Cybersecurity

Project 1 Technical Brief

Make a copy of this document before you begin. Place your answers below each question. This completed document will be your deliverable for Project 1. Submit it through Canvas when you're finished with the project at the end of the week.

Your Web Application

Enter the URL for the web application that you created:

```
cyberclouds.azurewebsites.net
```

Paste screenshots of your website created (Be sure to include your blog posts):

```
Screenshots
```

Blog Posts

The Role of Human Factors in Cybersecurity

Keywords: cybersecurity, human factors, security awareness, social engineering

Human beings are often considered the weakest link in cybersecurity, and for good reason. Despite advancements in technology and sophisticated security measures, human error remains a significant contributor to cyber incidents. From falling victim to phishing emails to inadvertently disclosing sensitive information on social media, individuals play a critical role in the cybersecurity landscape. However, recognizing the impact of human factors is the first step towards strengthening defenses and fostering a culture of security awareness within organizations.

In the realm of cybersecurity, social engineering techniques exploit human psychology to manipulate individuals into divulging confidential information or performing actions that compromise security. Whether it's through phishing emails, pretexting, or baiting, cybercriminals leverage human trust and curiosity to gain unauthorized access to systems and networks. Therefore, investing in comprehensive security awareness training programs is essential for equipping employees with the knowledge and skills to identify and respond to social engineering attacks effectively.

Moreover, fostering a culture of cybersecurity within organizations involves more than just training employees; it requires instilling a mindset of vigilance and accountability at all levels. Leadership support, clear policies, and regular communication are crucial components of creating an environment where security is prioritized and integrated into everyday practices. By promoting a shared responsibility for cybersecurity, organizations can mitigate the risks associated with human factors and build resilience against evolving cyber threats.

In conclusion, while human factors pose inherent challenges to cybersecurity, they also present opportunities for empowerment and collaboration. By acknowledging the pivotal role of individuals in safeguarding digital assets and fostering a culture of security awareness, organizations can fortify their defenses and adapt to the ever-changing threat landscape.

Day 1 Questions

General Questions

1. What option did you select for your domain (Azure free domain, GoDaddy domain)?

```
Azure Free Domain
```

2. What is your domain name?

```
CyberClouds
```

Networking Questions

1. What is the IP address of your webpage?

```
20.119.0.47
```

2. What is the location (city, state, country) of your IP address?

```
East US
```

3. Run a DNS lookup on your website. What does the NS record show?

```
C:\Users\efeli>nslookup cyberclouds.azurewebsites.net
Server:  G3100.myfiosgateway.com
Address:  192.168.1.1

Non-authoritative answer:
Name:    waws-prod-blu-517-7cbd.eastus.cloudapp.azure.com
Address:  20.119.0.47
Aliases:  cyberclouds.azurewebsites.net
          waws-prod-blu-517.sip.azurewebsites.windows.net
```

Web Development Questions

1. When creating your web app, you selected a runtime stack. What was it? Does it work on the front end or the back end?

```
PHP 8.2 back end
```

2. Inside the `/var/www/html` directory, there was another directory called assets. Explain what was inside that directory.

```
Inside the assets directory there is a css folder and images folder.
```

3. Consider your response to the above question. Does this work with the front end or back end?

```
Front end
```

Day 2 Questions

Cloud Questions

1. What is a cloud tenant?

```
In cloud computing, a cloud tenant refers to an individual or organization that uses the services and resources provided by a cloud service provider. A cloud tenant can be a user, a group of users, or an entire organization that subscribes to and consumes cloud services such as virtual machines, storage, networking, and applications from the cloud provider. Each cloud tenant operates within its own isolated environment within the cloud infrastructure, ensuring security and privacy for their data and applications.
```

