Threat Modeling for Careem System

Hajar Barrak Alharbi, Azaah Ali Kabbas, Nahla Aljojo

University of Jeddah, College of Computer Science and Engineering, Department of Cybersecurity , Jeddah, Saudi Arabia
University of Jeddah, College of Computer Science and Engineering, Department of Cybersecurity , Jeddah, Saudi Arabia
University of Jeddah, College of Computer Science and Engineering, Department of Information System and Technology, Jeddah, Saudi Arabia

Abstract
Mobility Service Provider (MSP), is system that connects customers with drivers through websites and mobile applications. So, working on Careem system as examples of transport companies in Saudi Arabia, also the first online platform in the Middle East. In this report, we provide Threat Modeling for Careem system, by understanding the logical architecture of the system to specify the attacks that faces this system and how to mitigate or apply control over these attacks. Then, explain Careem system security requirements.

Key words:
Architecture, Trust boundaries, Attack surface, Threat agent, Security requirement.

1. Introduction

The Transport Network Company (TNC), sometimes known as the Mobility Service Provider (MSP), is an organization that connects customers through websites and mobile applications with drivers that provide these services. Careem is one of the most popular examples of transport companies in Saudi Arabia. Careem is a Car Booking Service or Transportation network company with presence in almost 14 countries and 40 cities in Middle east, South Asia and Africa. Careem have very modern and advanced infrastructure and use Amazon as their Service provider. As per Careem management, 100 % of Careem infrastructure is deployed on AWS (Amazon Web Services). Careem uses AWS Elastic Beanstalk, Amazon S3,
and Amazon EC2 to host its mobile app, as well as Amazon RDS for databases and Amazon DynamoDB to store locations of its drivers ("Careem Case Study - Amazon Web Services (AWS)," 2019)

2. Logical Architecture of Careem Car Booking Service

Before we explained logical Architecture of Careem System, we present the general context diagram of car booking service as shown in figure 1 below.

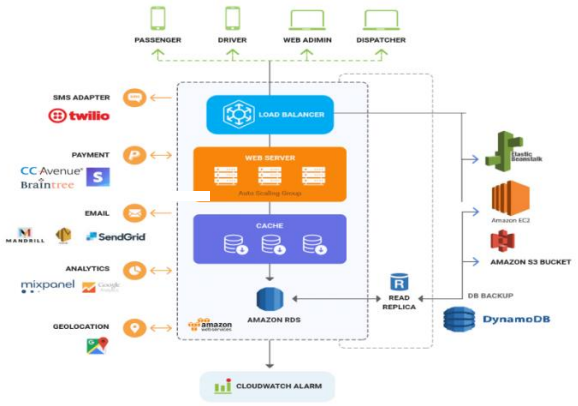

Fig. 1 Careem Context Diagram (Rahul Mathur, 2019)

In figure 2 we explained a logical architecture of the Careem car booking service with details. We will further divide the architecture based on trust boundaries ("Careem Case Study - Amazon Web Services (AWS)," 2019; Myagmar, Lee, & Yurcik, 2005; Chapple, James Michael Stewart, & Gibson, 2018, Pages, 28-35).
Careem is a Car booking service that provides online car booking service to customers by installing 'Careem' app in their mobiles. The app needs to be connected to internet and then to Careem servers which are mainly deployed on Amazon Web Services (AWS). Users can also login to their accounts/ profiles through internet and can add card details and access personal details like rides, card details and others user private information.
The external interactors of the Careem system includes:

- **Driver/ Pilot**: The drivers/ piolets register themselves by registering them with Careem office and then installing a Driver application in their mobiles. They can also log-in to their accounts through internet and see their details. the drivers have read access to financial databases for viewing payment history and bonuses. So, they can exploit any vulnerability to access private and also financial information.
- **Customers/ Careem users**: They can request for booking a Car by installing a Careem app in mobile and can also log-in to Careem through

Manuscript received January 5, 2020
Manuscript revised January 20, 2020

browser. They can view their details and can also enter their Credit card details and other private info. Means they have read access to Careem infrastructure.

- **Management interface/ stuff**: can be initiated by the management/ admin to the core Careem services, databases for management and operations purposes like administration, databases roles and authorizations, log analysis, validation and verifications etc.
- **Core Careem App**: The communication between the Cloud services like S3, EC2 with the DynamoDB and RDS databases and providing services ("Careem Case Study - Amazon Web Services (AWS)," 2019). The main business of Careem can be divided into certain functions: (1) Driver/ Pilot registration for booking (2) Customer installs the app and request for Car booking at a certain place (3) Careem Cloud applications select a Pilot for the customer through some processes (4) customer payment through Credit card or cash (5) Payment gateway for transaction/ electronic transactions (6) System management

