Technical Onboarding Playbook

Support Analyst - Level 2

Program:	Company X Enterprise Platform
Duration:	4 Weeks (30 days)
Author:	Gabriel Mazer
Role:	Senior Technical Support Engineer
Version:	2.1
Last Updated:	March, 2025
Status:	Active

⚠ Data Sanitization Notice

This document has been sanitized for portfolio purposes. All sensitive information including company names, client identifiers, internal URLs, and proprietary data have been replaced with generic placeholders (Company X, Client ABC, xxx.xxx, etc.). The structure, methodology, and didactic approach remain authentic.

Welcome to the Technical Onboarding Playbook for Support Analysts (Level 2) at Company X. This document is designed to accelerate your learning curve and provide you with the essential technical knowledge needed to effectively support our Enterprise Security & Device Management Platform.

Technical Context

What You'll Learn

- Core platform architecture (MDM, VPN, Content Filtering)
- Essential protocols: SSH, RDP, API REST
- Troubleshooting methodologies and decision frameworks
- Common customer scenarios and their resolutions
- Best practices for technical documentation and escalation

This playbook follows a **4-week structured learning path**. Each week builds upon the previous one, combining theoretical knowledge with practical labs.

Success Indicators

Learning Approach
70% Hands-on Practice — 20% Reading & Documentation — 10% Peer Learning
Work through each section sequentially, complete all labs, and don't hesitate to revisit earlier sections as needed.

By the end of this onboarding program, you should be able to:

1. **Independently troubleshoot** 70% of incoming L2 tickets
2. **Understand** the complete authentication and connection flow
3. **Utilize** internal tools, KB articles, and escalation procedures
4. **Communicate effectively** with both technical and non-technical stakeholders
5. **Document solutions** in the knowledge base following best practices

Key Learning Objectives

Week 1 Learning Objectives

- Understand the three-tier platform architecture
- Identify core components and their functions
- Learn authentication flow and security model
- Complete environment setup and first lab
- Achieve 40% readiness for basic ticket handling

Company X's platform consists of three integrated core components that work together to provide enterprise-grade security and device management.

1.2.1 Platform Architecture - Component Overview

Layer	Components & Functions
Layer 1: Client Side	• Mobile Apps (iOS, Android) • Desktop Clients (Windows, macOS, Linux) • Web Portal (Browser-based access)
Layer 2: Gateway	• VPN Gateway (secure tunnels) • MDM Engine (device management) • Content Filter (traffic inspection)
Layer 3: Backend	• Authentication Service (OAuth 2.0) • Policy Engine (compliance rules) • API Layer (integration endpoints) • Logging & Analytics

Table 1: Platform Architecture Layers

Technical Context

Data Flow Summary
User Device → connects to → **VPN Gateway**
VPN Gateway → routes through → **Content Filter** & **MDM Engine**
Both components → communicate with → **Backend Services**
Backend → validates via → **Authentication Service**
All traffic is logged and monitored in real-time.

1.3.1 1. Mobile Device Management (MDM)

The MDM component manages device enrollment, configuration profiles, and policy enforcement.

Technical Context

Key Functions

- **Device Enrollment**: iOS, Android, Windows, macOS
- **Policy Management**: Security policies, compliance rules
- **App Distribution**: Corporate app catalog and deployment
- **Remote Actions**: Lock, wipe, locate devices

Common Ports & Protocols:

- HTTPS (443) - Device communication
- APNS (2195, 2196) - Apple Push Notification Service
- FCM (443) - Firebase Cloud Messaging (Android)

Supported Platforms:

Platform	Min Version	Enrollment Method
iOS/iPadOS	14.0+	DEP/ABM or Manual
Android	8.0+	QR Code or Token
Windows	10 (1809+)	Azure AD Join
macOS	11.0+	DEP or Manual

1.3.2 2. VPN Gateway

Provides secure remote access to corporate resources through encrypted tunnels.

- **Protocols Supported**: OpenVPN, IPsec/IKEv2, WireGuard
- **Authentication**: Certificate-based, LDAP/AD integration, MFA
- **Split Tunneling**: Configurable per-app or per-domain
- **Default Ports**: UDP 1194 (OpenVPN), UDP 500/4500 (IPsec)

Connection Process:

1. User initiates VPN connection from client
2. Client presents certificate to gateway
3. Gateway validates certificate against CA
4. If valid, establish encrypted tunnel (TLS 1.3)
5. User traffic routes through tunnel
6. All traffic logged for compliance

1.3.3 3. Content Filtering Engine

Real-time content inspection and filtering for both browser and application traffic.