2. Why would an access policy be important on a key vault?

```
An access policy on a key vault is crucial for ensuring that only authorized users or applications can access sensitive cryptographic keys, certificates, and secrets stored within the vault. By defining specific access permissions and roles within the policy, organizations can control who can read, write, or manage these critical assets, thereby reducing the risk of unauthorized access, data breaches, and misuse of sensitive information. Additionally, access policies help enforce compliance with security standards and regulatory requirements by maintaining strict control over who can interact with the key vault's contents.
```

3. Within the key vault, what are the differences between keys, secrets, and certificates?

```
Within a key vault:

Keys:

- Keys are cryptographic objects used for encryption, decryption, and digital signing.
```

- These keys can be symmetric or asymmetric.
- Symmetric keys use the same key for both encryption and decryption, while asymmetric keys use a pair of public and private keys.
- Keys stored in a key vault can be used for securing sensitive data, securing communication channels, and verifying the integrity and authenticity of data.

Secrets:

- Secrets are sensitive data such as passwords, connection strings, API keys, and other confidential information.
- They are typically used for authentication, authorization, and accessing resources.
- Secrets are stored securely within the key vault to prevent unauthorized access.
- Access to secrets is controlled through access policies defined in the key vault.

Certificates:

- Certificates are digital documents that bind a public key to an identity.
- They are commonly used for secure communication over HTTPS, authentication, and code signing.
- Certificates stored in a key vault provide a centralized location for managing and distributing certificates across applications and services.
- Like keys and secrets, access to certificates is controlled through access policies to ensure their confidentiality and integrity.

In summary, keys are used for cryptographic operations, secrets store sensitive data, and certificates bind public keys to identities for secure communication. Each plays a crucial role in securing applications and data within a key vault.

Cryptography Questions

1. What are the advantages of a self-signed certificate?

Self-signed certificates have several advantages, particularly in certain scenarios:

Quick Setup: Self-signed certificates can be generated and deployed quickly without the need for involvement from a certificate authority (CA). This can be useful for testing, development, or setting up internal services where immediate deployment is required.

Cost-Effective: Since self-signed certificates are generated internally and do not involve a third-party CA, they are typically free of charge. This can be advantageous for organizations with limited budgets or for non-production environments where cost savings are important.

Offline Usage: Self-signed certificates can be used in environments that are isolated from the internet or where connectivity to external CAs is not available. This allows for secure communication within closed networks or in situations where internet access is restricted.

Flexibility: Self-signed certificates provide flexibility in terms of customization. Users have full control over the certificate parameters such as validity period, key size, and other attributes. This allows for tailoring certificates to specific requirements without relying on external CA policies.

Encryption: While self-signed certificates do not provide the same level of trust as certificates issued by reputable CAs, they still offer encryption capabilities. They can be used to establish secure communication channels and encrypt data transmitted over networks, providing a baseline level of security.

However, self-signed certificates lack the inherent trust provided by certificates issued by trusted CAs. They may trigger security warnings in web browsers and other client applications, as they are not automatically trusted by default. Additionally, they do not undergo the same rigorous validation processes as CA-signed certificates, potentially making them more susceptible to certain types of attacks, such as man-in-the-middle attacks. Therefore, self-signed certificates are typically more suitable for internal or non-production use cases where these limitations are acceptable.

2. What are the disadvantages of a self-signed certificate?

Self-signed certificates have several disadvantages compared to certificates signed by trusted certificate authorities (CAs):

Lack of Trust: Self-signed certificates are not inherently trusted by web browsers, operating systems, or other client applications. This means users may encounter security warnings or errors when accessing websites or services secured with self-signed certificates, leading to potential distrust and user experience issues.

Security Risks: Self-signed certificates do not undergo the same level of validation and authentication as certificates issued by reputable CAs. This increases the risk of man-in-the-middle attacks, where an attacker could intercept communications by impersonating the server using a self-signed certificate.

Limited Usability: Self-signed certificates may not be compatible with certain applications or platforms that expect certificates to be signed by trusted CAs. This can lead to interoperability issues and limitations in the deployment of services that require broader compatibility.

