Forcepoint

Advanced Malware Detection and Protection

2.0 - 2.1

On-Premises Migration Guide

Revision A

Contents

Introduction

Purpose of This Guide

This guide provides step-by-step instructions for migrating your AMDP On-Premises deployment from 2.0 (based on Ubuntu 20.04 LTS) to 2.1 (based on Ubuntu 24.04 LTS). The migration involves creating backups of your existing systems, performing fresh installations, and restoring your configurations and data.

Why Migrate?

Advanced Malware Detection and Protection On-Premises 2.1 is the latest version of Forcepoint's Advanced Threat Protection and detection sandbox.

Version 2.1 is an important security and technology update release that ensures continued stability and security updates for your AMDP On-Premises deployments.

Target Audience

This guide is intended for Customers' Security Administrators running AMDP On-Premises.

Prerequisites

- Root access to both Manager and Engine systems.
- Basic understanding of Linux command line operations.
- Familiarity with backup and restore procedures.
- Knowledge of PostgreSQL database operations (for Manager).

Migration Overview

Supported Upgrade Path

The migration follows a **backup-reinstall-restore** approach:

```
Phase 1: Backup - 2.0 (Ubuntu 20.04)

• Run backup_manager.sh on Manager (20.04)
• Run backup_engine.sh on Engine(s) (20.04)
• Transfer backup archives to a secure location

                                ↓


Phase 2: Fresh Installation - 2.1 (Ubuntu 24.04)

• Install Manager using new Ubuntu 24.04 ISO
• Install Engine(s) using new Ubuntu 24.04 ISO
• Complete initial setup wizard for both systems

                                ↓


Phase 3: Restore - 2.1 (Ubuntu 24.04)

• Transfer 2.0 backup archives to new systems
• Run restore_manager.sh on Manager (24.04)
• Run restore_engine.sh on Engine(s) (24.04)
• Regenerate VMs in Engine(s) & verify AMDP functionality
```

Important

This is **not** an in-place upgrade. Fresh installations using the new AMDP-OP 2.1 (Ubuntu 24.04) ISO need to be performed for both Manager and Engine systems.

High-Level Risks and Considerations

Risks

Risk	Impact	Mitigation
Data Loss	HIGH	Verify backup integrity before proceeding.
Service Downtime	MEDIUM	Schedule during maintenance window.
Database Corruption	HIGH	If feasible, test database restore in non-production environment first.

Risk	Impact	Mitigation
Configuration Errors	MEDIUM	Document all custom configurations before backup.
VM Compatibility	LOW	Sandbox VMs must be regenerated post-migration.

Key Considerations

- Sandbox VMs are NOT included in backups, only the VM configuration is backed up. So, sandbox VMs must be regenerated after restoring.
- Active sandbox jobs (if any) will be lost during migration.
- Network connectivity must be maintained for proper service restoration.
- After fresh install of Manager, the PostgreSQL database will be emptied and restored (user confirmation required).
- Forcepoint recommends the original 20.04 systems should remain available as a fallback until migration is fully validated, where this is feasible.

Pre-Migration Checklist

Pre-Migration Tasks

For Manager

- Document current database name (usually `triage-www`).

```
grep "dbname=" /var/lib/triage-frontend/frontend.yaml
```

- Verify Hatching Triage token exists.

```
grep "ht_token:" /opt/amd/tts/conf/configuration.yaml
```

- It is recommended to stop active sandbox submissions.
- Record the current Forcepoint Manager License Key in use.

```
grep license_key /etc/amd/amd-mgr.ini
```

- Record current system IP addresses and hostname.

```
hostname
ip addr show
```

For Engine

- Document VM configurations.

```
amd_setup sandbox --vmstatus
```

- Record the current Forcepoint Engine License Key in use.

```
grep license_key /etc/amd/amd-mgr.ini
```

- Record current system IP addresses and hostname.

```
hostname
ip addr show
```

General Preparation

- Download the latest Advanced Malware Detection and Protection 2.1 ISO file from **Downloads** > **AMDP On-Premises** section of Forcepoint Customer Hub.
- Prepare installation media.
- Document any custom firewall rules or network configurations.
- Backup any custom scripts or tools not included in standard installation.
- Prepare temporary storage location for backup archives:

Required Scripts

Download `forcepoint-amdp-2.0-to-2.1-migration.zip` from **Downloads** > **AMDP On-Premises** section of Forcepoint Customer Hub. Extract the following migration scripts from the downloaded archive.

