15 กรณีความผิดปกติของระบบเข้าออก (Access Control Anomalies)

🔒 Security Violations

1. การ์ดถูกปฏิเสธติดต่อกัน (Consecutive Card Rejections)

คำอธิบาย: การ์ดเดียวกันถูกปฏิเสธการเข้าหลายครั้งติดต่อกัน อาจเป็นการพยายาม brute force

```sql
-- n8n MySQL Node Query
SELECT
    "Card Number Hash",
    "Card Name",
    COUNT(*) as consecutive_fails,
    MIN("Date Time") as first_attempt,
    MAX("Date Time") as last_attempt,
    'CONSECUTIVE_CARD_REJECTION' as anomaly_type
FROM access_logs
WHERE "Allow" = false
    AND "Date Time" >= NOW() - INTERVAL 1 HOUR
GROUP BY "Card Number Hash", "Card Name"
HAVING consecutive_fails >= 5;
```

2. การเข้าออกนอกเวลาทำงาน (After Hours Access)

คำอธิบาย: การเข้าอาคารนอกเวลาทำงานปกติ (ก่อน 07:00 หรือหลัง 19:00)

```sql
SELECT
    "Transaction ID",
    "Date Time",
    "Card Name",
    "Location",
    "Direction",
    HOUR("Date Time") as access_hour,
    'AFTER_HOURS_ACCESS' as anomaly_type
FROM access_logs
WHERE "Allow" = true
    AND (HOUR("Date Time") < 7 OR HOUR("Date Time") > 19)
    AND DAYOFWEEK("Date Time") BETWEEN 2 AND 6  -- จันทร์-ศุกร์
    AND "Date Time" >= CURDATE();
```

3. การเข้าออกในวันหยุดสุดสัปดาห์ (Weekend Access)

คำอธิบาย: การเข้าอาคารในวันเสาร์-อาทิตย์ที่อาจไม่จำเป็น

sql

```
SELECT
    "Transaction ID",
    "Date Time",
    "Card Name",
    "Location",
    "Direction",
    DAYNAME("Date Time") as day_name,
    'WEEKEND_ACCESS' as anomaly_type
FROM access_logs
WHERE "Allow" = true
    AND DAYOFWEEK("Date Time") IN (1, 7)  -- อาทิตย์, เสาร์
    AND "Date Time" >= CURDATE() - INTERVAL 7 DAY;
```

4. Piggyback/Tailgating Detection

คำอธิบาย: คนหลายคนเข้าประตูเดียวกันในเวลาใกล้เคียงกัน (ภายใน 10 วินาที)

sql

```
SELECT
    a1."Door",
    a1."Date Time" as first_access,
    a2."Date Time" as second_access,
    a1."Card Name" as first_user,
    a2."Card Name" as second_user,
    ABS(TIMESTAMPDIFF(SECOND, a1."Date Time", a2."Date Time")) as time_diff,
    'POTENTIAL_TAILGATING' as anomaly_type
FROM access_logs a1
JOIN access_logs a2 ON a1."Door" = a2."Door"
WHERE a1."Allow" = true AND a2."Allow" = true
    AND a1."Direction" = a2."Direction"
    AND a1."Card Number Hash" != a2."Card Number Hash"
    AND ABS(TIMESTAMPDIFF(SECOND, a1."Date Time", a2."Date Time")) <= 10
    AND a1."Date Time" >= NOW() - INTERVAL 1 HOUR;
```

5. การเข้าออกผิดลำดับ (Invalid Entry/Exit Sequence)

คำอธิบาย: เข้าแล้วเข้าอีก หรือ ออกแล้วออกอีก โดยไม่มีการกลับทิศ

sql

```
WITH user_sequences AS (
  SELECT
    "Card Number Hash",
    "Card Name",
    "Direction",
    "Date Time",
    "Door",
    LAG("Direction") OVER (
      PARTITION BY "Card Number Hash"
      ORDER BY "Date Time"
    ) as prev_direction
  FROM access_logs
  WHERE "Allow" = true
    AND "Date Time" >= NOW() - INTERVAL 2 HOUR
)
SELECT
  "Card Number Hash",
  "Card Name",
  "Date Time",
  "Door",
  "Direction",
  prev_direction,
  'INVALID_ENTRY_EXIT_SEQUENCE' as anomaly_type
FROM user_sequences
WHERE "Direction" = prev_direction
  AND prev_direction IS NOT NULL;
```