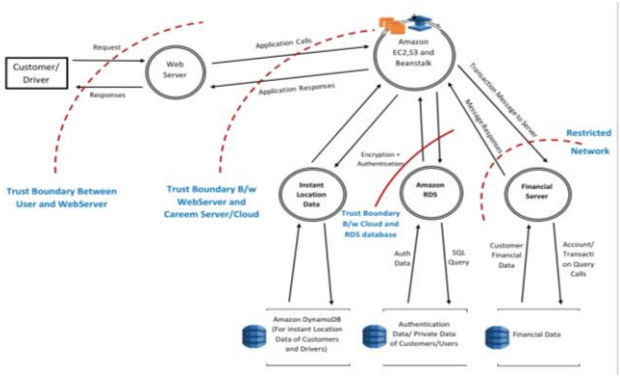

Fig. 2 Careem Logical Architecture

3. Decomposing the Architecture based on Trust Boundaries

The next step is to decompose the architecture to gain a greater understanding of the logic of the services and applications running and its operations with respect to the threats and risk because of external and internal elements (Chapple, James Michael Stewart, & Gibson, 2018, Pages, 28-35). The decomposed domains can be subroutines, or services of the main service. As we are working on Careem car booking service so our decomposition will be based on the tasks performed at the backend when a Driver/ pilot, Customer/ user or an administrator access Careem from outside. Furthermore, we will also decompose based on the trust boundaries w.r.t employees' access to private information.

The overall Careem architecture can be divided into several trust domains as given below:

3.1 User/ Customer and Web Server Boundary:

Customers can request for booking a Car by installing a Careem app in mobile and can also log-in to Careem through browser. They can view their details and can also enter their Credit card details and other private info. Means they have read access to Careem infrastructure and also a link to financial database.

Similarly, the drivers/ pilot registers themselves by registering them with Careem office and then installing a Driver application in their mobiles. They can also log-in to their accounts through internet and see their details.

The drivers have read access to financial databases for viewing payment history and bonuses. So, they can exploit any vulnerability to access private and also financial information. Also, their instant location information is saved in a DynamoDB. Similarly, the Driver app from the Mobile also have direct access to Careem AWS services and directly lands there. So, the trust level between customer/ driver and Web Server changes. So, the first boundary to analyze is the User/ Customer and Webserver Boundary, see figure 3.

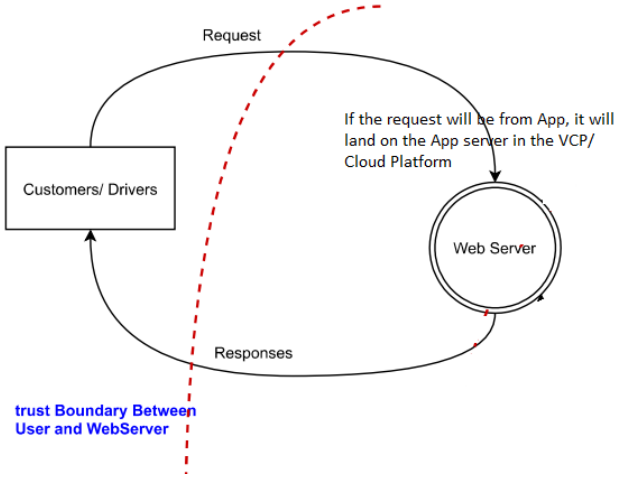

Fig. 3 User/ Customer and Webserver Boundary

3.2 Webserver and infrastructure boundary

The trust level again changes when the request traverses from webserver to internal Careem Cloud systems/ infrastructure. The attacker can bypass some authentication or through different attacks, elevate the privileges and get confidential information, see figure 4.

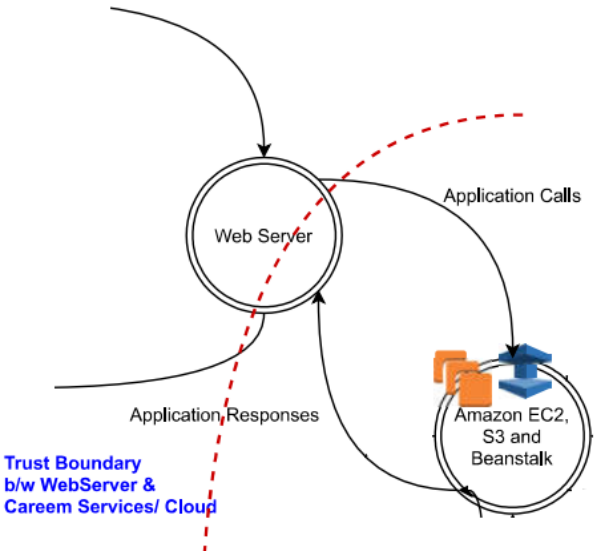

Fig. 4 Webserver and infrastructure boundary

3.3 Cloud Infrastructure and DynamoDB trust boundary:

As we go further in the system, the trust level increases and needs more trust to work. This trust boundary can be exploited by attackers to change the location info or get the location info of someone. If the attacker has access to this place, he can move further to get confidential and private information, see figure 5.

