Intrusion Detection in OAuth Protocol Flows

Florian Nehmer

May 24, 2023

1 Motivation

The OAuth 2.0 protocol [3] (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 [4]. 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 [5]. 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 aim to pursue the following research goals:

1. Create a comprehensive overview of currently known attack methodologies on OAuth to provide an understanding of the potential threats.

2. Implement and investigate the viability of anomaly-based intrusion detection using (Hidden-)Markov-chains in the context of OAuth, as applied in this reference work to other contexts [6] [2] [1].

3. Implement and investigate the viability of specification-based intrusion detection in the context of OAuth.

   - The main challenge is to specify patterns/rules for this approach given the complexity of the OAuth specification.

4. Compare the anomaly-based and specification-based approaches to each other regarding yield and accuracy.

3 Methodology

3.1 Literature research: Known attacks on OAuth

In order to gain a comprehensive understanding of the landscape of attacks on OAuth, a literature review will be conducted with the objective of creating a taxonomy that categorizes the various types of attacks. To facilitate this classification, threat and attacker models will be defined, serving as a foundation for organizing the vulnerabilities within the proposed taxonomy.

3.2 OAuth Intrusion Detection: Lab environment

To investigate intrusion detection on OAuth using the anomaly-based and specification-based approaches, a lab environment is implemented to generate data for analysis.

The lab environment consists of an OAuth Provider, a dummy web service, an Intrusion Detection System, and a Traffic Generator.

3.2.1 OAuth Provider

The OAuth provider is a web service that offers users the creation of an account for authentication at the provider. It also offers OAuth capabilities to provide authorization to the user's account data to other web services, if the user allows it. It offers all OAuth authorization flows at the same time (authorization code, client credentials, resource owner password, implicit). The OAuth Provider is implemented in Python using the libraries *flask*[1] and *authlib*[2].

3.2.2 Dummy Web Service

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

3.2.3 Intrusion Detection System

The intrusion detection system intercepts the traffic entering the lab and exports the traffic as a network log file. It offers functionality to analyze the traffic using at least the following two methods:

- Anomaly-based using Markov-chains implemented with the Python library *markovify*[3].
- Specification-based, by implementing rules and patterns for the usage of the OAuth protocol.

The open-source IDS *zeek*[4] and its Python connector *zat*[5] are used to load the network data into a Python *pandas*[6] dataframe. This data can then be fed into different detection models and methods to produce the alerts.

3.2.4 Generator

The generator executes valid OAuth interactions as well as malicious ones to generate traffic to be analyzed by the IDS. For the malicious interactions, different attack approaches on OAuth are implemented. The generator is implemented in Python utilizing the *requests*[7] library. After running an experiment, the detection methods are fine-tuned, and the experiments are run again.

4 Experiments

The experiments are run 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 produces network logs, which include the attack traffic.

3. The traffic logs are analyzed by the IDS with the implemented detection methods.

4. The IDS produces alerts for attacks it recognized.

5. The alerts are 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.

---

[1] https://flask.palletsprojects.com/en/2.3.x/
[2] https://docs.authlib.org/en/latest/
[3] https://pypi.org/project/markovify/
[4] https://zeek.org/
[5] https://pypi.org/project/zat/
[6] https://pandas.pydata.org/
[7] https://pypi.org/project/requests/

5 Novelty

The novelty is to contribute to the field of context-aware intrusion detection regarding protocols at the 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] AMATO, F., CASTIGLIONE, A., DE SANTO, A., MOSCATO, V., PICARIELLO, A., PERSIA, F., AND SPERLÍ, G. Recognizing human behaviours in online social networks. *Computers & Security 74* (2018), 355–370.

[2] CHEN, C.-M., GUAN, D.-J., HUANG, Y.-Z., AND OU, Y.-H. Anomaly network intrusion detection using hidden markov model. *Int. J. Innov. Comput. Inform. Control 12* (2016), 569–580.

[3] HARDT, D. The OAuth 2.0 Authorization Framework. RFC 6749, Oct. 2012.

[4] 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.

[5] LIU, H., AND LANG, B. Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences 9*, 20 (2019).

[6] 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.