Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

23. Januar 2024

Intrusion Detection for OAuth

Florian Nehmer

Einleitung (1)

- Open Authorization Framework 2.0 (OAuth)
  - Login-Prozess: "Weiter mit [Apple, Google, ...]“
    - Single Sign On
    - Smart Devices
  - RFC6749 (2012)
    - 30 RFCs die mit OAuth in Verbindung stehen
  - OAuth 2.1 in Arbeit

- Forschungsfragen
  - Welche Arten von Sicherheitsbedrohungen existieren aktuell für Implementierungen des OAuth Protokolls?
  - Wie kann man Anomalieerkennung einsetzen, um Angriffe auf das OAuth Protokoll zu erkennen?

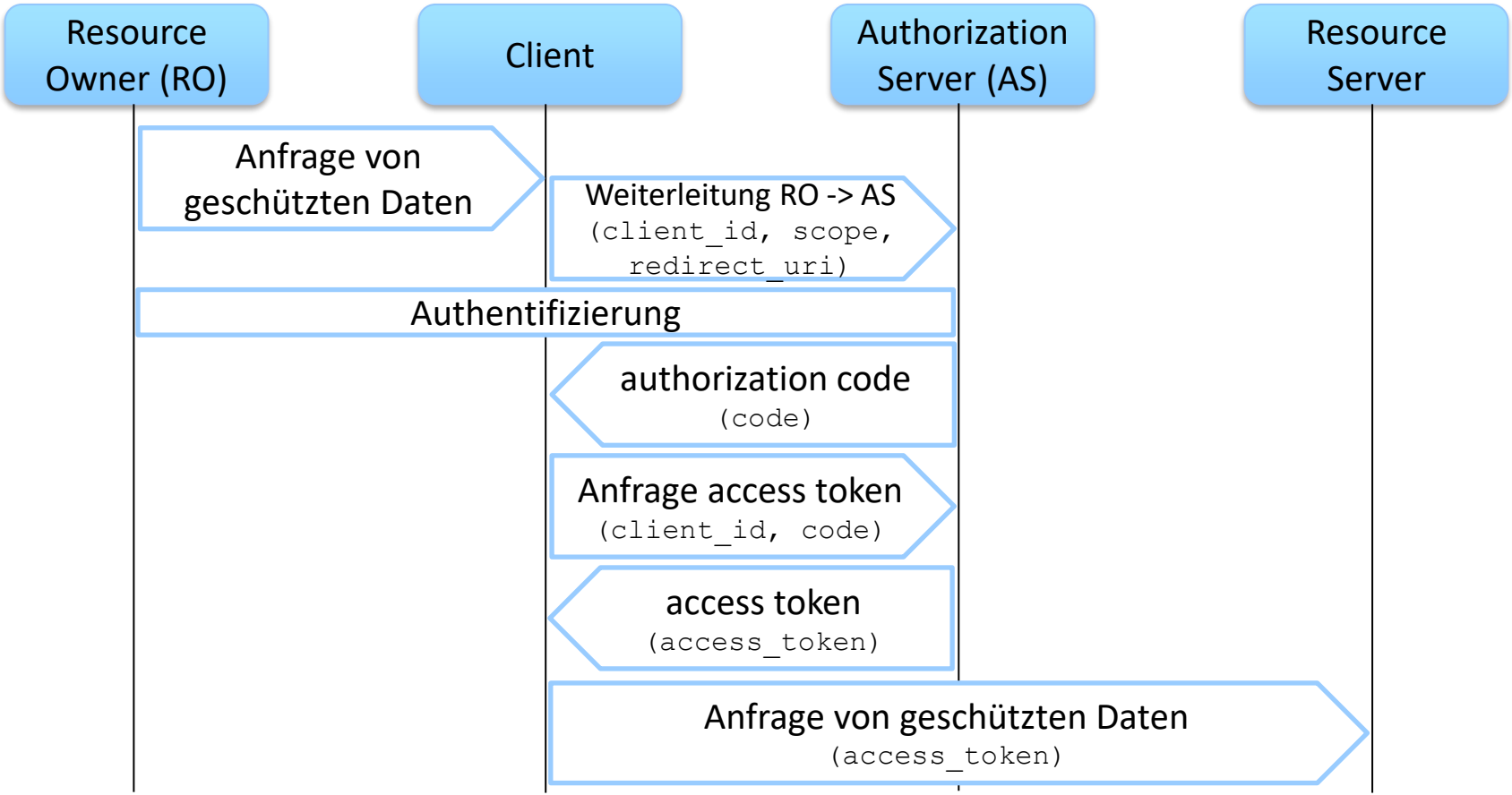
Einleitung (2)

- Methoden und Verfahren
  - Kategorisierung verschiedener OAuth Bedrohungen
  - Implementierung einer OAuth Testumgebung
  - Netzwerkdatengenerierung
  - Anomalie-Erkennung
    - Word2Vec Enkodierung von Netzwerkdaten
    - Clustering Algorithmen zur Anomalieerkennung

OAuth 2.0 Protokoll Funktionsweise

Authorization Code Grant

[Har12]



OAuth 2.0 Bedrohungen

Bedrohungslage von OAuth

1. Leak von Credentials & Session Übernahme
2. Manipulation von Weiterleitungen
3. Umgehung von Integritätsmaßnahmen
4. Credential Injection

[Lod+20]

1. Leak von Credentials & Session Übernahme

- Ausnutzung von Web Mechanismen und Umgebungen zur Übernahme von Session Daten
  - Autorisierungsdaten / Credentials sichtbar in Logdateien und Services
  - Clickjacking
  - 307 Redirect

[Lod+20]

2. Manipulation von Weiterleitungen

- Manipulation des Opfers zur Weiterleitung der Autorisierungsdaten zum Angreifer
  - Unzureichende Überprüfung des *redirect_uri* Parameters
  - AS leitet zu Phishing Seite weiter
  - Mix-Up Angriff

[Lod+20]

3. Umgehung von Integritätsmaßnahmen

- Maßnahmen, die die Integrität des Resource Owners bewahren sollen werden umgangen
  - PKCE Downgrade Angriff
  - Cross Site Request Forgery

[Lod+20]

4. Credential Injection

- Geklaute Credentials werden benutzt, um an geschützte Daten zu gelangen
  - Authorization Code Injection
  - Access Token Injection

