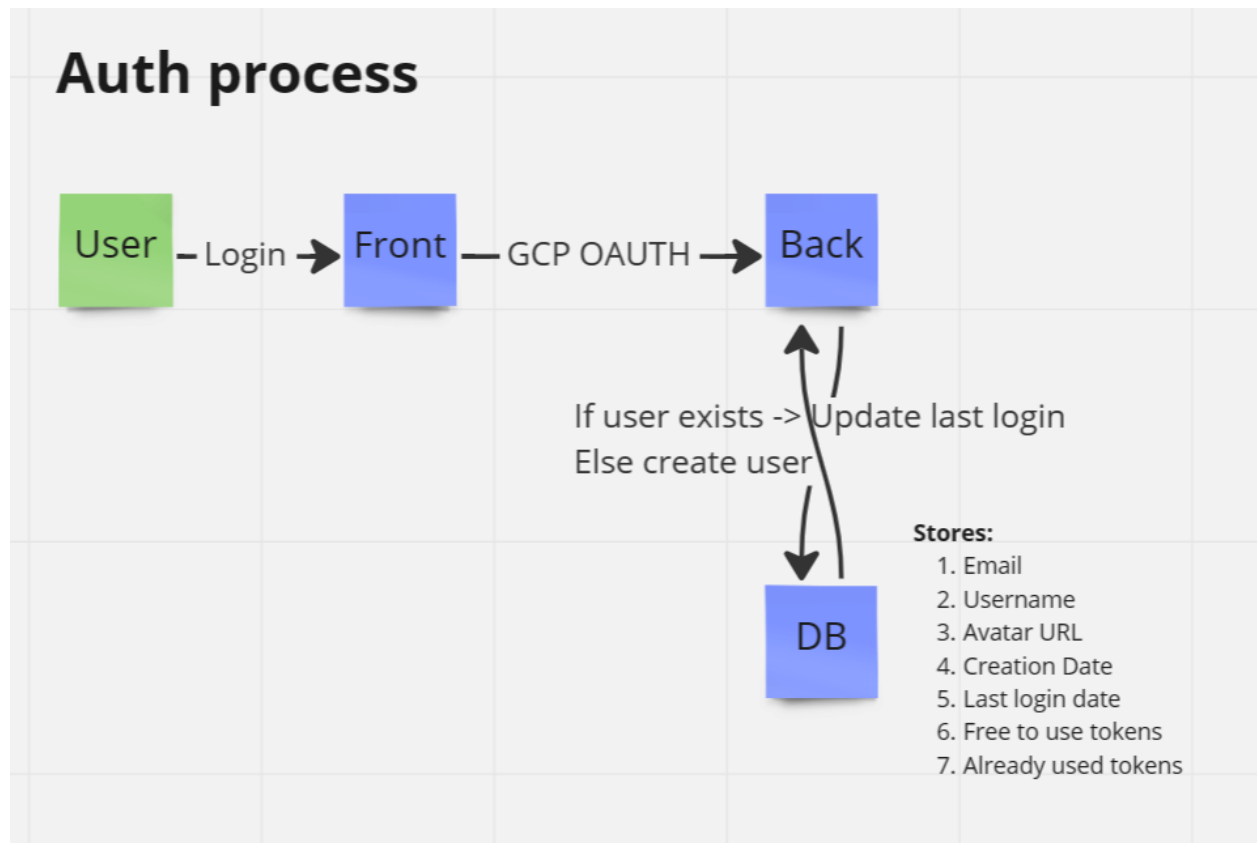


# TIPL Security Assessment

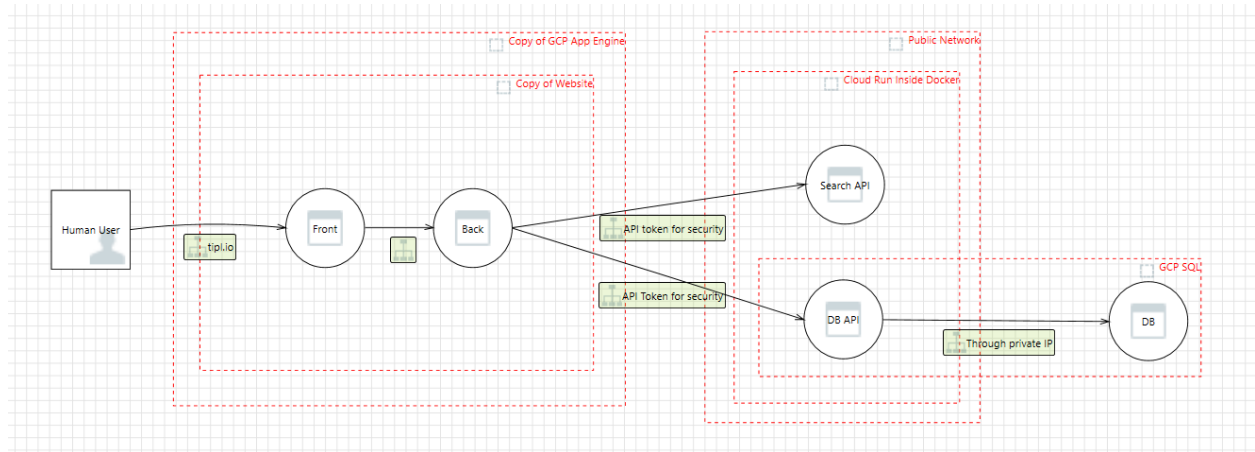
## What is TIPL?

TIPL is a search engine powered by the powerful GPT AI. Its main feature is its ability to search a stock and provide the user with the most relevant information available, giving the user a deeper insight into the stock. It also lets the user know whether they should invest or not.

## How does it work?



# Data Flow Diagram



## Overview

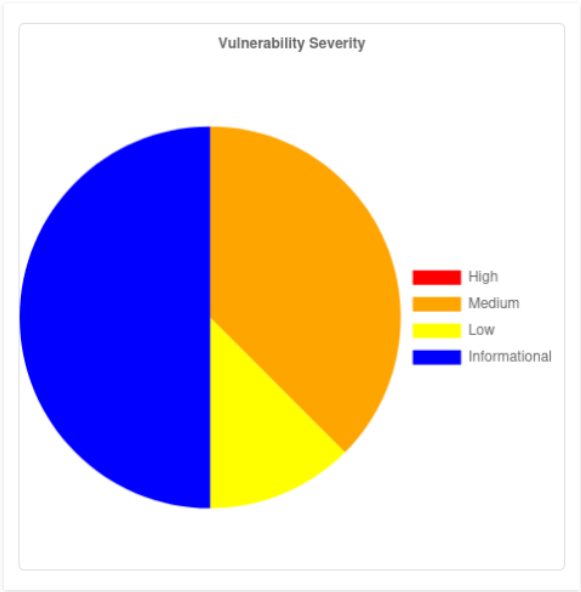
For my Capstone project, I did a complete security assessment for TIPL. The assessment is divided in the following steps:

1. Vulnerability Scan: I utilized a wide variety of tools to scan TIPL for any possible security flaws.
2. Penetration Testing: After running multiple scans, i attempted to exploit any vulnerabilities I could find
3. Secure Code Review: I went through all our lines of code and tried to identify any security flaws.
4. Google Cloud Platform: This is one of our most important assets, since everything is being hosted here. Any breaches to GCP could be fatal to our company.