Manual Management: Managing self-signed certificates requires manual effort, including generating, distributing, and renewing certificates. This can be cumbersome, especially in environments with a large number of certificates or when frequent updates are required.

No Chain of Trust: Self-signed certificates do not establish a chain of trust, as there is no hierarchical relationship between the certificate and a trusted root CA. This makes it difficult for users to verify the authenticity and integrity of the certificate, potentially undermining confidence in the security of the connection.

Not Suitable for Public-Facing Services: Self-signed certificates are generally not recommended for public-facing websites or services that require a high level of trust from users. Using self-signed certificates in such scenarios can negatively impact the organization's reputation and credibility.

Overall, while self-signed certificates offer a quick and cost-effective way to secure internal or non-production environments, they come with significant limitations and security risks that make them unsuitable for many production use cases. Organizations should carefully consider these disadvantages when deciding whether to use self-signed certificates.

3. What is a wildcard certificate?

```
A wildcard certificate is an SSL/TLS certificate that secures a domain and
all its subdomains by using a wildcard character (*) before the domain name,
allowing for simplified management and cost-effectiveness.
```

4. When binding a certificate to your website, Azure only provides TLS versions 1.0, 1.1, and 1.2. Explain why SSL 3.0 isn't provided.

```
SSL 3.0 isn't provided for binding certificates to websites in Azure because
it has known security vulnerabilities, including the POODLE attack, making
it obsolete and insecure compared to newer TLS versions such as 1.0, 1.1,
and 1.2, which offer improved security features and stronger encryption
algorithms.
```

5. After completing the Day 2 activities, view your SSL certificate and answer the following questions:

    a. Is your browser returning an error for your SSL certificate? Why or why not?

```
No, Azure has set up a secure SSL certificate
```

    b. What is the validity of your certificate (date range)?

```
Validity Period

Issued On   Monday, December 18, 2023 at 2:20:21 AM
Expires On  Thursday, June 27, 2024 at 7:59:59 PM
```

    c. Do you have an intermediate certificate? If so, what is it?

```
No
```

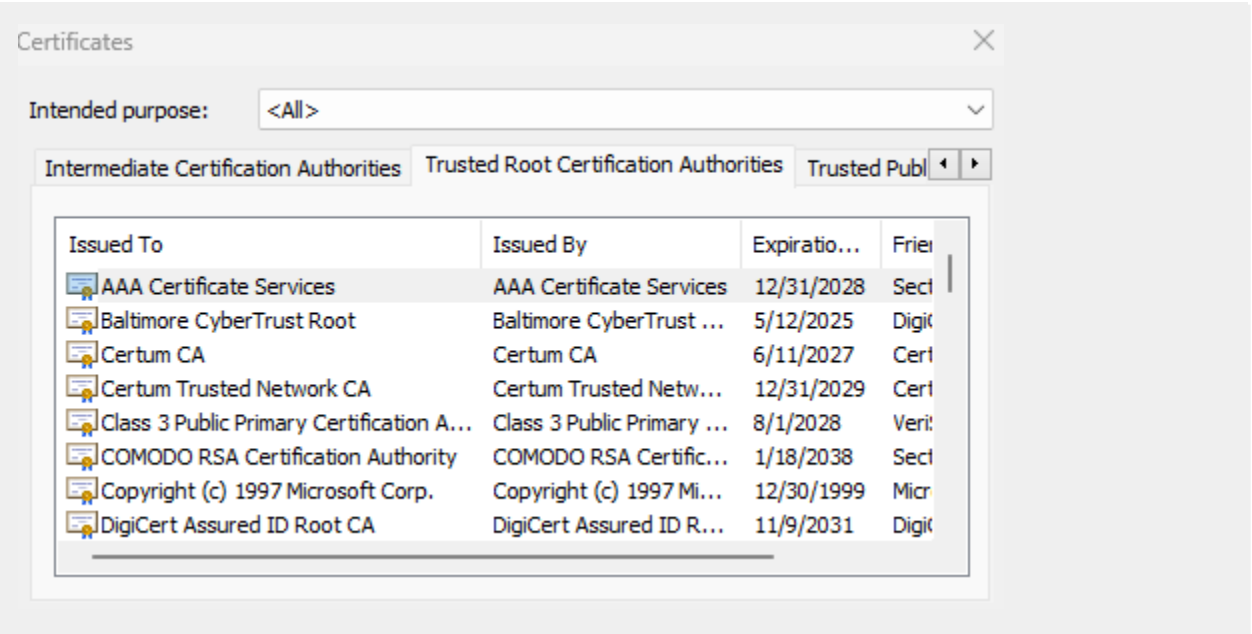
    d. Do you have a root certificate? If so, what is it?

```
Yes, DigiCert Global Root G2
```

