot or Logon Initialization Scripts	Execution Guardrails	Credentials from Password Stores	File and Directory Discovery	Lateral Tool Transfer	Audio Capture	Data Encoding	Exfiltration Over Alternative Protocol	Data Encrypted for Impact
Hardware Additions	Scheduled Task/Job	Browser Extensions	Create or Modify System Process	Exploitation for Defense Evasion	Exploitation for Credential Access	Network Service Scanning	Remote Service Session Hijacking	Automated Collection	Data Obfuscation	Exfiltration Over C2 Channel	Data Manipulation
Phishing	Software Deployment Tools	Compromise Client Software Binary	Escape to Host	File and Directory Permissions Modification	Forge Web Credentials	Network Share Discovery	Remote Services	Clipboard Data	Dynamic Resolution	Exfiltration Over Other Network Medium	Defacement
Supply Chain Compromise	User Execution	Create Account	Event Triggered Execution	Hide Artifacts	Input Capture	Network Sniffing	Software Deployment Tools	Data from Information Repositories	Encrypted Channel	Exfiltration Over Physical Medium	Disk Wipe
Trusted Relationship		Create or Modify System Process	Exploitation for Privilege Escalation	Hijack Execution Flow	Modify Authentication Process	Password Policy Discovery	Taint Shared Content	Data from Local System	Fallback Channels	Exfiltration Over Web Service	Endpoint Denial of Service
Valid Accounts		Event Triggered Execution	Hijack Execution Flow	Impair Defenses	Network Sniffing	Permission Groups Discovery		Data from Network Shared Drive	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
		External Remote Services	Process Injection	Indicator Removal on Host	OS Credential Dumping	Process Discovery		Data from Removable Media	Multi-Stage Channels		Inhibit System Recovery
		Hijack Execution Flow	Scheduled Task/Job	Masquerading	Steal or Forge Kerberos Tickets	Remote System Discovery		Data Staged	Non-Application Layer Protocol		Network Denial of Service
		Modify Authentication Process	Valid Accounts	Modify Authentication Process	Steal Web Session Cookie	Software Discovery		Email Collection	Non-Standard Port		Resource Hijacking
		Pre-OS Boot		Obfuscated Files or Information	Two-Factor Authentication Interception	System Information Discovery		Input Capture	Protocol Tunneling		Service Stop
		Scheduled Task/Job		Pre-OS Boot	Unsecured Credentials	System Location Discovery		Screen Capture	Proxy		System Shutdown/Reboot
		Server Software Component		Process Injection		System Network Configuration Discovery			Remote Access Software		
		Traffic Signaling		Reflective Code Loading		System Network Connections Discovery			Traffic Signaling		
		Valid Accounts		Rootkit		System Owner/User Discovery			Web Service		
				Subvert Trust Controls		Virtualization/Sandbox Evasion					
				Traffic Signaling							
				Valid Accounts							
				Virtualization/Sandbox Evasion							

Figure-6: MITRE ATT&CK Search Results

these vulnerabilities. MITRE provides CWE list and the description and related CWE with several classifications [10].

5.4. Risk Assessment and risk reduction

Based on the results of threat analysis, a risk management checklist is prepared. The checklist consists of the following items: evaluation of each threat, consideration of risk reduction methods, and evaluation of the implementation of risk reduction methods.

- Information Assets and Function
- Threats (STRIDE classification)
- Anticipated harm/risk conditions
- Risk Estimation and Assessment
- Risk reduction methods (mitigation measures)
- Evaluation after implementation of risk reduction methods
- Any new hazards/hazardous conditions that have arisen because of the implementation of risk reduction measures
- Availability/Reason for response

Since risk assessment for each threat depends on the knowledge and skills of the person in charge and it is difficult to obtain quantitative assessment results, we adopted an open, vendor-independent vulnerability assessment method proposed by FIRST (Forum of Incident Response and Security Teams) [11], CVSS (Common Vulnerability Scoring System) [12]. CVSS is open and vendor-independent vulnerability assessment method proposed by FIRST. CVSS is a common vulnerability scoring system that evaluates the severity of security vulnerabilities based on three criteria: basic evaluation criteria, current state evaluation criteria, and environmental evaluation criteria. By using CVSS, the severity of vulnerabilities can be quantitatively compared under the same criteria.

Risk reduction methods are examined. For example, for "data falsification," we will conduct a final risk assessment by implementing a mitigation method such as "applying security patches.