# Vulnerability Scan

## OWASP ZAP

		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	Medium	0 (0.0%)	2 (25.0%)	1 (12.5%)	0 (0.0%)	3 (37.5%)
	Low	0 (0.0%)	0 (0.0%)	1 (12.5%)	0 (0.0%)	1 (12.5%)
	Informational	0 (0.0%)	0 (0.0%)	3 (37.5%)	1 (12.5%)	4 (50.0%)
	Total	0 (0.0%)	2 (25.0%)	5 (62.5%)	1 (12.5%)	8 (100%)



# MEDIUM LEVEL

## Content Security Policy Header not set

1. Risk: Medium
2. Confidence: High
3. Description: Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
4. Solution: In order to fix this vulnerability, I configured the web server to return the Content Security Policy HTTP header.
5. Reference:
  - a. [https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing\\_Content\\_Security\\_Policy](https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy)
  - b. [https://cheatsheetseries.owasp.org/cheatsheets/Content\\_Security\\_Policy\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html)
  - c. <http://www.w3.org/TR/CSP/>
  - d. <http://w3c.github.io/webappsec/specs/content-security-policy/csp-specific-ation.dev.html>
  - e. <http://www.html5rocks.com/en/tutorials/security/content-security-policy/>
  - f. <http://caniuse.com/#feat=contentsecuritypolicy>
  - g. <http://content-security-policy.com/>

## Missing Anti-clickjacking Header

1. Risk Medium
2. Confidence Medium
3. Parameter: X Frame Options
4. Description: The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
5. Solution: This issue was resolved by setting the CSP header in the previous vulnerability.
6. Reference:
  - a. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options>

## X-Content-Type-Options Header Missing

1. Risk: Low
2. Confidence: Medium
3. Description: The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
4. Solution: I ensured that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.
5. References:

- a. <http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx>
- b. <https://owasp.org/www-community/Security-Headers>

## NMAP SCAN

IP:188.114.96.5

Nmap is an open-source tool that allows us to map networks, scan for vulnerabilities, port-scanning, and much more. In this case, I utilized nmap -Sv -Sc to try and find open ports and information about the services that are running on the system. Here is a breakdown of the command:

- Nmap will run a scan on the target, allowing us to see the open ports. Without additional arguments, it looks like this

```
$ nmap www.tipl.io
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-06 07:12 EDT
Nmap scan report for www.tipl.io (188.114.97.5)
Host is up (0.038s latency).
Other addresses for www.tipl.io (not scanned): 188.114.96.5 2a06
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
2000/tcp  open  cisco-sccp
5060/tcp  open  sip
8008/tcp  open  http
8080/tcp  open  http-proxy
8443/tcp  open  https-alt
```

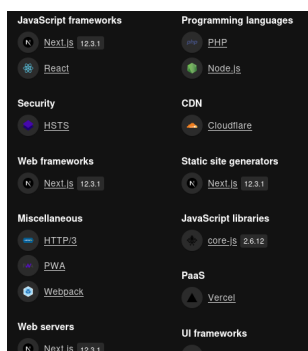
- 
- -sV option enables version detection, which means that Nmap will attempt to determine the version of the services running on the target. This will allow us to find known vulnerabilities for that specific version.
- -sC option enables the default script scan, in which Nmap will run a set of pre-configured scripts capable of finding common vulnerabilities as well as providing additional

