■ SAT WEBSITE

Security Enhancement Recommendations

Generated: January 17, 2026

Priority: HIGH

Status: RECOMMENDED ACTIONS

1. Current Security Status Assessment

Security Layer	Status	Grade
IAM Authentication	✓ Implemented	A
SSM-Based Deployment	✓ Implemented	A
GitHub Secrets Encryption	✓ Implemented	A
Environment Protection	✓ Implemented	A
Least Privilege Access	✓ Implemented	A-
TLS/SSL Encryption	✓ Implemented	A
Audit Logging	✓ Implemented	B+
Web Application Firewall	✗ Not Implemented	C
DDoS Protection	✗ Not Implemented	C
Security Headers	? Unknown	?
Rate Limiting	? Unknown	?
Intrusion Detection	✗ Not Implemented	C
Backup Encryption	? Unknown	?
Secret Rotation Policy	✗ Not Implemented	C
Multi-Factor Auth (MFA)	? Unknown	?

Overall Security Grade: B+

Your current deployment is SOLID with excellent authentication and deployment security. However, there are critical enhancements needed for production-grade web application security.

2. PRIORITY 1: Critical Security Enhancements

■■ IMPLEMENT IMMEDIATELY

2.1 AWS WAF (Web Application Firewall)

Risk: SQL injection, XSS attacks, bot traffic, DDoS

Impact: HIGH - Direct threat to application security

Effort: Medium (2-3 hours)

Benefits:

• Protection against OWASP Top 10 vulnerabilities

• Rate limiting to prevent abuse

• IP reputation blocking

• Custom rules for your application

• Real-time metrics and logging

Implementation Steps:

1. Go to AWS Console → WAF & Shield

2. Create Web ACL for us-east-1

3. Add AWS Managed Rule Groups:

- Core rule set (protects against common exploits)

- Known bad inputs (blocks malicious patterns)

- SQL database (prevents SQL injection)

4. Add rate limiting: 2000 requests per 5 minutes per IP

5. Associate with Application Load Balancer or CloudFront

6. Enable logging to CloudWatch

Estimated Cost: ~$5-10/month (minimal traffic)

2.2 Security Headers (HTTP Headers)

Risk: XSS, clickjacking, MIME sniffing attacks

Impact: HIGH - Browser-level security vulnerabilities

Effort: Low (30 minutes)

Required Headers:

```
# Add to Nginx configuration: /etc/nginx/sites-available/sat

server {
    ...

    # Security Headers
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-XSS-Protection "1; mode=block" always;
    add_header Referrer-Policy "strict-origin-when-cross-origin" always;
    add_header Content-Security-Policy "default-src 'self'; script-src 'self' 'unsafe-inli
ne'; style-src 'self' 'unsafe-inline';" always;
    add_header Permissions-Policy "geolocation=(), microphone=(), camera=()" always;

    # HSTS (uncomment after testing)
    # add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;

    ...
}
```

Test After Implementation:

Visit: `https://securityheaders.com/?q=https://www.sat.net.in/`

Target Grade: A or A+

2.3 Enable MFA on IAM Users

Risk: Account compromise if credentials leaked

Impact: HIGH - Prevents unauthorized AWS access

Effort: Low (10 minutes per user)

Implementation:

1. AWS Console → IAM → Users → github-actions-sat
2. Security credentials → Assign MFA device
3. Choose: Virtual MFA device (Google Authenticator, Authy)
4. Scan QR code with authenticator app
5. Enter two consecutive MFA codes
6. Save recovery codes securely
7. Repeat for root account and all IAM users

■■ **IMPORTANT:** Store MFA recovery codes in a secure location (1Password, LastPass, etc.)