e. Does your browser have the root certificate in its root store?

```
Yes
```

f. List one other root CA in your browser's root store.



Day 3 Questions

Cloud Security Questions

1. What are the similarities and differences between Azure Web Application Gateway and Azure Front Door?

```
Azure Web Application Gateway and Azure Front Door are both Azure services
```

that provide features for routing and load balancing traffic to backend resources, but they serve different purposes and have distinct functionalities:

Similarities:

- **Traffic Routing:** Both Azure Web Application Gateway and Azure Front Door allow for routing traffic to backend resources based on various criteria such as URL path, host headers, and geographic location.

- **SSL Offloading:** Both services support SSL termination, allowing them to offload SSL/TLS encryption and decryption tasks from backend servers, which can improve performance and reduce server load.

- **Security:** Both services offer features for enhancing security, such as Web Application Firewall (WAF) capabilities for protecting web applications against common web vulnerabilities like SQL injection and cross-site scripting (XSS) attacks.

Differences:

Use Case:

- Azure Web Application Gateway is primarily designed for load balancing and securing web applications hosted on Azure virtual machines or Azure App Services.
- Azure Front Door is a global, scalable content delivery network (CDN) service designed for accelerating the delivery of web applications and content to users worldwide. It provides features for global load balancing, dynamic site acceleration, and traffic optimization.

Global Load Balancing:

- Azure Front Door provides global load balancing capabilities, distributing user traffic across multiple regions to optimize performance and ensure high availability.
- Azure Web Application Gateway primarily operates within a single region and does not offer built-in global load balancing features.

CDN Functionality:

- Azure Front Door includes CDN functionality, allowing it to cache and

deliver static content closer to end-users, reducing latency and improving overall performance.
- Azure Web Application Gateway does not include built-in CDN functionality.

Layer of Operation:

- Azure Web Application Gateway operates at the application layer (Layer 7) of the OSI model, allowing it to inspect and route traffic based on application-level attributes such as URLs and HTTP headers.
- Azure Front Door operates at the network layer (Layer 4) of the OSI model, allowing it to route traffic based on IP address, TCP/UDP port, and geographic location without inspecting application-layer details.

In summary, while both Azure Web Application Gateway and Azure Front Door offer routing and load balancing capabilities, they target different use cases and operate at different layers of the networking stack, with Azure Front Door focusing on global content delivery and Azure Web Application Gateway focusing on application-level routing and security within a single region.

2. A feature of the Web Application Gateway and Front Door is “SSL Offloading.” What is SSL offloading? What are its benefits?

SSL offloading, also known as SSL termination, is a process in which SSL/TLS encryption and decryption tasks are performed by a frontend device or service (such as Azure Web Application Gateway or Azure Front Door) before the encrypted traffic is forwarded to backend servers. In SSL offloading, the frontend device terminates the SSL/TLS connection from the client, decrypts the traffic, and then forwards it to the backend servers in unencrypted form.