information about the services running on the target. The output was rather large, so i won't display all of it.

```
L-$ nmap -sV -sC 188.114.96.5
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-06 08:11 EDT
Stats: 0:00:00 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.50% done
Stats: 0:01:01 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 57.14% done; ETC: 08:13 (0:00:42 remaining)
Nmap scan report for 188.114.96.5
Host is up (0.038s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE        VERSION
80/tcp    open  http           Cloudflare http proxy
|_http-server-header: cloudflare
|_http-title: Site doesn't have a title (text/plain; charset=UTF-8).
443/tcp   open  ssl/http       Cloudflare http proxy
|_tls-alpn:
|_  h2
|_  http/1.1
|_http-server-header: cloudflare
|_ssl-date: TLS randomness does not represent time
|_http-title: 403 Forbidden
|_tls-nextprotoneg:
|_  h2
|_  http/1.1
|_ssl-cert: Subject: commonName=sni.cloudflaressl.com/organizationName=Cloudflare, Inc./stateOrProvinceName=Cloudflare, Inc./countryName=US
|_Subject Alternative Name: DNS:cdnjs.cloudflare.com, DNS:*.cdnjs.cloudflare.com, DNS:sni.cloudflaressl.com
|_Not valid before: 2022-08-03T00:00:00
|_Not valid after: 2023-08-02T23:59:59
2000/tcp  open  cisco-sccp?
5060/tcp  open  sip?
8008/tcp  open  http
|_fingerprint-strings:
|_  FourOhFourRequest:
|_    HTTP/1.1 302 Found
|_    Location: https://:8015/nice%20ports%2C/Tri%6Eity.txt%2ebak
|_    Connection: close
|_    X-Frame-Options: SAMEORIGIN
|_    X-XSS-Protection: 1; mode=block
|_    X-Content-Type-Options: nosniff
|_    Content-Security-Policy: frame-ancestors
|_    GenericLines, HTTPOptions, RTSPRequest, SIPOptions:
|_    HTTP/1.1 302 Found
|_    Location: https://:8015
```

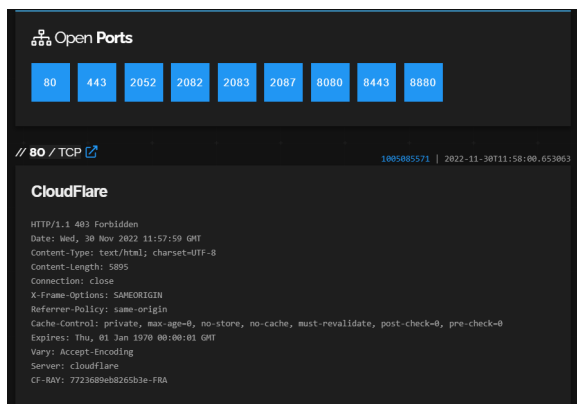
## Server Software and Technology Found

By utilizing Wappalyzer, a web browser extension that can identify the technologies used by a website, I found the following information regarding the site. Knowing what technologies the website is using allows us to identify potential vulnerabilities or attack vectors that may be found in those technologies.



# Shodan Network Scanner

Shodan is a useful tool for pen testing web applications because it can help identify potential vulnerabilities and attack vectors. It can also identify all the services and devices exposed to the internet, including web servers and databases, search for devices and services with known vulnerabilities, identify misconfigured or poorly secured systems, and map the attack surface of the target web application. These capabilities allow us to identify potential entry points and attack vectors that can be used to compromise the application.



# OWASP TOP 10

OWASP Top 10 is a list of the most critical security risks to web applications. The list is compiled by the Open Web Application Security Project (OWASP), a nonprofit organization that focuses on improving the security of software.

The OWASP Top 10 list provides a ranking of the top 10 security risks based on their prevalence and potential impact on web applications. The list is updated periodically to reflect new threats and changes in the security landscape.

These are the vulnerabilities on the list that could possibly affect TIPL.



# Cryptographic Failure

Cryptographic failure occurs whenever a web application exposes sensitive data due to a weak or a non-existent cryptographic algorithm.

At TIPL we circumvent this issue by utilizing google authentication, so no user passwords or PII (personal identifiable information) besides emails and usernames is being stored within our databases.

The only things our databases store are emails, number of tokens, and previous queries that the users have executed. On top of that, our SQL databases are encrypted with AES-256 bit encryption (regarded as one of the most reliable encryption algorithms), and we utilize a symmetric key which is stored safely. We also have a valid https certificate, which covers the following points:

- Confidentiality through encryption of all data transmitted between the user's browser and the web server
- Integrity by ensuring that the data transmitted between the user's browser and the web server is not tampered with during transmission.
- Authentication since a valid HTTPS certificate provides authentication by verifying the identity of the web server to the user's browser
- Trust by ensuring that the web server is authentic and trustworthy.