Fazit Bedrohungslage

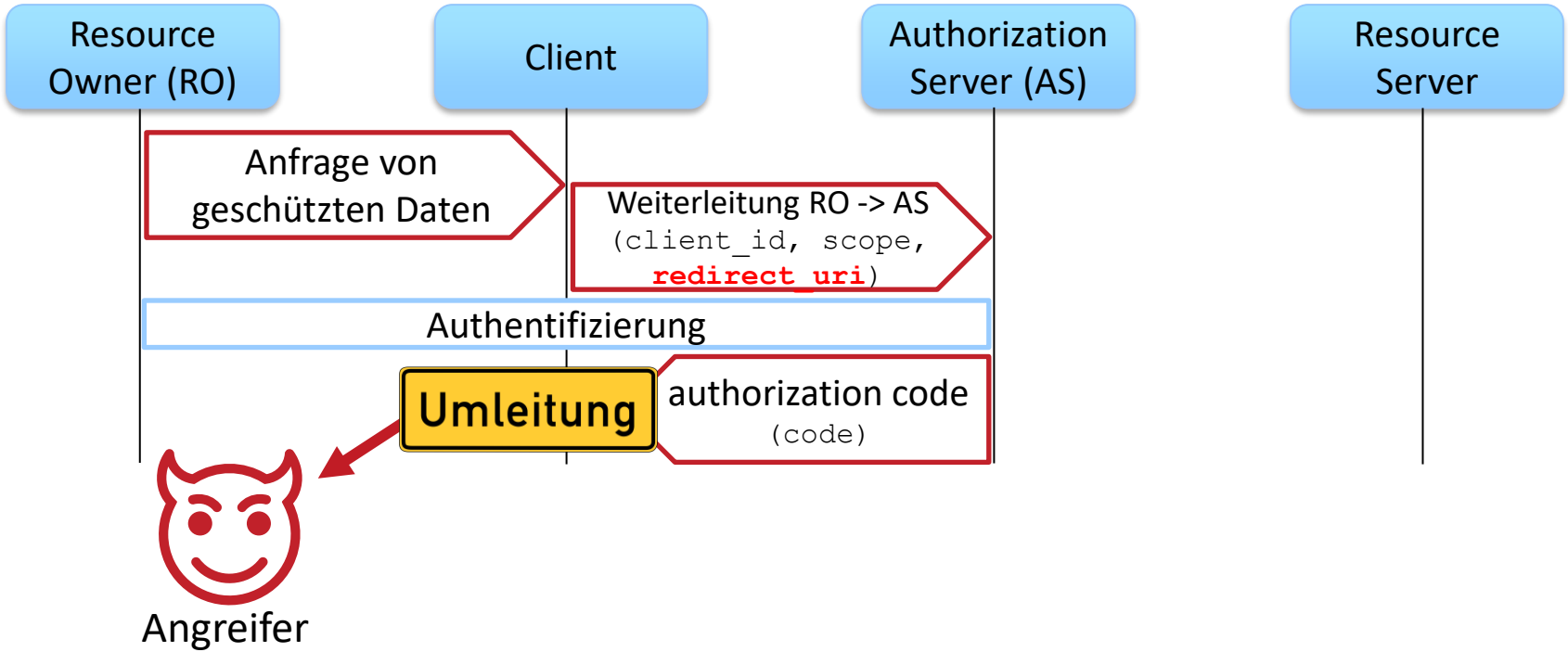
- Verschiedenste Angriffsvektoren sind denkbar
- Viele involvierte Komponenten
  - Mensch
  - Browser
  - Client Applikation
  - Authorization Server
- **Umgebung und Komponenten entwickeln sich weiter**
  - Möglicherweise immer wieder neue unbekannte Angriffe denkbar

Anomalie-Erkennung für Angriffe auf OAuth 2.0

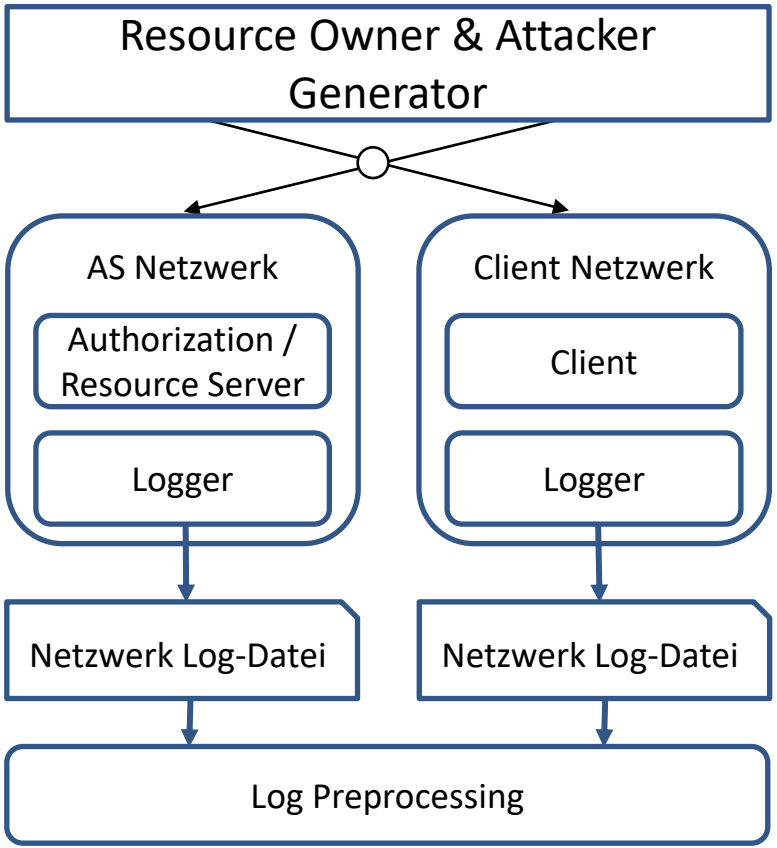
Anomalie-Erkennung für OAuth

- Bedrohungsart: Manipulation von Weiterleitungen

- Bedrohungsart: Manipulation von Weiterleitungen



Testdatengenerierung



- 3627 Logeinträge im Testdatensatz insgesamt
- 44 Einträge, die Teil eines Angriffes sind
  - Angriffsrate 1,12 %
  - 2 Arten von Angriffen

