File Upload Threat Model - Amazon Interview Scenario

Scenario Overview

Interviewer Prompt: "Threat model a web page that has file upload functionality"

Context: Amazon Prime members can upload profile pictures and documents to their account dashboard

Scale Considerations:

- 200+ million Prime members globally
- Expected 10M+ file uploads daily
- Files stored in Amazon S3
- Global CDN distribution via CloudFront

---

1. System Architecture & Data Flow

Components

```
[User Browser] → [CloudFront] → [ALB] → [EC2 Web Servers] → [S3 Storage]
                                    ↓
                               [Lambda Processors] → [DynamoDB Metadata]
                                    ↓
                               [Virus Scanning] → [Content Analysis]
```

Data Flow

1. User selects file in web browser
2. JavaScript validates file type/size client-side
3. File uploaded via HTTPS POST to ALB
4. Web server validates file headers and metadata
5. File temporarily stored in EC2 processing directory
6. Lambda function processes file (virus scan, content analysis)
7. Clean files moved to S3 with customer-specific access controls
8. Metadata stored in DynamoDB with user association
9. CDN serves files with signed URLs for authorized access

---

2. Assets & Trust Boundaries

Critical Assets

- **Customer PII**: Names, addresses, payment info in uploaded documents
- **System Integrity**: Web servers, databases, file storage
- **Service Availability**: Upload functionality for 200M+ users
- **Brand Reputation**: Customer trust in Amazon's security

Trust Boundaries

1. **Internet ↔ CloudFront**: Untrusted user content entering trusted AWS environment
2. **Web Servers ↔ S3**: Application layer to permanent storage
3. **Customer Data ↔ Amazon Systems**: Strict isolation between customer accounts

---

3. STRIDE Analysis

Spoofing (S)

Threat: Attacker uploads file claiming to be another user

- **Attack Vector**: Session hijacking, CSRF attacks
- **Business Impact**: Unauthorized access to customer accounts
- **Amazon Scale**: With 200M users, identity confusion could affect millions
- **Customer Impact**: Loss of trust in account security
- **Mitigation**: Strong session management, CSRF tokens, multi-factor auth

Tampering (T)

Threat: Malicious file modification during upload/storage

- **Attack Vector**: Man-in-the-middle attacks, storage corruption
- **Business Impact**: Data integrity issues, potential malware distribution
- **Amazon Scale**: Corrupted files could affect CDN distribution globally
- **Customer Impact**: Compromised documents, potential malware exposure
- **Mitigation**: HTTPS transport, file integrity checksums, signed S3 uploads

Repudiation (R)

Threat: User denies uploading malicious content

- **Attack Vector**: Account compromise followed by malicious uploads
- **Business Impact**: Legal liability, compliance violations
- **Amazon Scale**: Requires audit trails for 200M+ users
- **Customer Impact**: Account suspension, legal consequences
- **Mitigation**: Comprehensive audit logging, digital signatures, IP tracking

Information Disclosure (I)

Threat: Unauthorized access to uploaded files

- **Attack Vector**: Direct S3 access, misconfigured permissions, IDOR vulnerabilities
- **Business Impact**: Privacy violations, regulatory fines (GDPR: €20M+)
- **Amazon Scale**: One misconfiguration could expose millions of files
- **Customer Impact**: Personal documents exposed publicly
- **Mitigation**: S3 bucket policies, signed URLs, principle of least privilege

Denial of Service (D)

Threat: Upload functionality overwhelmed or disabled

- **Attack Vector**: Large file attacks, rapid upload attempts, storage exhaustion
- **Business Impact**: Service degradation for 200M+ Prime members
- **Amazon Scale**: Need to handle 10M+ daily uploads reliably
- **Customer Impact**: Cannot access Prime features requiring file uploads
- **Mitigation**: Rate limiting, file size restrictions, auto-scaling infrastructure

Elevation of Privilege (E)

Threat: File upload leads to system compromise

- **Attack Vector**: Malicious executables, script injection, path traversal
- **Business Impact**: Complete system compromise, data breach
- **Amazon Scale**: Could affect entire Prime platform infrastructure
- **Customer Impact**: Full account compromise, financial fraud
- **Mitigation**: Strict file type validation, sandboxed processing, WAF rules

