Fraunhofer FOKUS

Vertrauen stärken und Regulierung proaktiv umsetzen

Umfassende Sicherheits-tests leicht gemacht

FUZZINO

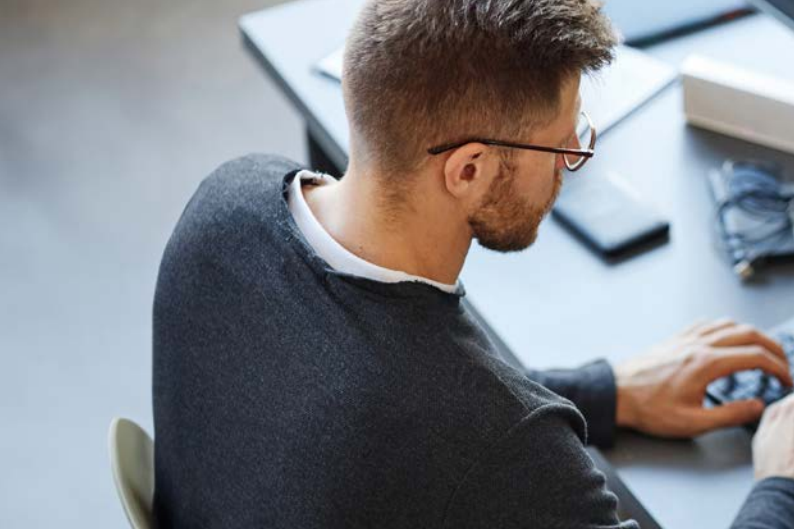
Umfassende Sicherheitstests leicht gemacht

Unsere Lösung für Sicherheitstests Fuzzino nutzt fortschrittliche Fuzzing-Techniken, um eine hervorragende Testabdeckung für alle Ihre Softwarekomponenten zu gewährleisten. Sie ermöglicht Ihnen, Schwachstellen in Ihren Produkten zu adressieren und Haftungsrisiken durch die Einhaltung des Cyber Resilience Acts zu minimieren, ohne dass dafür spezielle Kenntnisse über Sicherheitstests notwendig sind.

Die Zahl von Cyberangriffen auf Unternehmen steigt jedes Jahr. 2024 waren allein in Deutschland 60 Prozent der Unternehmen betroffen, wodurch ein Schaden von mehr als 200 Milliarden Euro entstanden ist. Die Europäische Union hat daher Vorschriften erlassen, bspw. den Cyber Resilience Act, um diesem Trend entgegenzuwirken, indem sie den Herstellern neue Verpflichtungen in Bezug auf die Produktsicherheit auferlegt. Dazu gehören auch regelmäßige Sicherheitstests zur Ermittlung von Schwachstellen – sowohl im Quellcode des Herstellers, als auch für eingebundene Softwarekomponenten

Die wichtigsten Verpflichtungen aus dem Cyber Resilience Act in Bezug auf Sicherheitstests:

1. Schwachstellen ermitteln und dokumentieren, auch für Software von Drittanbietern und Open-Source-Software.
2. Schwachstellen unverzüglich beheben, bspw. durch Sicherheitsupdates.
3. Die Sicherheit des Produkts regelmäßig und wirksam testen und überprüfen.



Fuzzino ist auf Benutzerfreundlichkeit ausgelegt:

Dritter. Mit der neuen Produkthaftungsrichtlinie werden Hersteller digitaler Produkte nicht nur für Schäden haftbar gemacht, die durch Sicherheitslücken verursacht werden, sondern auch für alle Schäden, die ein Angreifer unter Ausnutzung einer Sicherheitslücke verursacht. Die neuen Vorschriften gelten für viele Hersteller, die bisher nicht gesetzlich verpflichtet waren, Sicherheitstests durchzuführen. Auch Open-Source-Software ist von diesen neuen Regeln betroffen und stellt die Hersteller vor zusätzliche technische Herausforderungen.