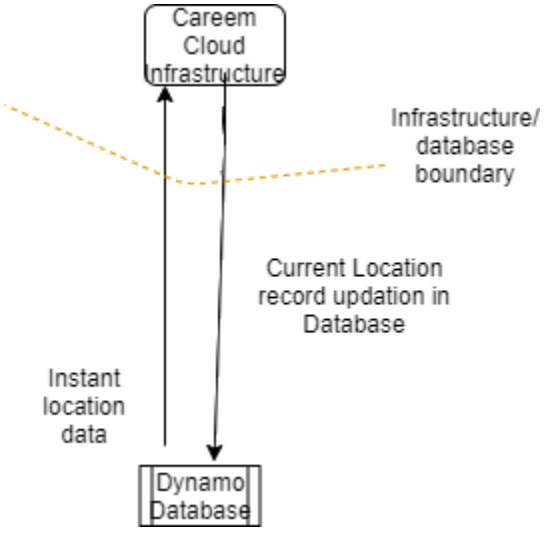

Fig. 5 Cloud Infrastructure and DynamoDB trust boundary

3.4 Trust Boundary between Cloud Infrastructure and Amazon RDS:

This boundary is critical and need to be secure because it is the target of attackers. This boundary contains the actual customers private information, see figure 6.

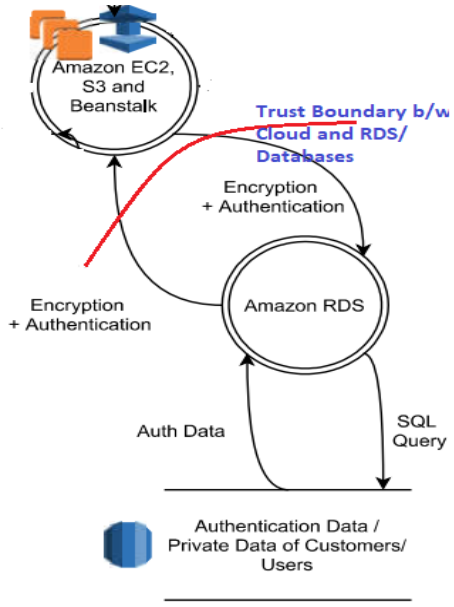

Fig. 6 Boundary between Cloud Infrastructure and Amazon RDS

3.5 Trust Boundary between Cloud Infrastructure and Financial Databases:

This is the most sensitive trust boundary that needs more care as this is the actual target of attackers. This boundary contains the customers financial information, see figure 7.

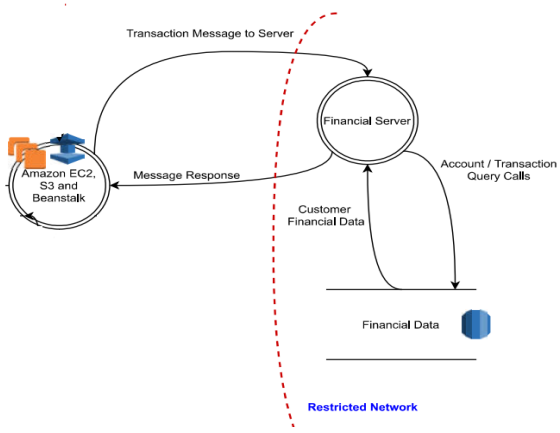

Fig. 7 Trust Boundary between Cloud Infrastructure and Financial Databases

3.6 Trust Boundary between Private Databases and Employees:

As employees have access to private information in their domain. This boundary also needs care. As human is the weakest link and most of the attacks are from insiders, so threats facing this trust boundary is more than other layers because of legitimate access to information, see figure 8.

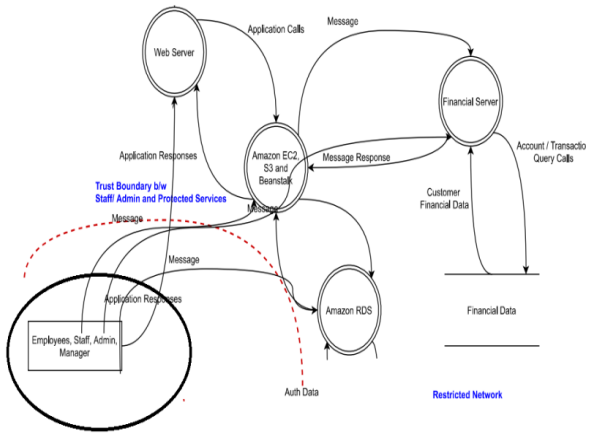

Fig. 8 Trust Boundary between Private Databases and Employees

4. Apply attack methods for expected goals to the attack surfaces

As Careem have web presence through App as well as through web servers and payment gateways so there are a lot of threats that can target Careem. We have focused on OWASP top 10 and also some generic threats that can be applied to Careem system as given below in Table 1, 2.

