R2.1: Stand der Testtechniken im Projekt

Abgleich Wissensstand im Projekt

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

aufgrund eines Beschlusses des Deutschen Bundestages

Version 1.0, Datum: 30.06.2017

Autoren:

Frank-Walter Jäkel	- Fraunhofer IPK
Theo Margraf	- AUDI AG
Stefan Stoelzle	- AUDI AG
Paul Hopton	- Relayr
Yuliya Brynzak	- Relayr
Michael Wagner	- Fraunhofer FOKUS
Axel Rennoch (Ed.)	- Fraunhofer FOKUS
Sascha Kretzschmann	- Fraunhofer FOKUS
Rutten, Stefan	- DEKRA
Andre Wardaschka	- DEKRA

Inhalt

1. Einleitung ... 3
1.1. Prüfobjekte und -ziele ... 3
1.2. Prüfarchitekturen ... 3
2. Stand der Prüfmethoden ... 6
2.1. Funktionalität / Konformität ... 6
2.1.1. Monitoring ... 6
2.1.2. Konnektivitätstests ... 6
2.1.3. Protokolltests ... 6
2.1.4. Anwendungsszenarien ... 7
2.1.5. Weitere (z.B. crowd-testing) ... 7
2.2. Interoperabilität ... 7
2.2.1. Plugfest ... 7
2.2.2. Test Suites ... 8
2.2.3. Referenz Implementierungen ... 8
2.3. Sicherheit ... 8
2.3.1. Schwachstellen Datenbanken und Scannen ... 8
2.3.2. Zero-Day/Fuzzing ... 8
2.3.3. Analyse des Verkehrs ... 8
2.3.4. Analyse des Source Codes ... 9
2.3.5. Social Engineering ... 9
2.4. Performanz / Robustheit ... 9
3. Prüflösungen und Werkzeuge (Open Source) ... 9
3.1. Funktionalität / Konformität ... 9
3.2. Interoperabilität ... 14
3.3. Sicherheit ... 18
3.4. Performanz ... 30
4. Zusammenfassung und Ausblick ... 30
5. Referenzen ... 31

1. Einleitung

Das Projekt „Ein Testlab und Testware für Internet der Dinge-Lösungen und -Geräte“ des BMWi kurz IoT-T hat als Ziel Firmen bei der Erstellung von IoT basierten Lösungen und Produkten in den Bereichen Qualitätssicherung und Zertifizierung zu unterstützen. Hierzu sind die Erstellung einer IoT Testware und die Etablierung mindestens eines IoT-Testlab geplant. Die IoT-Testware wird beim automatisierten testen von IoT relevanten Technologien wie z.B. Protokollen helfen und u.a. im IoT-Testlab zum Einsatz kommen. Dabei wird das IoT-Testlab Technologien wie CoAP und MQTT adressieren und gleichzeitig auf Standards wie z.B. TTCN3 aufsetzen. Das IoT-Testlab wird als praktisches Angebot für Firmen durch die DEKRA etabliert werden. Es soll Firmen ermöglichen auf IoT Testexpertise zugreifen zu können und Anwendungen zertifizieren zu lassen. Langfristig soll Firmen ermöglicht werden qualitative, sichere und interoperable IoT Lösungen zu erstellen.

Der vorliegende Report beinhaltet erste Ergebnisse bzgl. der Analyse von IoT-Prüfmethoden und Werkzeuge. Insbesondere werden die Open source Werkzeuge betrachtet. Die für die Prototypische Realisierung ausgewählten Ansätze werden im weiterführenden Report (R2.2) spezifiziert. Beteiligt an der Erstellung des Reports waren alle Partner des Projektes: Relayr, Audi, DEKRA, Fraunhofer FOKUS und Fraunhofer IPK.