Benefits of SSL offloading include:

Improved Performance: By offloading SSL/TLS encryption and decryption tasks from backend servers to a dedicated frontend device or service, SSL offloading can improve overall application performance and responsiveness. Backend servers can focus on processing application logic and serving content without the overhead of SSL/TLS encryption.

Reduced Server Load: SSL offloading helps reduce the computational burden on backend servers by handling SSL/TLS encryption and decryption centrally at

the frontend. This can lead to better scalability and resource utilization, allowing backend servers to handle more client requests efficiently.

Scalability: SSL offloading enables easier scalability of backend server infrastructure since SSL/TLS termination is handled by the frontend device or service. Additional backend servers can be added or removed without impacting SSL/TLS processing overhead, allowing for dynamic scaling based on demand.

Simplified Certificate Management: With SSL offloading, SSL/TLS certificates are managed centrally at the frontend device or service, reducing the complexity of certificate management on individual backend servers. This simplifies certificate deployment, renewal, and rotation, enhancing operational efficiency and security.

Enhanced Security Inspection: SSL offloading allows frontend devices or services to inspect decrypted traffic for security threats using features such as Web Application Firewalls (WAFs) or intrusion detection systems (IDS). This enables deeper visibility into traffic patterns and better protection against malicious attacks.

Overall, SSL offloading provides several benefits, including improved performance, reduced server load, simplified certificate management, scalability, and enhanced security inspection capabilities, making it a valuable feature for optimizing the delivery and security of web applications and services.

3. What OSI layer does a WAF work on?

A Web Application Firewall (WAF) typically operates at the application layer (Layer 7) of the OSI model. It examines and filters HTTP/HTTPS traffic, analyzing application-level data such as URLs, headers, and payloads to detect and block malicious or suspicious activity targeting web applications.

4. Select one of the WAF managed rules (e.g., directory traversal, SQL injection, etc.), and define it.

Cross-Site Scripting (XSS) is a common web application security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts can execute in the context of the

```
victim's browser, leading to unauthorized actions such as stealing session cookies, redirecting users to malicious websites, or modifying the content of web pages. XSS vulnerabilities typically arise when web applications fail to properly validate and sanitize user input, allowing attackers to inject scripts into HTML, JavaScript, or other client-side code executed by browsers. The Cross-Site Scripting (XSS) managed rule in a Web Application Firewall (WAF) helps detect and block XSS attacks by inspecting incoming HTTP requests and responses for suspicious script content, preventing malicious scripts from being executed in users' browsers and mitigating the risk of XSS exploitation.
```

5. Consider the rule that you selected. Could your website (as it is currently designed) be impacted by this vulnerability if Front Door wasn't enabled? Why or why not?

```
The effect of the chosen rule on the website is determined on the type of the vulnerability. For example, in our case where we made a rule to deny traffic if its not from the specified countries it had nothing to do with a specific vulnerability,
but we could block malicious traffic by identifying malicious IP addresses. Activating the rule through Front Door may give additional security if it solves a specific security issue. Yet, if the web page was not designed to prevent this flaw on its own, it may be vulnerable when Front Door is removed. The rule's effectiveness is determined by the website's intrinsic security mechanisms.
```

