# Project Charter: Operation Sentinel Purge (Forensic & Reverse Engineering)

**Project Name:** Operation Sentinel Purge **Target Device:** Samsung S24 Ultra (S24U Gen 2.2) **Host Environment:** Fedora COSMIC Atomic (Samsung Spin NP730QAA02US) **Root Workspace:** ~/storage/documents/S24U_SentinelVault/AIMS-S24U/ **Objective:** Execute a NIST-aligned forensic investigation, decompilation of malicious assets, and permanent service termination.

## 1. Operational Workflow (Order of Operations)

1. **Phase I: Preservation (Live State):** Capture real-time network traffic and DNS cache.
2. **Phase II: Deep Audit:** Identify the specific "Device Admin" package and policy flags.
3. **Phase III: Asset Discovery:** Map hidden directories and SQLite databases.
4. **Phase IV: Imaging:** Bit-for-bit userdata capture.
5. **Phase V: Forensic Extraction:** Pull target APKs and databases.
6. **Phase V.I: Reverse Engineering:** Decompile APKs to Java source using jadx.
7. **Phase VI: Verification:** Hash matching and account export validation.
8. **Phase VII: Termination (The Nuke):** Final device sanitation and package removal.

## 2. Phase I & II: Preservation & Deep Audit

### 2.1 Live Network Socket Monitor

```
cat >
~/storage/documents/S24U_SentinelVault/AIMS-S24U/scripts/01_network_monitor.sh << 'EOF'
#!/bin/bash
# AIMS Surveillance: Real-time Network Monitor
LOG_DIR="~/storage/documents/S24U_SentinelVault/AIMS-S24U/audit_logs/network"
mkdir -p "$LOG_DIR"
LOG_FILE="$LOG_DIR/sockets_$(date +%Y%m%d_%H%M%S).txt"

echo "[*] Monitoring active connections... Press Ctrl+C to stop."
while true; do
    echo "--- $(date) ---" >> "$LOG_FILE"
    adb shell "netstat -tunp 2>/dev/null || netstat -tun" | grep -v
"127.0.0.1" >> "$LOG_FILE"
    sleep 2
done
EOF
```

### 2.2 System Policy & Restriction Audit

```
cat >
~/storage/documents/S24U_SentinelVault/AIMS-S24U/scripts/02_restrictio
n_audit.sh << 'EOF'
#!/bin/bash
# AIMS Audit: Policy Restriction Discovery
OUT_DIR="~/storage/documents/S24U_SentinelVault/AIMS-S24U/audit_logs"
mkdir -p "$OUT_DIR"

echo "[*] Identifying Device Admins..."
adb shell dumpsys device_policy > "$OUT_DIR/device_policy.txt"

echo "[*] Checking for DISALLOW_CONFIG_NOTIFICATIONS..."
grep -i "DISALLOW_CONFIG_NOTIFICATIONS" "$OUT_DIR/device_policy.txt"

echo "[*] Auditing Authenticator AppOps..."
adb shell appops get com.google.android.apps.authenticator >
"$OUT_DIR/auth_appops.txt"

echo "[*] Dumping DNS Cache & ARP Table..."
adb shell "ip neigh" > "$OUT_DIR/arp_table.txt"
adb shell dumpsys dnsresolver > "$OUT_DIR/dns_cache.txt"
EOF
```

## 3. Phase III & IV: Asset Discovery & Imaging

### 3.1 Hidden Asset Hunt

```
cat >
~/storage/documents/S24U_SentinelVault/AIMS-S24U/scripts/03_asset_hunt
.sh << 'EOF'
#!/bin/bash
# AIMS Extraction: Hidden Asset Discovery
SCAN_DIR="~/storage/documents/S24U_SentinelVault/AIMS-S24U/audit_logs/
assets"
mkdir -p "$SCAN_DIR"

echo "[*] Scanning for hidden .nomedia folders..."
adb shell "find /sdcard -name '.nomedia'" >
"$SCAN_DIR/nomedia_locations.txt"

echo "[*] Mapping all SQLite databases..."
adb shell "find /sdcard -name '*.db' -o -name '*.sqlite'" >
"$SCAN_DIR/external_databases.txt"
```

```
echo "[*] Listing hidden directories ('.')..."
adb shell "find /sdcard -type d -name '.*'" >
"$SCAN_DIR/hidden_dirs.txt"
EOF
```