🌐 Multi-Location Anomalies

6. การใช้การ์ดพร้อมกันหลายตำแหน่ง (Simultaneous Multi-Location Access)

คำอธิบาย: การ์ดเดียวกันถูกใช้ในหลายสถานที่ในเวลาเดียวกัน

```
sql
```

```
SELECT
    a1."Card Number Hash",
    a1."Card Name",
    a1."Location" as location1,
    a2."Location" as location2,
    a1."Date Time" as time1,
    a2."Date Time" as time2,
    ABS(TIMESTAMPDIFF(MINUTE, a1."Date Time", a2."Date Time")) as time_diff,
    'SIMULTANEOUS_ACCESS' as anomaly_type
FROM access_logs a1
JOIN access_logs a2 ON a1."Card Number Hash" = a2."Card Number Hash"
WHERE a1."Allow" = true AND a2."Allow" = true
    AND a1."Location" != a2."Location"
    AND ABS(TIMESTAMPDIFF(MINUTE, a1."Date Time", a2."Date Time")) <= 5
    AND a1."Date Time" >= NOW() - INTERVAL 1 HOUR;
```

7. การเข้าออกด้วยการ์ดที่ถูก Clone/Duplicate (Potential Card Cloning)

คำอธิบาย: การ์ดเดียวกันใช้งานในหลายอุปกรณ์/สถานที่มากผิดปกติ

```
sql

SELECT
    "Card Number Hash",
    "Card Name",
    COUNT(DISTINCT "Device") as device_count,
    COUNT(DISTINCT "Location") as location_count,
    COUNT(*) as total_access,
    MIN("Date Time") as first_access,
    MAX("Date Time") as last_access,
    'POTENTIAL_CARD_CLONING' as anomaly_type
FROM access_logs
WHERE "Allow" = true
    AND "Date Time" >= NOW() - INTERVAL 1 HOUR
GROUP BY "Card Number Hash", "Card Name"
HAVING device_count > 3 AND location_count > 2;
```

📈 Access Pattern Anomalies

8. การเข้าออกยาวนานผิดปกติ (Unusually Long Stay Duration)

คำอธิบาย: บุคคลอยู่ในอาคารเกิน 12 ชั่วโมงโดยไม่ออก หรือไม่มีการบันทึกการออก

```
sql
```

```
WITH entry_exit_pairs AS (
  SELECT
    "Card Number Hash",
    "Card Name",
    "Location",
    MIN(CASE WHEN "Direction" = 'IN' THEN "Date Time" END) as entry_time,
    MAX(CASE WHEN "Direction" = 'OUT' THEN "Date Time" END) as exit_time
  FROM access_logs
  WHERE "Allow" = true
    AND "Direction" IN ('IN', 'OUT')
    AND "Date Time" >= CURDATE() - INTERVAL 2 DAY
  GROUP BY "Card Number Hash", "Card Name", "Location", DATE("Date Time")
)
SELECT
  "Card Number Hash",
  "Card Name",
  "Location",
  entry_time,
  exit_time,
  CASE
    WHEN exit_time IS NULL THEN TIMESTAMPDIFF(HOUR, entry_time, NOW())
    ELSE TIMESTAMPDIFF(HOUR, entry_time, exit_time)
  END as duration_hours,
  CASE
    WHEN exit_time IS NULL THEN 'NO_EXIT_RECORDED'
    ELSE 'EXTENDED_STAY'
  END as stay_type,
  'UNUSUALLY_LONG_STAY' as anomaly_type
FROM entry_exit_pairs
WHERE (
  exit_time IS NULL AND TIMESTAMPDIFF(HOUR, entry_time, NOW()) > 12
  OR TIMESTAMPDIFF(HOUR, entry_time, exit_time) > 12
);
```

📊 Behavioral Anomalies

9. การเข้าออกถี่ผิดปกติ (Excessive Access Frequency)

คำอธิบาย: การเข้าออกบ่อยเกินปกติในช่วงเวลาสั้นๆ

```sql
```

```
WITH hourly_access AS (
  SELECT
    "Card Number Hash",
    "Card Name",
    DATE_FORMAT("Date Time", '%Y-%m-%d %H:00:00') as hour_bucket,
    COUNT(*) as access_count
  FROM access_logs
  WHERE "Allow" = true
    AND "Date Time" >= NOW() - INTERVAL 24 HOUR
  GROUP BY "Card Number Hash", "Card Name", hour_bucket
)
SELECT
  "Card Number Hash",
  "Card Name",
  hour_bucket,
  access_count,
  'EXCESSIVE_ACCESS_FREQUENCY' as anomaly_type
FROM hourly_access
WHERE access_count > 20;
```