Below is a picture of our certificate.

General

Details

Issued To

Common Name (CN)

Organization (O)

Organizational Unit (OU)

\*.tipl.io

<Not Part Of Certificate>

<Not Part Of Certificate>

Issued By

Common Name (CN)

Organization (O)

Organizational Unit (OU)

GTS CA 1P5

Google Trust Services LLC

<Not Part Of Certificate>

Validity Period

Issued On

Expires On

Wednesday, March 29, 2023 at 2:51:42 AM

Tuesday, June 27, 2023 at 2:51:41 AM

Fingerprints

SHA-256 Fingerprint

SHA-1 Fingerprint

04 9E 6F 4C 3A 3F AE 04 71 87 30 F3 83 72 F8 74  
04 E6 92 43 40 98 D7 D5 B9 83 3B 31 E1 71 F7 28

FC D7 A8 4A 4E 98 3C 40 89 92 E7 C3 6C 07 7D A7  
CA 0F 30 F4

## Authentication and Identification Failure

Users sign in through Google Authentication, which is a widespread login method that is highly secure, and Google handles the data storage as well as the encryption. Since we don't directly store any passwords, using Google Authentication significantly reduces our workload. In the scenario that a hacker manages to acquire the google sign on for a user, they wouldn't be able to do much besides having access to the user's tokens and previous queries.

## Security Misconfiguration

In order to reduce risks, we only included relevant functionalities within the site. In order to ensure that every component is always up to date with the latest security patch, we implemented an automated update process. This helps us against any previously unpatched vulnerabilities.

The website had a vulnerability that was discovered earlier, where detailed error codes were being returned, potentially revealing sensitive information that could be exploited to gain unauthorized access to the site. This has been patched and a generic error message is now displayed.

Another vulnerability allowed attackers to list all the directories available within the site. This could be dangerous if an attacker manages to find a root directory which has no security configurations. This is not an issue since we ensured there are no directories that can be accessed through the site that may be utilized to further establish a foothold. However, it is still a risk that we have to address.

## Software and data Integrity Failure

Using public libraries provides many benefits, but it also paves the way for possible vulnerabilities. Malicious libraries, as well as attacks through existing libraries are always a possibility. This is why we removed unused libraries, as well as thoroughly research the libraries we choose.

Running constant updates to ensure the libraries are up to date is mandatory, since a new vulnerability might appear at any time. Using digital signatures or similar mechanisms to verify the software or data is from the expected source and has not been altered ensures we only use trustworthy software.

# Exploitation

## Cross Site Scripting (XSS)

Cross site scripting is a type of attack where the hacker inputs a malicious script into the site in an attempt to have it executed. Usually the best places to attempt this are search boxes and contact us request forms. Tipl only has the search function, which could be used to execute an XSS attack. However, since we implemented input sanitization, the code removes any unwanted characters before running the query. This way, hackers are unable to run scripts on our site. Here is an example.

tipl

🔍 <script>alert("hi")</script>

🕒 Loading... (takes around 7 seconds)

If the script had been successful, an alert would have popped up at the top of the screen.

# Denial of Service (DoS)

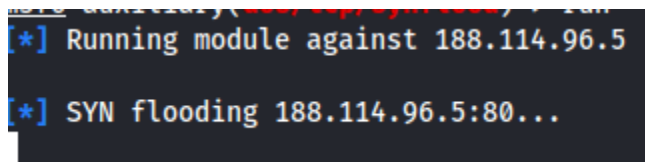
A Denial of Service (DoS) attack is a type of cyber attack in which an attacker attempts to make a network or website unavailable to its intended users. This is typically done by overwhelming the target system with a flood of traffic or requests, which can cause the system to become slow or unresponsive.