For Manager

- `backup_manager.sh` - Creates backup on Ubuntu 20.04.
- `restore_manager.sh` - Restores backup on Ubuntu 24.04.

For Engine

- `backup_engine.sh` - Creates backup on Ubuntu 20.04.
- `restore_engine.sh` - Restores backup on Ubuntu 24.04.

Migration Steps

Create Backups from Ubuntu 20.04

Manager Backup

Steps

1) Transfer backup script to Manager.

2) Make script executable and run backup.

```
# Make script executable
chmod +x /root/backup_manager.sh

# Run backup (as root)
./backup_manager.sh
```

Expected Output (verify for any errors):

```
[2025-12-09 10:00:00] ================================================
[2025-12-09 10:00:00] Starting AMDP On-Prem Manager Backup Process
[2025-12-09 10:00:00] ================================================
[2025-12-09 10:00:00] Backup directory: /root/amdp_mgr_backup_20251209_100000
[2025-12-09 10:00:00] Archive will be created at: /root/amdp_mgr_backup_20251209_100000.tar.gz

...
[2025-12-09 10:45:30] Backup completed successfully!
[2025-12-09 10:45:30] ================================================
[2025-12-09 10:45:30] Backup archive: /root/amdp_mgr_backup_20251209_100000.tar.gz
[2025-12-09 10:45:30] Log file: /root/amdp_mgr_backup_20251209_100000.log
[2025-12-09 10:45:30]
[2025-12-09 10:45:30] Please transfer this file to the Ubuntu 24.04 server for restoration.
[2025-12-09 10:45:30] You can use scp, rsync, or other file transfer methods.
```

3) Verify backup archive.

```
# Check backup file exists and size
ls -lh /root/amdp_mgr_backup_20251209_100000.tar.gz

# Verify backup log for any errors
tail -100 /root/amdp_mgr_backup_20251209_100000.log
```

4) Transfer backup to secure location.

```
# Example: Copy to network share
scp /root/amdp_mgr_backup_20251209_100000.tar.gz user@backup-server:/backups/amdp/

# Example: Copy to external storage
cp /root/amdp_mgr_backup_20251209_100000.tar.gz /mnt/external-storage/

# Verify transfer
sha256sum /root/amdp_mgr_backup_20251209_100000.tar.gz
# Compare checksum at destination
```

Engine Backup

Steps

1) Transfer backup script to Engine.

2) Make script executable and run backup.

```
# Make script executable
chmod +x /root/backup_engine.sh

# Run backup (as root)
./backup_engine.sh
```

Expected Output (verify for any errors):

```
[2025-12-09 11:00:00] ================================================
[2025-12-09 11:00:00] Starting AMDP On-Prem Engine Backup Process
[2025-12-09 11:00:00] ================================================
[2025-12-09 11:00:00] Backup directory: /root/amdp_eng_backup_20251209_110000
[2025-12-09 11:00:00] Archive will be created at: /root/amdp_eng_backup_20251209_110000.tar.gz
...
[2025-12-09 11:25:30] Backup completed successfully!
[2025-12-09 11:25:30] ================================================
[2025-12-09 11:25:30] Backup archive: /root/amdp_eng_backup_20251209_110000.tar.gz
[2025-12-09 11:25:30] Log file: /root/amdp_eng_backup_20251209_110000.log
[2025-12-09 11:25:30]
[2025-12-09 11:25:30] Please transfer this file to the Ubuntu 24.04 server for restoration.
[2025-12-09 11:25:30] You can use scp, rsync, or other file transfer methods.
```

3) Verify backup archive.

```
# Check backup file exists and size
ls -lh /root/amdp_eng_backup_20251209_100000.tar.gz

# Verify backup log for any errors
tail -100 /root/amdp_eng_backup_20251209_100000.log
```

4) Transfer backup to secure location.

```
# Example: Copy to network share
scp /root/amdp_eng_backup_20251209_100000.tar.gz user@backup-server:/backups/amdp/

# Verify transfer
sha256sum /root/amdp_eng_backup_20251209_100000.tar.gz
```

Backup Verification Checklist

Before proceeding to installation:

Manager Backup:

- Backup archive created successfully (`amdp_mgr_backup_*.tar.gz`).
- No errors in backup log file.
- Archive size is reasonable (not 0 bytes).
- Backup transferred to secure location.
- Checksum verified at destination.