Der nächste Schritt für mehr Sicherheit: Identifizierung von Schwachstellen tief im System

Sicherheitstests haben sich in den letzten 30 Jahren stark weiterentwickelt und umfassen heute eine Vielzahl von Methoden, Techniken und Werkzeugen. Ihr schiere Zahl kann dabei immer noch überwältigend sein. Fuzzing ist eine der am häufigsten und erfolgreichsten Techniken für den Sicherheitstest. Die Kernidee besteht darin, ein System mit zufällig generierten Eingaben auszuführen, um Schwachstellen aufzudecken. Auch wenn Fuzzing sehr effektiv sein kann, krankt der Einsatz vieler Fuzzing-Werkzeuge an verschiedensten Herausforderungen und Einschränkungen. Diese können insbesondere für

Es erfordert keine besonderen Fähigkeiten im Bereich Sicherheitstests, sodass jeder Tester davon profitieren kann

Unternehmen, denen es an Erfahrung oder Ressourcen für Sicherheitstests fehlt, eine große Herausforderung darstellen.

Fuzzino löst diese Herausforderungen und Einschränkungen mit einer Reihe innovativer Funktionen. Mit unserem Werkzeug werden Sicherheitstests erheblich effizienter und effektiver und die Benutzerfreundlichkeit deutlich verbessert. Diese Verbesserungen ermöglichen es Herstellern ohne spezielle und teure Schulungen, Sicherheitstests auf dem neuesten Stand der Technik durchzuführen.

Die Hauptmerkmale von Fuzzino umfassen:

- **Benutzerfreundlichkeit:** Fuzzing setzt oft spezifische Kenntnisse und Fähigkeiten für den effektiven Einsatz heutiger Werkzeuge voraus, um sie in eine Testumgebung zu integrieren und ihre Ergebnisse zu interpretieren. Dies erschwert oder verhindert den großflächigen Einsatz von Fuzzing. Um einen Fuzzer mit einer Komponente zur Übergabe der Fuzz-Testdaten zu verbinden, ist oft ein sogenanntes Fuzzing-Harness notwendig. Dessen Implementierung erfordert manuellen Aufwand und Kenntnisse über die zu testende Komponente sowie über das verwendete Fuzzing-Werkzeug. Unsere Lösung macht Fuzzing-Harnesse überflüssig, denn sie ermöglicht es

Ihnen, Ihren Adapter aus dem funktionalen Testen ohne Änderungen wiederzuverwenden.

- **Geschwindigkeit:** Heutige Fuzzing-Werkzeuge benötigen oft lange Laufzeiten, d. h. Stunden oder Tage, um Schwachstellen zu identifizieren. Dies macht die Integration in tägliche Tests, Build-Pipelines und DevOps-Prozesse schwierig. Mit unserer fortschrittlichen Testgenerierungs-Engine können Schwachstellen in Minuten statt in Stunden oder Tagen identifiziert werden.
- **Adaptive Erkennung von Sicherheitslücken:** Sicherheitslücken können sich in Abstürzen oder Speicherfehlern äußern. Bugs haben jedoch oft subtilere Auswirkungen. Bestehende Fuzzer erkennen diese Arten von Sicherheitslücken in der Regel nicht, da sie sich auf Speicherkorruption konzentrieren. Außerdem beschränken sich die meisten Fuzzing-Werkzeuge auf die Maximierung der Codeüberdeckung, die nicht unbedingt mit dem Aufdecken von Sicherheitslücken korreliert. Unsere Lösung kann über die Maximierung der Codeüberdeckung als Testziel hinaus jede beliebige Laufzeiteigenschaft nutzen, um z. B. Denial-of-Service-Schwachstellen zu identifizieren.
- **Zustandsabhängigkeit:** Schwachstellen sind oft tief in der Geschäftslogik des Systems versteckt. Viele Fuzzer können sie nicht finden, da ihre Eingaben in frühen

Künftig müssen Hersteller über den gesamten Lebenszyklus ihrer Produkte und Anwendungen die Verantwortung für deren Cybersicherheit übernehmen.«

Claudia Plattner, BSI-Präsidentin

Bildhaft ausgedrückt bedeutet Fuzzing, dass viele Pfeile auf eine zu testende Softwarekomponente abgeschossen werden, um Schwachstellen darin zu finden

Verarbeitungsphasen vom System abgewiesen werden und sie nicht zustandsorientiert arbeiten. Unsere Lösung bietet eine einfache und intuitive Möglichkeit, Nachrichtensequenzen und Systemzustände zu beschreiben. Auf der Grundlage dieser Beschreibungen findet unsere Lösung effektiv Sicherheitslücken tief im System, die andere Fuzzing-Werkzeuge übersehen.