In this case, I chose to attempt a basic synflood attack. This targets the TCP protocol, which is used to establish connections between devices over the internet. In a SYN flood attack, the attacker sends a large number of TCP connection requests (SYN packets) to the target device, with fake or spoofed source IP addresses.

GCP (Google Cloud Platform, which is thoroughly covered later) has security implementations in place in order to fend off such types of attacks.

GCP offers automated DDoS protection for all projects, including protection against SYN flood attacks, through the Google Cloud Armor service. Additionally, GCP provides various tools and features such as Cloud Load Balancing to protect against SYN flood attacks and other DDoS attacks. GCP also offers third-party DDoS protection solutions that can be integrated with GCP services.

From the attacker perspective, executing a synflood through the terminal looks like this:

A terminal window with a dark background and light blue text. The text shows a command being executed: "[\*] Running module against 188.114.96.5" followed by "[\*] SYN flooding 188.114.96.5:80...".

```
[*] Running module against 188.114.96.5  
[*] SYN flooding 188.114.96.5:80...
```

According to the GCP analytics, it received over 20,000 requests, which it was capable of shrugging off with ease. When we tried using the site at the same time as the attack was taking place, we didn't notice any loss of functionality, and the site ran at its usual speed.

# Directory Traversal

Directory traversal is a technique in which the attacker attempts to access files and directories that are not meant to be accessed. This can be accomplished by manipulating the path used to

access files by adding special characters like "." or "../" in order to traverse up the directory tree and access files outside the current directory.

In this case, to facilitate the process, I utilized a directory buster. It is a tool specialized in making directory traversal much faster. Directory busters work by sending a large number of HTTP requests to the target server, using a wordlist of common file and directory names as input. The tool then analyzes the HTTP responses to determine if the requested file or directory exists on the server.

There are no real ways of keeping an attacker from utilizing these tools. At TIPL, we covered this vulnerability by ensuring there were no accessible directories of files through the website besides the necessary ones, as well as strict access control. We also have a robust firewall which constantly monitors for any malicious traffic.

# Google Cloud Platform

## **Why did we choose GCP in terms of security?**

It is designed to provide security through the entire information processing lifecycle. This infrastructure ensures secure deployment and storage of services and data, prioritizing end-user privacy. It enables secure communication between services, maintains confidentiality and privacy of customer communication over the internet, and ensures safe operation by administrators.

## Roles

**Security leader:** It's essential to comprehend Google's core principles for cloud security and their practical application to secure your organization's deployment.

**Security engineer:** needs to understand how to configure and operate multiple different security controls so that they seamlessly interact with each other.

**Risk and compliance officer:** it's vital to be familiar with the controls offered by Google Cloud to ensure they align with your business requirements. Must know how to deploy these controls automatically and monitor any control drift or areas that require additional attention to meet the business's regulatory needs.

We adopted three security principles that are the core of Google security

- Executing defense in depth, at scale, by default.
- Adopting the BeyondProd approach to infrastructure and application security.
- De-risking cloud adoption by moving toward a shared fate relationship.

### Defense in depth

The principle of defense in depth emphasizes the importance of having multiple layers of protection between a potential attacker and a valuable target. Additionally, it's crucial for security measures to be easily scalable and enabled by default in order to strengthen overall security.

To ensure maximum data protection, multiple layers of defense are implemented by default, including policies and controls configured across various services such as networking, encryption, IAM, detection, logging, and monitoring. For instance, in a production project, data is automatically secured by three levels of network protection: VPC segmentation, VPC service perimeters, and firewall rules and policies. Additionally, IAM and access context levels are employed to further enhance data protection through multiple levels of access control.

## **BeyondProd**

The BeyondProd framework operates on the principle of zero trust, recognizing that perimeter-based security models are insufficient in preventing system breaches. By breaking down large, monolithic applications into microservices, BeyondProd increases segmentation and isolation, thereby reducing the affected area in the event of a breach. This approach also enhances operational efficiency and scalability.

