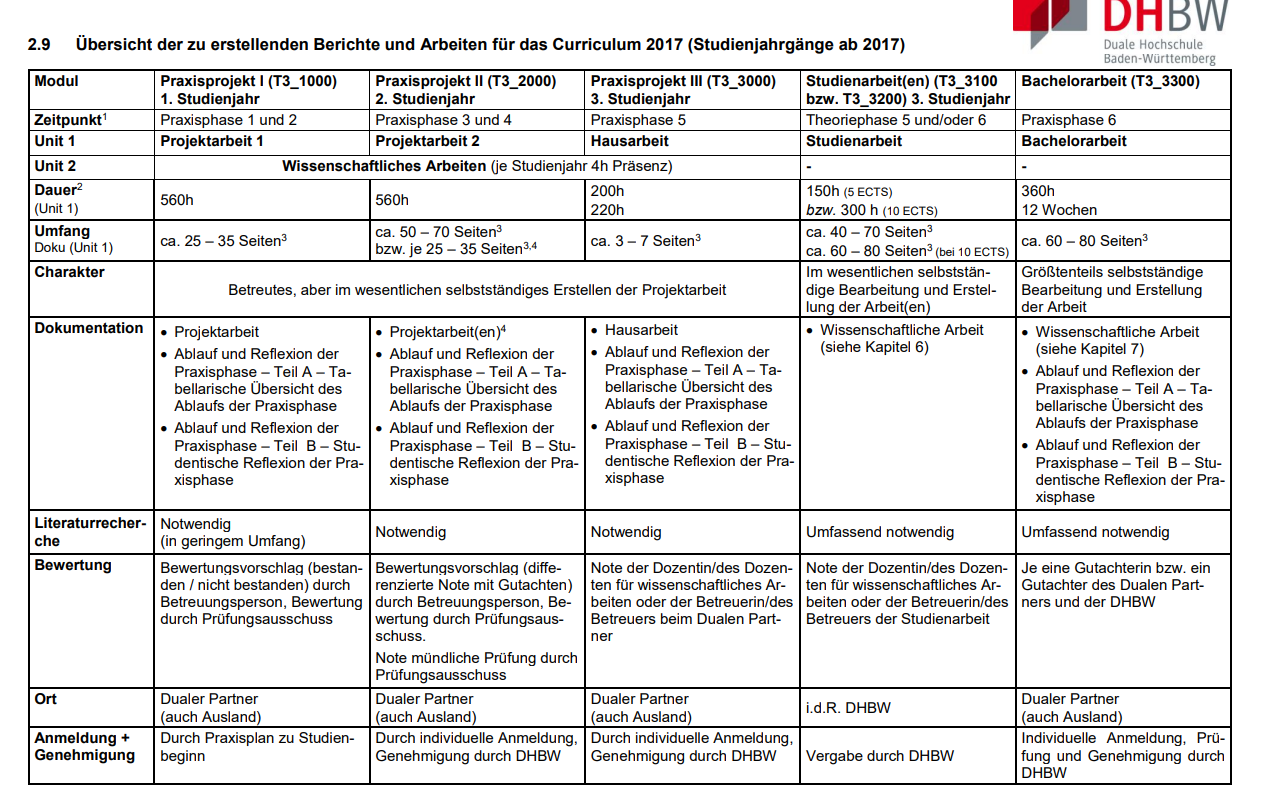
**Praxisarbeit II**

Themenideen:

* Erstellung einer Methode zur automatisierten Vulnerability-Scanning in Unternehmensnetzwerken/Web-Applikationen
  + Vergleich mehrerer tools
    - Pentesting - Generell
    - Vorgehen
* Automatisiertes Erkennung von Schwachstellen in Web Applikationen
* Analyse von Malware-Infektionsketten und Entwicklung von Erkennungsmechanismen
* Untersuchung von Angriffstechniken auf Cloud-Umgebungen und Entwicklung von Schutzmaßnahmen
  + Einzelne Themen, Vergleich bei verschiedenen Providern
  + Prüfen auf wissen. Input
* Analyse von Sicherheitslücken in Web-Applikationen/mobile Applikationen/andere Systeme und Entwicklung von Gegenmaßnahmen

