Guideline to Cyber Threat Modelling for Risk Assessment

Critical Information Infrastructure

CSB
CYBER SECURITY BRUNEI

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 Importance of Threat Modelling

Due to finite resources of the system owner, it is difficult to mitigate every vulnerability within a system. Therefore, system owners must prioritise risks and treat them accordingly. A key step in determining risk is identifying threat events, which contribute to the likelihood and impact of risk. A threat event refers to any event during which a threat actor[1], by means of threat vector[2], acts against an asset in a manner that has the potential to cause harm. In the context of cybersecurity, threat events can be characterised by the tactics, techniques, and procedures (TTP) employed by threat actors.

Threat modelling helps owners comprehensively identify threat events that are relevant to the system, so that owners can focus on implementing effective control measures to protect key components within the system. This makes it harder for the adversary to compromise key components by establishing a foothold, pivoting, and moving laterally within the system. Consequently, system owners can stem and curtail the kill-chain before the adversary reaches the crown jewels. With a threat model, system owners can also avoid blind spots in identifying threat events.

1.2 Purpose of Document

CSB issued the Guidelines for Auditing and Conducting Cybersecurity Risk Assessment for Critical Information Infrastructure in April 2024. The document provided guidance to Critical Information Infrastructure Owners (CIIOs) on performing a proper cybersecurity risk assessment, and briefly covered steps for threat modelling as part of the risk assessment.

This document supplements the document by elaborating on threat modelling, and aims to provide a practical and systematic way to identify threat events that can be used in a cybersecurity risk assessment.

It will introduce various approaches and methods of threat modelling, and provide a suggested framework, coupled with practical examples, for individuals and groups to adopt to derive a robust system threat model and relevant threat events. System owners can then incorporate these threat events into their cybersecurity risk assessment to develop and prioritise effective controls. Ultimately, this exercise aims to cultivate a customised threat perspective in system owners that goes beyond meeting minimum generic standards.

1.3 Scope

This document is for individuals or groups who would like to build a threat model for their system(s). They can use the results of the threat model as inputs to other assessments, such as cybersecurity risk assessments, to prioritise risk controls. Individuals and groups using this guidance (subsequently collectively termed as Users) may include, but are not limited to, the following:

- Internal stakeholders e.g., system owners, business unit heads, Chief Information Security Officers, and personnel involved in IT risk assessment and management within any organisation, including Critical Information Infrastructure Owners;
- External consultants or service providers engaged to conduct threat modelling on behalf of system owners; and
- Red team members, blue team defenders, and purple team members.

The guidance set out in this document focuses on the key areas of technical scoping, system decomposition, threat identification and attack modelling. Other areas such as cyber threat intelligence monitoring and

[1] Threat actor refers to a person or entity that is responsible for an event that has the potential to cause harm.
[2] Threat vector refers to the path or route that a threat actor uses to attack a target.

studying geo-political threats, which are under the wider domain of threat monitoring and analysis, are beyond the scope of this document.

2.0 APPROACH

2.1 System Level Approach

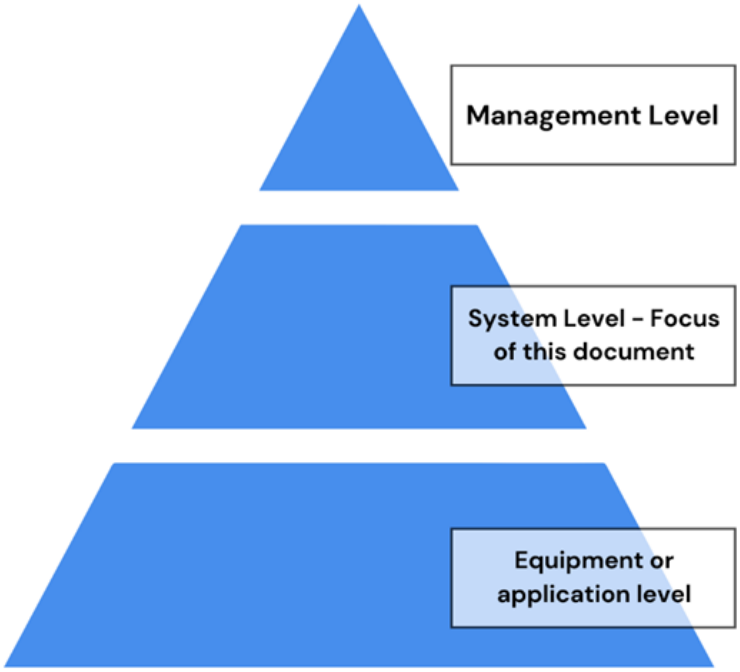

Figure 1: Hierarchy of Threat Analysis

Users can approach threat analysis at three different tiers – from a management perspective, from a system perspective, and from an equipment or application perspective. Below is a general description of each tier. The focus of this documents is at the system level.

- ***Management level (out of scope)* -** intelligence feeds and trends are analysed with focus on external factors such as geo-politics. Analysts typically conduct adversary centric profiling, considering adversary's broad motives and actions pre- and post-intrusion. Organisations do this from a high-level perspective typically for management's consumption. An example of such an approach is the Common Cyber Threat Framework used by the US Office of Director of National Intelligence (ODNI), which follows across the four horizontal "threat lifecycle" stages of preparation, engagement, presence and consequences, while expanding on the verticals of objectives, actions and indicators.

- ***System level (in scope)* –** this approach considers system constructs and relationships, as well as system behaviours, in the form of components, architectural layers and data flows. The User first models' assets, data flows and boundaries within an environment, before determining relevant threat events to the system. We will focus on systems level in this document.

- ***Equipment or application level (out of scope)* –** threat analysis at this level is the most granular. It often involves cyber threat hunting, log correlation, detailed data triaging, advanced analytics, and heuristic techniques etc. to scan for detailed evasion and exploitation of published vulnerabilities.

As mentioned, there is no one right or wrong approach to threat modelling. Users must choose the approach based on context of the business needs, greater enterprise environment, situation, and audience.

2.2 Common Missteps in Threat Modelling

While system owners seek to model threat events for their systems, some pitfalls hinder their process or diminish the effectiveness of their threat model. Some of the common problems include:

- ***Misdirected or unbalanced threat focus*** **–** in some cases where scope of threat model is ill-defined, system owners may derive non-existent or irrelevant threats. Similarly, systems owners' attention may be overly occupied with the "flavour of the day" (events that are prevalent in the news) and neglect other scenarios which, although are not as current, also pose realistic threats to the system.

- ***Not considering subsequent stages of attack*** **–** most attacks occur in stages, often starting first with reconnaissance, followed by intermediate hops, and ultimately ending with impact to the crown jewels. It is crucial that system owners consider each stage of an attack and not only address the initial threat. This will enable defence in depth with multi-layered controls.

- ***Taking a "once and done" approach*** **–** the threat model of a system should be treated as a living document, bearing in mind its shelf life. With ever-evolving external threats and new internal vulnerabilities discovered, it is important to refresh threat models constantly, especially whenever there are material changes to the system.

A threat model is only useful when conducted in a systematic manner with well-defined scope. On the contrary, an ineffective threat model will result in poor prioritisation of resources to address cybersecurity risks, and system owner being ill-prepared for a cyber-attack. Hence, it is important for organisations to adopt good practices and avoid the common missteps when conducting threat modelling.

2.3 Integrating Threat Modelling into Risk Assessment Process

As mentioned in Section 1.2 above, CSB issued the Guidelines for Auditing and Conducting Cybersecurity Risk Assessment for Critical Information Infrastructure, which provides guidance on performing a proper cybersecurity risk assessment. This document supplements the risk assessment process by providing a systematic way to identify threat events as part of the cybersecurity risk assessment.

Threat modelling augments the risk assessment process by generating contextualised threat events with well-described sequence of actions, activities, and scenarios that the attacker may take to compromise the system. With more relevant threat events, risk assessments conducted by Users will be more rigorous and robust, resulting in more targeted controls and effective layered defence. This addresses the first two common missteps mentioned in the previous section. In addition, since risk assessment is a continuous cycle, the threat model should also be regularly updated, addressing the third common misstep mentioned above.

