Owasp top 10 2022 REST-API

Im Laufe des Moduls „HDA WS2022/2023 – Security of Web Applications” habe ich eine REST-API in der Programmiersprache Java mithilfe des Frameworks „SpringBoot“ entwickelt. Die API wurde mithilfe von Postman auf ihre Funktionalität überprüft und auf Github hochgeladen: <https://github.com/AliIgnis/spring-boot-tutorial>. Im folgenden wird die API auf die OWASP-Top 10 überprüft.

1. Broken Access Control

Meine Anwendung enthält diese Schwachstelle. Zwar wird beim Löschen eines Benutzers, ein Token benötigt, um zu verifizieren, dass man Autorisiert ist diese Funktionalität durchzuführen, aber es findet hier keine Kontrolle von Rechten oder Rollen Statt. Außerdem, wird an keiner anderer Stelle überprüft, ob ein Nutzer Rechte hat, die Schnittstellen Funktionen aufzurufen. Beispielsweise können die Nutzerinformationen von dritten angefragt werden ohne das eine Zugansberechtigung stattfindet. Daher könnte jeder mit der richtigen URL beispielsweise meine E-Mail auslesen, solange meine meine ID bekannt ist oder geraten wird.

1. Cryptographic failures

Meine Anwendung ist von dieser Schwachstelle betroffen. Es werden keine Informationen verschlüsselt und alles per Klartext übertragen. Daher kann jede Person einfach mithören und die jeweiligen Informationen auslesen. Auch Personengeschützte Daten wie Vornamen oder Nachnamen werden nicht geschützt.

1. Injection

Meine Anwendung ist generell von dieser Schwachstelle betroffen. Es findet in meinem Code keine Überprüfung der eingelesenen Informationen statt. Weshalb der Input des Nutzers dazu führen kann ungewollte Aktionen mit der Anwendung durchzuführen. Selbst, wenn jetzt kein großes Problem besteht, könnte bei der weiter Verwendung der Anwendung und ausbauen das Risiko steigen.

1. Insecure Design

Meine Anwendung ist auch von dieser Schwachstelle betroffen, da keine Überprüfung des Standes gegenüber Potenziellen Bedrohungen stattfindet. Außerdem habe ich keine Sicherheitskontrollen durchgeführt.

1. Security misconfiguration

Meine Anwendung ist auch von dieser Schwachstelle betroffen. Meine Anwendung enthält nicht benötigte Endpunkte. Außerdem könnten weitere Schnittstellen enthalten sein, da keine Sicherheitseinstellungen angepasst wurden und die Default Einstellungen verwendet werden.

1. Vulnerable outdated components

Meine Anwendung ist von dieser Schwachstelle nicht betroffen. Innerhalb von Intelij kann überprüft werden, up ou dated Funktionen genutzt werden und hierfür bestand keine Warnung innerhalb des Projektes. Außerdem wurden die neusten Versionen der libraries innerhalb der .pom geladen.

### Identification and authentication failures

Meine Anwendung ist von dieser Schwachstelle dahingehend nicht betroffen da nur an einer Stelle eine Überprüfung stattfindet. Dies findet bei der Löschung des Nutzers statt dort wird der Token überprüft. Falls dieser Falsch ist, kann der jeweilige Nutzer nicht gelöscht werden.

### Software and data integrity failures

Meine Anwendung ist im allgemein von dieser Schwachstelle betroffen. Dies liegt daran das keine Überprüfung von neuen Libraries oder Versionen eingerichtet ist. Dementsprechend könnten neue Versionen und Libraries mit Potenziellen Risiken ohne Hinweis eingepflegt werden.

### Security logging and monitoring failures

Meine Anwendung ist im Allgemeinen von dieser Schwachstelle betroffen. Dies liegt daran das ich kein logging betreibe. Dementsprechend könnten auch Angriffe nicht nachvollzogen werden. Dies macht es umso schwerer potenziellen Schwachstellen zu finden und herauszufinden, welche Systemstellen den Angriff ermöglicht haben.

### Server-side request forgery (SSRF)

Meine Anwendung könnte auch von diesen Schwachstellen betroffen sein, da die API keine Firewall enthält. Bei Aufrufen einer nicht festgelegten URL, wird zwar eine Error Page angezeigt, aber die Url wird vorerst ausgeführt. Es besteht auch keine Deny-by-default policy und Dateneingaben werden weder überprüft und auch nicht gereinigt.