Umfang: 50-70 Seiten,/ 25-30 Seiten je



Zunächst soll eine wissenschaftliche Grundlage durch eine gründliche Literaturrecherche zum Thema Automatisierte Erkennung und  
Ausnutzung von Schwachstellen in Web-Applikationen geschaffen werden, um einen Überblick über die aktuelle Forschungslage zu erhalten. Dann wird der Unterschied zwischen manueller und automatisierter Schwachstellenerkennung anhand einer Reihe von Anwendungsbeispielen untersucht. Daraufhin werden bereits vorhandene Vorgehensweisen und Methoden im Bereich des automatisierten Schwachstellenerkennung analysiert. Die Ergebnisse werden anschließend interpretiert, um Rückschlüsse auf die Effektivität der untersuchten Methoden ziehen zu können. Anhand der Ergebnisse sollen anschließend Empfehlungen für die automatisierte Erkennung und Ausnutzung von Schwachstellen in Web-Applikationen gegeben werden.

Web-Applikationen sind ein wichtiger Bestandteil des modernen Geschäftsbetriebs und bieten oft Zugang zu sensiblen Daten und Funktionalitäten. Eine erfolgreiche Ausnutzung von Schwachstellen in diesen Anwendungen kann jedoch zu ernsthaften Sicherheitsproblemen führen. Es gibt bereits viele verschiedene Methoden und Werkzeuge zur manuellen und automatisierten Erkennung und Ausnutzung von Schwachstellen in Web-Applikationen. Daher ist nur schwer ersichtlich, welche Praktiken und zugehörige Tools geeignet sind um automatisiert Schwachstelen in Web-Applikationen zu erkennen und auszunutzen Ziel der Arbeit ist eine Ermittlung und Evaluierung der Effektivität einzelner Praktiken bei der automatisierten Erkennung und Ausnutzung verschiedener Schwachstellen in Web-Applikationen. Insgesamt soll dadurch eine zielgerichtetere Erkennung von Schwachstellen in Web-Applikationen erhöht werden.

Web-Applikationen sind ein wichtiger Bestandteil des modernen Geschäftsbetriebs und bieten oft Zugang zu sensiblen Daten und Funktionalitäten. Eine erfolgreiche Ausnutzung von Schwachstellen in diesen Anwendungen kann jedoch zu ernsthaften Sicherheitsproblemen führen. Es gibt bereits viele verschiedene Methoden und Werkzeuge zur manuellen und automatisierten Erkennung und Ausnutzung von Schwachstellen in Web-Applikationen. Daher ist nur schwer ersichtlich, welche Praktiken und zugehörige Tools geeignet sind um automatisiert Schwachstelen in Web-Applikationen zu erkennen und auszunutzen Ziel der Arbeit ist eine Ermittlung und Evaluierung der Effektivität einzelner Praktiken bei der automatisierten Erkennung und Ausnutzung verschiedener Schwachtellen in Web-Applikationen. Insgesamt soll dadurch eine zielgerichtetere Erkennung von Schwaschstellen in Web-Applikationen erhöht werden.

Automatisierte Erkennung und Ausnutzung von Schwachstellen in Web-Applikationen

1. Einleitung
   1. Einordnung des Themas
   2. Ziel der Arbeit
   3. Aufbau der Arbeit
2. Grundlagen
   1. Pentesting
   2. Owasp top 10
   3. Automatisiertes Pentesting
   4. Verschiedene Tools

Automatisiertes Pentesting unter Verwendung von Deep Learning  
<http://repository.kpi.kharkov.ua/bitstream/KhPI-Press/54956/1/AIS_2021_5_3_Semenov_Automated_penetration.pdf>

1. Attack tree paterns untersuchen und Informationen sammeln , wie z.B. Shodan
   * Nutzt shadow kanälu und stellt ununterbrochen anfragen und sucht nach vulnerabilities
2. Matrix aus attack tree paterns bilden (Mulval)
   * Transformation eines attack trees in eine cyber intrusion matrix via an Algorithm