In the BeyondProd model, security is not an add-on feature, but is instead holistically integrated into the system. While edge network protection is still important, it is not the primary defense point. The framework operates on the principle of no inherent mutual trust between services, and only trusted machines are used to run code with known provenance.

Logical choke points are employed for consistent policy enforcement across services, such as for authorized data access. Change rollout is simple, automated, and standardized, while isolation between workloads is strictly enforced and continuously monitored.

## **Shared fate**

Google Cloud Platform has transitioned from a shared responsibility model to a shared fate model, which significantly reduces the workload for us. In the shared responsibility model, TIPL (the customers) and GCP are responsible for different assets, whereas in the shared fate model, the focus is on meeting TIPL's needs. This approach leverages the cloud provider's expertise to help TIPL achieve optimal security in the cloud.

The shared fate approach allows TIPL to manage risks by utilizing secure-by-default configurations, secure blueprints, secure policy hierarchies, consistent availability of advanced security features, and high assurance attestation of controls. By tapping into these resources, TIPL can more effectively address their security needs without becoming overwhelmed by the responsibility.



# Secure Code Review

Secure code review is a crucial part of web app development. At TIPL, we are strong believers in security by design. Ensuring that we prioritize security from the beginning is the best way to handle the situation, since it may be costly and time consuming to implement it later. The main points that have to be covered in secure code review are the following:

## **Failures in identification, authentication and access control**

Mishandling of identification and authentication can prove fatal for any web app out there. If an attacker manages to authenticate as a root user, the damage they can cause could be irreversible. However, with proper access control policies, even if they do manage to acquire root credentials, the damage could be kept to a minimum.

With zero trust policies, each account is limited to things they need and nothing more. This prevents lateral movement and reduces the attack surface. Furthermore, proper implementation of access control can help security personnel pinpoint the source of a breach, and react accordingly.

At TIPL, users login with their google accounts. Since we do not handle the authentication process, hackers would be unable to acquire the users passwords in the scenario that our systems are breached. Google has robust security implementations that make their authentication process extremely secure and hard to circumvent.

Internally, we have a clear separation of duties through role based access control. Within our Google Cloud Platform (which is the true backbone of TIPL), we meticulously assigned roles based on the different areas we need covered. Each service account has limited access based only on what they truly need, nothing more.

## **Potential exposure of sensitive data;**

Exposure of sensitive data is a huge risk that every company has to address no matter what. If a hacker gets their hands on sensitive data, it could spell the end for any company. With all the hardcore regulations and policies the government has in place to ensure companies protect PII, it leaves no choice but to protect it at all costs.

At TIPL, we store as little PII as possible, this being . In case of a breach, the attacker would only acquire the user's emails, which while still being PII, the damage done is kept to a minimum. Furthermore, all of our databases are encrypted with SHA-256, one of the most robust encryption algorithms currently available.

### **Inadequate error handling**

Inadequate error handling by itself does not cause much damage, if any. However, displaying too much information regarding an error could provide an attacker with the necessary data to carry out an attack. By knowing what is causing an error, an attacker can get a better idea of vulnerabilities or angles of attack they can use to exploit other system flaws.

In order to protect our site against this issue, we opted for displaying general error codes with minimal information. This way, an attacker will not be informed on what is causing an error code.

### **Injection flaws**

Like many other vulnerabilities ,injections carry a serious risk. Through injections, hackers can execute a plethora of attacks ,such as leaking private data, modifying information stored in databases, injecting malware, and many other malicious actions. However, we have implemented security measures to defend against this sort of attack.

SQL injections occur when an attacker executes SQL commands in order to exploit the database. At TIPL, we have safeguards against this.

The database is not accessible to the public, since it resides within the GCP. The only way an attacker could access it is by acquiring the service account login in charge of managing the database. We have also implemented security features in order to keep this from happening,

such as strict password policies.

