Context-aware Intrusion Detection in OAuth Protocol Flows

Florian Nehmer

May 10, 2023

1 Background & Motivation

The OAuth 2.0 protocol [1] (referred to as OAuth) is widely used in various contexts, such as social logins, single sign-on services for companies, and sharing specific resources for mobile apps like calendars. All these contexts share a common feature: they involve accessing sensitive data. Consequently, it is crucial that the authentication and authorization mechanisms provided by the OAuth standard are securely implemented. Unfortunately, experience has shown that this is often not the case [2]. The OAuth standard offers a range of possibilities for different use cases, making the implementation of OAuth for a specific use case a complex task, and misconfigurations may occur. Attackers attempt to exploit these misconfigurations by misusing the OAuth protocol in malicious ways. To detect such attacks, intrusion detection systems can be employed to analyze network traffic in real-time through flow analysis or packet inspection [3]. This approach offers at least two benefits: first, it enables the detection of attacks, preventing further exploitation; second, it facilitates the gathering of intelligence on the attacks, allowing for the validation of whether the current implementation of the OAuth service in use is adequately hardened against such threats.

2 Objectives

In my work I want to pursue the following research goals:

1. Implement and investigate the viability of anomaly-based intrusion detection using (Hidden-)Markov-chains in the context of OAuth like it was applied in this reference work to other contexts [4]
2. Implement and investigate the viability of specification-based intrusion detection in the context of OAuth.
    - The main challenge is to specifiy patterns/rules for this approach given the complexity of the OAuth specification.
3. Compare the anomaly-based and Specification-based approach to each other regarding yield and accuracy.

3 Methodology

In order to investigate intrusion detection on OAuth using the anomaly-based and specification-based approach a lab environment gets implemented, which generates data, which afterwards gets analysed.

3.1 Lab environment

The Lab environment consists of different entites.

3.1.1 OAuth Provider

The OAuth provider is a web service, which offers users the creation of an account for authentication at the provider. It also offers OAuth capabilities to provide authorization to the users account data to other web services, if the user is allowing it. It offers all OAuth authorization flows at the same time (authorization code, client credentials, resource owner password, implicit). The OAuth Provider will be implemented in Python using the libraries *flask* and *authlib*.

3.1.2 Dummy Web Service

The dummy web service, is utilising the different OAuth capabilities of the OAuth provider. It is implemented in Python using the *flask* library.

3.1.3 Intrusion Detection System

The intrusion detection system is intercepting the traffic entering the lab and exports the traffic as a network log file. It offers functionality to analyse the traffic using two methods:

- Anomaly-based using Markov-chains implemented with the Python library *markovify*
- Specification-based, by implementing rules and patterns for the usage of the OAuth protocol.

The open-source IDS "zeek" and its Python connector "zat" are used to load the network data into a Python "pandas" dataframe. This data then can be fed into different detection models and methods to produce the alerts.

3.1.4 Generator

The Generator is executing valid OAuth interactions aswell as malicious ones, to generate traffic to be analysed by the IDS. For the malicious interactions different attack approaches on OAuth are getting implemented. The Generator is implemented in Python utilising the *requests* library. After running an experiment the detection methods are getting fine-tuned and the experiments are ran again.

4 Experiments

The experiments are ran with the following recipe:

1. The Generator runs attacks at random over a fixed period of time and produces logs of which attack it executed at which time
2. The IDS meanwhile will produce network logs, which include the attack traffic
3. The traffic logs are analysed by the IDS with the implemented detection methods
4. The IDS produces alerts for attacks it recognised
5. The alerts are getting compared with the logs of the generator

Every experiment is analyzed regarding yield and accuracy on the detection of the different types of attacks on the OAuth protocol.

5 Novelty

The novelty is to contribute to the field of context aware intrusion detection, regarding protocols at application layer.

- A suitable method to better detect attacks on OAuth in networks.
- Data on how well different anomaly-based and specification-based detection methods for attacks on OAuth perform.

References

[1] Hardt, D. The OAuth 2.0 Authorization Framework. RFC 6749, Oct. 2012.

[2] Li, W., Mitchell, C. J., and Chen, T. Oauthguard: Protecting user security and privacy with oauth 2.0 and openid connect. In *Proceedings of the 5th ACM Workshop on Security Standardisation Research Workshop* (New York, NY, USA, 2019), SSR'19, Association for Computing Machinery, p. 35–44.

[3] Liu, H., and Lang, B. Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences 9*, 20 (2019).

[4] Sperotto, A., and Pras, A. Flow-based intrusion detection. In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops* (2011), pp. 958–963.