---

4. Detailed Threat Scenarios

High-Risk Scenario: Malicious File Execution

Attack Chain:

1. Attacker crafts PHP shell disguised as image file
2. Bypasses client-side validation with double extension (shell.php.jpg)
3. Server-side validation checks only final extension (.jpg)
4. File uploaded to web-accessible directory
5. Attacker accesses shell.php.jpg, executes server commands
6. Gains access to customer database and S3 credentials

Business Impact:

- **Direct Cost**: $200M+ in breach response (industry average: $4.24M per breach)
- **Customer Impact**: 200M Prime members' PII exposed
- **Regulatory**: GDPR fines up to €20M, state-level breach notifications
- **Reputation**: Customer trust loss, potential Prime subscription cancellations

Amazon-Scale Considerations:

- Single vulnerability could affect entire Prime platform
- Global impact across all AWS regions
- Potential cascade failures to other Amazon services

Medium-Risk Scenario: Storage-Based DoS Attack

Attack Chain:

1. Attacker creates automated script to upload maximum-size files
2. Generates thousands of accounts to bypass rate limiting

3. Exhausts S3 storage quotas and increases costs
4. Legitimate users receive "storage full" errors
5. Service degradation affects customer experience

Business Impact:

- **Direct Cost**: Unexpected S3 storage costs (could reach millions)
- **Customer Impact**: Service unavailable for legitimate file uploads
- **Competitive**: Customers may switch to competitors during outage

Low-Risk Scenario: Metadata Information Leakage

Attack Chain:

1. User uploads image with embedded GPS coordinates
2. Other users access file via direct link sharing
3. Metadata reveals personal location information
4. Privacy violation occurs without user awareness

Business Impact:

- **Regulatory**: GDPR privacy violation complaints
- **Customer Impact**: Unintended privacy exposure
- **Reputation**: Media coverage of privacy issues

---

5. Amazon-Scale Mitigations

Infrastructure-Level Controls

```
CloudFront Configuration:
  - WAF rules blocking malicious file types
  - Rate limiting: 100 uploads per user per hour
  - Geographic restrictions based on risk assessment

Application Load Balancer:
  - SSL/TLS termination with perfect forward secrecy
  - Request size limits: 100MB maximum file size
  - Request rate limiting: 1000 requests per minute per IP

EC2 Web Servers:
  - Auto-scaling groups for handling upload spikes
  - Sandboxed file processing in isolated containers
  - No execution permissions on upload directories
  - Regular security patching automated via Systems Manager
```

Application-Level Controls

```python
# Secure Upload Validation (Example)
ALLOWED_EXTENSIONS = {'.jpg', '.jpeg', '.png', '.pdf', '.docx'}
MAX_FILE_SIZE = 100 * 1024 * 1024  # 100MB

def validate_upload(file, user_context):
    # Multi-layer validation
    if not user_context.is_authenticated():
        raise SecurityError("Authentication required")

    # File extension validation
    ext = get_file_extension(file.filename).lower()
    if ext not in ALLOWED_EXTENSIONS:
        raise ValidationError(f"File type {ext} not allowed")

    # Magic number validation (prevents extension spoofing)
    magic_number = file.read(8)
    if not validate_file_signature(magic_number, ext):
        raise ValidationError("File content doesn't match extension")

    # Size validation
    if file.size > MAX_FILE_SIZE:
        raise ValidationError("File exceeds size limit")

    # Virus scanning
    scan_result = antivirus_scan(file)
    if scan_result.threat_detected:
        log_security_event(user_context.user_id, "malware_upload", scan_result)
        raise SecurityError("Malicious content detected")

    return True
```