- **Skalierbare Protokollbeschreibungen:** Viele Fuzzer benötigen wohlgeformte Beispieleingaben, sogenannte Seeds. Es ist jedoch nicht trivial, einen Datensatz zu erhalten, der erschöpfend und repräsentativ ist. Unsere Lösung setzt skalierbare Protokollbeschreibungen ein. Diese ermöglichen es Ihnen, protokollkonforme Eingaben nicht nur durch Beispieleingaben (d. h. Seeds) zu beschreiben, sondern auch durch die Beschreibung ihrer Datenstrukturen und der Regeln, denen sie folgen. Solche Beschreibungen können oft direkt aus offiziellen Spezifikationen bezogen oder aus diesen abgeleitet werden.

Entwicklerinnen und Tester ohne Kenntnisse im Bereich Sicherheitstests profitieren von diesen Funktionen, um umfassende Sicherheitstests für alle Softwarekomponenten eines Produkts durchzuführen.

Open-Source-Fallstudien

Die Effektivität von Fuzzino wurde anhand von zwei Open-Source-Fallstudien bestätigt: Eclipse® Mosquitto und NanoMQ. Bei beiden handelt es sich um Broker für das MQTT-Protokoll, das im Internet der Dinge häufig zum Austausch von Nachrichten, z. B. in der Fertigung, der Automobilindustrie oder der Landwirtschaft, verwendet wird. Wir haben diese Broker mit Fuzz-Testdaten an ihren funktionalen MQTT-Schnittstellen getestet, ohne einen speziellen Adapter für Sicherheitstests zu entwickeln. Fuzzino generierte und sendete gefuzzte MQTT-Nachrichten an die Broker. Diese Nachrichten berücksichtigten auch den Zustand des MQTT-Brokers, was es Fuzzino ermöglichte, komplexere Interaktionen als mit herkömmlichen Fuzzern zu erzeugen. Dadurch wurden mehrere Zero-Day-Schwachstellen in beiden Brokern gefunden.

In Mosquitto hat Fuzzino eine hochgradig gefährliche Schwachstelle (CVE-2024-8376) aufgedeckt, die Ressourcenüberbeanspruchungen und ungültige Speicherzugriffe ermöglicht und den gesamten Broker zum Absturz bringen kann. Diese Schwachstelle wurde in nur wenigen Minuten aufgedeckt. Im Gegensatz dazu hat Google OSS Fuzz die von uns getestete Mosquitto-Version mehr als ein Jahr lang mit mehreren Open-Source-Fuzzern überprüft, ohne die Schwachstelle zu finden. Bei NanoMQ hat Fuzzino zwei Schwachstellen aufgedeckt, die jeweils in wenigen Minuten zu ungültigen Speicherzugriffen führen. Dies ist eine drastische Verbesserung im Vergleich zu den in der Branche üblichen stunden- oder tagelangen Fuzzing-Prozessen.

Unsere Lösung für Sicherheitstests Fuzzino steigert die Effizienz, Effektivität und Benutzerfreundlichkeit von Sicherheitstests erheblich. Durch die Berücksichtigung des Serverzustands während der Generierung und Ausführung von Fuzz-Testfällen werden komplexe, hochgefährliche Schwachstellen äußerst schnell identifiziert. Dieser Ansatz übertrifft herkömmliche Fuzzer, indem er die Erkennungszeit von Stunden oder Tagen auf nur wenige Minuten reduziert und insbesondere kritische Probleme aufdeckt.

Kontakt

Dipl.-Inform. Martin Schneider
Gruppenleiter Testen
Geschäftsbereich Quality Engineering
Tel. +49 30 3463-7383
martin.schneider@fokus.fraunhofer.de

Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin

www.fokus.fraunhofer.de/de/sqc/security_testing

www.fokus.fraunhofer.de

Wir vernetzen alles

iku | 2502 (Fotos: Titel: AI-generated / Fraunhofer FOKUS,
S. 3-4: istock / SeventyFour; S. 6: AI-generated / adobe stock)