1.1. Prüfobjekte und -ziele

Da der Einsatz von IoT Prüfmethoden von den jeweiligen Prüfobjekten und -zielen abhängig ist müssen zunächst die hauptsächlichen Betrachtungsgegenstände und mögliche Testansätze benannt werden. Es sollen zunächst folgende Prüfobjekte unterschieden werden, u.a.:

- Ein einzelner Micro Controller, der Messdaten von ein oder mehreren Sensoren erhält und an einer Kommunikationsschnittstelle bereithält.
- Kommunikationsprotokolle, die zwischen einem „Think“ (z.B. Micro Controller) und einem Server oder Gateway aktiv sind, z.B. CoAP, MQTT etc.
- IoT Gateway, die Daten vor Ort von mehreren Datenquellen einsammeln, aufbereiten und in der Cloud zur Verfügung stellen.
- Server in der Cloud, die Daten aus mehreren Anwendungen und Gebieten aufbereiten und für den Endbenutzer zur Verfügung stellen.
- Endgeräte der Benutzer.
- IoT Infrastrukturen (Plattformen für die Anbindung von IoT Geräten, IoT-Servicelayer über unterschiedlichen Protokollstapel, etc.)
- IoT Lösungen für bestimmte Kundenanforderungen oder Produktionsprozesse

1.2. Prüfarchitekturen

Die folgenden Abbildungen illustrieren Beispiele von Prüfarchitekturen zu ausgewählten Prüfobjekten.

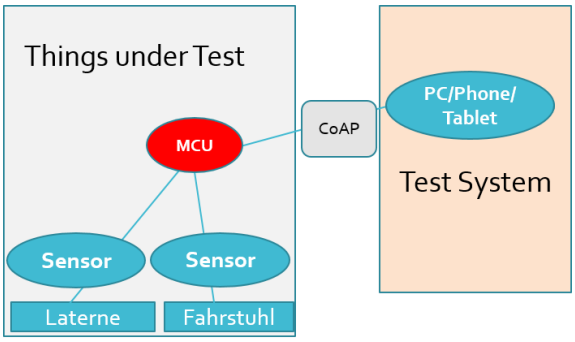

Fig. 1: Einfache Testkonfiguration für einen Microcontroller (under test)

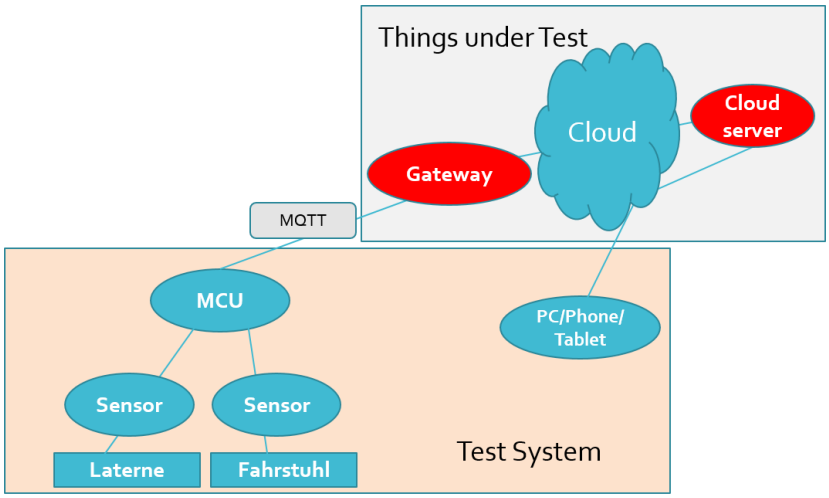

Fig. 2: Testkonfiguration für ein Gateway unter Beteiligung eines Servers in der Cloud (under test)

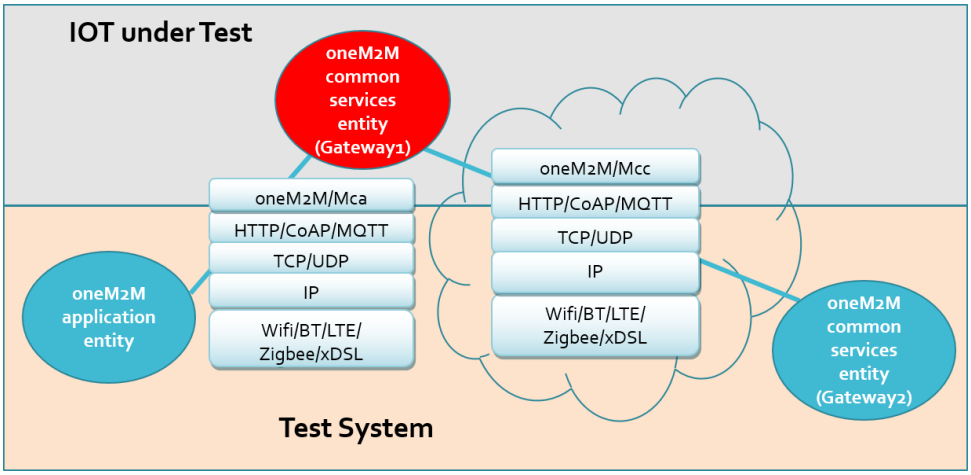

Fig. 3: Einfache Testkonfiguration für eine IoT Middleware (under test)