2.4 Implement Secret Rotation Policy

Risk: Long-lived credentials increase compromise risk

Impact: MEDIUM - Limits exposure window

Effort: Medium (1 hour setup + recurring)

Rotation Schedule:

Credential Type	Rotation Frequency	Method
IAM Access Keys	Every 90 days	Manual via AWS Console
GitHub Secrets	After each key rotation	Manual via GitHub Settings
Database Passwords	Every 90 days	AWS Secrets Manager (future)
SSL/TLS Certificates	Auto-renewal (Let's Encrypt)	Certbot (if applicable)

Automation Option: Use AWS Secrets Manager for automatic rotation (additional cost: ~$0.40/secret/month)

3. PRIORITY 2: High Priority Enhancements

■ IMPLEMENT WITHIN 30 DAYS

3.1 CloudWatch Alarms & Monitoring

Purpose: Real-time alerting for security and performance issues

Effort: Medium (2 hours)

Metric	Threshold	Action
EC2 CPU Utilization	> 80% for 5 min	Email alert
EC2 StatusCheckFailed	≥ 1	Email + SMS alert
Failed SSH Attempts	> 5 attempts/min	Email alert (if SSH enabled)
Deployment Failures	≥ 1 in GitHub Actions	Email alert
Website Downtime	HTTP 5xx errors	Email + SMS alert
Disk Usage	> 85%	Email alert

3.2 Automated Encrypted Backups

Purpose: Disaster recovery and data protection

Effort: Medium (1-2 hours)

Backup Type	Frequency	Retention	Method
Code Repository	Real-time	Unlimited	GitHub (already done ✓)
Database	Daily at 2 AM UTC	30 days	AWS Backup or pg_dump
EC2 AMI Snapshot	Weekly	4 weeks	AWS Backup
Configuration Files	Weekly	12 weeks	S3 bucket (encrypted)
Application Data	Daily	30 days	S3 bucket (encrypted)

Implementation: Use AWS Backup for automated, encrypted backups (~$0.05/GB/month)

3.3 Enable VPC Flow Logs

Purpose: Network traffic monitoring and intrusion detection

Effort: Low (30 minutes)

1. AWS Console → VPC → Your VPC → Flow logs

2. Create flow log

3. Filter: All traffic (Accept, Reject)

4. Destination: CloudWatch Logs

5. Create new log group: /aws/vpc/flowlogs

6. Set retention: 30 days

7. Review logs weekly for suspicious activity

Cost: ~$0.50-2/month for small traffic

4. PRIORITY 3: Medium Priority Enhancements

■ IMPLEMENT WITHIN 90 DAYS

4.1 Enable AWS GuardDuty

Purpose: Intelligent threat detection using machine learning

Effort: Low (15 minutes)

Detection Capabilities:

• Detects compromised EC2 instances

• Identifies malicious IP addresses

• Monitors for cryptocurrency mining

• Detects unusual API calls

• Monitors for data exfiltration attempts

Implementation: AWS Console → GuardDuty → Enable (30-day free trial)

Cost After Trial: ~$1-5/month based on usage

4.2 Enable AWS Config

Purpose: Configuration compliance and change tracking

Effort: Medium (1 hour)

Compliance Rules to Monitor:

• EC2 instances must have SSM agent installed ✓

• IAM users must have MFA enabled

• S3 buckets must be encrypted

• Security groups cannot allow 0.0.0.0/0 on port 22

• EC2 instances must have approved AMIs only

• IAM policies follow least privilege

Cost: ~$2-10/month depending on resources

4.3 Application-Level Security (Flask)

Purpose: Harden the Flask application itself

Effort: Medium (3-4 hours)

Security Measure	Implementation	Priority
Input validation	Validate all user inputs	Critical
SQL injection protection	Use parameterized queries	Critical
CSRF protection	Flask-WTF with CSRF tokens	High
Session security	Secure cookies, HTTPOnly, SameSite	High
Rate limiting	Flask-Limiter package	High
Logging & monitoring	Log security events	Medium
Error handling	Don't expose stack traces	Medium