The attack(s) pertains to the following boundaries:

1. User and Webserver
2. Webserver and Careem Private Cloud infrastructure

Table 1: attack methods for expected goals to the attack surfaces (Schoenfield 2015)

#	Specific Attack	System Objectives	Attack Surface	Threat Agent
1	SQL and Command Injection attacks	unauthorized disclosure of data complete host takeover denial of access Data loss and corruption	Web applications/ HTTP	Cyber Criminals/ Competitors
2	Broken Authentication attacks through session hijacking and MITM	-Identity theft -disclosure of highly sensitive unauthorized information -spoofing	Web applications/ HTTP	Cyber Criminals/ Competitors
3	Cookie stealing through XSS	-Identity theft -disclosure of highly sensitive unauthorized information -spoofing	Web applications/ HTTP	Cyber Criminals/ Competitors
4	Sensitive Data Exposure - Execute a MITM attack, or steal clear text data from the server, while in transit, or from the user's client through different means	-Compromise of PII (personal identifiable information), personal data and records, user credentials and credit card details	Web applications/ HTTP	Cyber Criminals/ Competitors
5	XML External Entities (XXE) - Exploiting vulnerable XML by uploading XML and incorporating malicious content in the XML document, or exploiting a vulnerable XML code	-extract private data from target systems, -execute a remote request from the web server, -perform a denial-of-service attack, or orchestrate and execute other attacks	None Available	Cyber Criminals/ Competitors
6	Buffer overflow attack	- Disclosure of highly sensitive and private unauthorized info. - Data corruption	Web servers, web applications, Databases front/ backend	Cyber Criminals/ Competitors
7	Cross-Site Scripting (XSS)	- Execute scripts in victim's browser -Hijack user sessions by cookie stealing -Credential stealing -Deface websites through persistent XSS -Redirect the victim to malicious websites	Web servers, web applications, HTTP	Cyber Criminals/ Competitors
8	Exposed direct Object/ Directory References	-disclosure of highly sensitive unauthorized information	Web servers, web applications, HTTP responses	Cyber Criminals/ Competitors
9	Denial of Service attack by flooding the target web server with unnecessary traffic	-denial of access and service for legitimate users	Web servers, web applications, HTTP	Cyber Criminals/ Competitors

The attack(s) pertains to the following boundaries

1. Staff/ Employees and Careem Protected Servers, Private and Financial Databases
2. Trust boundary between Careem Private Cloud infrastructure and Protected Databases

Table 2: attack methods for expected goals to the attack surfaces (Schoenfield 2015

#	Specific Attack	System Objectives	Attack Surface	Threat Agent
10	**Spoofing another user identity by illegally obtaining certificate/ stolen identity**	-identity theft - disclosure of highly sensitive - unauthorized information - complete host/ account takeover -denial of access	Web servers, web applications, HTTP, Databases front/ backend	Cyber Criminals/Disgruntled employees
11	**Disgruntled Insiders steal Customer information and publish/ sale it**	- Disclosure of highly sensitive and private unauthorized information of customers and company employees - Financial gain	Web servers, web applications, Databases front/ backend	Malicious/Disgruntled employees
12	**Careem web servers and databases are hacked and information is leaked by exploiting some vulnerability**	- Disclosure of highly sensitive and private unauthorized information of customers and company employees - Financial gain	Web servers, web applications, Databases front/ backend	Cyber Criminals/Competitors
13	**Broken Access Control by manually testing and bypassing access control vulnerability**	-Identity theft - unauthorized disclosure of data - complete account takeover -denial of access	Web servers, web applications, HTTP	Cyber Criminals/Disgruntled employees/ Competitors

5. Threats agents who have no attack surfaces

In the current threat landscape, there is always some attack surface exist for realizing any attack. So, the threats we have mentioned above have CAV's exists for all threats. Because there is no such threat mentioned in the above table that don't have any CAV except "XML External Entities (XXE) - Exploiting vulnerable XML by uploading XML and incorporating malicious content in the XML document, or exploiting a vulnerable XML code". Through normal means, realization of this attack is very difficult.