Netzwerkdaten Beispiel ohne Angriff

Method	URI	Is_attack
GET	/?next=http://localhost:5123/oauth/authorize ?response_type=code &client_id=A30qfzW7fbvhAN4OtIq4ZNFR &redirect_uri=http://localhost:8080/index.html	0
GET	/	0
POST	/	0
POST	/create_client	0

Netzwerkdaten Beispiel mit Angriff

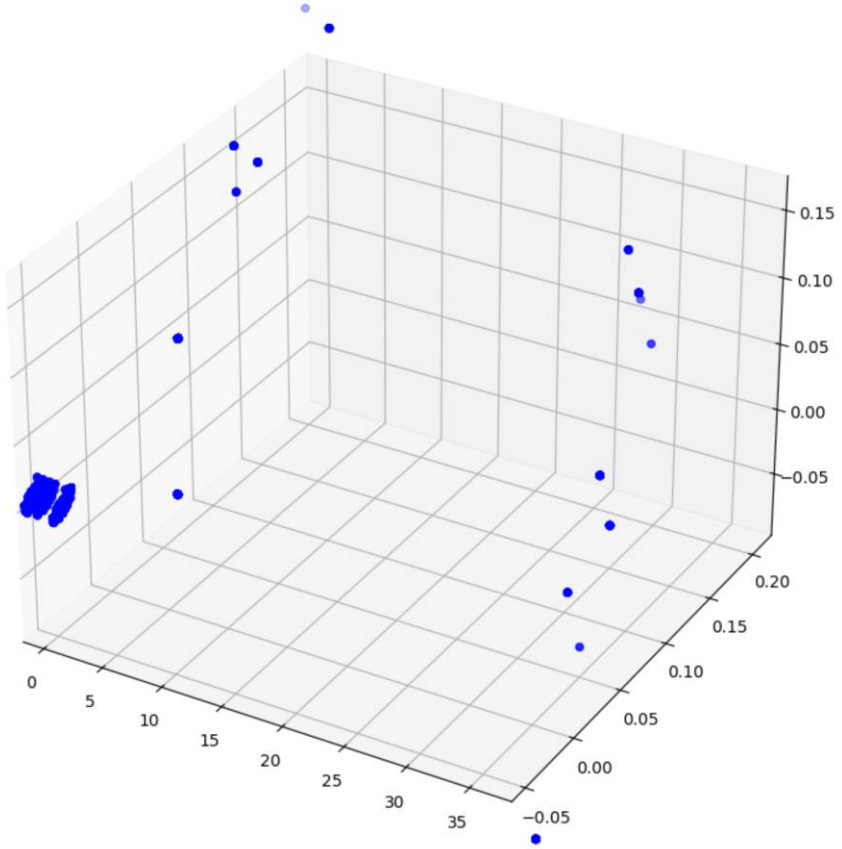
Method	URI	Is_attack
GET	/oauth/authorize ?response_type=code &client_id=A30qfzW7fbvhAN 4OtIq4ZNFR &redirect_uri=http://evil-server.com	1

Enkodierung der textuellen Daten

[Mik+13]

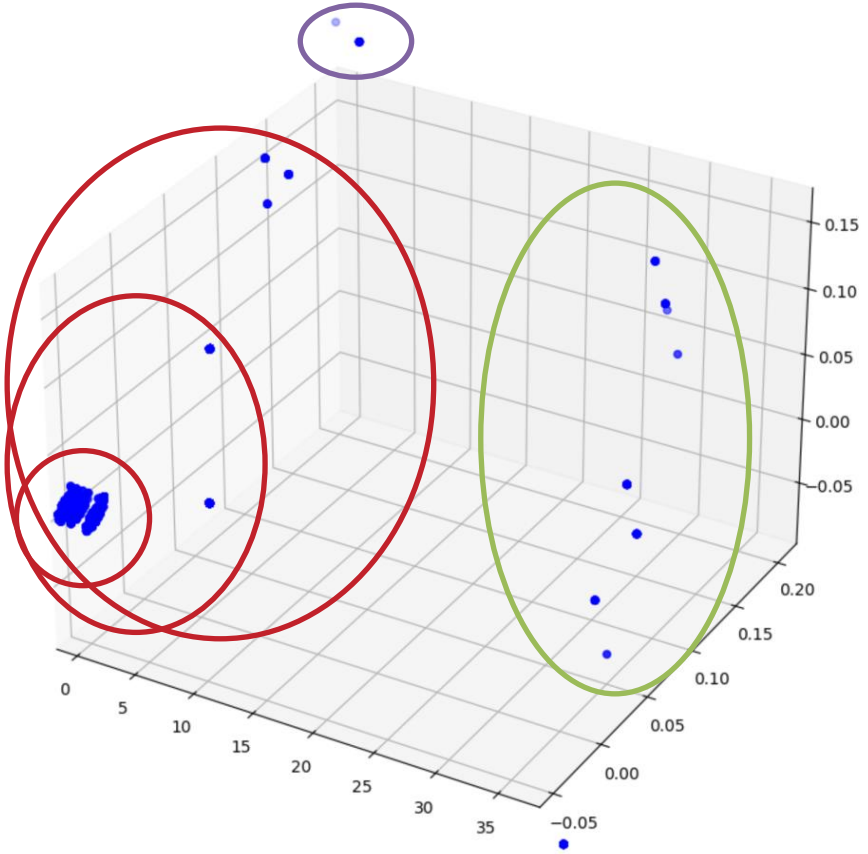
- Clustering benötigt numerische Repräsentation der Daten
- Word2Vec
  - Methode aus der natürlichen Sprachverarbeitung
  - Wörter, die häufig zusammen auftauchen, stehen in einer Beziehung
  - Continuous Bag of Words (CBOW) vs. Skip-Gram
  - Wichtiger Hyperparameter: Fenstergröße
- Input Beispiele:
  - ['GET', 'api', 'v1.0', 'users’]
  - ['GET', 'oauth', 'authorize', 'response_type', 'code', 'client_id', 'A30qfzW7fbvhAN4OtIq4ZNFR', 'redirect_uri', 'http:', 'evil-server.com']