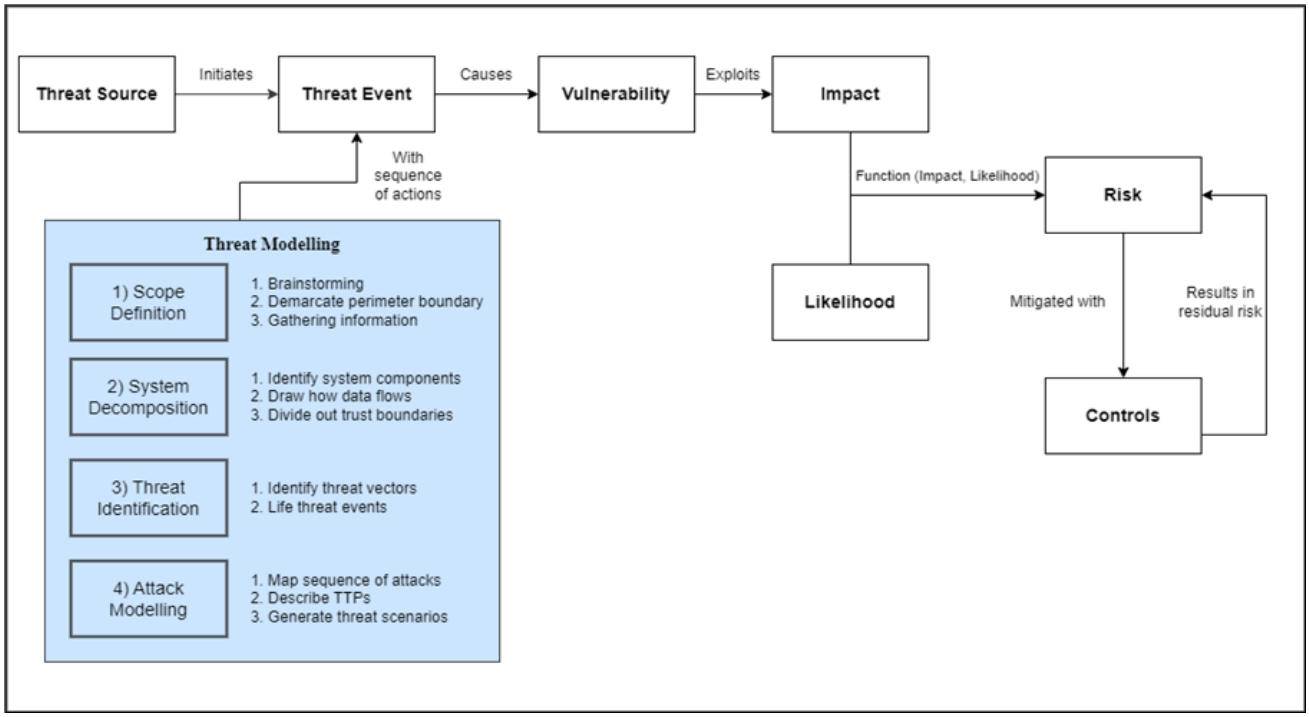

Figure 2: How Threat Modelling Fits into Risk Assessment

3.0 METHODOLOGY

3.1 Overview of Method

The threat modelling method proposed in this guide comprises broadly the following 4 steps:

- **Step 1 –** Scope Definition, which involves gathering information and demarcating perimeter boundary;
- **Step 2 –** System Decomposition, which involves identifying system components, drawing how data flows, and dividing out trust boundaries;
- **Step 3 –** Threat Identification, which involves identifying threat vectors and listing threat events; and
- **Step 4 –** Attack Modelling, which involves mapping sequence of attack, describing tactics, techniques, and procedures.

Finally, at the end of the section, we will walk-through an example to bring everything together.

3.2 Step 1: Preliminaries and Scope Definition

Users should establish the technical scope, system architecture, and system components before performing threat modelling for a system. Users should also examine the security perimeters, interfaces, and data flows to characterise the attack surface.

Task A: Gather Information

Users should gather information pertaining to the system architecture and dependences by referring to system operations manual, software design document (SDD), technical specification or any system-related documentation. Users can also interview the system custodian, system administrator, and database administrator to get their input on the system architecture.

Task B: Demarcate Perimeter Boundary

Based on existing network diagrams and architecture drawings, Users should demarcate the perimeter boundary[3] to determine the scope for threat modelling. Some examples of guiding principles to determine what component is within the perimeter boundary may include, but are not limited, to the following:

- Components deployed behind data diodes or “demilitarised zones” (DMZs);
- Components that support the functioning and running of the system at any point in time e.g., servers, databases, client workstations, hosts, switches, routers etc.; and
- Components that support the cybersecurity of the system e.g., firewalls, intrusion detection systems (IDS), intrusion prevention systems (IPS).

[3] CIIOs may consider using the CII boundary as the perimeter boundary to determine the scope for threat modelling.

3.3 Step 2: System Decomposition

System decomposition is the process of breaking down a system into its different components. Components refer to external entities, users, processes, or data stores as seen in Figure 3 below. By understanding the data flows among components, Users can construct a data flow diagram (DFD). DFDs help Users to gain better insights to the system by giving a visual representation that shows how the "nuts and bolts" of the system work. DFDs aim to focus on how data moves through the system. System decomposition consists of three tasks as follow:

Task A: Identify System Components

Users should identify system components prone to attack, i.e., the components, which a potential attacker may be interested in, categorised as follows:

- Data or functions essential to the business mission of the system; and
- Data or functions that are of special interest to an attacker

After identifying the system components, Users should document the component types e.g., database, transactional system etc., locate where these components reside within the system, and draw the DFD by denoting the system component as shown in the figure below:

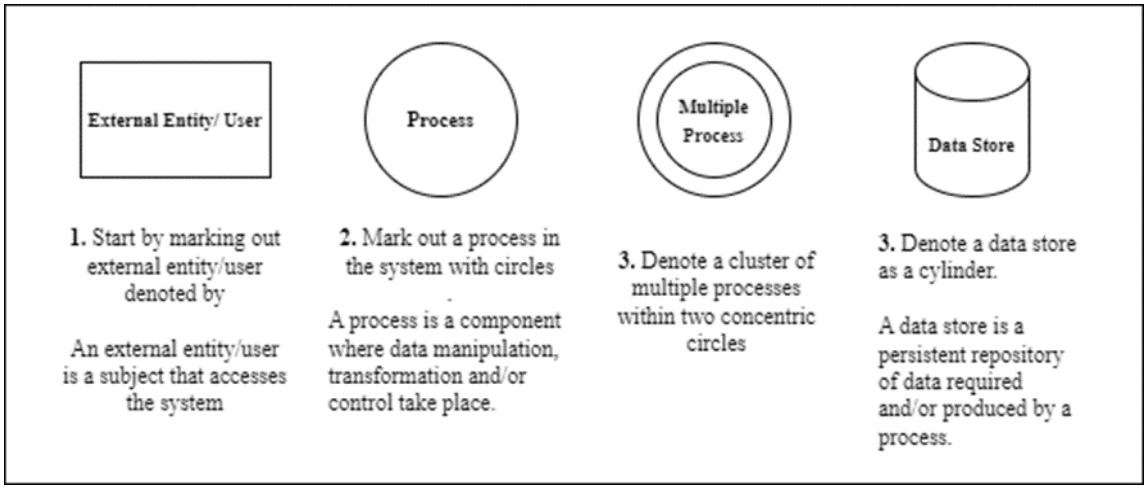

Figure 3: Legend for System Components

Task B: Draw How Data Flows Once

Users have identified system components, Users should map out how the components communicate with one another, or provide some form of access to one another. With a completed DFD, Users will be able to visualize how data flows through the system and which components are involved, as shown below:

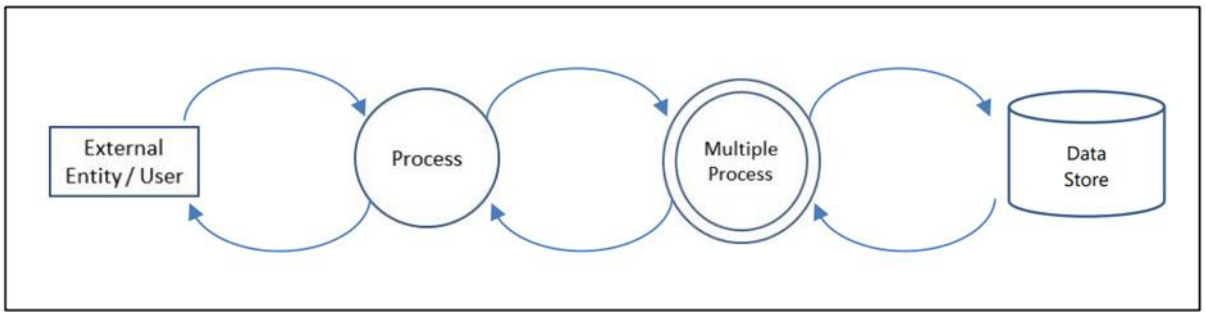

Figure 4: Example of Data Flow

Task C: Divide Out Trust Boundaries

Finally, Users should identify the respective limits of access, as well as the required levels of authorisation e.g., trust levels, granted to subjects. In other words, a trust boundary can be used to represent the change of trust levels as the data flows through the system. The trust boundaries provide the User with various intersections of data flow, which helps the User to identify attack surfaces where an attacker can traverse across secure or unsecure zones. Dashed lines in the DFD below denote trust boundaries:

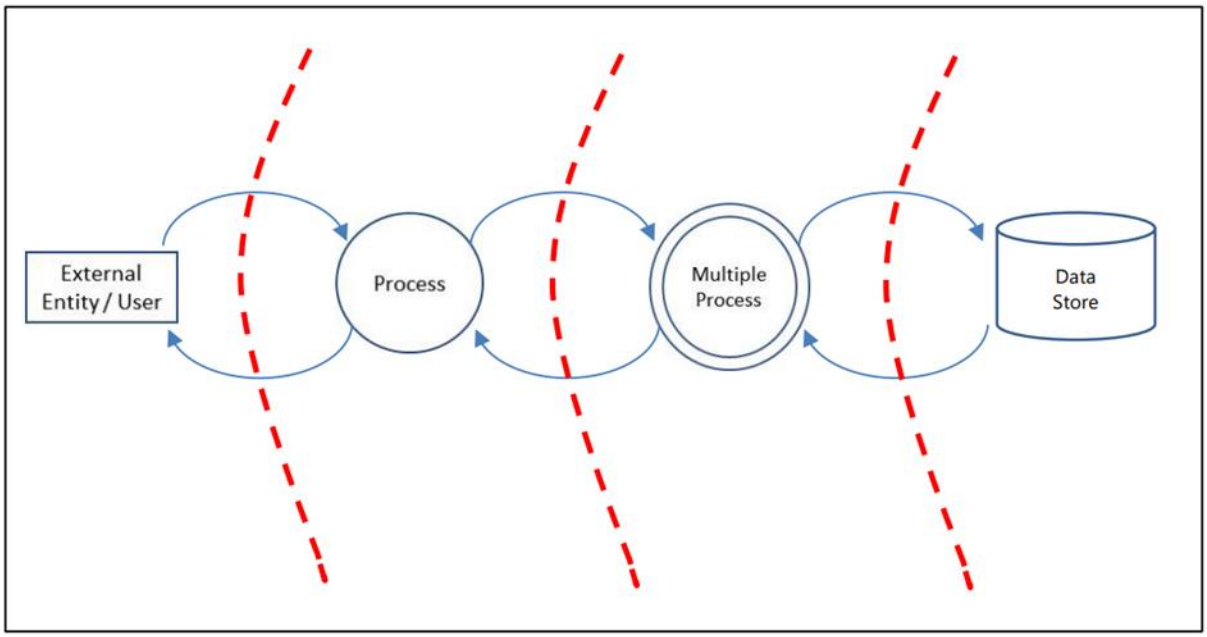

Figure 5: Example of Trust Boundaries

3.4 Step 3: Threat Identification

Step 3 builds on the information gained from the previous steps by providing systematic methods to identify threat events relevant to the system. This consists of three tasks – brainstorming, identifying threats vectors, and documenting possible threat events. One common approach that can be used in this step is the STRIDE-LM methodology. Users can also refer to NIST SP 800-30 Appendix D and E for threat sources and events. For threats related web applications, the OWASP Automated Threat Handbook[4] is a useful source of information providing 21 threats. Information Sharing and Analysis Centres (ISACs)[5] are also useful places to obtain threat information that is relevant to your organisation or sector. This can be complemented with information from threat intelligence sources to provide a more holistic picture. Some points on what to look out for when using threat intelligence sources are covered in the next section.

Threats are constantly evolving, and it is important to rely on either open source or paid threat intelligence sources to gain knowledge on the latest threats and learn how attackers target systems. Although subscribing to multiple threat intelligence sources broadens the Users' perspective, Users must first be clear about its objectives to collect specific threat intelligence. It is counterproductive to have too many sources leading to fatigue in teams examining feeds. Useful threat information typically has the following characteristics:

- Timely – Information should be received in a timely manner as information that is outdated is useless to Users;
- Relevant – Information needs to be relevant to the context of the Users. For example, industrial control systems may have different priorities compared to financial institutions; and
- Actionable – Information should be actionable for the correct group of users. Users must be able to react to information at the appropriate level e.g., tactical, or strategical level.

Task A: Identify Threat Vectors

Threat vectors are paths through which an attacker can exploit to penetrate a system component or bypass defences. For this task, Users identify threat vectors for individual components enumerated previously. From the system's DFD generated in the previous step, Users identify specific threat vectors that an attacker can use to compromise the individual system component. Vulnerability studies and intelligence on threat actors and their modus operandi should support this task.

Task B: List Possible Threat Events

An attacker may, by means of a vector, act against the system to cause harm or steal information. Threat events can be characterised by the tactics, techniques and procedures (TTP) employed by the attacker. To list all possible threat events, TTPs and attack sources comprehensively, it is useful to categorise threat events using methods such as STRIDE-LM, NIST SP 800-154, Security Cards or OCTAVE. We provide elaboration on STRIDE-LM in this guide, but organisations should choose a method that suits their purpose. STRIDE-LM was chosen in this guide as it is deemed a simple and systematic way to identify threats. The information derived from such an exercise will highlight areas of focus to implement controls that mitigate threats.

[4] Details on the OWASP Automated Threat Handbook can be found at the OWASP Automated Threats to Web Applications portal at https://owasp.org/www-project-automated-threats-to-web-applications/.
[5] There are numerous global ISACs. Some examples include FS-ISAC for financial services, H-ISAC for healthcare, OT-ISAC for operational technology etc.

STRIDE-LM

Kohnfelder and Garg (1999) developed STRIDE as a mnemonic to identify various threats types – spoofing, tampering, repudiation, information disclosure, denial of service and escalation of privilege. Muckin and Fitch (2019) further proposed to add LM into the mnemonic to include Lateral Movement. The table below captures the threat categories, the security property that they compromise and their definitions:

STRIDE-LM	THREAT	PROPERTY	DEFINITION
S	**S**poofing	Authentication	Impersonating someone or something
T	**T**ampering	Integrity / Access Controls	Modifying data or code
R	**R**epudiation	Non-repudiation	Claiming to have not performed a specific action
I	**I**nformation Disclosure	Confidentiality	Exposing information or data to unauthorized individuals or roles
D	**D**enial of Service	Availability	Deny or degrade service
E	**E**levation of Privilege	Authorization / Least Privilege	Gain capabilities without proper authorization
LM	**L**ateral **M**ovement	Segmentation / Least Privilege	Expand influence post-compromise; often dependent on Elevation of Privilege

Table 1: Definition of Threat Categories Using STRIDE-LM (Muckin and Fitch, 2019)

- **Spoofing**

Spoofing is pretending to be someone or something else. Spoofing typically compromises authenticity. Examples of spoofing threats include impersonating a legitimate user or machine in the network, or creating a fake file or process on a machine. Common examples of controls to mitigate such threats include having strong authentication mechanisms.

- **Tampering**

Tampering is to modify something without authorisation. Tampering typically compromises integrity. Examples of tampering include altering a file, memory space, or network configuration or communication. Common examples of controls to mitigate such threats include encryption and proper access checks.

- **Repudiation**

Repudiation is claiming not to have done something, or not being responsible for something that happened. Repudiation is typically associated with users, whether authorised or unauthorised, denying performing an action without other parties being able to prove otherwise. Examples of repudiation include a user perceived not to have performed an activity even if they did so, or claims to be innocent despite fraudulent. Common examples of controls to mitigate such threats include robust logging and digital signatures.

- **Information Disclosure**

Information disclosure is allowing people to see information that they are not authorised to see. Such attacks are usually breaches in confidentiality. While information disclosure mostly affects files and data stores, it can also happen on side channels like processes and networks. Common examples of controls to mitigate such threats include encryption of data in storage, use or transit.

- **Denial of Service (DoS)**

DoS is disabling or degrading a service so much so that users cannot access the service normally. DoS is usually an attack on availability. An attacker may carry out DoS by abnormally consuming service resources. Examples of such resources include memory, processing power, network resources, and data storage. Common examples of controls to mitigate such threats include proper checking of legitimate traffic and putting in place redundancy mechanisms.