Table 3:Threats agents who have no attack surfaces (Schoenfield 2015)

#	Specific Attack	System Objectives	Threat Agent	Attack Surface
1	Bypass the no-execute page protection policy to execute code	- Execute code of the attacker's choosing within the context of the currently logged user and a running application	Security Researchers	None available
2	XML External Entities (XXE) - Exploiting vulnerable XML by uploading XML and incorporating malicious content in the XML document, or exploiting a vulnerable XML code	-extract private data from target systems, -execute a remote request from the web server, -perform a denial-of-service attack, or orchestrate and execute other attacks	Cyber Criminals/ Competitors	None available

6. Security controls for each attack surface

There is no Silver bullet for security and there is no one control that can thwart a threat. As a defense in depth, several controls can be applied to a single vulnerability and threat combination to prevent from realization. Based on our knowledge and research ("Top 10-2017 Top 10 - OWASP," 2017), we have listed some security controls that we are expecting Careem have developed after analysis.

The list is not exhaustive. There can be other controls that can be also applied to certain threats and attack combination but not obvious to us.

One way to find the implemented controls is manual testing and analysis of the target website and server. Based on our manual analysis of the Careem website and Webserver, we have found out that the website is very good and based on secure standards. They have TLS certificate installed, No direct input point for XSS and SQL injections. Also, the administration panel is not

visible and after a lot of tries, we have not found the admin configuration panel.

As we are not using any method to engage with the target company, so we are not using any vulnerability assessment tool. We have only used two OSINT tools without engagement with the Careem company. One is ImmuniWeb ("ImmuniWeb® - Web and Mobile Security Testing, Application Penetration Testing, Security Ratings," 2019) and the other one is Shodan ("www.careem.com - Shodan Search," 2019). Shodan is a search engine which can be used to find about specific servers on the internet, their open ports, service used and sometimes vulnerabilities, shown in figure 9.

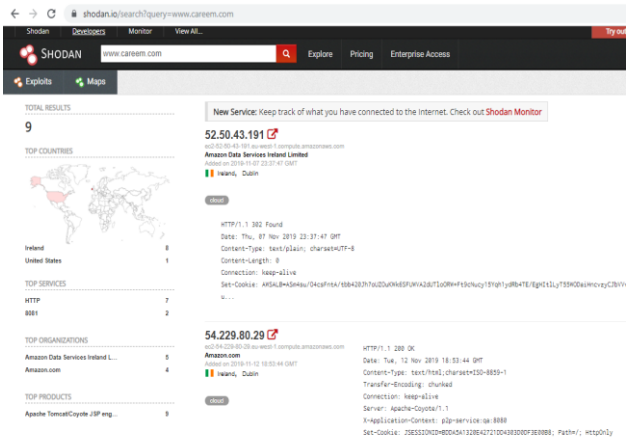

Figure. 9 Shodan Results for Careem – 1

Shodan showing 9 results for Careem means there are nine different subdomains of Careem. We can search for all of them by just clicking the IP. All the IPs are from Amazon Web services, see figure 10.

Fig. 10 Shodan Results for a Single IP/ Webserver

The above result is showing that the server 54.229.80.29 is using only 3 Open Ports and their services. But we are not seeing any specific vulnerability attached to this webserver. So, we can stat that based on this result the Web server is in a good state.

Now let's search for the other one in figure 11.

Fig. 11 Results for 52.50.43.191

The above snapshot is the result for Webserver 52.50.43.191. But with Open Ports and services we are also seeing so vulnerabilities. Below in figure 12 is a list of vulnerabilities of the above webserver.

Fig. 12 Vulnerabilities in Webserver 52.50.43.191

We search for each and every CVE, we have found that they are deals with using older versions of Software's

packages. If they upgrade their server's application and frameworks like Php, Apache, Tomcat, JSP etc., they can be secure.

We have also tested Careem on ImmuniWeb. The results are given below in figures 13, 14, 15.