Word2Vec Ergebnisse



Clustering

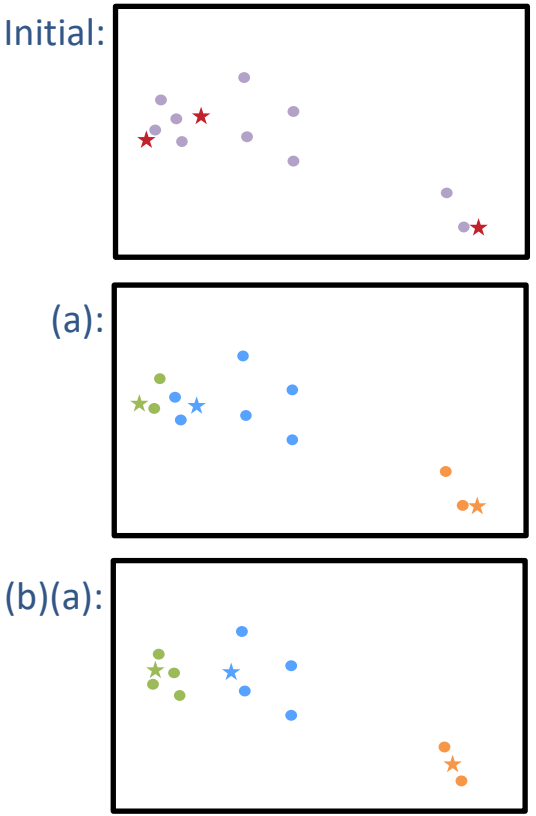
- Anomalien erkennen durch Clusterbildung
- Definition von Schwelle für Anomaliecluster
  - >= 5% zählt als Anomalie
- Zwei Algorithmen wurden getestet
  - k-Means
  - Self-Organizing Maps



Clustering (2)

[Llo82]

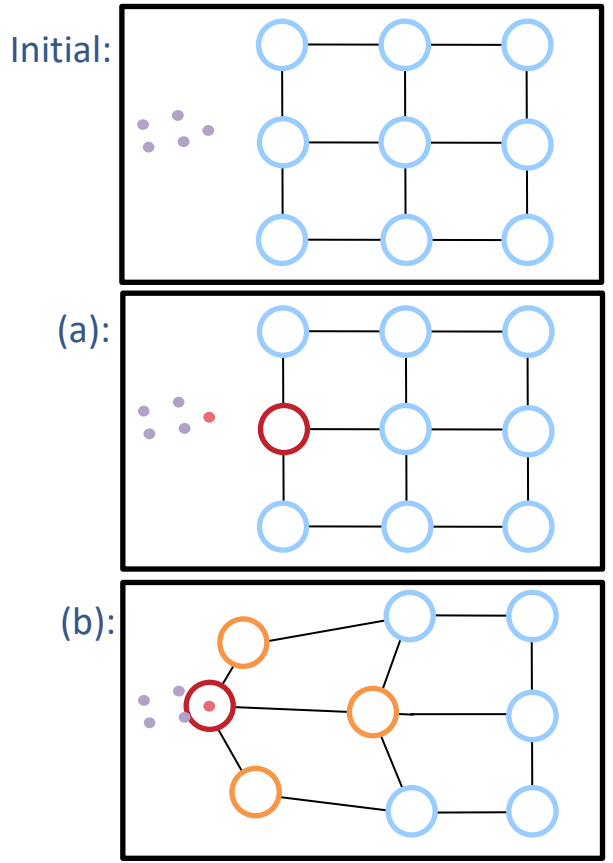
- k-Means
  - Vorbestimmte Anzahl von Clustern *(k)*
  - Initial: Feste Anzahl (*k*) von zufälligen Cluster Mittelpunkten (*Centroids*) wird gewählt
  - (a) Vektoren werden anhand der euklidischen Distanz dem nächsten Centroid zugeordnet – es bilden sich Cluster
  - (b) Zu jedem entstandenen Cluster wird der tatsächliche Mittelpunkt berechnet
  - Schritte a und b werden wiederholt, bis sich die Centroids nicht mehr ändern



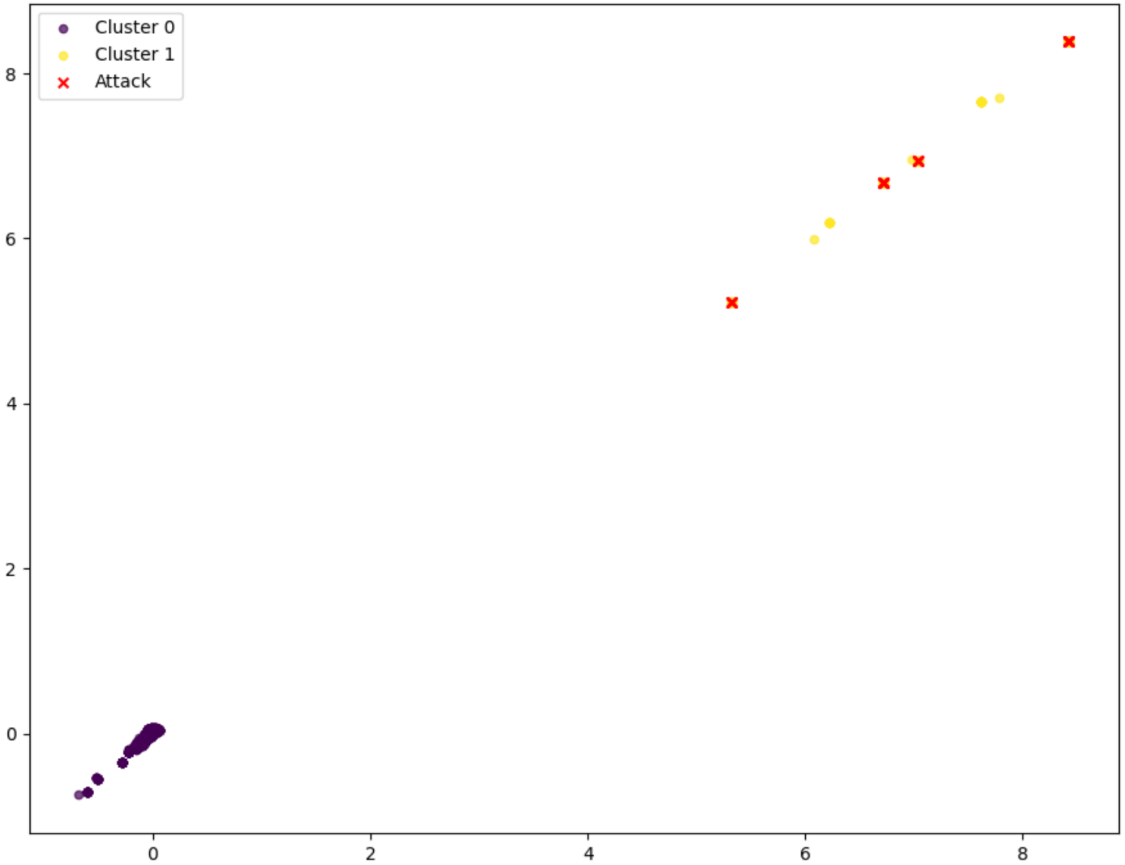
Clustering (3)