### 3.2 Physical Forensic Imaging

```
cat >
~/storage/documents/S24U_SentinelVault/AIMS-S24U/scripts/04_imaging.sh
<< 'EOF'
#!/bin/bash
# AIMS Forensic: Physical Image Stream
BACKUP_PATH="~/storage/documents/S24U_SentinelVault/AIMS-S24U/backups"
mkdir -p "$BACKUP_PATH"
TS=$(date +%Y%m%d)

echo "[*] Streaming userdata partition..."
adb shell su -c "dd if=/dev/block/by-name/userdata" | gzip -c >
"$BACKUP_PATH/S24U_userdata_$TS.img.gz"

echo "[*] Generating Checksum..."
sha256sum "$BACKUP_PATH/S24U_userdata_$TS.img.gz" >
"$BACKUP_PATH/checksum.txt"
EOF
```

## 4. Phase V & V.I: Extraction & Reverse Engineering

### 4.1 Forensic Pull (DBs & APKs)

```
cat >
~/storage/documents/S24U_SentinelVault/AIMS-S24U/scripts/05_extraction
.sh << 'EOF'
#!/bin/bash
# AIMS Extraction: Database & APK Pull
DB_LOG="~/storage/documents/S24U_SentinelVault/AIMS-S24U/audit_logs/as
sets/external_databases.txt"
OUT_DIR="~/storage/documents/S24U_SentinelVault/AIMS-S24U/extracted"
mkdir -p "$OUT_DIR/db" "$OUT_DIR/apks"

echo "[*] Pulling SQLite Databases..."
while read -r remote; do
    adb pull "$remote" "$OUT_DIR/db/$(echo "$remote" | tr '/' '_')"
done < "$DB_LOG"

echo "[*] Summarizing database tables..."
```

```
for db in "$OUT_DIR/db"/*; do
    echo "--- $db ---" >> "$OUT_DIR/db_analysis.txt"
    sqlite3 "$db" ".tables" >> "$OUT_DIR/db_analysis.txt" 2>/dev/null
done
EOF
```

### 4.2 JADX Decompilation

```
cat >
~/storage/documents/S24U_SentinelVault/AIMS-S24U/scripts/05b_decompile
_apk.sh << 'EOF'
#!/bin/bash
# AIMS Reverse Engineering: APK to Java Decompiler
TARGET_PKG=$1
EXTRACT_DIR="~/storage/documents/S24U_SentinelVault/AIMS-S24U/extracte
d/apks"
DECOMPILE_DIR="~/storage/documents/S24U_SentinelVault/AIMS-S24U/analys
is/source_$TARGET_PKG"

if [ -z "$TARGET_PKG" ]; then
    echo "Usage: ./05b_decompile_apk.sh <package_name>"
    exit 1
fi

mkdir -p "$DECOMPILE_DIR"
echo "[*] Decompiling $TARGET_PKG.apk..."
jadx -d "$DECOMPILE_DIR" "$EXTRACT_DIR/${TARGET_PKG}.apk"

echo "[*] Searching source for C2 endpoints & notification blocks..."
grep -rEi
"http|https|socket|DISALLOW_CONFIG_NOTIFICATIONS|DevicePolicyManager"
"$DECOMPILE_DIR" > "$DECOMPILE_DIR/investigation_hits.txt"
EOF
```

## 5. Phase VII: The Final Nuke

```
cat >
~/storage/documents/S24U_SentinelVault/AIMS-S24U/scripts/06_final_nuke
.sh << 'EOF'
#!/bin/bash
# AIMS: Secure Data Purge & Service Uninstall
BACKUP_PATH="~/storage/documents/S24U_SentinelVault/AIMS-S24U/backups"

cd "$BACKUP_PATH" || exit
if sha256sum -c checksum.txt; then
```

```
    echo "[!] Image Verified. Purging S24U..."
    adb shell "rm -rf /sdcard/*"
    PKGS=("com.google.android.gms" "com.google.android.gsf"
"com.microsoft.office.outlook" "com.microsoft.teams"
"com.microsoft.onedrive")
    for pkg in "${PKGS[@]}"; do
        adb shell pm uninstall -k --user 0 "$pkg"
    done
    echo "[+] Operation Complete."
else
    echo "[X] VERIFICATION FAILED. DO NOT PURGE."
    exit 1
fi
EOF
```