- **Categories**: 50+ content categories (malware, adult, gambling, etc.)
- **Granularity**: URL-level, domain-level, IP-level filtering
- **Reporting**: Real-time logs and historical analytics
- **Bypass Rules**: Configurable exceptions for trusted domains

Understanding the authentication flow is critical for troubleshooting access issues.

Key Learning Objectives

Authentication Process - Step by Step

Step 1: Initial Login Attempt
- User opens client application
- System checks device enrollment status
- If not enrolled → Redirect to enrollment flow
- If enrolled → Proceed to Step 2

Step 2: Credential Validation
- User enters username/password
- Credentials sent to Authentication Service (encrypted)
- Service queries LDAP/AD or local database
- Invalid credentials → Deny access, log attempt
- Valid credentials → Proceed to Step 3

Step 3: Multi-Factor Authentication (if required)
- System checks if MFA is mandatory for user/group
- If yes → Request MFA token (SMS, authenticator app, push)
- User provides token
- System validates token (time-sensitive, 30-60 sec window)
- Invalid token → Deny access
- Valid token → Proceed to Step 4

Step 4: Device Compliance Check
- System evaluates device health (OS version, encryption, jailbreak)
- Checks policy compliance (required apps, security settings)
- Non-compliant → Block access, notify admin
- Compliant → Proceed to Step 5

Step 5: Access Granted
- Issue session token (JWT)
- Establish VPN tunnel
- Apply content filtering rules
- User gains access to corporate resources
- All activity logged

Success Indicators

Lab 1.1: Configure Your Testing Environment
Objective: Set up your local environment to simulate customer scenarios
Requirements:

- Virtual machine with Windows 10/11 (VirtualBox/VMware)
- Virtual machine with Ubuntu 22.04 LTS
- Access to Company X staging environment
- Testing devices enrolled in test organization ”TEST-ORG-XXX”

Steps:

1. Install VMs using provided OVA files from internal repository
2. Configure network adapters (NAT + Host-Only)
3. Install Company X agent/client (version 4.2.x)
4. Verify connectivity to staging API: `api-staging.xxx.com`
5. Complete enrollment flow and document any issues encountered

Expected Outcome: Successfully enrolled test devices with active VPN connection
Validation Checklist:
VMs running and accessible
Company X client installed
Enrollment completed successfully
VPN tunnel established
Can access internal test resources

Important Notes

Before Proceeding to Week 2
Ensure you can answer these questions:

1. What are the three main platform components?
2. What ports does the VPN gateway use?
3. What happens if MFA validation fails?
4. How many authentication steps are there?
5. What’s the difference between supervised and non-supervised iOS devices?

If you can’t answer 80% of these, review Week 1 materials before continuing.

Key Learning Objectives

Week 2 Learning Objectives

- Master SSH and RDP for remote troubleshooting
- Understand API REST architecture and common endpoints
- Learn systematic troubleshooting methodologies
- Complete 3 practical labs with real scenarios
- Achieve 65% readiness for ticket handling

As a Level 2 Support Analyst, you'll frequently troubleshoot remote access issues. Mastering SSH and RDP is essential.

2.2.1 SSH (Secure Shell)

Use Cases in Our Environment:

- Troubleshooting Linux-based VPN gateways
- Accessing backend infrastructure for log analysis
- Executing remote commands for diagnostics
- Verifying certificate validity

Technical Context

Common SSH Commands for Support

```
# Check VPN service status
ssh admin@xxx.gateway.com "systemctl status openvpn"

# Tail real-time logs
ssh admin@xxx.gateway.com "tail -f /var/log/vpn/access.log"

# Check active connections
ssh admin@xxx.gateway.com "netstat -an | grep :1194"

# Verify certificate validity
ssh admin@xxx.gateway.com "openssl x509 -in /etc/ssl/server.crt \
  -noout -dates"
```

Troubleshooting SSH Issues:

Error	Solution
Connection refused	Check if SSH service is running, verify firewall rules (port 22)
Permission denied	Verify SSH key is correct, check user permissions on server
Host key verification failed	Remove old key from known_hosts, verify server identity
Timeout	Check network connectivity, verify security groups/ACLs

2.2.2 RDP (Remote Desktop Protocol)

Use Cases:

- Supporting Windows-based customer environments
- Troubleshooting client application issues
- Verifying policy application on Windows devices
- Screen-sharing for guided troubleshooting

Key Ports: 3389 (default), 3390-3395 (custom configurations)
Connection String Format:

```
mstsc /v:hostname.xxx.com:3389 /admin
```

2.2.3 API REST Integration

Our platform exposes a comprehensive REST API for automation and integration.