[Koh90]

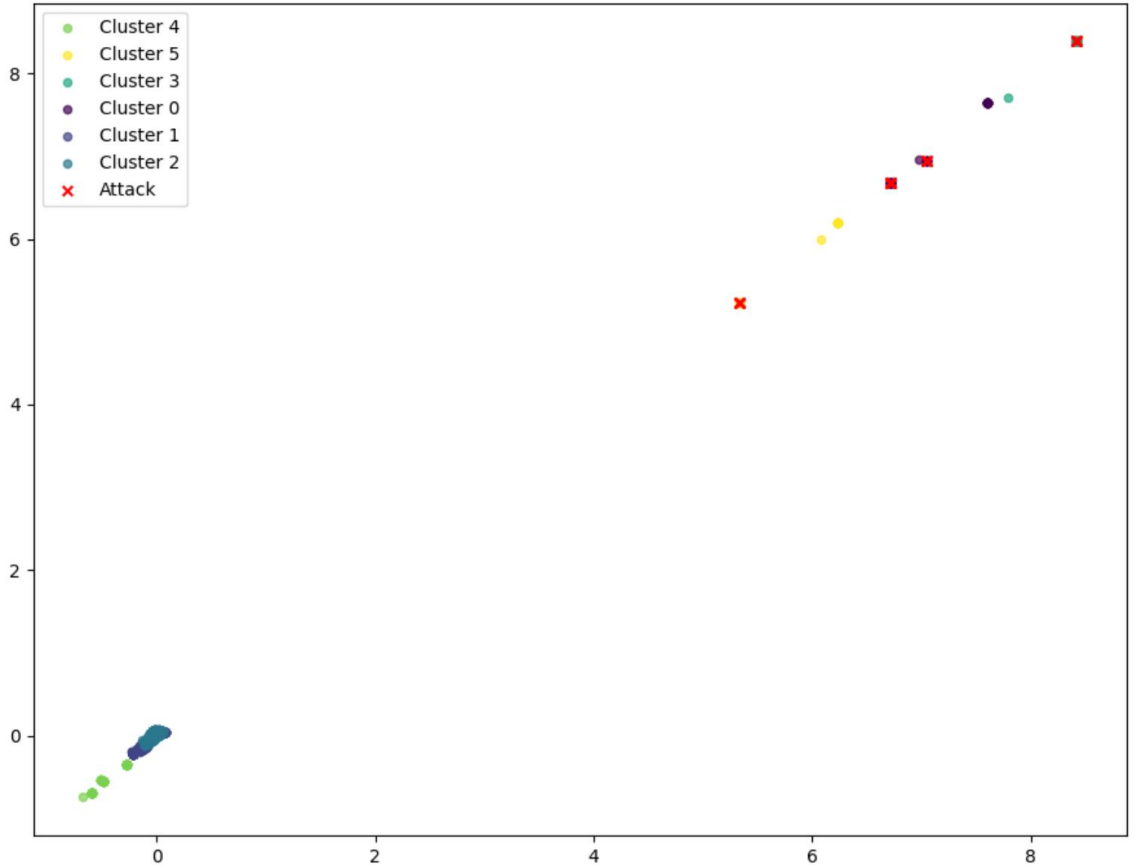
- Self-Organizing-Maps
  - Clusteranzahl ergibt sich durch Algorithmus
  - Initial: 2-dimensionales Raster von Neuronen wird initialisiert
  - (a) Vektoren werden anhand der euklidischen Distanz, dem nächsten/ähnlichsten Neuron aus dem Raster zugeordnet -> Best Matching Unit (BMU)
  - (b) Eine Nachbarschaftsfunktion wird angewendet um das Gewicht des BMU und benachbarten Neuronen anzupassen
  - Schritte a und b werden für alle Vektoren mehrmals wiederholt für eine beg renzte Anzahl von Epochen