- **Elevation of Privilege**

Elevation of privilege is gaining access that exceeds the authorisation of a subject. Common ways to elevate privileges include corrupting processes such that it an unauthorised user can perform operations that he or she prohibited to do so. An attacker can also bypass checks to gain elevated privileges. Common examples of controls to mitigate such threats include applying the principles of least privilege and need to know, and to have strong privileged account protection mechanisms.

- **Lateral Movement**

Lateral movement is pivoting across the network usually with the eventual goal of reaching the crown jewels of the system to inflict damage or for data exfiltration. Lateral movement is typically coupled with elevation of privilege to gain access to other parts of the network. Common examples of controls to mitigate such threats include proper segmentation of network with strong firewall rules.

While *STRIDE-LM* can aid Users in accurately categorising threats, this should not be the only focus. Emphasis should be on using STRIDE-LM as a thought process to brainstorm all possible threats to a system. For example, how can an attacker use spoofing to attack the system? Could the attacker spoof as a user, a file process, or an endpoint? How will the attacker tamper with the system? Could the attacker tamper with the database, the configuration files, or the logs? In addition, certain types of attacks may overlap multiple categories so Users must seek to avoid blind spots.

STRIDE-LM provides a good baseline. However, users may have additional threats identified through other means. Users should then use STRIDE-LM to complement their existing threat inventory to see if they have overlooked anything. Even if no new threats are identified using STRIDE-LM, the process of using STRIDE-LM to complement their existing threat inventory provides an additional layer of assurance to the organisation.

Finally, Users should add the threat events generated on the DFD for easy visualisation of the threats at each location:

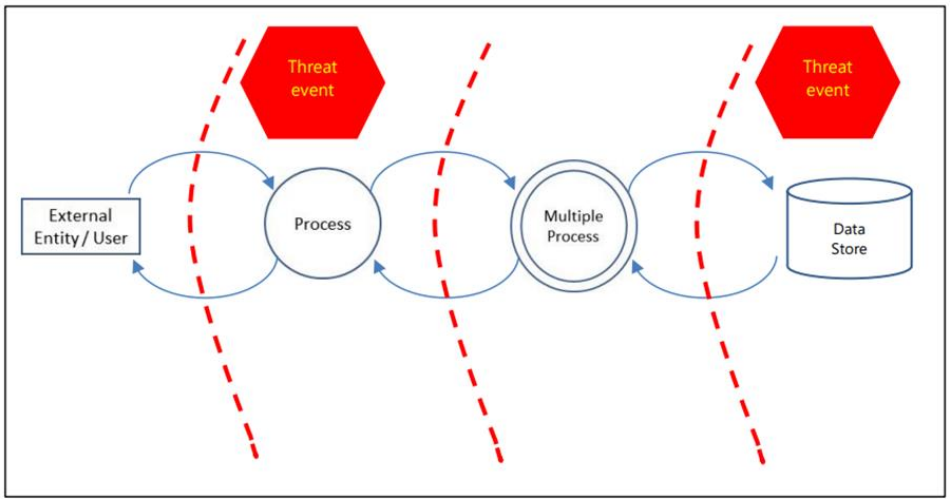

Figure 6: Example of Threat Events Incorporated into the DFD

3.5 Step 4: Attack Modelling

After identifying threat events relevant to the system, Users should link these events into a possible sequence of attack. Attack modelling describes an attacker's intrusion approach so that Users can identify mitigation controls needed to defend the system and prioritise its implementation. Users can use either MITRE ATT&CK or Lockheed Martin Kill Chain to model the attack.

These methods provide a thought process to map possible sequence of attack used by attackers to exploit resources of organisations to achieve their malicious intent, at the same time; the modelling allows the users to understand the set of conditions required, for a threat to be successful. As such, the user could develop strategies and prioritise resources to defend against the threats.

By understanding the attack vectors and security risks at various stages, these models provide useful insights for the user to apply multi-layered controls to prevent attacks, interrupt on-going attacks, and minimise the impact of an attack.

MITRE ATT&CK

ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) by the MITRE Corporation[6] is a publicly accessible knowledge base used as a foundation for the development of specific threat models and methodologies. The framework provides a comprehensive matrix of tactics and techniques used by attackers to compromise an organisation's system. With the same understanding of an attacker's tactic and strategy, Users can defend their systems effectively.

ATT&CK consists of 12 tactical categories, which is the "why" of an attack, listed below:

MITRE ID	NAME	DESCRIBTION
TA0001	Initial Access	To enter the system
TA0002	Execution	To run the malicious code
TA0003	Persistence	To maintain a foothold
TA0004	Privilege Escalation	To gain higher-level permissions
TA0005	Defence Evasion	To avoid being detected
TA0006	Credential Access	To steal account names and passwords
TA0007	Discovery	To learn about the system environment
TA0008	Lateral Movement	To traverse through the system environment
TA0009	Collection	To gather information of interest
TA0011	Command and Control	To control a compromised system
TA0010	Exfiltration	To steal data
TA0040	Impact	To manipulate, interrupt, or destroy your systems and data

Table 2: ATT&CK Categories of Tactics

Each tactic contains an array of techniques that the attacker can use. This is the "how" of the ATT&CK framework. At the time of writing, there are 266 techniques in the ATT&CK matrix for Enterprise[7]. ATT&CK is by no means a static framework. MITRE regularly updates ATT&CK with the latest tactics and techniques discovered by security researchers. Users should customise the relevant ATT&CK matrices to their context. Suggested matrices for Information Technology (IT) systems and Operation Technology (OT) systems respectively can be found in the Annex of this document.

[6] Attack matrices can be found at https://attack.mitre.org/matrices/
[7] There is also an ATT&CK matrix for Industrial Control Systems (ICS)

Lockheed Martin Cyber Kill Chain

The Lockheed Martin Cyber Kill Chain (Kill Chain) framework was derived from military offensive strategy. The framework is presented as a series of well-defined sequence of stages that the attackers are likely to complete to achieve their end objective. With the same understanding, Users can defend their systems effectively.

The Kill Chain consists of 7 stages – reconnaissance, weaponisation, delivery, exploitation, installation, command and control, and actions on objectives, spread across 3 phases – pre-compromise, compromise, and post-compromise. Each stage of the Kill Chain is elaborated below:

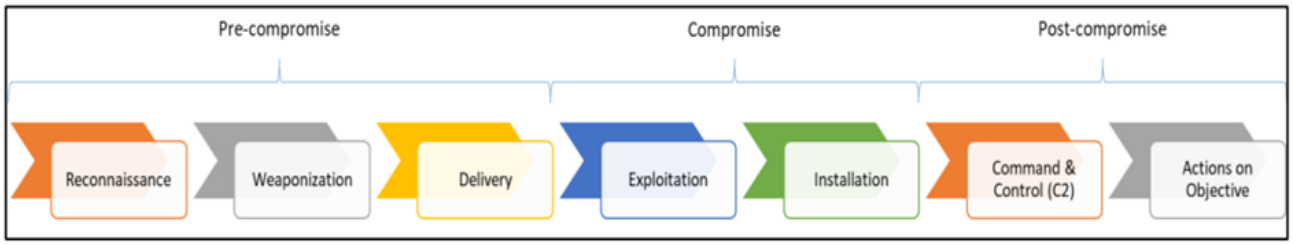

Figure 7: Lockheed Martin Kill Chain Lifecycle

- Reconnaissance - research, identification, and selection of targets
- Weaponisation - pairing malware containing exploits with a deliverable payload e.g., Adobe PDF and Microsoft Office files
- Delivery – transmitting the weapon to the target e.g., via email attachments, websites, or USB drives
- Exploitation - once delivered, the weapon's code is triggered, exploiting vulnerable applications or systems
- Installation - the weapon may install a backdoor on a target's system allowing persistent access
- Command and Control (C2) – malicious external servers communicate with the system through call-backs e.g., remote shell
- Actions on Objective - attacker works to reach the crown jewels and achieve the objective of the intrusion e.g., exfiltration or destruction of data, disruption of processes

As can be observed, the Kill Chain only describes the sequence of attack. Users should hence populate each stage of attack with specific threats derived from Step 3 – Threat Identification, explained previously.

3.6 Step 5: Bringing Everything Together

Now that we have introduced the concepts of each step-in threat modelling, we will walk through how Users can apply them using an example of a web application and database system. Various tools can be used for this exercise. Some examples include Microsoft Visio and Microsoft Threat Modelling Tool.