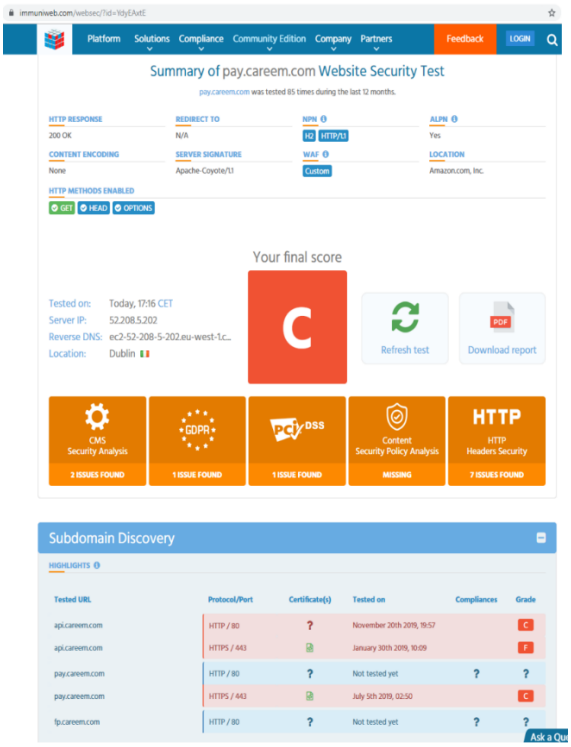

Fig. 13 Scanning Careem Webservers for Vulnerabilities

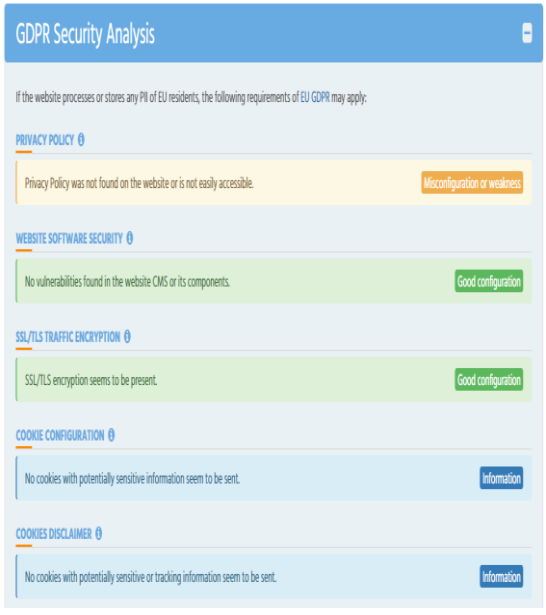

Fig. 14 Careem GDPR Security Analysis

Fig. 15 Careem PCI-DSS Security Analysis

From the above and below results, we can see that Careem is mostly in compliance with GDPR and PCI-DSS, which means that they have implemented most of the controls related to GDPR and PCI-DSS.

Another benchmark to test Careem for implemented controls is PCI-DSS (PCI-DSS, 2018). As Careem is also dealing with Credit cards and Payment gateways, so it is obviously PCI-DSS compliance. Without PCI-DSS compliance, credit card processing is not allowed by regulators, so we are expecting that all PCI-DSS based controls are implemented.

The below table is a list of attacks where Careem needs protection. As Careem implemented PCI-DSS, so they have many controls deployed to prevent from most of the mentioned attacks as given below in figure 16.

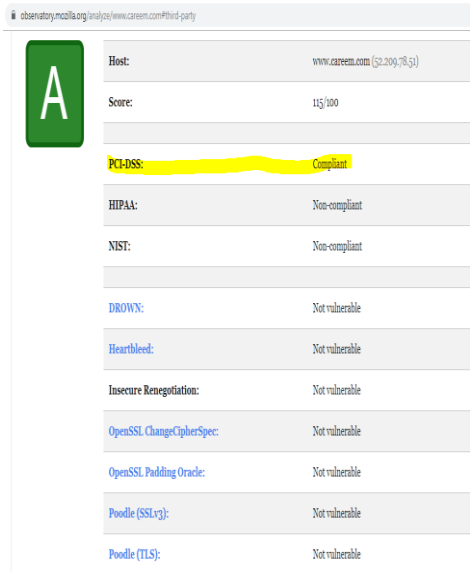

Fig. 16 Careem PCI Compliance

For SQL, LDAP, and Command injection, as they are in compliance with PCI-DSS (PCI-DSS, 2018), so they have implemented Clause 6.5.1 of PCI-DSS for preventing injection flaws. Also, we have tested online that either the site is vulnerable to SQL injection directly, we have got the result that Careem is protected by AWS WAF (Amazon Web Services Web Application Firewall) and no SQL injection results. So, we stat that the control for SQL injection is implemented, see figure 17.

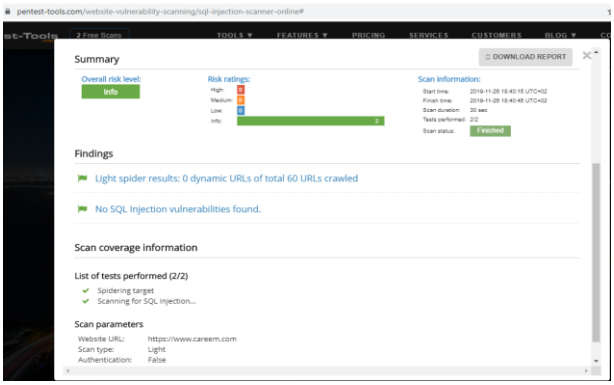

Fig. 17 Testing SQL Injection Vulnerability

In table 4, we list possible existing security controls based on the above analysis that Careem have implemented for attack protection. We are also considering Careem data breech of 2018 which affected 14 million Careem users (Information Security Buzz, 2018). Another document for getting information about the existing controls is the report named "Ride-Sharing Apps and Privacy in Pakistan: A Detailed Study on the Practices of Uber and Careem" (Kamran & Rehman, 2019).

