Seminar Softwareentwicklung mit Dev(Sec)Ops

Dynamic Application Security Testing (DAST)

Tobias Kiehnlein

Inhaltsverzeichnis

1 Einleitung

Mit der Verbreitung agiler Methoden und moderner Softwareentwicklung wird stets das Ziel verfolgt in kurzen Abständen Änderungen am Code vorzunehmen und diesen zu deployen. Nicht selten stellt sich ein Deployment als vergleichsweise komplex dar. Dies hat zur Folge, dass release cycles länger werden, deployments eine unangenehme Last werden und security oftmals völlig außer Acht gelassen wird.

Gerade deshalb sollte man sich über die Prozesse Gedanken machen, die nach der Implementation des Codes stehen. Diese Aufgabe stellt sich ebenfalls als nicht ganz trivial heraus. Warum sollte man sich diesen Aufwand also machen? Hier spielen viele Faktoren eine Rolle, allerdings wird sich diese Arbeit primär auf den Aspekt der Anwendungssicherheit fokusieren. Nicht selten haben Entwickler keine oder nur wenig Ahnung von IT-Sicherheit. Nicht grundlos befinden sich seit Jahren bekannte Sicherheitslücken wie SQL-Injections, XSS oder XSRF immernoch unter den häufigsten Sicherheitslücken im Web.[1]

Wie lässt sich nun ein derart fundamentales Problem in der IT Industrie lösen? Die einzige Möglichkeit ist, die eigene Software regelmäßig auf Schwachstellen zu untersuchen, um gerade häufigen Sicherheitslücken vorbeugen zu können. Dies ist jedoch häufig mit hohen Kosten verbunden, da Sicherheitsexperten für hohes Budget wiederholte triviale Tests durchführen müssen. Die Lösung scheint also einfach: Automatische Sicherheitstest, welche die Anwendung auf die bekanntesten Lücken prüft und den Entwicklern und Sicherheitsexperten direktes Feedback gibt, um IT-Security effizient zu gestalten.

2 Vergleich zwischen Burp Suite, OWASP ZAP und GitLab für den Einsatz im DAST

2.1 Burp Suite

2.2 OWASP ZAP

2.3 GitLab

2.4 Fazit

3 Analyse eines optimalen Dec(Sec)Ops Prozesses

Literatur

[1] *The Invicti AppSec Indicator*. 2021. URL: https://www.acunetix.com/white-papers/acunetix-web-application-vulnerability-report-2021/ [20. Aug. 2021].

Abbildungsverzeichnis