AWS Bedrock Guardrails: Alignment with OWASP Top 10 for LLMs

Executive Summary

This document maps the AWS Bedrock guardrails implemented in our CloudFormation template to the OWASP Top 10 for Large Language Model Applications. The OWASP framework provides a recognized standard for identifying and mitigating security risks in LLM applications. Our implementation addresses all ten vulnerabilities through a defense-in-depth approach with multiple layers of security controls.

OWASP Top 10 for LLMs - Alignment Matrix

1. Prompt Injection

Risk Description:
Attackers manipulate LLMs through crafted inputs, causing unintended actions. This can lead to data exfiltration, social engineering, and other security issues.

Our Controls:

- **Content Filtering Guardrails:** Implements filters to detect and block malicious prompts
- **Least Privilege IAM Policy:** Restricts which models can be accessed and what actions they can perform
- **Approved Prompts Database:** DynamoDB table for storing and managing vetted prompt templates
- **Input Validation:** Lambda function to validate inputs before processing

CloudFormation Resources:

yaml Copy

```yaml
BedrockGuardrails:
  Type: Custom::BedrockGuardrails
  # Implements content filtering through custom resource

ApprovedPromptsTable:
  Type: AWS::DynamoDB::Table
  # Stores vetted prompts to prevent injection

BedrockRestrictedPolicy:
  Type: AWS::IAM::ManagedPolicy
  # Applies least privilege principle
```

2. Insecure Output Handling

Risk Description:
This vulnerability occurs when an LLM output is accepted without scrutiny, exposing backend systems to potential attacks including XSS, CSRF, SSRF, privilege escalation, or remote code execution.

Our Controls:

- **Edge Case Review Function:** Lambda function that validates outputs before they're acted upon
- **Content Safety Guardrails:** Prevents generation of harmful content
- **Standardized Templates:** SSM Parameter Store for safe prompt templates that establish proper boundaries

CloudFormation Resources:

yaml Copy

```yaml
EdgeCaseReviewFunction:
  Type: AWS::Lambda::Function
  # Validates outputs for security issues

BedrockGuardrails:
  # Implements content filtering

SafePromptTemplates:
  Type: AWS::SSM::Parameter
  # Stores standardized templates
```

3. Data & Model Poisoning

Risk Description:
Occurs when LLM training data is tampered with, introducing vulnerabilities or biases that compromise security, effectiveness, or ethical behavior.

Our Controls:

- **Private Network Access:** VPC endpoint provides isolated access to prevent tampering
- **Anomaly Detection:** Edge case review function detects potentially compromised outputs
- **Comprehensive Logging:** CloudTrail integration logs all API activity for audit

CloudFormation Resources:

yaml Copy

```yaml
BedrockVPCEndpoint:
  Type: AWS::EC2::VPCEndpoint
  # Provides private network access

EdgeCaseReviewFunction:
  # Detects suspicious outputs

# Integration with existing CloudTrail
# Records all API calls for audit purposes
```

4. Sensitive Information Disclosure

Risk Description:
LLMs may reveal sensitive information, proprietary algorithms, or confidential details through their output, leading to unauthorized access to sensitive data, intellectual property, and privacy violations.

Our Controls:

- **PII Filtering:** Bedrock Guardrails configuration to mask/block emails, credit cards, SSNs
- **Custom Word Lists:** Blocks organization-specific sensitive terms
- **Monitoring:** CloudWatch alerts for potential information leakage

CloudFormation Resources:

yaml Copy

```yaml
BedrockGuardrails:
  # PII detection and filtering configuration
  # sensitiveInformationPolicyConfig with PII entities
  # customWordLists for organization-specific terms

BedrockLogGroup:
  Type: AWS::Logs::LogGroup
  # Centralized logging for detection
```

5. Supply Chain Vulnerabilities

Risk Description:
Risks introduced by open-access LLMs, fine-tuning techniques, and models sourced from public repositories or collaborative platforms.

Our Controls:

- **Approved Models Only:** IAM policy restricts to specific vetted foundation models
- **Environment Tagging:** Ensures proper model versioning and tracking
- **Comprehensive Auditing:** Integration with CloudTrail for complete visibility

CloudFormation Resources:

yaml Copy

```yaml
BedrockRestrictedPolicy:
  # Restricts to specific foundation models:
  # - anthropic.claude-3-sonnet-20240229-v1:0
  # - amazon.titan-text-express-v1

# Environment tagging enforced throughout resources
# Integration with existing CloudTrail
```

6. Improper Output Handling

Risk Description:
Similar but distinct from #2, this vulnerability focuses specifically on how LLM outputs are used in other systems, which can lead to security breaches when the output isn't properly validated.

