

# OAuth Abuse Detection & Incident Response Report

**Author:** Emmanuel Ajayi

**Date:** 30-Dec-2025

## Executive Summary

This lab demonstrates how valid OAuth tokens can be **misused outside their intended context**, bypassing authentication controls.

Key takeaways:

- Token theft can grant access to sensitive data.
- Context-aware monitoring is critical.
- Mitigations include short-lived tokens, strict scope control, and behavioral monitoring.

## Attack Narrative

### Step 1: Legitimate OAuth Login

- User authenticates successfully via Flask OAuth app.

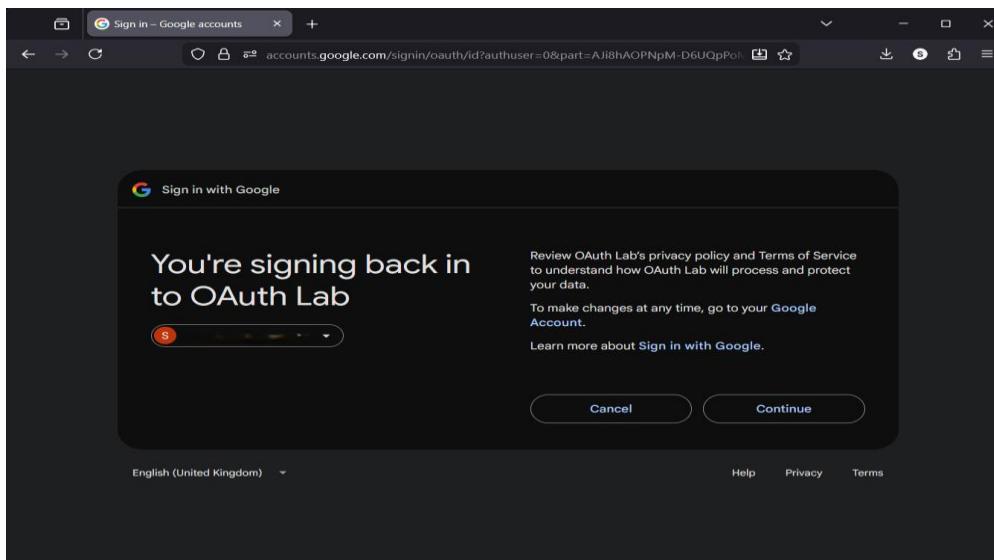


Figure 1: Successful OAuth login

### Step 2: Access Token Issued

- OAuth server issues a valid access token with proper scopes.

### Step 3: Token Exposure

- Token is accidentally exposed in logs/debug output.
  - Simulated attacker extracts token.

```
python_OAuth
  logs
    auth.log
      +--> auth.log
        +--> auth.log
          +--> auth.log
            +--> auth.log
              +--> auth.log
                +--> auth.log
                  +--> auth.log
                    +--> auth.log
                      +--> auth.log
                        +--> auth.log
                          +--> auth.log
                            +--> auth.log
                              +--> auth.log
                                +--> auth.log
                                  +--> auth.log
                                    +--> auth.log
                                      +--> auth.log
                                        +--> auth.log
                                          +--> auth.log
                                            +--> auth.log
                                              +--> auth.log
                                                +--> auth.log
                                                  +--> auth.log
                                                    +--> auth.log
                                                      +--> auth.log
                                                        +--> auth.log
                                                          +--> auth.log
                                                            +--> auth.log
                                                              +--> auth.log
                                                                +--> auth.log
                                                                  +--> auth.log
                                                                    +--> auth.log
                                                                      +--> auth.log
                                                                        +--> auth.log
              requirements.txt
```

*Figure 3: Token visible in client logs*

## Step 4: Token Abuse

- Attacker reuses token to access protected API endpoints.
  - Access successful without login.

*Figure 4: Token reused via API call*

## Indicators of Compromise (IOCs)

- Reuse of the same OAuth access token outside its original context
  - API access occurring without a corresponding recent login event
  - Token usage inconsistent with expected user behavior

## Detection & Mitigation

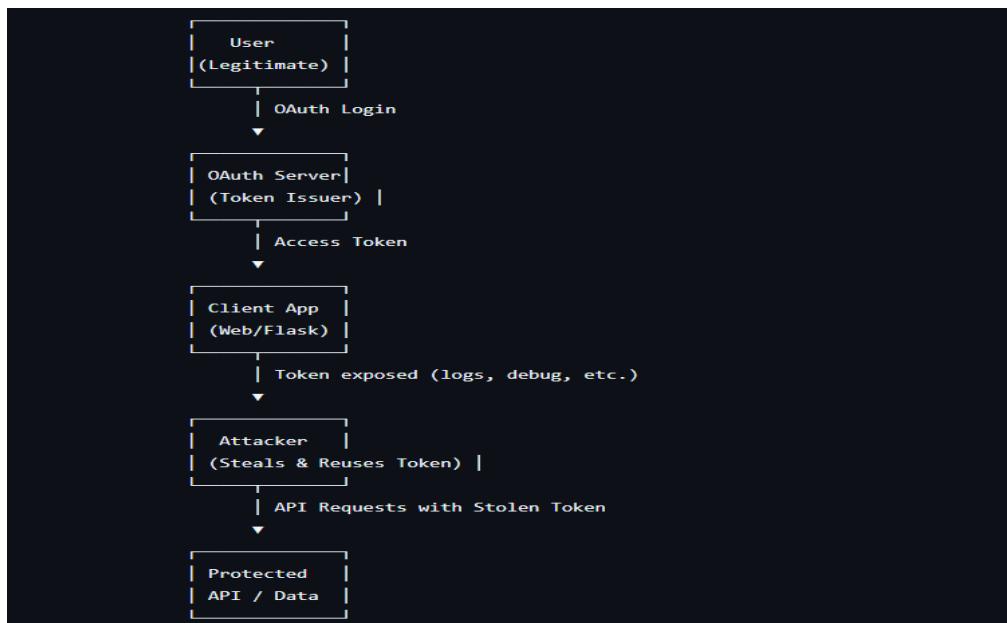
## Detection Strategies:

- Monitor token usage patterns (IP, device, frequency)
- Trigger alerts for abnormal behavior

## Mitigation Measures:

- Short-lived access tokens
- Enforce strict scopes
- Eliminate plaintext token logging
- Revoke suspicious tokens immediately
- Implement behavior-based monitoring

## Flow Diagram



Flow-diagram.png

## Conclusion

- Authentication success does not equal security.
- Tokens can be abused even when login is legitimate.
- Continuous monitoring, anomaly detection, and proper token hygiene are key to defense.