**Akilandeshwari Srinivasan (451036)**

**CLCM3604 – Lab 6**

**Due by – 17th March, 2024**

**Description:**

Implementation of security measures against SQL injection, XSS, and DDoS attacks targeting different tiers of the application, ensuring comprehensive protection and resilience.

**Scenario 1:**

SQL Injection Protection Rule:

Description: This rule is designed to detect and block HTTP requests that contain common SQL injection keywords and symbols in the URI, query string, or request body. By blocking these requests, we prevent attackers from attempting to manipulate the website's database through SQL injection attacks.

Justification: SQL injection attacks pose a significant threat to web applications, allowing attackers to execute malicious SQL queries and potentially gain unauthorized access to sensitive data. By implementing this rule, we mitigate the risk of SQL injection vulnerabilities being exploited.

Rate-based Rule for SQL Injection:

Description: This rate-based rule monitors the rate of HTTP requests containing potential SQL injection payloads from individual IP addresses. If the rate exceeds the defined threshold within a specified time window, further requests from that IP address are blocked. This helps to mitigate the impact of automated SQL injection attacks.

Justification: Rate-based rules provide an additional layer of protection against SQL injection attacks by limiting the number of requests that can be made within a certain timeframe. By enforcing rate limits, we reduce the likelihood of successful SQL injection attacks.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Scenario 2:**

Front-end Web Server:

AWS WAF:

Deploy AWS WAF in front of the Elastic Load Balancer (ELB) serving the front-end web server.

Configure AWS WAF rules to filter and block HTTP requests containing XSS attack payloads.

Utilize AWS Managed Rules for AWS WAF to protect against common XSS attack patterns.

AWS Shield:

Enable AWS Shield Standard for the ELB to protect against common DDoS attacks targeting the front-end web server.

Application Server:

AWS WAF:

Deploy AWS WAF on EC2 instances hosting the application server.

Configure AWS WAF rules to detect and block SQL injection attempts targeting the application layer.

Implement custom rules to filter out malicious SQL injection payloads from HTTP requests.

AWS Shield:

Enable AWS Shield Advanced for additional protection against sophisticated DDoS attacks targeting the application server.

Configure DDoS protection policies and mitigation controls specific to the application layer.

Database Server:

AWS WAF:

Implement input validation and parameterized queries within the application code to prevent SQL injection vulnerabilities.

AWS Shield:

Enable AWS Shield Advanced for the application server, which indirectly protects the database server by mitigating DDoS attacks targeting the application layer.

Explanation and Justification:

Front-end Web Server: AWS WAF is deployed in front of the ELB to filter and block HTTP requests containing XSS attack payloads, while AWS Shield Standard provides basic protection against common DDoS attacks.

Application Server: AWS WAF is deployed on EC2 instances hosting the application server to detect and block SQL injection attempts, while AWS Shield Advanced provides advanced DDoS protection tailored to the application layer.

Database Server: While AWS WAF does not directly protect RDS instances, implementing input validation and parameterized queries within the application code helps prevent SQL injection vulnerabilities. AWS Shield Advanced for the application server indirectly protects the database server by mitigating DDoS attacks targeting the application layer.

**Reference:**

www.google.com