Storage-Level Controls

```
{
  "S3_bucket_policy": {
    "Statement": [
      {
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::prime-uploads/*",
        "Condition": {
          "StringNotEquals": {
            "s3:ExistingObjectTag/AccessLevel": "authorized"
          }
        }
      }
    ]
  },
  "lifecycle_policy": {
```

```
      "Rules": [
        {
          "Status": "Enabled",
          "Filter": {"Tag": {"Key": "ContentType", "Value": "temporary"}},
          "Expiration": {"Days": 30}
        }
      ]
    }
  }
```

Monitoring & Detection

```
CloudWatch Alerts:
  - Upload rate spikes (>150% of normal)
  - File size anomalies (multiple large files from single user)
  - Failed validation attempts (>10 per user per hour)

GuardDuty Integration:
  - Malicious IP detection
  - Cryptocurrency mining detection in uploaded files
  - Data exfiltration pattern detection

Custom Lambda Monitors:
  - File type distribution analysis
  - User behavior anomaly detection
  - Metadata extraction for compliance scanning
```

---

6. Business Impact Quantification

Customer Trust Metrics

- **Breach Impact**: $165 per customer record × 200M customers = $33B potential exposure
- **Prime Cancellations**: 5% churn rate post-breach = 10M lost subscriptions = $1.4B annual revenue loss
- **Recovery Time**: 12-18 months for full trust restoration based on industry benchmarks

Regulatory Compliance Costs

- **GDPR Maximum**: €20M fine for major privacy violations
- **State Breach Notifications**: $0.50-$2.00 per customer = $100M-$400M notification costs
- **Legal Defense**: $50M-$100M in breach litigation costs

Operational Impact

- **Service Downtime**: 1 hour of Prime service disruption = $10M lost revenue
- **Storage Costs**: Malicious uploads could increase S3 costs by 200-500%
- **Engineering Response**: Emergency fix deployment = $500K-$1M in overtime costs

---

7. Interview Key Points

Technical Competence Demonstration

1. **Systematic Approach**: Used STRIDE methodology comprehensively
2. **Scale Awareness**: Considered 200M+ user implications for each threat
3. **AWS Integration**: Leveraged CloudFront, S3, Lambda for scalable security
4. **Defense in Depth**: Multiple validation layers from client to storage

Business Impact Understanding

1. **Quantified Risk**: Specific dollar amounts for breach scenarios
2. **Customer Focus**: Connected technical threats to customer trust impact
3. **Regulatory Awareness**: GDPR and compliance implications
4. **Competitive Positioning**: Service availability affects market position

Communication Excellence

1. **Executive Summary**: Can explain threat model in 2-3 minutes
2. **Technical Details**: Detailed implementation for engineering teams
3. **Risk Prioritization**: Clear high/medium/low risk categorization
4. **Actionable Recommendations**: Specific AWS services and configurations

Amazon Leadership Principles Alignment

- **Customer Obsession**: Every threat connects to customer impact
- **Ownership**: Comprehensive approach to security responsibility
- **Invent and Simplify**: Scalable solutions using AWS services
- **Are Right, A Lot**: Data-driven risk assessment and mitigation
- **Think Big**: Solutions designed for Amazon's global scale

---

8. Follow-up Questions & Responses

Q: "How would you prioritize these threats for immediate action?" A: "High priority: File execution vulnerabilities due to $33B potential exposure. Medium priority: DoS attacks affecting 200M users. Low priority: Metadata leakage with limited customer impact. I'd allocate 70% of resources to preventing file execution, 25% to DoS protection, 5% to metadata controls."

Q: "What metrics would you use to measure security effectiveness?" A: "Primary metrics: Zero successful malicious file executions, 99.99% upload availability, <0.01% false positive rate in virus scanning. Business metrics: Maintained customer trust scores, zero regulatory violations, security response time <4 hours for critical issues."

Q: "How does this scale to Amazon's other services?" A: "This threat model applies to any AWS service with user-generated content: S3 direct uploads, EFS file sharing, WorkDocs collaboration. I'd create reusable CloudFormation templates with these security controls, enabling consistent protection across all Amazon services with minimal engineering overhead."

This comprehensive threat model demonstrates the systematic thinking, business acumen, and technical depth that Amazon expects from Application Security Engineers, while specifically addressing the file upload scenario mentioned by the recruiter.