First, to draw the DFD, we enumerate all components within the web application and database system. Importantly, we should denote the crown jewel with a star. This component holds all the critical information or is crucial for critical functions. In this case, we identify the crown jewel is as the database.

a) Users
b) Enterprise workstation
c) Network equipment e.g., firewall, routers
d) Web server
e) Application server
f) Database server
g) Database (crown jewel)

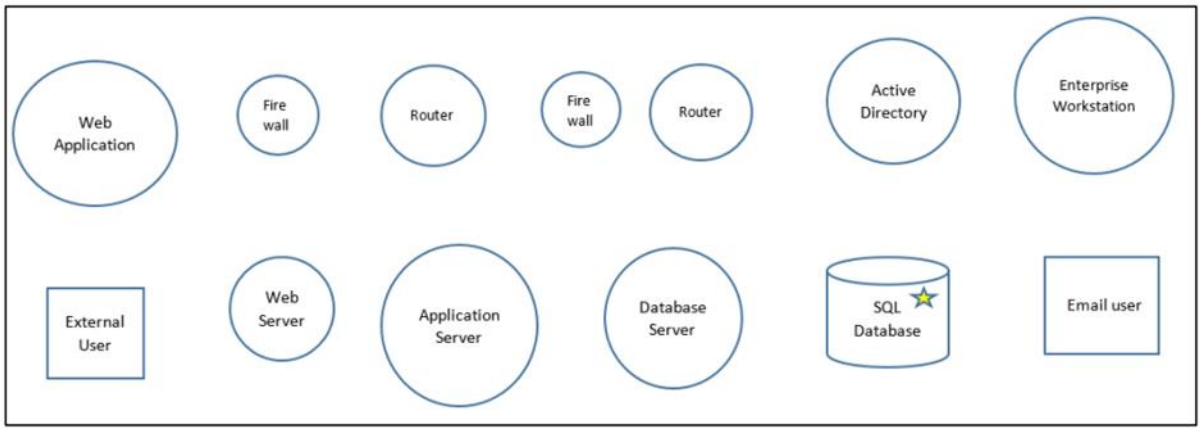

Figure 8: Component Identification

Next, we draw out all the data flows between components in our system. The flow starts from an external user initiating HTTP request to the web server. The web server listens for client request and responds with HTTP reply, hence the bidirectional arrow. The web server communicates with the application server via a router, which in turn communicates with the database server through API calls. The database server finally sends SQL queries to the database. In addition, there are also enterprise workstations providing services including email.

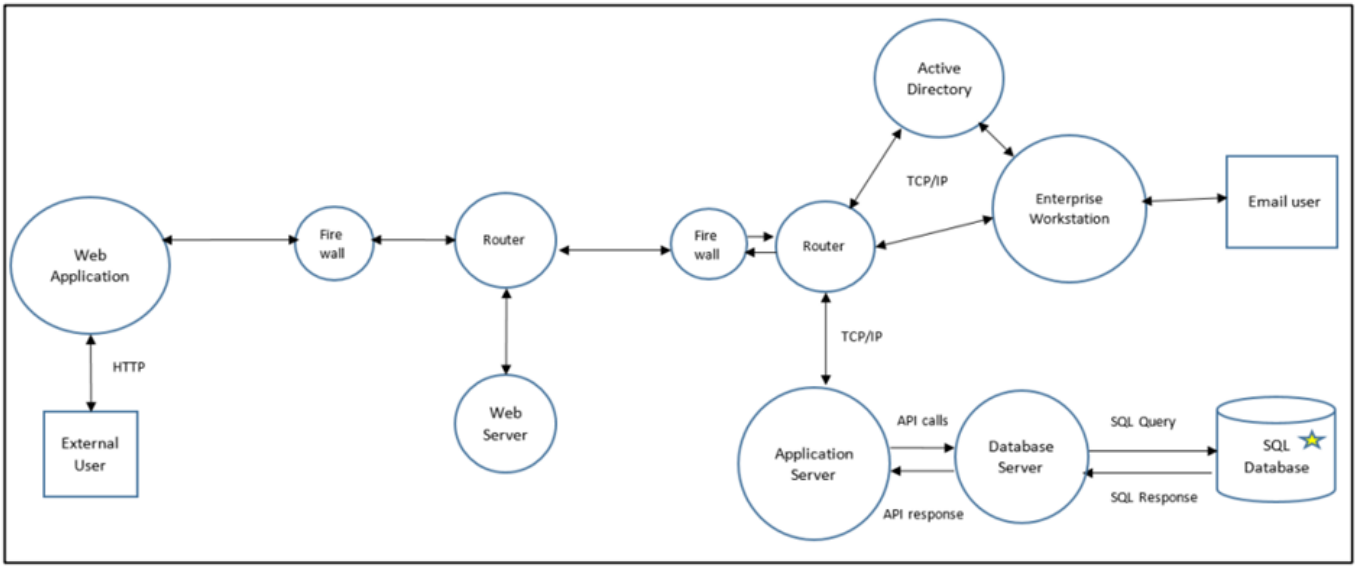

Figure 9: Data Flow Diagram

We the divide the system based on trust boundaries. Trust boundaries separate domains with different rights and privilege assignments. For example, the rights and privileges of subjects from the internet are far less than the rights and privileges of subjects residing within the corporate network. For our web application and database system, we draw a trust boundary between the internet and Demilitarised Zone (DMZ), and another trust boundary between DMZ and corporate network.

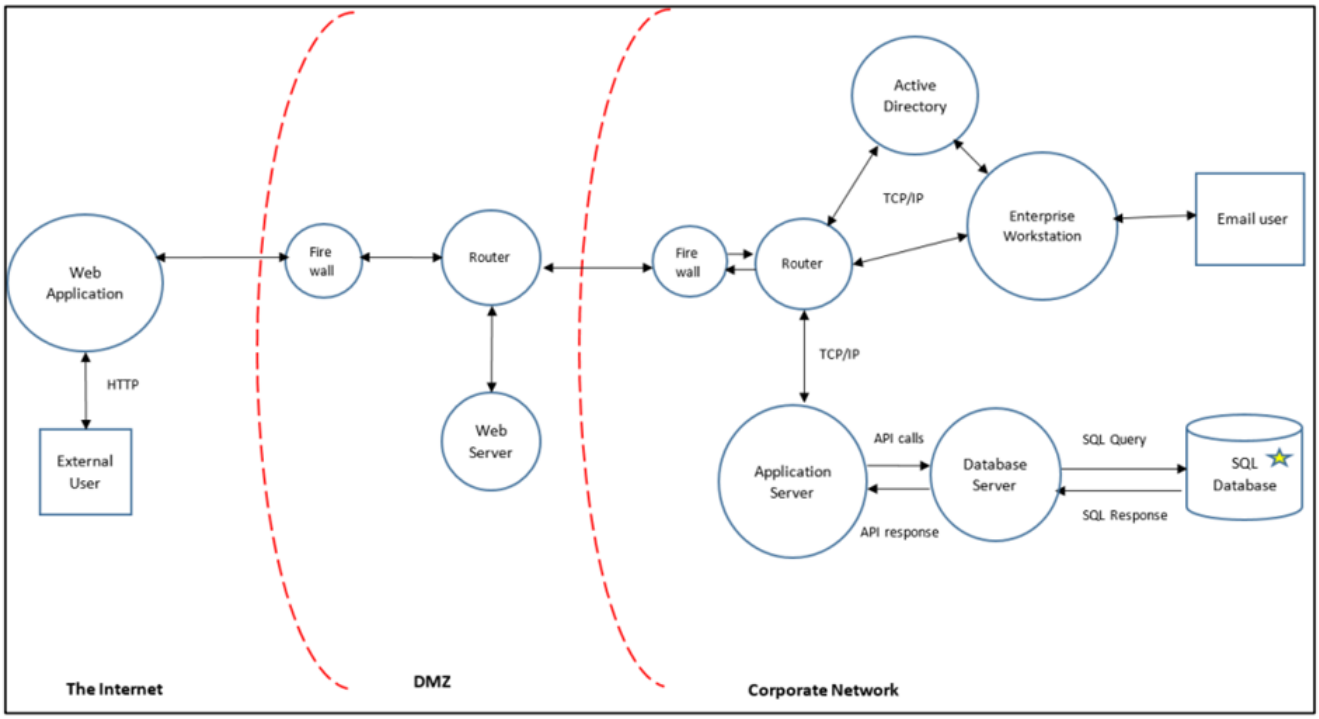

Figure 10: Trust Boundaries

After completing the DFD, we identify the threats. As an example, we will look specifically at two components together - the backend application and database servers. Broadly, attacks may include compromising confidentiality, integrity and availability. With that in mind, we dive deeper by going through the STRIDE-LM mnemonic.