Technical Context

API Basics
Base URL: `https://api.xxx.com/v2/`
Authentication: Bearer Token (OAuth 2.0)
Common Endpoints:

- `GET /devices` - List enrolled devices
- `POST /devices/{id}/actions` - Execute device actions
- `GET /policies` - Retrieve policy configurations
- `GET /logs/vpn` - Query VPN connection logs
- `POST /users` - Create new user accounts

Example API Call (using curl):

```
curl -X GET "https://api.xxx.com/v2/devices?org=CLIENT-ABC" \
  -H "Authorization: Bearer xxxTOKENxxx" \
  -H "Content-Type: application/json"
```

Common API Response Codes:

- **200 OK**: Request successful
- **400 Bad Request**: Invalid parameters
- **401 Unauthorized**: Invalid or expired token
- **403 Forbidden**: Insufficient permissions

- **429 Too Many Requests**: Rate limit exceeded
- **500 Internal Server Error**: Server-side issue

When a customer reports an issue, follow this systematic approach:

Key Learning Objectives

Universal Troubleshooting Framework
Phase 1: Information Gathering (5-10 minutes)

1. What is the exact error message or symptom?
2. When did the issue start?
3. Is it affecting one user or multiple?
4. What changed recently? (updates, config changes)
5. Can you reproduce the issue?

Phase 2: Categorization (2-3 minutes)

- **Category A**: Connectivity Issues (network, VPN, firewall)
- **Category B**: Authentication Issues (credentials, MFA, tokens)
- **Category C**: Performance Issues (latency, timeouts, slowness)
- **Category D**: Configuration Issues (policies, settings, profiles)

Phase 3: Investigation (15-30 minutes)

1. Check service status (are all systems operational?)
2. Review logs (application, system, network)
3. Test connectivity (ping, traceroute, telnet)
4. Validate configuration (compare with working environment)
5. Isolate variables (test in controlled environment)

Phase 4: Resolution (varies)

1. Implement fix based on findings
2. Test thoroughly before confirming with customer
3. Document solution in ticket
4. If unable to resolve $\rightarrow$ Escalate to L3 with full context

Phase 5: Documentation (5 minutes)

1. Update ticket with root cause and resolution
2. Create KB article if issue is likely to recur
3. Note any workarounds or temporary fixes

Symptoms	Likely Category	First Actions
”Cannot connect”, ”Timeout”, ”Unreachable”	Connectivity (A)	Check network, ping gateway, verify firewall
”Invalid password”, ”Access denied”, ”401 error”	Authentication (B)	Verify credentials, check MFA, review token
”Slow”, ”Takes forever”, ”Laggy”	Performance (C)	Check latency, review bandwidth, analyze logs
”Feature not working”, ”Settings wrong”	Configuration (D)	Compare policies, review profile, check enrollment

Success Indicators

Lab 2.1: VPN Connection Failure
Scenario: Customer from Client ABC reports: ”Users cannot establish VPN connection. Error: ’Authentication failed (error code 401)’”
Your Task:

1. Use SSH to access VPN gateway: `ssh admin@vpn-gw-abc.xxx.com`
2. Check authentication logs: `tail -100 /var/log/auth.log`
3. Verify LDAP connectivity: `ldapsearch -x -h ldap.xxx.com`
4. Check user’s MFA status via API
5. Document findings and resolution steps

Expected Resolution Time: 15-20 minutes
Hint: 401 errors typically indicate authentication service issues. Check if LDAP is responding and if MFA service is operational.

Important Notes

Lab 2.2: API Rate Limiting
Scenario: Customer reports their integration script is failing with ”429 Too Many Requests”
Investigation Steps:

- Query API logs for customer org-id: `CLIENT-XYZ`
- Identify request patterns and frequency
- Check current rate limits: 100 req/min per org
- Recommend implementation of exponential backoff
- Provide code sample for retry logic

Learning Point: Rate limiting is a protective measure. Educate customers on best practices rather than just increasing limits.