Clustering Ergebnis (k-Means) k=2



Clustering Ergebnis (k-Means) k=6

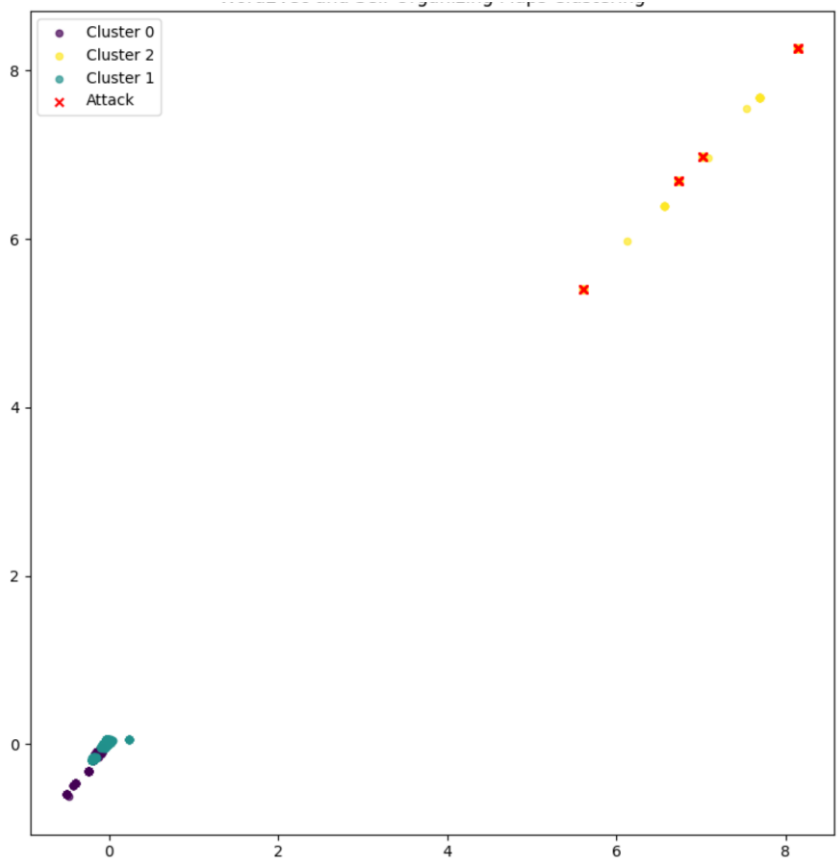


Clustering Ergebnis (k-Means) Clusteranzahl

- Verhalten bei Erhöhung der Clusteranzahl; Anomalieschwelle = 5%

k	2	3	4	5	6	7	8	17	20	24
Anomalie Cluster	1	2	2	2	3	4	5	10	12	18
Accuracy	0,99	0,99	0,99	0,99	0,99	0,99	0,97	0,92	0,89	0,71
Precision	0,63	0,63	0,63	0,63	0,63	0,63	0,34	0,13	0,11	0,05

Clustering Ergebnis (SOM)



Ergebnisse beider Clustering Algorithmen

	Word2Vec + K-Means	Word2Vec + SOM
Accuracy	0,993	0,993
Yield	1.0	1.0
Precision	0,637	0,637

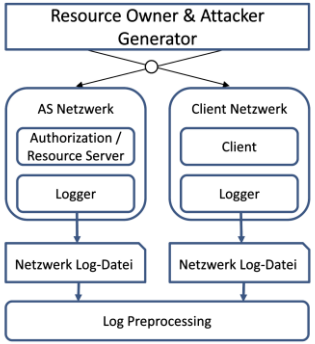
- Enkodierung durch Word2Vec hat die Ergebnisse maßgeblich beeinflusst
- Clustering Algorithmen entdecken offensichtliche Cluster
- Vorgehen trennt z.T. nur OAuth Netzwerkdaten von anderen Netzwerkdaten

Diskussion & zukünftige Ansatzpunkte

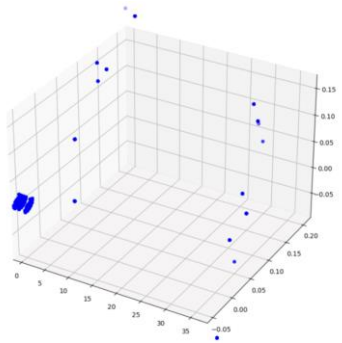
- Modellierung der Netzwerkdaten konzentriert sich auf Anwendungsdaten
  - Aggregierung von realen Daten
- Word2Vec Performance
  - Untersuchung von anderen NLP-Methoden anhand von realen Daten
- Klassifizierung
  - Untersuchung von Supervised Learning Methoden auf realen Daten

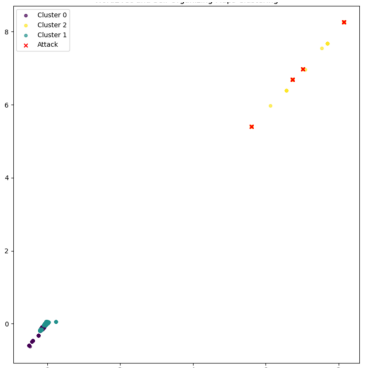
Zusammenfassung

1. Leak von Credentials & Session Übernahme
2. Manipulation von Weiterleitungen
3. Umgehung von Integritätsmaßnahmen
4. Credential Injection



- 3627 Logeinträge im Testdatensatz insgesamt
- 44 Einträge, die Teil eines Angriffes sind
  - Angriffsrate 1,12 %
  - 2 Arten von Angriffen





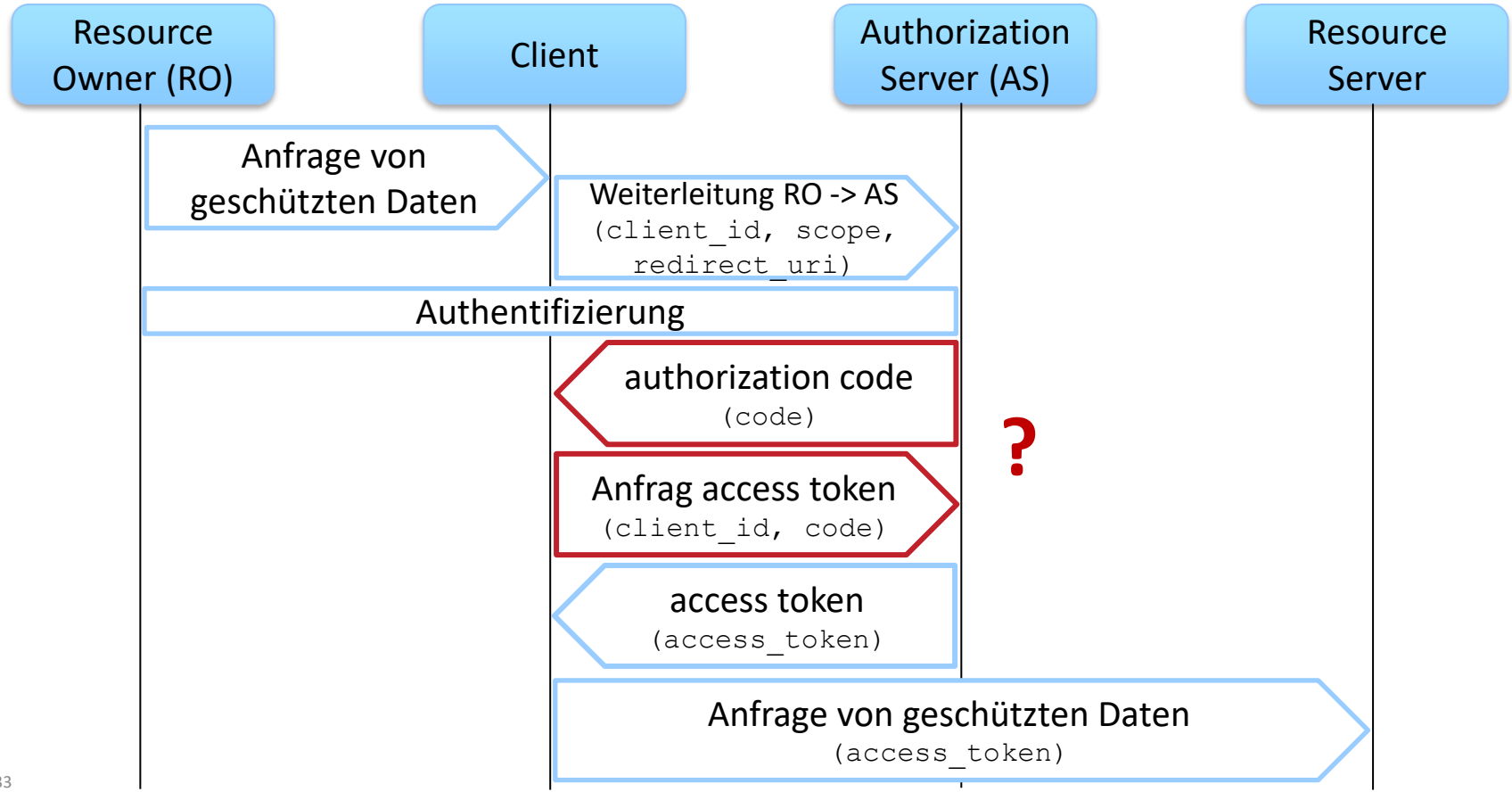
	K-Means	SOM
Accuracy	0,993	0,993
Yield	1.0	1.0
Precision	0,637	0,637