10. Anti-passback Violation

คำอธิบาย: การละเมิดกฎ anti-passback (เข้าแล้วไม่ออก หรือ ออกแล้วไม่เข้า)

```
sql
```

```
WITH user_location_status AS (
  SELECT
    "Card Number Hash",
    "Card Name",
    "Location",
    SUM(CASE WHEN "Direction" = 'IN' THEN 1 ELSE -1 END) as net_entries
  FROM access_logs
  WHERE "Allow" = true
    AND "Date Time" >= NOW() - INTERVAL 8 HOUR
    AND "Direction" IN ('IN', 'OUT')
  GROUP BY "Card Number Hash", "Card Name", "Location"
)
SELECT
  "Card Number Hash",
  "Card Name",
  "Location",
  net_entries,
  CASE
    WHEN net_entries > 2 THEN 'MULTIPLE_ENTRIES_NO_EXIT'
    WHEN net_entries < -2 THEN 'MULTIPLE_EXITS_NO_ENTRY'
    ELSE 'ANTI_PASSBACK_VIOLATION'
  END as violation_type,
  'ANTI_PASSBACK_VIOLATION' as anomaly_type
FROM user_location_status
WHERE ABS(net_entries) > 1;
```

👤 Authorization Anomalies

11. การใช้สิทธิ์ไม่ตรงกับ User Type (Unauthorized Access by User Type)

คำอธิบาย: User Type ไม่ควรเข้าถึงพื้นที่นั้นๆ ได้

sql

```
SELECT
  "Transaction ID",
  "Date Time",
  "Card Name",
  "User Type",
  "Location",
  "Permission",
  CASE
    WHEN "User Type" = 'Guest' AND "Location" IN ('Server Room', 'Executive Floor') THEN 'GUEST_RESTRICTE
    WHEN "User Type" = 'Contractor' AND "Location" LIKE '%Finance%' THEN 'CONTRACTOR_FINANCE_ACCE
    WHEN "User Type" = 'Intern' AND "Location" IN ('Data Center', 'Security Office') THEN 'INTERN_SENSITIVE_
    ELSE 'UNAUTHORIZED_USER_TYPE'
  END as violation_detail,
  'UNAUTHORIZED_USER_TYPE_ACCESS' as anomaly_type
FROM access_logs
WHERE "Allow" = true
  AND (
    ("User Type" = 'Guest' AND "Location" IN ('Server Room', 'Executive Floor', 'Finance Department'))
    OR ("User Type" = 'Contractor' AND "Location" LIKE '%Finance%')
    OR ("User Type" = 'Intern' AND "Location" IN ('Data Center', 'Security Office'))
  )
  AND "Date Time" >= NOW() - INTERVAL 24 HOUR;
```

12. การ์ดหมดอายุหรือถูกยกเลิกแล้วยังใช้ได้ (Expired/Revoked Card Usage)

คำอธิบาย: การ์ดที่ควรจะใช้ไม่ได้แล้วยังสามารถเข้าออกได้

sql

```
-- สมมติมี table revoked_cards แยก
SELECT
    a."Transaction ID",
    a."Date Time",
    a."Card Name",
    a."Card Number Hash",
    r.revoked_date,
    r.reason as revoke_reason,
    'EXPIRED_CARD_USAGE' as anomaly_type
FROM access_logs a
LEFT JOIN revoked_cards r ON a."Card Number Hash" = r.card_hash
WHERE a."Allow" = true
    AND (
        r.revoked_date IS NOT NULL
        OR r.expiry_date < CURDATE()
    )
    AND a."Date Time" >= NOW() - INTERVAL 24 HOUR;

-- หรือถ้าไม่มี table แยก ใช้การตรวจสอบจาก log เก่า
-- ที่มี Reason เป็น 'Card Expired' หรือ 'Card Revoked'
```

🔧 System & Technical Anomalies

13. อุปกรณ์ออฟไลน์แล้วมีการเข้าออก (Access During Device Offline)