Our Controls:

- **Human Oversight:** Edge case review function provides validation for suspicious outputs
- **Alert System:** SNS topic alerts for potentially harmful content requiring review
- **Least Privilege:** LLM access role limited to necessary capabilities only

CloudFormation Resources:

yaml Copy

```yaml
EdgeCaseReviewFunction:
  # Provides review workflow

BedrockAlertsTopic:
  Type: AWS::SNS::Topic
  # Notification system for concerning outputs

BedrockAccessRole:
  # Implements least privilege
```

7. Excessive Agency

Risk Description:
LLM-based systems granted too much functionality, rights, or independence can lead to unforeseen negative results.

Our Controls:

- **Restricted Permissions:** IAM policy limits what the model can access and do
- **Network Controls:** Security group restricts network access
- **Human in the Loop:** Oversight workflow for sensitive operations

CloudFormation Resources:

yaml Copy

```yaml
BedrockRestrictedPolicy:
  # Limits model capabilities

BedrockSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  # Restricts network access

EdgeCaseReviewFunction:
  # Human oversight
```

8. Misinformation

Risk Description:
LLMs can generate incorrect or misleading information, posing risks to applications that rely on their output for critical decisions.

Our Controls:

- **Human Validation:** Edge case review with human verification
- **Pattern Management:** DynamoDB table for storing vetted prompt patterns
- **Anomaly Detection:** CloudWatch monitoring for unusual patterns

CloudFormation Resources:

yaml Copy

```yaml
EdgeCaseReviewFunction:
  # Human review workflow

ApprovedPromptsTable:
  # Vetted prompt patterns

BedrockLogGroup:
  # Monitoring for unusual patterns
```

9. Unbounded Consumption

Risk Description:
LLMs can consume excessive resources (compute, tokens, cost) without appropriate limits, leading to service degradation or financial impacts.

Our Controls:

- **Budget Controls:** AWS Budget with alerts for cost management
- **Usage Monitoring:** CloudWatch alarms detect unusual usage patterns
- **Rate Limiting:** IAM policy provides access boundaries

CloudFormation Resources:

yaml Copy

```yaml
BedrockBudget:
  Type: AWS::Budgets::Budget
  # Cost management and alerting

BedrockCostAlarm:
  Type: AWS::CloudWatch::Alarm
  # Detects usage spikes

BedrockRestrictedPolicy:
  # Implements access boundaries
```

10. Model Theft

Risk Description:
Unauthorized access, copying, or exfiltration of proprietary LLM models by malicious actors, leading to economic losses and compromised competitive advantage.

Our Controls:

- **Private Network:** VPC endpoint for secure access only
- **Traffic Restrictions:** Security group limiting ingress/egress
- **Activity Monitoring:** CloudTrail for detecting unusual access patterns
- **Least Privilege:** IAM role restricts access to minimum necessary

CloudFormation Resources:

yaml Copy

```yaml
BedrockVPCEndpoint:
  # Private access only

BedrockSecurityGroup:
  # Traffic restrictions

# Integration with existing CloudTrail
# Detects unusual access patterns

BedrockAccessRole:
  # Least privilege principle
```

Additional Security Measures

Beyond the OWASP Top 10 mapping, our template includes:

Comprehensive Monitoring

- CloudWatch logs with 30-day retention policy
- SNS notifications for critical security events
- Cost alarms for usage spikes

Cost Controls

- Monthly budget with notification threshold
- Alarm for high-cost model invocations

Human Oversight

- Review workflow for edge cases
- Approval mechanisms for sensitive operations

Conclusion

Our AWS Bedrock guardrails implementation provides a comprehensive security framework that addresses all ten of the OWASP Top 10 vulnerabilities for LLMs. This defense-in-depth approach applies multiple layers of controls at each level of the stack:

1. **Network Level:** VPC endpoints, security groups
2. **Identity Level:** IAM roles and policies with least privilege
3. **Content Level:** Guardrails, PII detection, custom word lists
4. **Operational Level:** Monitoring, logging, human oversight

This multi-layered approach ensures that our Bedrock implementation follows industry best practices for securing LLM applications, creating a robust foundation for responsible AI deployment.

References

1. OWASP Top 10 for Large Language Model Applications
2. AWS Security Best Practices for Bedrock
3. NIST AI Risk Management Framework