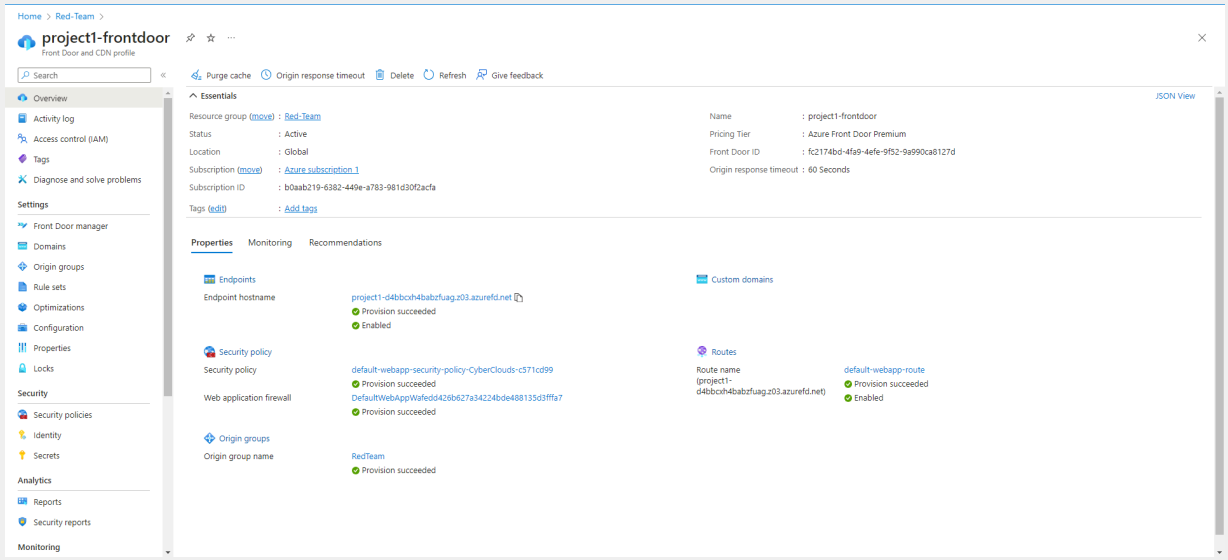
6. Hypothetically, say that you create a custom WAF rule to block all traffic from Canada. Does that mean that anyone who resides in Canada would not be able to access your website? Why or why not?

```
Setting a custom Web Application Firewall (WAF) rule to restrict all Canadian traffic would not necessarily prevent Canadian residents from accessing the site. WAF rules typically rely on IP addresses rather than user physical locations. While it's possible to deny traffic from specific geographic locations based on IP address ranges, this method is not foolproof and may inadvertently block legitimate users who happen to be using VPNs or proxy servers located outside of Canada. Additionally, IP addresses can be dynamic and may not always accurately reflect a user's physical location.
```

7. Include screenshots below to demonstrate that your web app has the following:

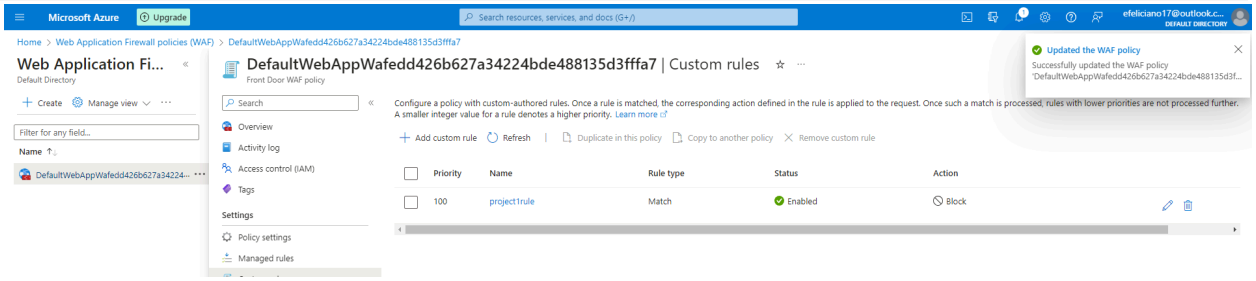
    a. Azure Front Door enabled

Screenshot



b. A WAF custom rule

Screenshot



Disclaimer on Future Charges

Please type "**YES**" after one of the following options:

- ***Maintaining website after project conclusion****: I am aware that I am responsible for any charges that I incur by maintaining my website. I have reviewed the guidance for minimizing costs and monitoring Azure charges.*

  YES
- ***Disabling website after project conclusion****: I am aware that I am responsible for deleting all of my project resources as soon as I have gathered all of my web application screen shots and completed this document.*

  YES