Important Notes

Competency Check
Before Week 3, ensure you can:

1. SSH into a server and navigate logs
2. Identify the difference between 401, 403, and 429 errors
3. Follow the troubleshooting framework independently
4. Categorize an issue within 3 minutes
5. Know when to escalate vs. continue investigating

Key Learning Objectives

Week 3 Learning Objectives

- Handle complex VPN connectivity issues
- Diagnose content filter bypass attempts
- Troubleshoot device enrollment failures across platforms
- Apply advanced debugging techniques
- Achieve 85% readiness for independent ticket handling

3.2.1 Common VPN Failure Modes

Error Code	Description	Resolution
E-VPN-001	Certificate expired	Regenerate client certificate via admin portal
E-VPN-002	DNS resolution failure	Check DNS servers in VPN config, verify connectivity
E-VPN-003	Split tunnel misconfiguration	Review routing table, adjust split-include rules
E-VPN-004	MTU size mismatch	Set MTU to 1400 or lower, test with ping -f -l 1400
E-VPN-005	Firewall blocking UDP 1194	Work with customer's network team to whitelist

3.2.2 Case Study: Enterprise Client VPN Rollout

Technical Context

Real Scenario (Sanitized)
Client: Large financial institution (5,000+ employees)
Issue: 30% of MacOS users experiencing intermittent VPN disconnections
Timeline: Reported on Day 1 of production rollout
Investigation Process:

1. Analyzed VPN gateway logs - identified pattern: disconnects every 3600 seconds
2. Checked client config: `reneg-sec 3600` (default)
3. Found conflict with corporate proxy auto-configuration (PAC)
4. Root cause: Renegotiation triggered proxy re-evaluation, causing timeout

Solution:

- Updated client config: `reneg-sec 0` (disable automatic renegotiation)
- Implemented keep-alive: `ping 10, ping-restart 60`
- Pushed updated profile to affected MacOS devices
- Issue resolved within 4 hours of identification

Lessons Learned: Always consider enterprise network infrastructure (proxies, DLP) when troubleshooting VPN at scale.

3.3.1 Detection & Response

Customers may report that users are bypassing content filtering. Common methods include:

1. **VPN within VPN**: User installs personal VPN app
2. **DNS Tunneling**: Using alternative DNS resolvers (8.8.8.8, 1.1.1.1)
3. **Proxy Services**: Web-based proxies or browser extensions
4. **Encrypted SNI**: Modern TLS techniques to hide destination

How to Investigate:

```
# Query suspicious network patterns
curl -X POST "https://api.xxx.com/v2/analytics/query" \
  -H "Authorization: Bearer xxxTOKENxxx" \
  -d '{
    "org_id": "CLIENT-ABC",
    "query_type": "anomaly_detection",
    "time_range": "last_24h",
    "filters": ["high_encrypted_traffic", "non_standard_ports"]
  }'
```

Recommended Policies:

- Block known VPN/proxy applications via MDM
- Enforce DNS resolution through corporate DNS only
- Enable SSL/TLS inspection (requires enterprise certificate deployment)
- Alert on traffic to known proxy domains

3.4.1 Platform-Specific Issues

iOS/iPadOS Enrollment:

- **Supervised vs. Non-Supervised**: Verify enrollment type matches policy requirements
- **ABM/ASM Integration**: Check Apple Business Manager token validity
- **Profile Installation**: Users must install profile via Settings app (not Safari)
- **Common Error**: ”Profile Installation Failed” → Check certificate chain

Android Enrollment:

- **Work Profile vs. Fully Managed**: Understand deployment mode
- **Google Play Services**: Verify GMS is available and updated
- **DPC Identifier**: Confirm correct DPC token for enrollment
- **Common Error**: ”Device not compatible” → Check Android version (min 8.0)

Windows Enrollment:

- **Azure AD Join vs. Hybrid Join**: Check identity configuration
- **Autopilot Integration**: Verify device is registered in Autopilot
- **Group Policy Conflicts**: Check for conflicting on-prem GPOs
- **Common Error**: ”Something went wrong” → Check event logs (Event Viewer)

Important Notes

Critical: Data Privacy Considerations
When troubleshooting enrollment issues, be mindful of employee privacy:

- Only access personal device data when explicitly authorized
- Follow GDPR/LGPD compliance requirements for EU/Brazil customers
- Inform users about data collection scope during enrollment
- Document consent in support tickets

Success Indicators

Lab 3.1: Multi-Factor Authentication Issue
Scenario: Customer reports: ”10 users in Finance department cannot complete MFA enrollment. They receive SMS codes but system rejects them as invalid.”
Investigation Checklist:

- Check MFA provider status dashboard
- Verify phone numbers in user directory (format: +1-XXX-XXX-XXXX)
- Check time sync on authentication servers (NTP)
- Review MFA policy for Finance department org unit
- Test with your own device to reproduce issue
- Check API logs for TOTP validation failures

Root Cause Discovery: Time drift on authentication server (45 seconds behind). TOTP tokens generated by SMS were valid for user’s actual time but rejected by server’s clock.
Resolution: Synchronize NTP on auth servers, implement monitoring alert for time drift ¿ 10 seconds.
Time to Resolution: 45-60 minutes (including investigation)

Important Notes

Advanced Competency Check
You should now be able to:

1. Diagnose VPN issues without supervision
2. Identify platform-specific enrollment problems
3. Handle privacy concerns appropriately
4. Recognize when time sync causes auth failures
5. Escalate complex issues with full diagnostic context

Target: 85% of L2 tickets handled independently

Key Learning Objectives

Week 4 Learning Objectives

- Master professional customer communication
- Learn escalation procedures and criteria
- Contribute to knowledge base effectively
- Complete final assessment
- Achieve 95% readiness for autonomous work

4.2.1 Technical Communication Framework

When communicating with customers, follow the **CLEAR** framework:

- **C**oncise: Get to the point quickly, respect customer's time
- **L**ucid: Use clear language, avoid jargon unless customer is technical
- **E**mpathetic: Acknowledge impact of issue on their business
- **A**ctionable: Always provide next steps, even if escalating
- **R**esponsive: Set clear expectations on response times

4.2.2 Email Template: Initial Response

Subject: Re: [Ticket #12345] VPN Connection Issues - Investigation Update
Dear [Customer Name],
Thank you for contacting Company X Support. I'm Gabriel from the Level 2 Technical team, and I'll be assisting you with the VPN connection issues reported for your organization.
Issue Summary: Users experiencing authentication failures when connecting to VPN (Error: E-VPN-001)
Initial Findings:
- Reviewed logs from VPN gateway vpn-gw-xxx.xxx.com
- Identified certificate expiration for 15 user accounts
- Root cause: Bulk certificate renewal process incomplete

Next Steps:
- **[In Progress]** Regenerating certificates for affected users (ETA: 30 minutes)
- **[Scheduled]** Pushing updated certificates via MDM (ETA: 1 hour)
- **[Planned]** Implementing automated renewal alerts to prevent recurrence

I'll update you within 2 hours with resolution confirmation. If you have any questions in the meantime, please reply to this email or call our support hotline.
Best regards,
Gabriel Mazer
Senior Support Engineer, Level 2
Company X — Support Team

Know when to escalate and who to escalate to:

Issue Type	Escalate To	When to Escalate
Platform bug/outage	L3 - Engineering	Issue affects multiple customers, or root cause is code-related
Infrastructure issue	L3 - DevOps/SRE	Server/network problems, scaling issues
Security incident	Security Team	Suspected breach, unauthorized access, data leak
Billing/licensing	Customer Success	Contract questions, licensing counts, upgrades
Feature request	Product Management	Customer needs functionality not currently available

Escalation Best Practices:

1. Always exhaust L2 options before escalating
2. Provide complete context in escalation (logs, reproduction steps, timeline)
3. Stay on the ticket after escalation to assist L3 if needed
4. Never tell customer ”I’m escalating because I don’t know” - use professional language

4.4.1 When to Create a KB Article

Create a new article when:

- You’ve resolved the same issue 3+ times
- Issue resolution is not documented elsewhere
- Solution involves non-obvious steps
- Customer feedback indicates gap in documentation

4.4.2 KB Article Structure

Every KB article should follow this template:

1. **Title**: Clear, searchable, includes error code if applicable
2. **Symptoms**: Exactly what the user experiences
3. **Environment**: OS, platform version, affected components
4. **Root Cause**: Technical explanation (can be hidden for end-users)
5. **Resolution Steps**: Numbered, with screenshots
6. **Verification**: How to confirm the issue is resolved
7. **Related Articles**: Links to similar topics
8. **Tags**: For searchability (e.g., vpn, ios, authentication)

Technical Context

KB Quality Standards
A good KB article should:

- Be understandable by someone with basic technical knowledge
- Include command examples with actual syntax
- Have screenshots for GUI-based procedures
- Be tested by at least one peer before publishing
- Be updated when product changes affect the solution

Success Indicators

Week 4 Competency Assessment
By the end of Week 4, you should be able to check off ALL of these items:
Technical Skills:

- Explain platform architecture without referring to notes
- SSH into test environments and navigate logs confidently
- Use API endpoints to query device/user information
- Troubleshoot VPN issues independently for 80% of cases
- Identify when to escalate vs. continue investigation

Communication & Process:

- Write professional customer-facing emails
- Document ticket resolutions with sufficient detail
- Create at least one KB article following the template
- Successfully complete shadowing sessions with senior analysts
- Understand escalation paths and when to use them

Tools & Systems:

- Navigate internal KB with ease
- Use ticketing system effectively (create, update, close)
- Access monitoring dashboards and interpret metrics
- Utilize collaboration tools (Slack, documentation wiki)

Shadowing Completed:

- Observed 5+ ticket resolutions by senior analysts
- Handled 3+ tickets with supervision
- Participated in team daily standup meetings
- Completed peer review of your documentation

After completing this 4-week program, you'll transition to handling tickets independently with the following guidelines:

Technical Context

Week 5+ Expectations
Ticket Load: Start with 5-8 tickets per day, gradually increase to 12-15
Response Times:

- Initial response: Within 2 hours (business hours)
- Status updates: Every 24 hours for open tickets
- Resolution target: 80% of L2 tickets within 48 hours

Support Structure:

- Weekly 1-on-1 with team lead
- Access to #l2-support Slack channel for quick questions
- Monthly knowledge-sharing sessions
- Quarterly performance reviews

5.1.1 Linux/Unix Commands

Command	Purpose
`tail -f /var/log/xxx.log`	Monitor log file in real-time
`grep "ERROR" /var/log/xxx.log`	Search for errors in log file
`netstat -tuln`	Show listening ports and connections
`systemctl status service-name`	Check service status
`journalctl -u service-name -f`	Follow service logs (systemd)
`df -h`	Check disk space usage
`top` or `htop`	Monitor system resources
`ping -c 4 xxx.xxx.com`	Test network connectivity
`traceroute xxx.xxx.com`	Trace network path
`nslookup xxx.xxx.com`	DNS lookup

5.1.2 OpenVPN Specific

```
# Check OpenVPN service
systemctl status openvpn@server

# View active connections
cat /var/log/openvpn/openvpn-status.log

# Test configuration
openvpn --config /etc/openvpn/server.conf --test-crypto

# Generate client certificate
./easyrsa build-client-full client-name nopass
```

5.1.3 API Quick Reference

```
# Get device details
curl -H "Authorization: Bearer $TOKEN" \
  https://api.xxx.com/v2/devices/{device_id}

# List VPN logs for organization
curl -H "Authorization: Bearer $TOKEN" \
  "https://api.xxx.com/v2/logs/vpn?org_id=CLIENT-ABC&limit=100"

# Trigger device sync
curl -X POST -H "Authorization: Bearer $TOKEN" \
  https://api.xxx.com/v2/devices/{device_id}/actions/sync

# Check API rate limit status
curl -H "Authorization: Bearer $TOKEN" \
  https://api.xxx.com/v2/rate-limit/status
```

5.2.1 Platform Error Codes

Code	Category	Meaning
E-AUTH-001	Authentication	Invalid credentials
E-AUTH-002	Authentication	MFA token expired
E-AUTH-003	Authentication	Account locked (too many attempts)
E-VPN-001	VPN	Certificate expired or invalid
E-VPN-002	VPN	DNS resolution failure
E-VPN-003	VPN	Split tunnel configuration error
E-VPN-004	VPN	MTU size mismatch
E-VPN-005	VPN	Port blocked by firewall
E-MDM-001	MDM	Device enrollment failed
E-MDM-002	MDM	Policy application timeout
E-MDM-003	MDM	Certificate provisioning error
E-API-001	API	Invalid API token
E-API-002	API	Rate limit exceeded (429)
E-API-003	API	Insufficient permissions (403)
E-CF-001	Content Filter	Category list update failed
E-CF-002	Content Filter	SSL inspection certificate error

