PART TWO – STRIDE

1. Data flows

Data flow	***Type of Threat***	***Description***	***Mitigation***
1	Tampering	Attacker modifies links on Legacy Web Application to redirect traffic to his own machine	Approve modifications to Legacy Web application before they go live. Prevents unauthorized modifications.
2	Tampering	Attacker creates process on API server that impersonates another process but functions like a fork bomb, slowing down the data flow.	Prevent duplicate processes from being created on the API server. Authorize all changes to API server. Reduces risk of malicious processes being created.
3	Tampering	Attacker modifies the code of the service application to make maintenance crew members do things they did not intend to do	Keep logs to see what changes to the code has been made. Prevents unauthorized changes to code.
4	Tampering	Attacker sends deauthentication frame to Wi-Fi router and sets up his own Wi-Fi router for the maintenance crew to connect to, so that he can read the traffic	Use Wireless Intrusion Prevention Systems to detect evil-twin access points and get rid of them – this makes sure that maintenance crew members won't unknowingly connect to a non-genuine Wi-Fi router
1	Information Disclosure	Attacker modifies links on Legacy Web Application to redirect traffic to his own machine	Approve modifications to Legacy Web application before they go live. Prevents unauthorized modifications that will put information at risk.
2	Information Disclosure	Packet sniffing tool used to snoop traffic	Use SSL encryption when interacting with legacy web application – ensures that packets won't be in plain-text and cannot be viewed
3	Information Disclosure	Attacker watches DNS records to see what information maintenance crew members are sending to the service application	Encrypt all communication using HTTPS, RSA keys, etc. So that intruders cannot see confidential information.
4	Information Disclosure	Attacker sends deauthentication frame to Wi-Fi router and sets up his own Wi-Fi router for the maintenance crew to connect to, so that he can read the traffic	Use Wireless Intrusion Prevention Systems to detect evil-twin access points and get rid of them – this makes sure that maintenance crew members won't unknowingly connect to a non-genuine Wi-Fi router

1	Denial-of-service	Attacker sits in airport and spams requests to legacy web application, slowing down traffic.	Implement third-party DoS protection such as Cloudflare. This detects IPs where DoS attacks are occuring and block them, resuming traffic flow as normal.
2	Denial-of-service	Attacker creates process on API server that impersonates another process but functions like a fork bomb, slowing down the data flow.	Implement a fork-bomb detection system that shuts off any fork bombs in process. Prevents fork bombs from succeeding.
3	Denial-of-service	Ping flood: the attacker, who is a maintenance crew member, sends lots of ping requests to the legacy web applciation without waiting, overloading the requests for packets and slowing down the application	Disallow ping requests from everyone except those authorized to do so, or limit the size of ping requests. Will slow down floods.
4	Denial-of-service	SYN flood attack: the attacker sends a TCP packet with a SYN flag. The servcie applciation sends a SYN-ACK packet to acknowledge the packet, but the attacker does not send an ACK packet back, forcing the application to keep sending SYN-ACK packets and deny service to other users while this happens.	Keep logs to see what changes to the code has been made. Prevents unauthorized changes to code.

2. Data stores

Data stores	***Type of Threat***	***Description***	***Mitigation***
Data centre database	Tampering	Attacker uses an SQL injection to modify a database	Sanitize, control and restrict inputs so that only authorized people can change the database. Reduces risk of unwanted changes to database
Tablet database	Tampering	Maintenance crew member clicks link on an email on tablet, and the email downloads an app that contains malware that modifies the tablet database.	Restrict the apps that are downloaded on tablets so that only organizational or trusted apps are able to be downloaded. That way malware won't be downloaded and databases can remain intact
Data centre database	Information Disclosure	Attacker sends a query to the database that displays all its contents	Encrypt data using different encryption algorithms so that the data being shown can only be read with a key. Prevents unauthorized people from viewing contents of database
Tablet	Information Disclosure	Maintenance crew member downloads an app	Implement an intrusion detection system to detect

Data stores	***Type of Threat***	***Description***	***Mitigation***
database		from the app store, not knowing that it is malware. The app sends everything in the database to the owners of the app.	abnormal activity and alert senior members. Reduces chance of attacks succeeding.
Data centre database	Denial-of-service	Attacker sends thousands of requests to the database, slowing down the performance of the database and eventually making it stop responding.	Use a third-party DoS protection service like Cloudflare, as preventing these attacks can be tricky on its own. This will detect which IP addresses are being used and block them, reducing load on the database.
Tablet database	Denial-of-service	Attacker adds lots of fake entries to the data base, so much so that the tablet cannot handle the memory and crashes.	Limit amount of entries being put into the database every day. Prevents unauthorized tampering and doesn't slow down tablet.

