Fachhochschule
Nordwestschweiz

Application Security (apsi)

Lecture at FHNW

Lecture 14 AS 2021

Arno Wagner, Michael Schläpfer, Rolf Wagner

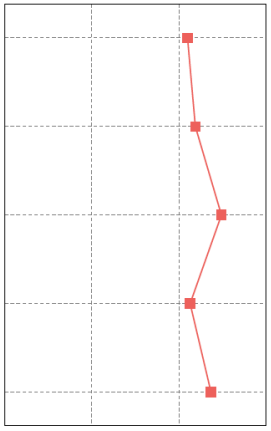
<arno@wagner.name>, <{michael.schlaepfer,rolf.wagner}@fort-it.ch>

Agenda

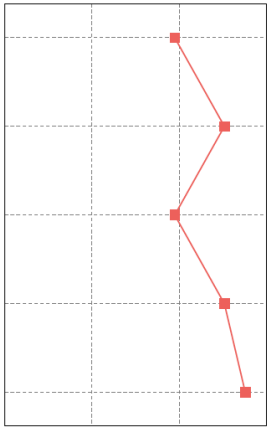
- **Lecture Evaluation Results and Improvement Suggestions**
- API Security
- Automated Application Security Testing
- Guest Talk: Web Application Firewall (WAF) / Identity Federation

Lecture Evaluation Results

Profil-Linie

1. Grosser Beitrag der Studierenden	trifft nicht zu		trifft zu	M: 3.1 n: 16
2. Gutes Lernklima	trifft nicht zu		trifft zu	M: 3.2 n: 16
3. Grosse didaktische Versiertheit	trifft nicht zu		trifft zu	M: 3.5 n: 16
4. Gute Unterrichtsorganisation	trifft nicht zu		trifft zu	M: 3.1 n: 16
5. Grosser Lerngewinn	trifft nicht zu		trifft zu	M: 3.4 n: 15

Profil-Linie

1. Grosser Beitrag der Studierenden	trifft nicht zu		trifft zu	M: 3 n: 7
2. Gutes Lernklima	trifft nicht zu		trifft zu	M: 3.5 n: 7
3. Grosse didaktische Versiertheit	trifft nicht zu		trifft zu	M: 3 n: 7
4. Gute Unterrichtsorganisation	trifft nicht zu		trifft zu	M: 3.5 n: 7
5. Grosser Lerngewinn	trifft nicht zu		trifft zu	M: 3.8 n: 7