Firstly, we examine **Spoofing.** An attacker, who may also be an insider, can harvest credentials, capture tokens, hijack session IDs and steal keys through man-in-the middle attacks or squatting. He then stuffs these credentials and injects codes into RESTful API headers and parameters to masquerade as another user. In that way, he is able make SQL queries to the database to exfiltrate data leading to **Information Disclosure**. Even if he is not able to see all the data initially, he may be able to do so by using SQL injection in the form of blind SQL injection (true/false testing for error response) or table aggregation. He may also **Tamper** with database records by using various SQL commands such as INSERT INTO, DELETE, DROP etc.

If he can inject administrator credentials or bypass authentication and authorisation checks, is will be able to Elevate Privileges and gain rights to access data that are more sensitive.

The attacker can also conduct application layer Denial of Service attack by flooding the server with requests, although this is typically done upstream through the web server.

In the case above, it is likely that event and process logs will capture the attacker's activity. However, if he using a stolen trusted account there is a chance of Repudiation.

Once the attacker is done with this database, if there are still other targets in the system, he may conduct Lateral Movement to compromise other servers.

Once we have enumerated the threats, we can include them into the DFD.

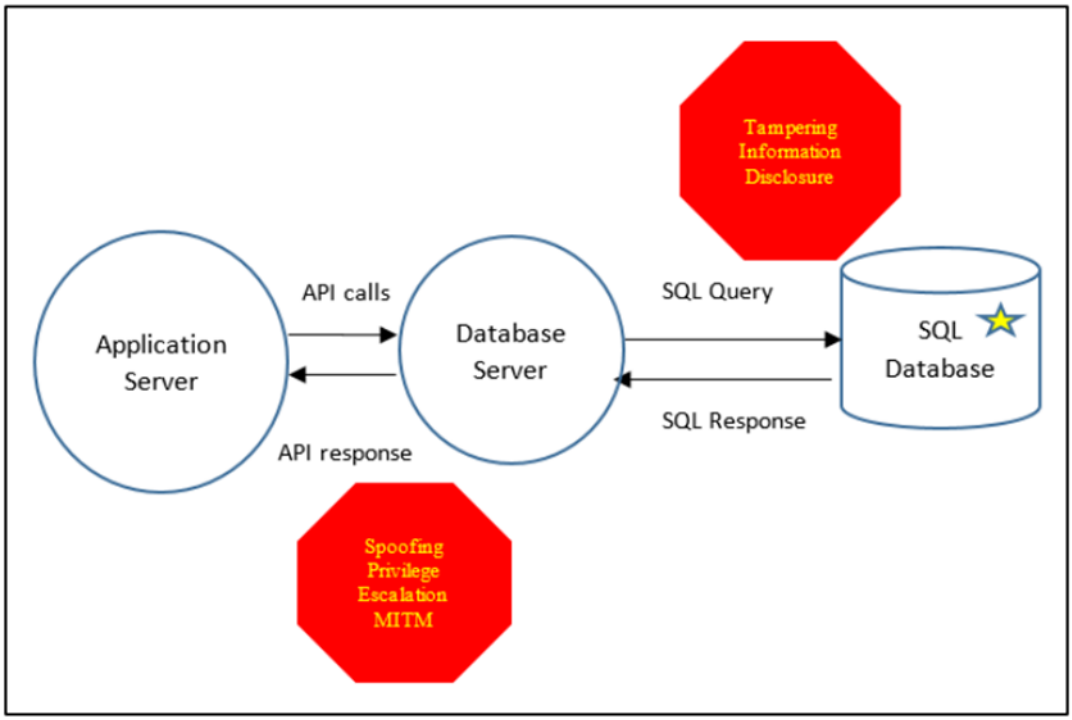

Figure 11: Potential threat events at application and database server (non-exhaustive)

We should repeat the STRIDE-LM process for every component within the DFD. The final product is shown below.

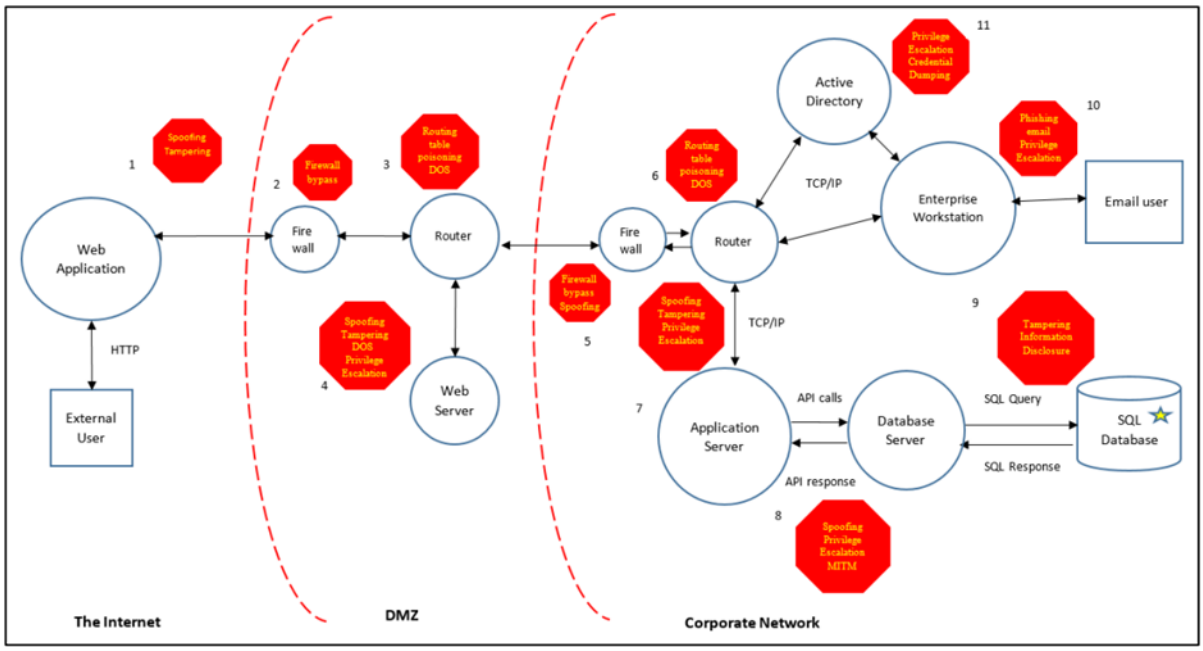

Figure 12: Complete DFD enumerated with possible threat events

Once we have plotted the DFD with threat events enumerated, we can sequence the threat events accordingly. We use the table format shown in the samples to list various sequence of attacks. The table has the following headers:

a) Point of Entry – where the system is first compromised

b) Threat Actor(s) – perpetrators of the attack. This may include Advanced Persistent Threat (APT) groups, cyber criminals or privileged insiders

c) Sequence of Attack – order of threat events, which we will walk through with an example below

d) Threat Description – how the attack takes place

e) Examples

Taking the Sample Threat Model for ICT – Web Application and Database System, as an example, we map the threat sequence according to MITRE ATT&CK. Firstly, the attacker can access to a client with the web application [TA0001] from the internet. Next, he exploits vulnerabilities to run malicious code [TA0002] via attacks such as cross-site scripting.

Through which, he can login to the application, perhaps even as an administrator [TA0004]. Once in the system, he may establish persistence so that he will not be booted out [TA0003]. All the while, he operates as a trusted user to evade detection [TA0005]. Through credential dumping [TA0006], as well as network and host enumeration [TA0007], he can know how to move laterally and pivot through the system [TA0008] towards the crown jewels. Finally, from the last hop, the bulk queries tables from the SQL database, adopting methods such as aggregation to steal desired information [TA0009] [TA0010]. This information if disclosed or sold may have detrimental effect on an organisation's reputation. It may result in financial loss or even regulatory penalties. [TA0040].

We can also categorise the attack sequence into the three phases of the Kill Chain. For example, before the attacker enters the system from the web application client, he would need have needed to conduct some reconnaissance, in the pre-compromise phase, about the application design, operating system, hosting server etc. to find vulnerabilities. Once that is done, he knows how to craft the malicious code to exploit the application using various methods like cross-site scripting. After injecting the script, he compromises the system by being able to achieve unauthorised access and traverse within the system. Post-compromise, he moves towards the crown jewels to execute SQL queries and steals data. This is an action on objective for the attacker.