Quellen

- [Lod+20] - Torsten Lodderstedt et al. OAuth 2.0 security best current practice. In: IETF Web Authorization Protocol, Tech. Rep. draft-ietf-oauth-security-topics-16 (2020).
- [Har12] - Dick Hardt. Rfc 6749: The oauth 2.0 authorization framework. 2012.
- [Mik+13] - Tomas Mikolov et al. Efficient estimation of word representations in vector space. In: arXiv preprint arXiv:1301.3781 (2013).
- [Llo82] - S. Lloyd. Least squares quantization in PCM. In: IEEE Transactions on Information Theory 28.2 (1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489.
- [Koh90] - T. Kohonen. The self-organizing map. In: Proceedings of the IEEE 78.9 (1990), pp. 1464–1480. DOI: 10.1109/5.58325.

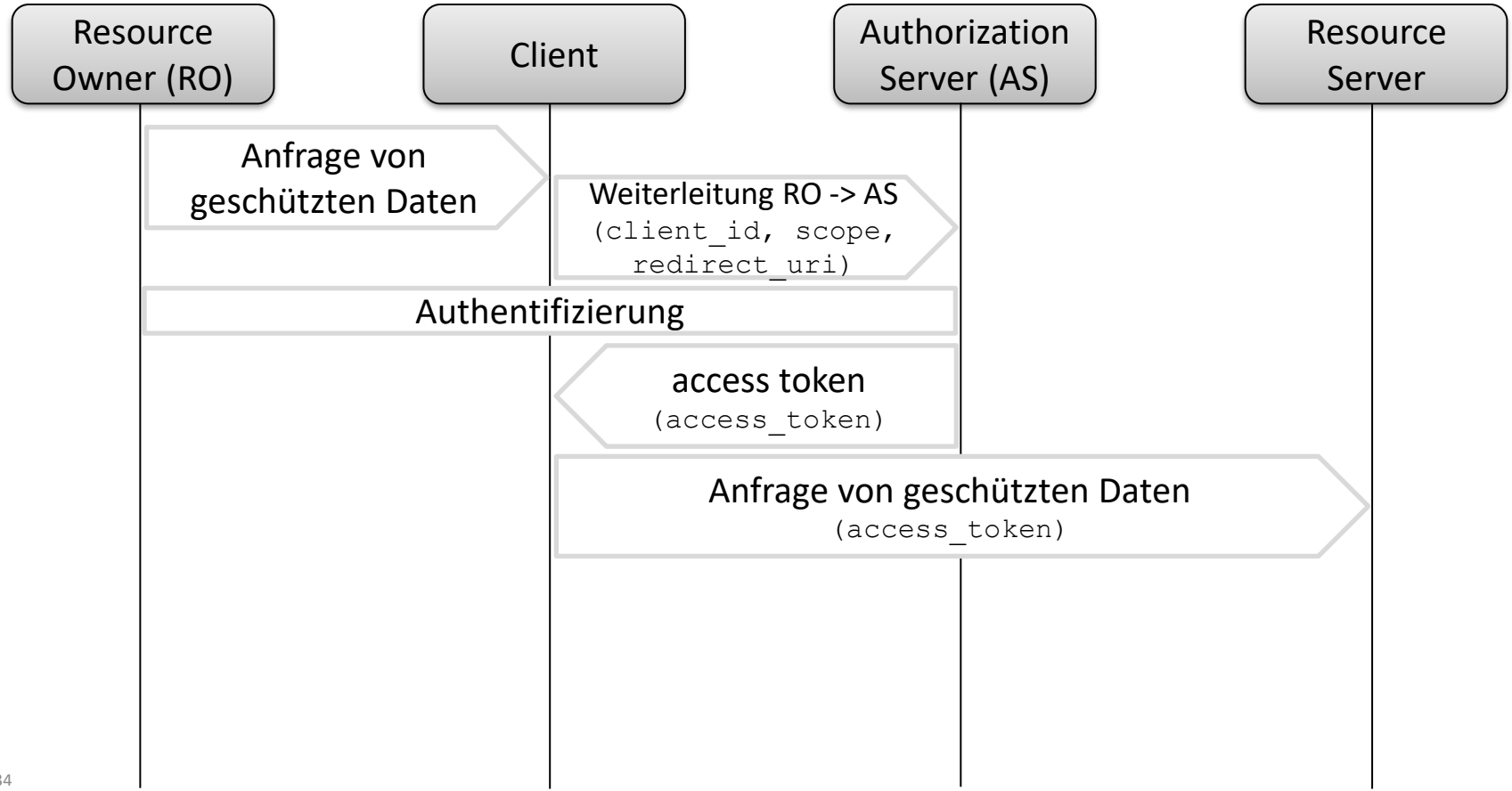
Authorization Code Grant

[Har12]