XXS (cross site scripting, which was previously covered) is another type of injection attack where malicious code is executed through functions encountered within the site. As previously stated, we have methods of preventing this, such as input sanitation.'

There are many other types of injection attacks, such as remote code execution, header injections, and many more. Rest assured, we have implemented the necessary precautions to avoid falling victim to such moves.

# Compliance

Compliance is often overlooked when designing a website. Compliance with GDPR regulations and policies is of paramount importance. GDPR has strict requirements for collecting, using, and protecting personal data for individuals located in the European Union. GDPR aims to protect the privacy and rights of users by giving them control over their own personal data. Failure to comply with the GDPR can result in significant fines, reputational damage, and legal action.

## **Privacy policy**

Under GDPR guidelines, any website that recollects user information must have a privacy policy that clearly states the information that is being recollected. It must also include how the data will be processed and who will have access to it. At TIPL, we strongly believe in transparency, and do not sell or distribute private information to anyone. Here are the main points of our privacy policy:

## **Personal data we collect**

We collect only the username and email address of our users. We do not collect any other personal data.

### **How we use your personal data**

We use your personal data to create and manage your account on our website. We do not use your personal data for any other purpose.

### **How we protect your personal data**

We take appropriate technical and organizational measures to protect your personal data from unauthorized access, use, or disclosure. We limit access to your personal data to those employees who have a need to know.

### **Your rights**

You have the right to access, correct, and delete your personal data. You can do this by logging in to your account and making the necessary changes.

## **My Roles**

My main role in the creation of TIPL was as the only member of the purple team. I conducted both offensive and defensive activities.

It was my first time undertaking a project of this type, so it was daunting at first. I decided to split the workload into segments to make things easier to manage. My first priority was secure code development. Through clear communication with the developers, I ensured that they were following security best practices, such as input validation, avoiding dead code, and looking for instances where a buffer overflow may occur. This was crucial, as it is much more efficient and effective to focus on security early on, rather than implementing it later.

The next stage was securing Google Cloud Platform (GCP). This was the backend of our web app, and where all the magic happened.

First, I configured the services we were running on GCP, such as the MySQL database, the firewall, and so on. Although we were not storing any personally identifiable information (PII) other than emails, any and all PII is heavily protected by regulations such as GDPR. I made sure that the database was not accessible to anyone other than the services that needed access (also known as least privilege). By using stored procedures, we greatly reduced the risk of SQL injections. The GCP firewall further prevented unwanted connections, provided filtering, and helped keep our cloud safe and protected.

Finally, setting up the IAM permissions was the most tedious task when it came to GCP. I had to create roles for each service account in order to make sure we were sticking to least privilege. Every service account would have access to whatever service they managed and nothing more. This would make sure that if an account was breached, the attack surface would be greatly limited to only what that account had access to. This made lateral movement much harder as well.

Once GCP was properly configured, I moved to the final phase: securing the web application. Implementing a web application firewall (WAF) was common sense, as this would provide additional filtering and block malicious connections. I ensured that we had input validation in our search bar to avoid any cross-site scripting (XSS) attacks. I also implemented X-XSRF-Tokens to help mitigate the risks of cross-site request forgery (CSRF) attacks.

After setting up the security for the web app, I attempted a penetration test. The site held up marvelously. I ran a denial-of-service (DoS) attack that peaked at over 20,000 requests per second, and it had no problems handling the load. I also ran multiple fuzzing tests, such as looking for possible XSS vulnerabilities or the opportunity for a buffer overflow, among other tests.

Here are some general tips in order to enjoy TIPL safely:

1. Always keep your passwords secure. Make sure they are at least 8 characters long and include uppercase letters, lowercase letters, numbers, and special characters.

2. If possible, use a VPN to keep your data encrypted and safe.
3. Ensure that the TIPL website you are using has a valid HTTPS certificate. This will prove that the site is reliable and that the communication is being encrypted.
4. Always double-check the URL to avoid typosquatting.