Ultimately, we populate the result from attack modelling into the table mentioned above, in a similar way as the entry below. The star at the end of the sequence of attack indicates that crown jewel i.e., database, is impacted.

S/N.	Point of Entry	Threat Actor(s)	Sequence of Attack	Threat Event	Examples
3.2 – Web App Vuln	Web Application	APT Group Cyber criminals	1,4,7,8,9 ✶	Logging into the administrator or user account through broken authentication or injection attack, to access web application to query information from database.	Cross-site scripting (XSS), SQL injection. Broken authentication – insecure coding of application authentication, credential stuffing.

Table 1: Extract from Sample Threat Model for ICT -Web Application and Database System

4.0 REFERENCES

1. Bodeau D.J., McCollum, C. D. Homeland Security Systems Engineering & Development Institute. (2018). Cyber Threat Modelling: Survey, Assessment and Representative Framework.
2. Kohnfelder, L., & Garg, P. (1999). The Threats to Our Products.
3. Morana M. M., UcedaVelez T. (2015). Risk Centric Threat Modelling: Process for Attack Simulation and Threat Analysis.
4. Muckin, M., & Fitch S. C. (2019). Threat-Driven Approach to Cyber Security.
5. National Institute of Standard and Technology. (2012). Guide for Conducting Risk Assessments https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final.
6. National Institute of Standard and Technology. (2016). Guide to Data-Centric System Threat Modelling https://csrc.nist.gov/publications/detail/sp/800-154/draft.
7. Shostack, A. (2014). Threat Modelling – Designing for Security.
8. Shevcenko, Nataliya. Software Engineering Institute (SEI). Threat Modelling: 12 Available Methods. https://insights.sei.cmu.edu/blog/threat-modeling-12-available-methods/

ANNEX A – SAMPLE THREAT MODELS

A1 – Information Communication Technology (ICT) – Web Application and Database System

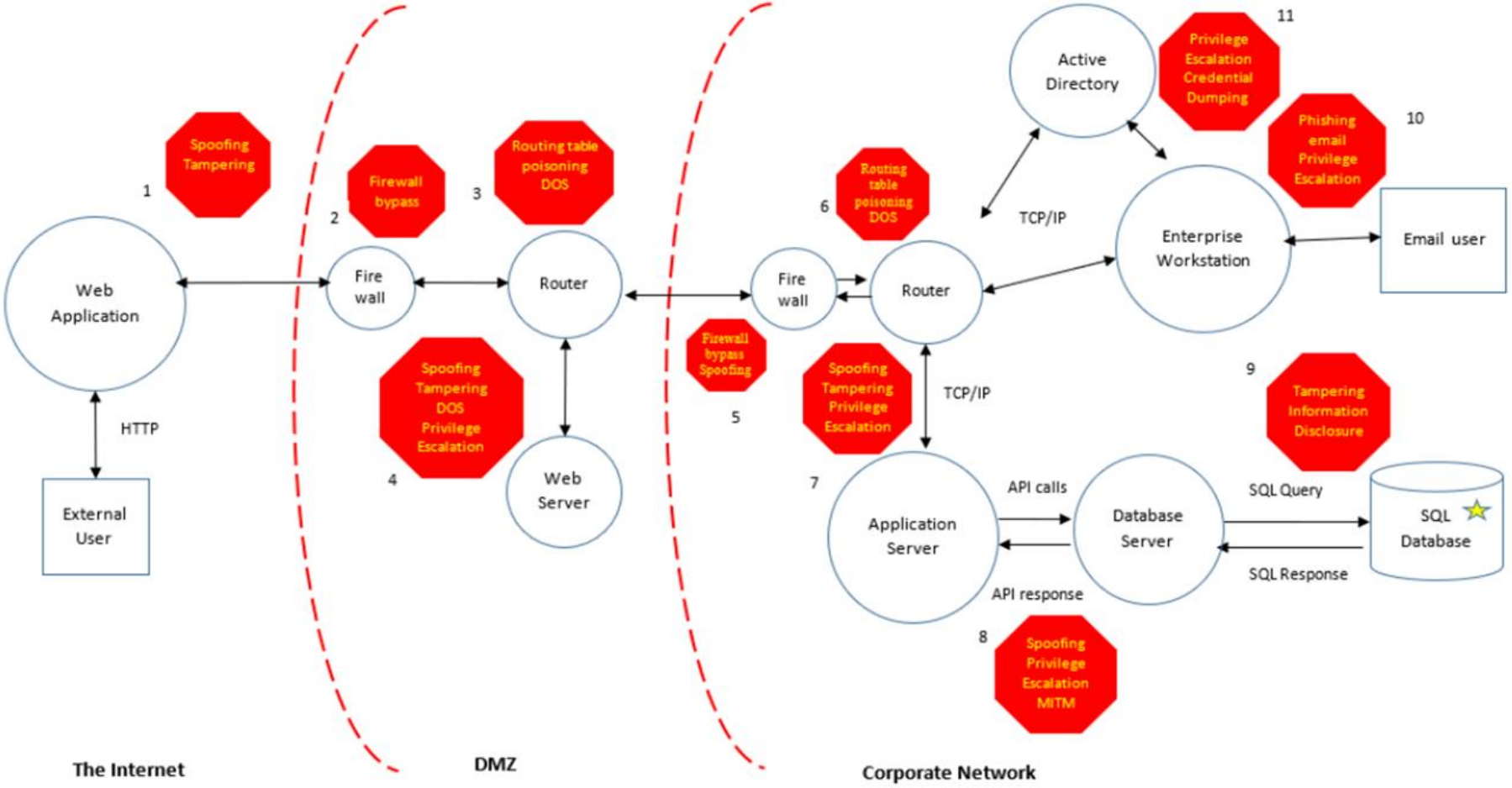

Table of Attack

S/N.	Point of Entry	Threat Actor(s)	Sequence of Attack	Threat Event	Examples
1.1 – Recon	Web Server	APT Group Cyber criminals	2,4	Exploit misconfigured firewall and use network discovery tools on the web server to discover open ports, server OS, application name and version number, and map out any additional host names or subdomains. Attackers may use such information to exploit known vulnerabilities.	Reconnaissance and scan using NMAP tool's port scanning, ping sweeps, OS fingerprinting
1.2 – Recon	Web Server	APT Group Cyber criminals	4	Perform scanning using web vulnerability scanners to discover known web application and script vulnerabilities on the web server URL. Such information is then used to exploit known vulnerabilities to gain unauthorised access to the files and folders in the system and exfiltrate sensitive information.	Reconnaissance and scan using Nikto vulnerability scanner to extract PHP information to provide more information regarding the installation of PHP versions, paths and installed extensions and scans for vulnerabilities in the web server.
2 - DDOS	Web Server	Cyber criminals	4	Overwhelming the web server with HTTP requests and cause a denial of service (DoS).	HTTP flood attacks e.g., GET or POST requests
3.1 – Web App Vuln	Web Application	APT Group Cyber criminals	1,4,7,8,9 ☆	Intercepting traffic to steal credentials, and use credentials to log in as administrator or user account to access web application and query information from database	Burpsuite

S/N.	Point of Entry	Threat Actor(s)	Sequence of Attack	Threat Event	Examples
3.2 – Web App Vuln	Web Application	APT Group Cyber criminals	1,4,7,8,9	Logging into administrator or user account through broken authentication or injection attack, to access web application to query information from database.	Cross-site scripting (XSS), SQL injection. Broken authentication – insecure coding of application authentication; credential stuffing.
3.3 – Web App Vuln	Web Application	APT Group Cyber criminals	1,4,7,8,9	Exploiting vulnerable code, dependencies, or integrations to execute a remote request from the server and extract sensitive data.	XML External Entities (XXE) Attack. Exploit vulnerability in XML processors to upload XML or include hostile content in an XML document
4.1 - Malware in email	Enterprise Workstation	APT Group Cyber criminals	10,11,7,8,9	Installing Remote Access Trojan (RAT), which establishes a backdoor, attacker pivots to AD and dumps credentials. He further moves laterally to exfiltrate valuable information from database.	Phishing, whaling, spear phishing.
5.1 -MITM	Enterprise Workstation	Insider	7,8,9	Man-in-the-middle attack to manipulate communications between the application and the database.	Manipulating API application message headers or body
6.1 -Tamper	Enterprise Workstation	Insider	8,9	Attacker escalates privilege to tamper with database, resulting in modification or unavailability of data.	Stealing of privileged database admin credentials to modify or drop tables.
7.1	Firewall/Router	APT Group Cyber criminals	2,3	Attacker performs DOS attack on the router and cause disruption to the network.	Exploit misconfiguration in firewall or flood the router with ICMP packets
7.2	Router	Insider	6	Attacker modifies the routing table of the router to conduct man-in-the middle attacks	ARP poisoning. Sending malicious/wrong routing table updates to redirect the traffic, or delete the configuration of the router