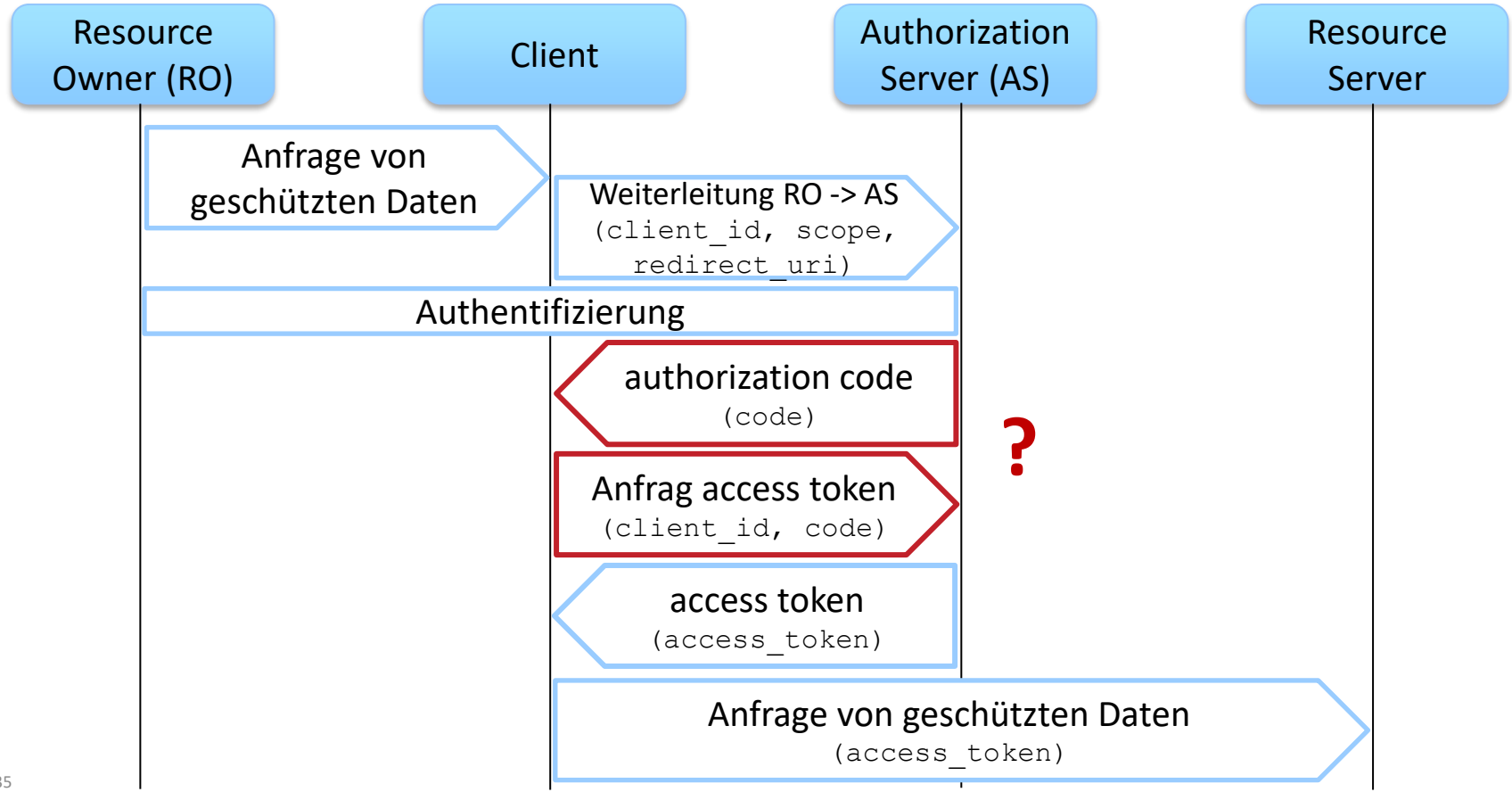
[Har12]

Implicit Grant



Authorization Code Grant

[Har12]



Anomalie Erkennung für OAuth

- Ansatzpunkt *redirect_uri* Parameter und HTTP Methode
  - Textuelle Repräsentation
  - Zwischen weiteren anderen Query-Parametern
  - Durch Implementierung in Clients sehr wenige Unterschiede in Netzwerkdaten

- Anomalie-Erkennung anhand der URI und der Methode von HTTP Requests
  - Testdatengenerierung
  - Enkodierung der textuellen Daten
  - Clustering

Anhang (1)

- Self-Organizing Maps
  - Euklidische Distanz:
    - $D(w,x) = \sqrt{\sum_{i=1}^{N}(w_i - x_i)^2} = \|r^2\|$
      - w = Gewichtsvektor, x = Inputvektor, N = Anzahl an Dimensionen von w und x
  - Nachbarschaftsfunktion:
    - $h(r,t) = \exp\left(-\frac{\|r^2\|}{2\sigma^2(t)}\right)$
      - $\|r^2\|$ = Euklidische Distanz, t = Zeit/Iteration, σ^2 = Nachbarschaftsradius

- Xiaofei Qu et al. A survey on the development of self-organizing maps for unsupervised intrusion detection. In: Mobile networks and applications 26 (2021), pp. 808–829
- Corby Ziesman. Self-Organizing Maps. https://slideplayer.com/slide/7798627/ (Abgerufen: 10.11.2023)

Anhang (2)

- k-Means
  - Centroid mit der minimalen Distanz
    - $\arg min_j \left\| x_i - \mu_j \right\|^2$
      - μ = Centroid, x = Inputvektor
  - Mittelpunkt berechnen
    - $c_j = \frac{1}{|\mu_j|} \sum_{x_i \in \mu_i} x_i$
      - c = Mittelpunkt = neuer μ

- Chris Piech. K Means. 2013. URL: https://stanford.edu/~cpiech/cs221/handouts/kmeans.html (Abgerufen: 15.10.2023).
- Jeremy Watt, Reza Borhani, Aggelos K. Katsaggelos. Machine Learning Refined: Notes, Exercises, Presentations, and Sample Chapters. URL: https://jermwatt.github.io/machine_learning_refined/notes/8_Linear_unsupervised_learning/8_5_Kmeans.html (Abgerufen: 15.10.2023)