3. Processes

Processes	***Type of Threat***	***Description***	***Mitigation***
Legacy Web Application	Spoofing	Attacker sets up his own Legacy Web Application and redirects network traffic to use his application	Use Intrusion Prevention Systems to detect evil-twin access points and get rid of them – this makes sure that employees won't unknowingly connect to a non-authentic application
API Server	Spoofing	Attacker creates a process that has the same name as another process on the API server so that the attacker's process gets executed instead	Ban certain names for files, processes, etc. That share the same or similar names to existing processes. Reduces risk of a malicious process being executed this way.
Service Application	Spoofing	Attacker spoofs the IP that the service application is on, making maintenance crew members send information somewhere else.	Use Ipv6 instead of Ipv4 and use authentication keys to validate IP addresses. Reduces risk of IP spoofing and makes sure the correct application is being used/
Legacy Web Application	Tampering	Attacker sets up his own Legacy Web Application and redirects network traffic to use his application	Use Intrusion Prevention Systems to detect evil-twin access points and get rid of them – this makes sure that employees won't unknowingly connect to a non-authentic application
API Server	Tampering	Attacker creates a process that has the same name as another process on the API server so that the attacker's process gets executed instead	Ban certain names for files, processes, etc. That share the same or similar names to existing processes. Reduces risk of a malicious process being executed this way.
Service	Tampering	Attacker modifies the code of the service	Keep logs to see what changes to the code has been made.

Processes	***Type of Threat***	***Description***	***Mitigation***
Application		application to make maintenance crew members do things they did not intend to do	Prevents unauthorized changes to code.
Legacy Web Application	Repudiation	Attacker uses a XSS attack to steal an office member/maintenance crew's login cookie to access the application	Add SSL encryption to encode cookie. This way, the attacker cannot use it.
API Server	Repudiation	Attacker notices there are no logs of what commands are sent to the server and does attacks knowing that it will not be found.	Implement logs to check what is being sent to server. Can use to check suspicious activity.
Service Application	Repudiation	Attacker corrupts logs to cover up his other attacks.	Control who has access to logs. Prevents unauthorized usage.
Legacy Web Application	Information Disclosure	Attacker sets up his own Legacy Web Application and redirects network traffic to use his application	Use Intrusion Prevention Systems to detect evil-twin access points and get rid of them – this makes sure that employees won't unknowingly connect to a non-authentic application
API Server	Information Disclosure	Attacker sends request to API that results in a detailed error that exposes how the server works internally.	Keep error messages non-verbose. This way you do not give away thing that an attacker can use against you.
Service Application	Information Disclosure	Attacker takes advantage of the fact that the application is running on root, and steals the password file by typing its url by brute-force.	Use a separate account for the application and give it access to only public files. This will avoid any chance of traversing the file directory to get to the root user.
Legacy Web Application	Denial-of-service	HTTP flood attack: attacker spans HTTP GET and POST requests to the application, making it use up all its resources and slowing down its performance.	Keep track of abnormal activity to deny service to certain Ips. Will at least reduce the time of an attack.
API Server	Denial-of-service	Ping flood: the attacker sends lots of ping requests without waiting, overloading the requests for packets and slowing down the API server.	Disallow ping requests from everyone except those authorized to do so, or limit the size of ping requests. Will slow down floods.
Service Application	Denial-of-service	SYN flood attack: the attacker sends a TCP packet with a SYN flag. The servcie applciation sends a SYN-ACK packet to acknowledge the packet, but the attacker does not send an ACK	Upgrade application to handle lots of SYN packets, and use cryptographic hashing to verify ACK packets. This way the application won't have to constantly send ACK packets back to consume resources.

Processes	***Type of Threat***	***Description***	***Mitigation***
		packet back, forcing the application to keep sending SYN-ACK packets and deny service to other users while this happens.	
Legacy Web Application	Elevation of privilege	Office member uses a Broken Access Control attack on a member with more privileges, inserting their login credentials in a URL request, thus gaining the privileges on their account.	Deny all actions by default, require higher approval for any sensitive action. This way, every action requries an extra step to be done, reducing the risk of attacks succeeding.
API Server	Elevation of privilege	The attacker sends an API request with garbage characters, causing the API server to not handle the error correctly and give the attacker elevated access.	Regular code reviews to check if invalid inputs are handled correctly. Makes sure that bad inputs do not compromise the system.
Service Application	Elevation of privilege	Attacker modifies the code of the service application to make maintenance crew members do things they did not intend to do	Keep logs to see what changes to the code has been made. Prevents unauthorized changes to code.

4. Interactors

Interactors/ Actors	***Type of Threat***	***Description***	***Mitigation***
Office Support	Spoofing	Attacker makes fake LinkedIn account of an office support member, messages a manager to request information.	Request employees to contact each other in person or through email only. Reduces risk of harm happening through impersonation.
Maintenance Crew	Spoofing	Attacker dresses up in maintenance crew uniform, shows up to place of work and starts doing things.	Require everyone to sign in and out, do a head count to see who should and shouldn't be here. Reduces risk of intruders.
Office Support	Repudiation	Attacker sets email display name as office support member and says they have been locked out of their original email account.	Request employees to raise issues in person only. Reduces risk of harm happening through impersonation.
Maintenance Crew	Repudiation	Attacker, who is a maintenance crew member, claims not to have received instruction manuals.	Set up logs to keep track of what is being sent where to prevent repudiation.