A2 – Industrial Control System (ICS) – According to Purdue Model

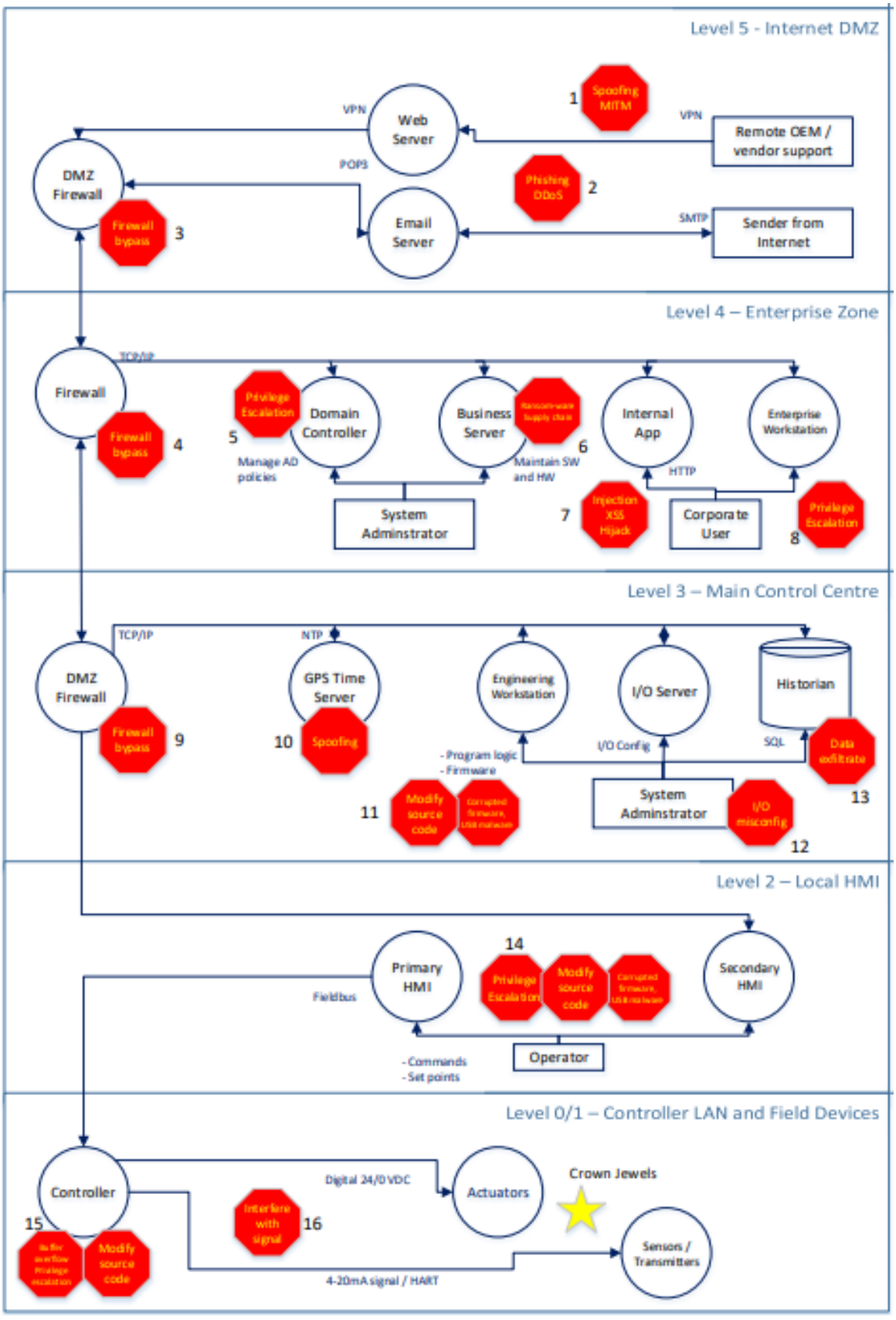

Table of Attack

S/N.	Point of Entry	Threat Actor(s)	Sequence of Attack	Threat Event	Examples
1.1	Email	Cyber criminals	2	Flooding email server causing DDoS. Phishing email harvesting credentials.	Estonia 2007
1.2	Email	Cyber criminals	2,5,6	Ransomware through email, encrypting files needed for enterprise operations	EKANS, WannaCry
1.3	Email	APT actor	2,8,11,15 ☆	External party establishes call back to C2 server and injects RAT for keylogging or disrupting physical operations	Duqu, Havex (Dragonfly), Triton
2	Domain Controller	Privileged insider	5,8,11 ☆	Compromised AD allows privileged escalation from enterprise workstation pivoting into engineering workstation disrupting physical operations	DCShadow
3	Business server	Supplier	6	Supply chain flaw in hardware compromises enterprise business operations.	NotPetya/M.E. Doc
4	Application	Insider	7,6	Script injection through application compromises enterprise business operations.	ICSA-19-050-04 (CWE-79)
5	Enterprise workstation	Insider	8,11,15 ☆	Privileged escalation from enterprise workstation allows pivoting into engineering workstation disrupting physical operations.	BlackEnergy
6	Time server	Contractor Insider	10 ☆	Asynchronous clock disrupts time dependent protocols and operations	ICSA-14-345-01 CVE-2015-7871
7	Engineering workstation	Contractor Privileged Insider	11,15 ☆	Modified code, corrupted firmware or malware through removable device disrupts physical operations.	ICSA-16-138-01A
8	I/O server	Contractor Privileged Insider	12 ☆	Misconfigured I/Os disrupt communication and operations	ICSA-15-337-03

S/N.	Point of Entry	Threat Actor(s)	Sequence of Attack	Threat Event	Examples
9	Historian	Privileged insider	13	Extraction of critical operational data.	CVE-2015-7903
10	Operator HMI	Insider	14,15 ☆	Privileged escalation from operator HMI allows disruption of operations.	Cisco Blogs (Aug 2009): Lessons from an Insider Attack on SCADA Systems - GhostExodus
11	Controller	Contractor Insider	15 ☆	Modified programme logic disrupts physical operations.	Maroochy Shire sewage spill
12	Signal communication	Insider	16 ☆	Jamming or intercepting of signals disrupt communication and operations.	Electromagnetic interference (EMI); Radiofrequency interference (RFI)
13	Firewall	Insider	3 or 4 or 9	Firewall misconfiguration allowing malicious communication to pass bypass.	Communication between unauthorised network ports. Unauthorised application calls
14	Remote access	APT actor	1,15 ☆ (direct access)	MITM hijacks remote session to communicate with controller.	Remote vendor access

ANNEX B – OTHER METHODS

- ***Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE)***

OCTAVE resources.sei.cmu.edu/library/Asset is a risk-based strategic assessment and planning method for cybersecurity.

- ***Common Vulnerability Scoring System (CVSS)***

CVSS cvss.specification/document is a threat scoring system developed by the National Institute of Standards and Technology (NIST) and maintained by the Forum of Incident Response and Security Teams (FIRST).

- ***Process for Attack Stimulation and Threat Analysis (PASTA)***

PASTA processattack.simulation.threatanalysis.pdf is the Process for Attack Simulation and Threat Analysis and is a risk-based threat modelling methodology aimed at identifying viable threat patterns against an application or system environment.

- ***Security Cards Security Cards***

securitycards.cs.washington.edu is a brainstorming toolkit, which encourages creative thinking about cybersecurity threats. It uses a deck of 42 cards in four dimensions (suites) to facilitate threat discovery.

- ***Attack Trees***

Introduced by Bruce Schneier, an Attack Tree is a conceptual hierarchy that shows how a system may be attacked. Attack Trees provide a holistic way to analyse the security of a system and its processes. At the top of the diagram is a root node, defining the attacker's ultimate objective e.g., obtain administrative privileges. Below that are leaf nodes that describe different ways of achieving the ultimate objective defined in the root node. Each leaf node then becomes a subsidiary objective of the root node with child nodes that further expand how the attacker can achieve the objective.

QUERIES AND FEEDBACK

Questions and feedback on this document may be submitted to:

CII_feedback@csb.gov.bn