- ht_token was successfully backed up (check log).
- Database dump is not empty (check log for size).

Engine Backup:

- Backup archive created successfully (`amdp_eng_backup_*.tar.gz`).
- No errors in backup log file.
- Archive size is reasonable (not 0 bytes).
- Backup transferred to secure location.
- Checksum verified at destination.

Fresh Installation Using Ubuntu 24.04 ISO

- Refer AMDP On-Premises 2.1 Deployment Guide for fresh manager and engine installation.
- Ensure the same IPs and FQDNs are entered during manager and engine installation.

Restore Backups to Ubuntu 24.04

Manager Restore

Steps

1) Transfer backup archive and restore script.

2) Prepare for restore.

```
# Make restore script executable
chmod +x /root/restore_manager.sh
```

3) Run restore script.

```
# Run restore (as root)
./restore_manager.sh /home/admin/amdp_mgr_backup_20251209_100000.tar.gz
```

4) Follow interactive prompts.
   The script will:
   a) Extract and validate the backup archive.
   b) Display the backup manifest.
   c) Check if TTS configuration and `ht_token` exist.
   d) Check database existence and status.
   e) **If database is not empty**, prompt for action:

```
WARNING: Database 'triage-www' is not empty (contains XX tables)

Choose an option:
  1) Empty the database and continue with restore
  2) Cancel restore

Enter your choice (1/2):
```

      i) Choose **1** to empty the database.
      ii) If you choose 1, you'll need to type `DELETE ALL` to confirm

Expected Output:

```
[2025-12-09 14:00:00] ================================================
[2025-12-09 14:00:00] Starting AMDP On-Prem Manager Restore Process
[2025-12-09 14:00:00] ================================================
[2025-12-09 14:00:00] Backup archive: /home/admin/amdp_mgr_backup_20251209_100000.tar.gz
[2025-12-09 14:00:00] Target OS: Ubuntu 24.04
...
[2025-12-09 14:35:00] Restore completed successfully!
[2025-12-09 14:35:00] ================================================
[2025-12-09 14:35:00] Log file: /root/amdp_mgr_restore_20251209_140000.log
[2025-12-09 14:35:00]
[2025-12-09 14:35:00] Next steps:
[2025-12-09 14:35:00]   1. Reboot the system to ensure all services start cleanly
[2025-12-09 14:35:00]   2. Test AMDP functionality
```

5) Reboot manager.

6) Test AMDP functionality after connecting to an engine.

Engine Restore

Steps

1) Transfer backup archive and restore script.

2) Prepare for restore.

```
# Make restore script executable
chmod +x /root/restore_engine.sh
```

3) Run restore script.

```
# Run restore (as root)
./restore_engine.sh /root/amdp_eng_backup_20251209_110000.tar.gz
```

4) Follow interactive prompts.

The script will:

a) Extract and validate the backup archive.

b) Display the backup manifest.

c) Show final confirmation:

```
WARNING: This restore process will:
  1. Overwrite existing configuration files
  2. Overwrite existing data in /var/lib/triage and /var/lib/sandbox

Do you want to continue? (yes/no):
```

- Type **yes** to proceed.

Expected Output:

```
[2025-12-09 15:00:00] ==================================================
[2025-12-09 15:00:00] Starting AMDP On-Prem Engine Restore Process
[2025-12-09 15:00:00] ==================================================
[2025-12-09 15:00:00] Backup archive: /root/amdp_eng_backup_20251209_110000.tar.gz
[2025-12-09 15:00:00] Target OS: Ubuntu 24.04
...
[2025-12-09 15:25:00] Restore completed successfully!
[2025-12-09 15:25:00] ==================================================
[2025-12-09 15:25:00] Log file: /root/amdp_eng_restore_20251209_150000.log
[2025-12-09 15:25:00]
[2025-12-09 15:25:00] Next steps:
[2025-12-09 15:25:00]   1. Reboot the system to ensure all services start cleanly
[2025-12-09 15:25:00]   2. Regenerate sandbox VMs (they are not included in the backup)
[2025-12-09 15:25:00]   3. Connect to the same AMDP Manager as before and test sandboxing
 functionality
```

5) Reboot engine.