Table 4: security controls for each attack surface (Schoenfield 2015)

#	Specific Attack	Attack Surface	System Objectives	Controls	
1	SQL and Command Injection attacks	Web applications/ HTTP	unauthorized disclosure of data complete host takeover denial of access Data loss and corruption	The major control is to keep the data away from commands and queries that the user is entering. Never take the input directly. Instead used prepared statements. Input validation and sanitization by accepting only authorized characters and sends to interpreter. E.g. escape single quotes Use safe API that does not use the interpreter Character Escaping by only accepting whitelisted characters Use of parameterized queries	
2	Broken Authentication attacks through session hijacking and MITM.	Web applications/ HTTP	identity theft disclosure of highly sensitive unauthorized information spoofing	Deploy multi-factor authentication where required and necessary, especially for staff. This will prevent reuse stolen credentials, brute-force attacks and credential stuffing Encrypting session IDs and not exposing in URLs Never use default credentials in any case. Use strong password and never share with anyone. Rigorous session management implementation with time-based sessions Implementation of time-outs and rotation of session IDs after a successful login and after some time	
3	Cookie stealing through XSS	Web applications/ HTTP	identity theft disclosure of highly sensitive unauthorized information spoofing	The first and foremost preventive measure is to validate and sanitize the input for any script separate the input from the active browser content Encrypt the cookies and sessions makes is invaluable for the attacker Apply Secure and HTTP Only flags set for cookies Use encrypted sessions for communicating identity credentials and information	
4	Sensitive Data Exposure - Execute a MIT attack, or steal clear text data from the server, while in transit, or from the user's client through different means	Web applications/ HTTP	Compromise of PII (personal identifiable information), personal data and records, user credentials and credit card details	Encrypt sensitive data according to classification policy. Apply encryption at rest and in transmission both. Never store sensitive data that no longer required Disable auto-fill form controls to avoid leakage or personal info	
5	Broken Access Control by manually testing and bypassing access control vulnerability	Web servers, web applications, HTTP	Identity theft unauthorized disclosure of data complete account takeover denial of access	Deploy multi-factor authentication. This will prevent reuse stolen credentials, brute-force attacks and credential stuffing Encrypting session IDs and not exposing in URLs Never use default credentials in any case. Use Strong password and never share with anyone Strictly implement password policy Implementation of time-outs and rotation of session IDs after a successful login and after some time	

7. Filter out attack surfaces for which there is sufficient existing protection

In the table 3, we have listed existing security controls and security measures that we have analyzed and Careem claims to have implemented. In Table 5, only shown the areas where Careem needs improvement in terms of security effectiveness and enhancements.

Table 5: attack surfaces for which there is sufficient existing (Schoenfield 2015)

#	Specific Attack	Attack Surface	System Objectives	Threat agent
1	Denial of Service attack by flooding the target web server with unnecessary traffic	Web servers, web applications, HTTP	denial of access and service for legitimate users	Cyber criminals
2	Disgruntled Insiders steal Customer information and publish/ sale it	Web servers, web applications, Databases front/ backend	Disclosure of highly sensitive and private unauthorized information of customers and company employees Financial gain	Cyber criminals

8. Security controls to the set of attack services for which there isn't sufficient mitigation

Security is not only technology issue but it is a management issue (NIST SP 800-50, 2003). The results we have obtained in number (6) shows that on technical side, Careem has implemented most of the security controls related to the above attacks, but they are lacking controls that are related to "disgruntled employees, insiders, Patch management, system and component upgradation, employee awareness, use of components with known vulnerabilities".
Recommended controls (ISO 27001:2013 ISMS; NIST SP 800-53) for attack services for which there isn't sufficient mitigation are list in the tale 6.

Table 6: Security controls to the set of attack services for which there isn't sufficient mitigation (Schoenfield 2015)

#	Specific Attack	Attack Surface	System Objectives	Recommended Controls
1	Denial of Service attack by flooding the target web server with unnecessary traffic	Web servers, web applications, HTTP	-denial of access and service for legitimate users	1. Regularly check and Update the web server for any vulnerability that can crash the server 2. Use load balancers and redundant servers for sustaining the load 3. Deploy anti-DDoS mechanism to thwart DDoS attacks
2	Disgruntled Insiders steal Customer information and publish/ sale it	Web servers, web applications, Databases front/ backend	-Disclosure of highly sensitive and private unauthorized information of customers and company employees -Financial gain	1. Strict implementation of Need to know and principle of least privilege 2. Implement Segregation of Duties for the critical and important positions 3. Implement two-person control for the administration and management of critical services 4. Rigorous monitoring and implementation of Access controls 5. Multi-factor authentication mechanisms for login 6. Access to data should be role based only

9. Security requirements for the system

Security requirement defining what level of security is expected from the system with respect to some type of threat or malicious attack.3 security requirement have things to do with access control, data integrity, authentication, authorization accountability and is related specifically about the kind of vulnerabilities to prevent. The security requirements (ISO 27001:2013 ISMS) that needs to be implemented for effective security on Careem are listed below:

Security Requirement

- Administrative:

1. Conduct vulnerability scans at least monthly
2. Strict authentication and access controls mechanisms should be implemented on management interface, like lockout policy, session expiry, brute force prevention, salting and hashing
3. Regularly check and Update the web server for any vulnerability that can crash the server
4. Multi-factor authentication mechanisms for (admin/ staff) login
5. Only need to know based access control
6. Implement Segregation of Duties for the critical and important positions
7. Implement Principle of least privilege
8. Rigorous monitoring and implementation of Access controls
9. Implement two-person control for the administration and management of critical services
10. Data at Rest must be in encrypted format and only accessible to the person having need to know for his/ work
11. Terminate access or elevated privileges promptly upon role change
12. Control access based on authorization, least privilege, and limited duration
13. Limit admin privileges to owners and to those they specifically authorize
14. Conduct risk assessments annually
15. Conduct a risk assessment soon after a serious IT security incident

- Applications

1. The payment application must validate and verify the correctness of every message received from the payment processing service
2. Avoid dynamic inclusion of software
3. Validate application input
4. Customer Service Identification and Authentication considerations

5. Execute proper error handling
6. Implement two-factor authentication
7. Control access based on roles and the principle of least privilege
8. Encrypt external transmission of data
9. Conduct code security reviews/audits for new or changed applications

- Networking

1. Decoys, Honeypots, and other devices for detection and delay
2. Configure each network boundary control (firewall, switch, router, gateway, or other network control device or system)
3. Implement Multi-Factor Authentication to each component of the System that supports Multi-Factor Authentication
4. Customer financial data between the payment application and the third-party payment processing service will traverse a bidirectionally authenticated VPN
5. Ensure that only personnel assigned to Trusted Roles have access to Secure Zones and High Security Zones
6. Require employees and contractors to observe the principle of "least privilege
7. Restrict remote administration or access to an Issuing System

10. Conclusion

In this report we presented a process of architecture Careem system and threat modeling that begins with architecture, by identifying attack types and attack surfaces, and then applies security controls, or mitigations, to build a defense-in-depth. Finally, we recommended a security requirement for Careem system that should be implemented.

References

[1] Careem Case Study - Amazon Web Services (AWS). (2019). Retrieved November 26, 2019, from Amazon Web Services, Inc. website: https://aws.amazon.com/solutions/case-studies/careem/

[2] Chapple, M., James Michael Stewart, & Gibson, D. (2018). (ISC)2 CISSP certified information systems security professional : official study guide. Indianapolis, Indiana: John Wiley & Sons. Pages, 28-35, Chap 1

[3] Schoenfield, B. S. E., 2015. Securing Systems: Applied Security Architecture and Threat Models. s.l.:CRC Press.

[4] S. Myagmar, A. J. Lee, and W. Yurcik. Threat Modeling as a Basis for Security Requirements (SREIS). In Symposium on Requirements Engineering for Information Security, 2005

[5] Payment Card Industry (PCI) Data Security Standard, v3.2.1, PCI Security Standards Council, LLC

[6] Top 10-2017 Top 10 - OWASP. (2017). Retrieved from Owasp.org website: https://www.owasp.org/index.php/Top_10-2017_Top_10

[7] ImmuniWeb® - Web and Mobile Security Testing, Application Penetration Testing, Security Ratings. (2019). Retrieved November 26, 2019, from Immuniweb.com website: http://www.immuniweb.com

[8] www.careem.com - Shodan Search. (2019). Retrieved November 26, 2019, from Shodan.io website: https://www.shodan.io/search?query=www.careem.com

[9] NIST SP 800-50 (2003). National Institute of Standards and Technology (NIST). Building an information technology security awareness and training program (NIST SP 800-50). Washington, DC: US Department of Commerce.

[10] Kamran, H., & Rehman, Z. (2019). Ride-Sharing Apps and Privacy in Pakistan: A Detailed Study on the Practices of Uber and Careem. Retrieved from Digital Rights Foundation (DRF) website: https://digitalrightsfoundation.pk/wp-content/uploads/2019/01/Careem-Uber-Research.pdf

[11] Information Security Buzz. (2018, April 29). Dubai-Based Ride Hailing App Careem Breached, Affecting 14M | Information Security Buzz. Retrieved November 26, 2019, from Information Security Buzz website: https://www.informationsecuritybuzz.com/expert-comments/dubai-based-ride-hailing-app/

[12] ISO/IEC 27001:2013(en) Information technology — Security techniques — Information security management systems — Requirements

[13] J. T. Force and T. Initiative. 2013. Security and privacy controls for federal information systems and organizations. NIST Special Publication 800-53 (2013).