Improvement Suggestions

Content

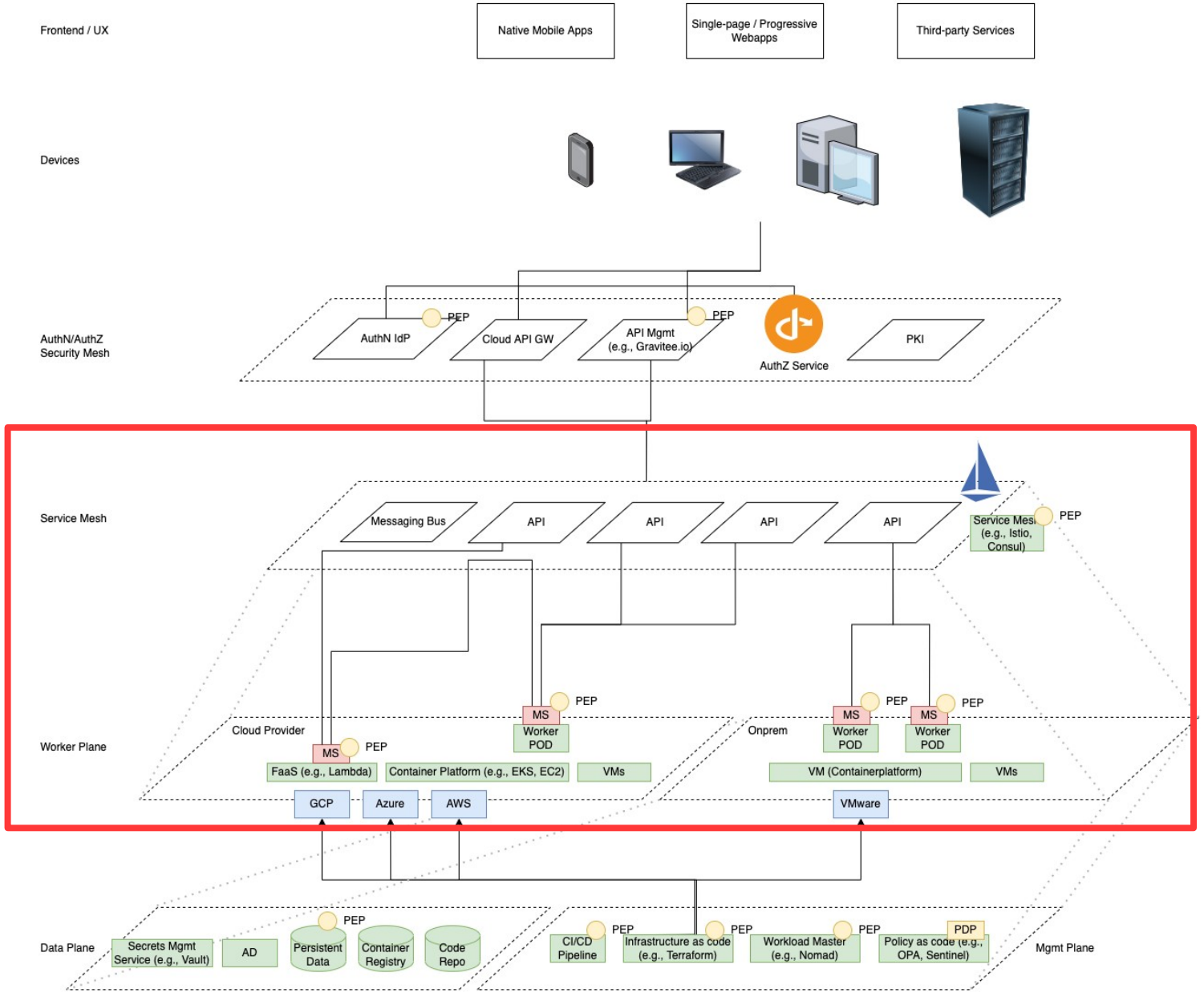
- More practical examples/exercises
- Container Security instead of CGI basics
- “Side-project” that is going to be extended throughout the lectures

Organization/Administration

- AD instead of GitHub
- More central MS Teams integration

Agenda

- Lecture Evaluation Results and Improvement Suggestions
- **API Security**
- Automated Application Security Testing
- Guest Talk: Web Application Firewall (WAF) / Identity Federation



OWASP API Security Project

- OWASP API Security Top 10 2019
- Updates announced every couple of years (last news in 2020)
- https://owasp.org/www-project-api-security/
- Good overview on security best practices
- Cheat Sheets for Devs:
  https://github.com/OWASP/CheatSheetSeries/tree/master/cheatsheets

OWASP API Security Top 10 2019

OWASP API Security Top 10 2019

The Ten Most Critical API Security Risks

Threat Agents	Attack Vectors	Security Weakness		Impacts	
API Specific	**Exploitability: 3**	**Prevalence: 3**	**Detectability: 2**	**Technical: 3**	**Business Specific**
Attackers can exploit API endpoints that are vulnerable to broken object level authorization by manipulating the ID of an object that is sent within the request. This may lead to unauthorized access to sensitive data. This issue is extremely common in API-based applications because the server compone track the instead, r paramete are sent f which ob		This has been the most common and impactful attack on APIs. Authorization and access control mechanisms in modern applications are complex and wide-spread. Even if the application implements a proper infrastructure for authorization checks, developers might forget to use these checks before accessing a sensitive object. Access control detection is not typically amenable to automated static or dynamic testing.		Unauthorized access can result in data disclosure to unauthorized parties, data loss, or data manipulation. Unauthorized access to objects can also lead to full account takeover.	

Threat Agents	Attack Vectors	Security Weakness		Impacts	
API Specific	**Exploitability: 3**	**Prevalence: 2**	**Detectability: 2**	**Technical: 2**	**Business Specific**
Exploitation of Excessive Data Exposure is simple, and is usually performed by sniffing the traffic to analyze the API responses, looking for sensitive data exposure that should not be returned to the user.		APIs rely on clients to perform the data filtering. Since APIs are used as data sources, sometimes developers try to implement them in a generic way without thinking about the sensitivity of the exposed data. Automatic tools usually can't detect this type of vulnerability because it's hard to differentiate between legitimate data returned from the API, and sensitive data that should not be returned without a deep understanding of the application.		Excessive Data Exposure commonly leads to exposure of sensitive data.	