5.3.1 Internal Resources

- **Knowledge Base**: `https://kb.xxx.com` (1,500+ articles)
- **Internal Wiki**: `https://wiki.xxx.com/support`
- **API Documentation**: `https://docs.api.xxx.com`
- **Monitoring Dashboard**: `https://monitor.xxx.com`
- **Status Page**: `https://status.xxx.com` (for customer-facing incidents)

5.3.2 Team Contacts

Team	Contact	Escalation Path
L2 Support Lead	xxx@xxx.com	Direct escalation for guidance
L3 Engineering	l3-eng@xxx.com	Technical escalation
DevOps/SRE	devops@xxx.com	Infrastructure issues
Security Team	security@xxx.com	Security incidents
Customer Success	cs@xxx.com	Account/billing issues

5.3.3 On-Call Rotation

For after-hours critical issues (P1/P0), use the on-call rotation:

- **PagerDuty**: Critical alerts auto-page on-call engineer
- **Slack**: `@oncall-l3` mentions on-call engineer
- **Phone**: Emergency hotline +XX-XXX-XXX-XXXX (P0 only)

ABM/ASM Apple Business Manager / Apple School Manager - Apple's device enrollment programs

API Application Programming Interface - programmatic access to platform features

APNS Apple Push Notification Service - iOS notification delivery system

DPC Device Policy Controller - Android enterprise management component

FCM Firebase Cloud Messaging - Android notification system

IKEv2 Internet Key Exchange version 2 - VPN protocol

LDAP Lightweight Directory Access Protocol - directory service protocol

MDM Mobile Device Management - device enrollment and management system

MFA Multi-Factor Authentication - additional authentication beyond password

MTU Maximum Transmission Unit - maximum packet size

OAuth Open Authorization - token-based authentication standard

PAC Proxy Auto-Configuration - automatic proxy configuration

RDP Remote Desktop Protocol - Windows remote access protocol

SNI Server Name Indication - TLS extension for hostname identification

SSH Secure Shell - encrypted remote access protocol

SSL/TLS Secure Sockets Layer / Transport Layer Security - encryption protocols

TOTP Time-based One-Time Password - MFA token generation method

VPN Virtual Private Network - secure remote access technology

6.2.1 Recommended Reading

- **Books**:
  - "The Practice of System and Network Administration" - Limoncelli et al.
  - "Site Reliability Engineering" - Google SRE Team
  - "The Phoenix Project" - Gene Kim (DevOps principles)
- **Online Courses**:
  - Linux Academy - "Linux System Administrator"
  - Udemy - "OpenVPN: Complete Guide"
  - Coursera - "Network Security & Database Vulnerabilities"
- **Technical Documentation**:
  - OpenVPN Official Documentation
  - RFC 4251 (SSH Protocol)
  - RFC 6749 (OAuth 2.0)

6.2.2 Certifications to Consider

Building these certifications can enhance your career growth:

- **CompTIA Security+** - Foundational security knowledge
- **CCNA** - Networking fundamentals
- **Linux Professional Institute Certification (LPIC-1)** - Linux administration
- **Certified Information Systems Security Professional (CISSP)** - Advanced security

Version	Date	Changes
1.0	Oct 2024	Initial release
1.5	Dec 2024	Added API reference section, updated error codes
2.0	Feb 2025	Major update: Added Week 3 case studies, expanded troubleshooting frameworks
2.1	Mar 2025	Current version: Added MFA troubleshooting, updated contact information

This playbook is a living document. Your feedback is valuable for improving the onboarding experience for future analysts.

Technical Context

Submit Feedback
How to contribute:

- Found an error or outdated information? Email: `kb-feedback@xxx.com`
- Have a suggestion for additional content? Submit via internal wiki
- Want to share a case study? Contact your team lead

Recognition: Contributors who significantly improve this playbook will be acknowledged in the version history and receive team recognition.

Welcome to the Team!

Remember: You're not alone in this journey. Reach out to your colleagues, ask questions, and don't hesitate to seek help. We're here to support your growth.

Company X Support Team