For threats that require countermeasures, check the "Microsoft Threat Modeling Tool mitigations" based on the STRIDE classification of the Threat Modeling Tool and the MITRE ATT & CK mitigations for each attack method, and then consider and plan tactics.

From the results of the risk assessment, the existence of countermeasures and threats to be prioritized can be identified, and the security requirements can be determined if they have been met by inspecting the threats in testing-related activities.

6. IMPLEMENTATION RESULTS

Using the Threat Modeling Tool, mechanical threat analysis can be performed without any security skills as long as you can draw DFD diagrams. The discussion by looking at the DFD diagrams is also effective to examine where the threats are in the system configuration, and it is commended that the quality requirements are secured from the threat analysis results.

In addition, MITRE ATT&CK can visually identify threats, and when used in conjunction with the Threat Modeling Tool, it is an effective complement to threat analysis. The MITRE ATT&CK can also be used as a database for investigating mitigation measures against attack methods.

The risk management check sheet, which is a deliverable of the design process, can be used as a check sheet for code review in coding and as a vulnerability test item in testing. By implementing security measures upstream, the security activities to be implemented in the subsequent processes have been clarified.

If risk management check sheets had not been prepared in the design process, security activities would have been implemented in other processes without uniformity, and unnecessary man-hours would have been spent due to duplication of work.

In addition, since the CWE of the target threats are clarified, it can be checked against the results of vulnerability testing tools such as sonarqube [13], Fortify [14], etc. to confirm that the target threats have not been detected. You can prove that confidentiality and integrity are ensured.

7. REMAINING ISSUES

The implementation of a threat analysis tool is time-consuming due to the large number of analysis results that are detected and scrutinized. Even if the threats can be identified mechanically by tools, it takes time to evaluate and filter them one by one, which requires security skill to do. For example, if there are three servers, the same threats will be output for all three servers. The question is whether to exclude these results as the same threat or treat them as individual threats. The method of filtering and evaluating the analysis results is an issue to be addressed in the future.

In order to deal with threat analysis, it was necessary to educate the staff about security design and architecture as well as on threat analysis methods and tools. Since many developers implicitly feel that security is difficult, it is necessary to eliminate this image by constantly providing appropriate education.

CONCLUSION

Security is "must be quality" and it is possible to classify security requirements by using a quality model in SQuaRE. However, it is difficult to set the goal of how security should be analyzed. There are various methods of threat analysis, and threat analysis needs a lot of costs. In order to solve this problem, it is considered that clarification of security requirements and evaluation of them for each quality sub-characteristic by using a quality model.

It is also considered necessary to set security ranks for the target systems and to create indicators of which security measures should be implemented for each rank and to what extent time and cost should be spent.

In the current situation where new attack methods emerge and threats continue to emerge, continuous security analysis and tactics are necessary, and this also leads to risk management for the entire system by considering which quality is affected at the sub-characteristic level.

By utilizing SDL's security lifecycle and SQuaRE, we expect that continuous security activities can be taken to provide safe and secure products.

8. References

[1] ISO/IEC 27034-1:2011 “Information technology — Security techniques — Application security — Part 1: Overview and concepts”.

[2] "Manual for Developing Security Requirements in Government Procurement for Information Systems," Cabinet Cyber Security Center (NISC), 2019, in Japanese.

[3] Microsoft Threat Modeling Tool, https://docs.microsoft.com/ja-jp/azure/security/develop/threat-modeling-tool .

[4] MITRE ATT&CK, https://attack.mitre.org/ .

[5] Microsoft Security Update Guide, https://portal.msrc.microsoft.com/en-us/security-guidance

[6] MITRE ATT&CK® Navigator, https://mitre-attack.github.io/attack-navigator

[7] Kusaka H., Nagata, T., Futagawa, Y.: "The Role of QA in SDL Considering Security Quality (Upstream Process)", 2017, https://www.juse.or.jp/sqip/community/bucyo/8/files/shiryou_seika7.pdf, in Japanese.

[8] ISO/IEC 25010:2011 “Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models”.

[9] ISO/IEC 25012:2008 “ Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model”.

[10] CWE, https://cwe.mitre.org/ .

[11] FIRST, https://www.first.org/ .

[12] CVSS, https://www.first.og/cvss/ .

[13] sonarqube, https://www.sonarqube.org/ .

[14] Fortify, https://www.microfocus.com/en-us/cyberres/application-security/ .

[15] Iwasaki,D., Yasuda.K, Kato, D.: Security Design Methodology Considering Threat Analysis in SDL, The 51st Symposium Reliability, Mainteinability and Sefety, 2022, in Japanese.