Post-Restore Configuration

Manager Post-Restore Tasks

Steps

1) Verify Admin Web Portal access.
   Open browser to: `https://<manager-ip>/`

2) Verify Engine-Manager connectivity.
   Open the Machines tab in the Admin Web Portal.

3) Connect product to new AMDP manager.
   Refer to the below documentation:
   a) Web: Web Security licensing and configuration
   b) NGFW: Configuring AMDP on Secure SD-WAN via SMC GUI
   c) Email: Selecting advanced file analysis platform

Engine Post-Restore Tasks

Steps

1) Regenerate sandbox VMs.
   Sandbox VMs are NOT included in backups, only the VM configuration is backed up. Therefore, sandbox VMs must be regenerated after a restore.

   Check VM build status and trigger VM build:

```
# amd_setup sandbox --vmstatus
# amd_setup sandbox --vmbuild
```

2) Enable AMDP sandboxing in the product.
   Once VMs are built, test AMDP sandboxing functionality by enabling sandboxing in the product.

Troubleshooting

Backup Issues

Issue: Backup script fails with "ht_token not found"

Symptoms:

```
[2025-12-09 10:00:05] ERROR: ht_token value not found or is default in /opt/amd/tts/conf/
configuration.yaml
```

Cause: The TTS configuration file doesn't have a valid ht_token configured.

Resolution:

1) Check if the file exists:

```
ls -la /opt/amd/tts/conf/configuration.yaml
```

2) Check token value:

```
grep "ht_token:" /opt/amd/tts/conf/configuration.yaml
```

3) If missing or default value, that indicates an incomplete installation or corrupted `configuration.yaml` file. In this case, a re-install will be required.

Issue: PostgreSQL dump fails

Symptoms:

```
[2025-12-09 10:15:00] ERROR: Failed to create PostgreSQL dump
```

Cause: Database connection issues or insufficient permissions.

Resolution:

1) Verify PostgreSQL is running:

```
systemctl status postgresql
```

2) Check database exists:

```
sudo -u postgres psql -l | grep triage-www
```

3) Test database connection:

```
sudo -u postgres psql -d triage-www -c "SELECT version();"
```

4) If `triage-www` database does not exist, a re-install will be required.

Restore Issues

Issue: Database restore fails with permission denied

Symptoms:

```
psql: error: /tmp/tmp.XXXXXXX: Permission denied
```

Cause: Temporary file permissions issue with PostgreSQL user.

Resolution: The restore script should handle this automatically by using the correct database user (`triage-www`).

If the issue persists:

1) Verify database user exists:

```
id triage-www
```

2) Check PostgreSQL service:

```
systemctl status postgresql
```

Issue: ht_token validation fails

Symptoms:

```
[2025-12-09 14:05:00] ERROR: ht_token value in backup is empty or invalid
```

Cause: Backup archive contains invalid or missing ht_token.

Resolution: This indicates a corrupted backup. Use a different backup archive.

Issue: Restore validation fails - missing directories

Symptoms:

```
[2025-12-09 14:02:00] ERROR: Missing required directory: config/triage
[2025-12-09 14:02:00] ERROR: Backup validation failed
```

Cause: Corrupted or incomplete backup archive.

Resolution:

1) Verify backup archive integrity:

```
tar -tzf amdp_mgr_backup_*.tar.gz | head -20
```

2) Try re-transferring backup from source location.

3) Create new backup from original 20.04 system if still available.

Issue: Configuration files not restored

Symptoms:

```
cp: cannot stat '/root/.../config/triage/*.yaml': No such file or directory
```

Cause: Backup extraction issue or missing files in backup.

Resolution:

1) Manually extract and inspect backup:

```
cd /tmp
tar -xzf /root/amdp_mgr_backup_*.tar.gz
ls -laR amdp_mgr_backup_*/
```

2) Verify backup contains expected files.

3) If files missing, create new backup from source system.

Issue: rsync fails during data restore

Symptoms:

```
rsync: failed to set permissions on ...: Operation not permitted
```

Cause: File ownership or permission issues.

Resolution:

1) Ensure running as root:

```
whoami  # Should show: root
```

2) Check target directory permissions:

```
ls -la /var/lib/triage/
```

3) Fix ownership if needed:

```
chown -R triage:triage /var/lib/triage/
```

Published 19 December 2025