คำอธิบาย: มีการบันทึกการเข้าออกขณะที่อุปกรณ์ควรจะออฟไลน์

```sql
-- สมมติมี table device_status แยก
SELECT
    a."Transaction ID",
    a."Date Time",
    a."Device",
    a."Location",
    a."Card Name",
    d.offline_start,
    d.offline_end,
    'ACCESS_DURING_DEVICE_OFFLINE' as anomaly_type
FROM access_logs a
JOIN device_status d ON a."Device" = d.device_id
WHERE a."Allow" = true
    AND d.status = 'offline'
    AND a."Date Time" BETWEEN d.offline_start AND COALESCE(d.offline_end, NOW())
    AND a."Date Time" >= NOW() - INTERVAL 24 HOUR;
```

14. Force Door / Emergency Exit ใช้งาน (Emergency Exit Usage)

คำอธิบาย: การใช้ประตูฉุกเฉินหรือการบังคับเปิดประตู

```
sql

SELECT
  "Transaction ID",
  "Date Time",
  "Door",
  "Location",
  "Card Name",
  "Reason",
  CASE
    WHEN "Reason" LIKE '%emergency%' THEN 'EMERGENCY_EXIT'
    WHEN "Reason" LIKE '%force%' THEN 'FORCED_DOOR'
    WHEN "Door" LIKE '%emergency%' THEN 'EMERGENCY_DOOR_USED'
    ELSE 'UNUSUAL_DOOR_ACCESS'
  END as emergency_type,
  'EMERGENCY_EXIT_USED' as anomaly_type
FROM access_logs
WHERE "Allow" = true
  AND (
    "Reason" LIKE '%emergency%'
    OR "Reason" LIKE '%force%'
    OR "Door" LIKE '%emergency%'
    OR "Channel" = 'Emergency'
  )
  AND "Date Time" >= NOW() - INTERVAL 24 HOUR;
```

15. การเข้าออกในช่วงที่ระบบกำลัง Maintenance (Access During Maintenance)

คำอธิบาย: มีการเข้าออกขณะที่ระบบกำลังอยู่ในโหมด maintenance

```
sql
```

```
-- สมมติมี table maintenance_schedules แยก
SELECT
    a."Transaction ID",
    a."Date Time",
    a."Device",
    a."Location",
    a."Card Name",
    m.maintenance_type,
    m.start_time,
    m.end_time,
    'ACCESS_DURING_MAINTENANCE' as anomaly_type
FROM access_logs a
JOIN maintenance_schedules m ON a."Location" = m.location
WHERE a."Allow" = true
    AND a."Date Time" BETWEEN m.start_time AND m.end_time
    AND m.status = 'active'
    AND a."Date Time" >= NOW() - INTERVAL 24 HOUR;

-- หรือใช้การตรวจสอบจาก Reason field
SELECT
    "Transaction ID",
    "Date Time",
    "Device",
    "Location",
    "Card Name",
    "Reason",
    'ACCESS_DURING_MAINTENANCE' as anomaly_type
FROM access_logs
WHERE "Allow" = true
    AND "Reason" LIKE '%maintenance%'
    AND "Date Time" >= NOW() - INTERVAL 24 HOUR;
```

📊 Severity Classification

Critical (Level 3) 🔴

- Potential Card Cloning
- Simultaneous Multi-Location Access
- Emergency Exit Usage
- Expired/Revoked Card Usage

Warning (Level 2) 🟡

- Consecutive Card Rejections

- After Hours Access
- Unusually Long Stay Duration
- Unauthorized User Type Access
- Anti-passback Violation

Info (Level 1) 🟢

- Weekend Access
- Potential Tailgating
- Invalid Entry/Exit Sequence
- Excessive Access Frequency
- Access During Device Offline
- Access During Maintenance

🎯 Implementation in n8n

Workflow Structure

```
Trigger (Webhook)
  ↓
For Each Anomaly Type
  ↓
Execute SQL Query
  ↓
If Results Found
  ↓
Insert into Anomalies Table
  ↓
Send Alert (if Critical/Warning)
  ↓
Update Dashboard Cache
```

Alert Thresholds

- **Critical**: Immediate notification
- **Warning**: Notification within 15 minutes
- **Info**: Daily summary report

SQL queries เหล่านี้สามารถนำไปใช้ใน n8n MySQL nodes โดยตรง และปรับแต่ง threshold ได้ตามความเหมาะสมของแต่ละองค์กร