5. PRIORITY 4: Nice-to-Have Enhancements

■ CONSIDER FOR FUTURE

Enhancement	Benefit	Effort	Cost/Month
AWS CloudFront CDN	Faster global delivery, DDoS protection	Medium	$5-20
AWS Certificate Manager	Free SSL/TLS certificates	Low	Free
Elasticsearch + Kibana	Advanced log analysis	High	$50-100
AWS Shield Standard	DDoS protection (free tier)	Low	Free
Penetration Testing	Find vulnerabilities proactively	High	$500-2000
Bug Bounty Program	Crowdsourced security testing	Medium	Variable
SOC 2 Compliance	If handling sensitive data	Very High	$15k-50k

6. 90-Day Implementation Roadmap

Week	Task	Priority	Time
1-2	Implement Security Headers in Nginx	P1	30 min
1-2	Enable MFA on all IAM users	P1	1 hour
3-4	Set up AWS WAF with managed rules	P1	3 hours
3-4	Implement secret rotation policy	P1	1 hour
5-6	Configure CloudWatch alarms	P2	2 hours
5-6	Enable VPC Flow Logs	P2	30 min
7-8	Set up automated encrypted backups	P2	2 hours
9-10	Enable AWS GuardDuty	P3	15 min
11-12	Enable AWS Config with compliance rules	P3	1 hour
On-going	Application security hardening	P3	4 hours

Total Time Investment: ~15 hours over 90 days

Total Monthly Cost: ~$15-30/month (minimal for the security gained)

7. Cost-Benefit Analysis

Security Enhancement	Monthly Cost	Annual Cost	Value
AWS WAF	$5-10	$60-120	Very High
CloudWatch Alarms	$1-3	$12-36	High
VPC Flow Logs	$0.50-2	$6-24	Medium
AWS Backup	$2-5	$24-60	Very High
AWS GuardDuty	$1-5	$12-60	High
AWS Config	$2-10	$24-120	Medium
Security Headers	Free	Free	Very High
MFA	Free	Free	Very High
TOTAL	**$11.50-35**	**$138-420**	

Return on Investment:

• Cost of a security breach: $1,000 - $50,000+ (downtime, reputation, data loss)

• Cost of prevention: $138-420/year

• ROI: Prevents potentially catastrophic losses

• Compliance: Meets industry security standards

• Peace of mind: 24/7 automated monitoring and protection

8. Final Recommendations

Immediate Actions (This Week):

✓ Add security headers to Nginx configuration (30 minutes)

✓ Enable MFA on github-actions-sat IAM user (10 minutes)

✓ Document your current backup strategy

✓ Test your current disaster recovery plan

Priority Order:

1. **Security Headers** (Free, Fast, High Impact)
2. **MFA on IAM Users** (Free, Fast, Critical Protection)
3. **AWS WAF** (Low Cost, High Protection)
4. **CloudWatch Alarms** (Low Cost, Early Detection)
5. **Automated Backups** (Medium Cost, Essential for DR)

Security is a Journey, Not a Destination

Your current infrastructure is solid with A-grade authentication security. These recommendations will elevate you to enterprise-grade, production-ready security. Start with the free/low-cost, high-impact items first, then expand over time.

Questions? Review each section and prioritize based on your risk tolerance and budget.

Appendix: Quick Implementation Commands

A1: Add Security Headers to Nginx

```
# Edit Nginx configuration
sudo nano /etc/nginx/sites-available/sat

# Add security headers inside server block
# (See section 2.2 for complete headers)

# Test configuration
sudo nginx -t

# Reload Nginx
sudo systemctl reload nginx

# Verify headers
curl -I https://www.sat.net.in/
```

A2: Rotate IAM Access Keys

```
# AWS Console process:
1. IAM → Users → github-actions-sat
2. Security credentials → Create access key
3. Save new keys immediately
4. Update GitHub Secrets with new keys
5. Test deployment works with new keys
6. Deactivate old access key (don't delete yet)
7. After 24 hours, if no issues, delete old key
```

A3: Check Security Headers

Test your headers: `https://securityheaders.com/?q=https://www.sat.net.in/`

Document Version: 1.0 | **Generated:** January 17, 2026

Status: Recommended Actions - Pending Implementation