REST Security

https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html

- Use HTTPS
- Use a central IdP for AuthN
- Access Control at each API endpoint using security tokens (e.g., JWT)
- Limit access also to public REST services (rate limiting, API Keys)
- Restrict HTTP methods
- Input validation
- Validate content types
- Avoid exposing management endpoints
- Use generic error messages
- Audit log generation
- Set proper response headers
- Avoid sensitive information in HTTP requests

GraphQL Security

https://cheatsheetseries.owasp.org/cheatsheets/GraphQL_Cheat_Sheet.html

- Input validation
- DoS prevention (e.g., query limiting, rate limiting, etc.)
- Ensure proper Access Control
- Disable introspection queries and schema exploration tools (e.g., GraphiQL)
- Disable “talkative” error messages
- Mitigate batching attacks (e.g., rate limiting, prevent batching for sensitive objects, etc.)

Agenda

- Lecture Evaluation Results and Improvement Suggestions
- API Security
- **Automated Application Security Testing**
- Guest Talk: Web Application Firewall (WAF) / Identity Federation

Application Testing

- SAST (Static application security testing)
  - Whitebox Codeanalysis –> runtime vulnerabilities?
  - Integrates early in SDLC (Software Development Life Cycle)
  - Limitations with dynamically typed languages
- DAST (Dynamic application security testing)
  - Blackbox testing –> Limitations regarding coding standards
  - Less expensive than SAST
  - Supports all kinds of languages and frameworks
- IAST (Interactive application security testing)
  - Software instrumentation –> Potential performance impact
  - “Agent-like” approach using “sensors” –> code and behavior of application

OWASP GitHub Action

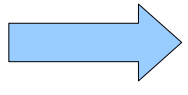
- GitHub action from OWASP:
  https://github.com/marketplace?type=actions&query=owasp

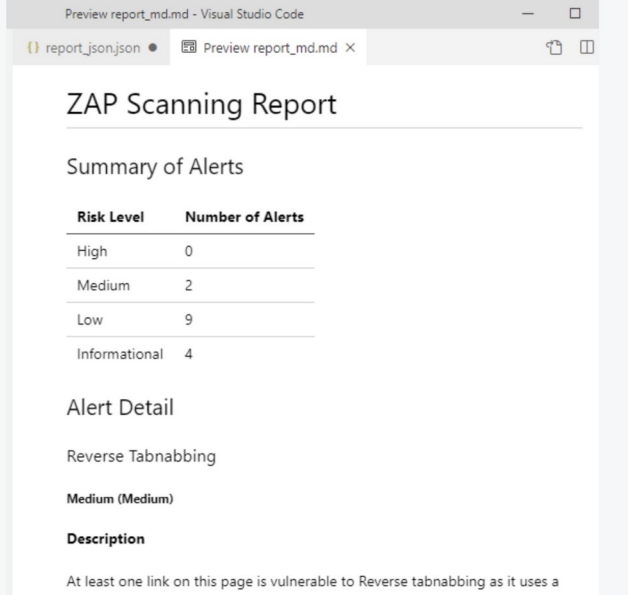
OWASP ZAP Baseline Scan

- Easy to implement (GitHub Action)
- Automatic testing for most common (web app) vulnerabilities
- Extensive reports possible with many false positives
- Easy tutorial:
  https://zimmergren.net/security-scanning-with-github-actions-and-owasp-zap/

```yaml
name: site-security-scan
on:
  push:
    branches: [ master ]
  pull_request:
    branches: [ master ]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
    - name: OWASP ZAP Baseline Scan
      uses: zaproxy/action-baseline@v0.3.0
      with:
        target: "https://your-website.com/"
```





Agenda

- Lecture Evaluation Results and Improvement Suggestions
- API Security
- Automated Application Security Testing
- **Guest Talk: Web Application Firewall (WAF) / Identity Federation**

10.01.22: Guest Speaker about WAF/Identity Federation

UNITED SECURITY PROVIDERS

- Christoph Schulthess, Head of Value Stream Security Products @ United Security Providers AG, Mitglied der Geschäftsleitung
- Corporate Networks and Application Level Security using Web Application Firewalls
- Integration with web applications and consequences on software development/deployment for you