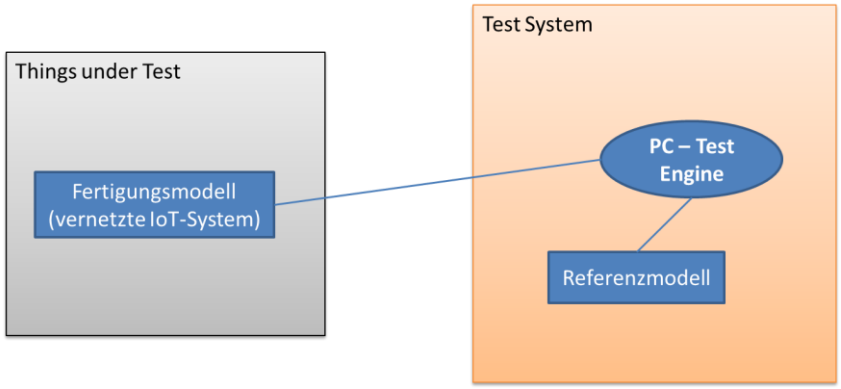

Fig. 4: Test von Services von vernetzten IoT-Systemen (Komptabilitätstest)

Um sicher zu stellen das vernetzte IoT-Systeme korrekt zusammenarbeiten können, werden die in Abbildung 4 und Abbildung 5 dargestellten Testkonfigurationen benutzt. Im ersten Test wird ein Fertigungsmodell gegen ein Referenzmodell getestet. Bei diesem Komptabilitätstest soll sichergestellt werden, dass nur bestimmte IoT-Systeme miteinander verknüpft werden können. Wenn der erste Test erfolgreich absolviert worden ist, können in einem zweiten Test die Schnittstellen der einzelnen IoT-Systeme gegeneinander getestet werden. Um die Schnittstellen zweier verbunder IoT-Systeme zu testen, werden die Input- und Output-Parameter verglichen.

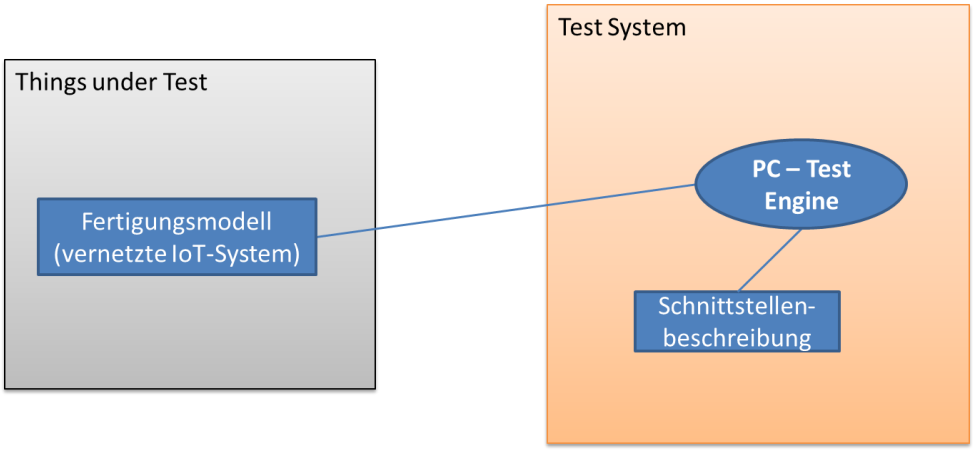

Fig. 5: Test von Services von vernetzten IoT-Systemen (Schnittstellentest)

Die Abbildung 6 illustriert eine Testkonfiguration für einen Verfügbarkeitstest von Services zu einer Anlage als IoT System (hier Roboter). Dieser Test soll sicherstellen, dass die spezifizierten Services einer Anlage zur Verfügung stehen. Dieser Test muss von einem Anlagen- oder Systemlieferant erfüllt werden und basiert auf vorgegebenen Spezifikationen, welche zukünftig auch standardisiert werden könnten. Hier soll unteranderem sichergestellt werden, dass sich neue IoT Systeme sicher in bestehende IoT Infrastrukturen und Netzwerke einfügen können.

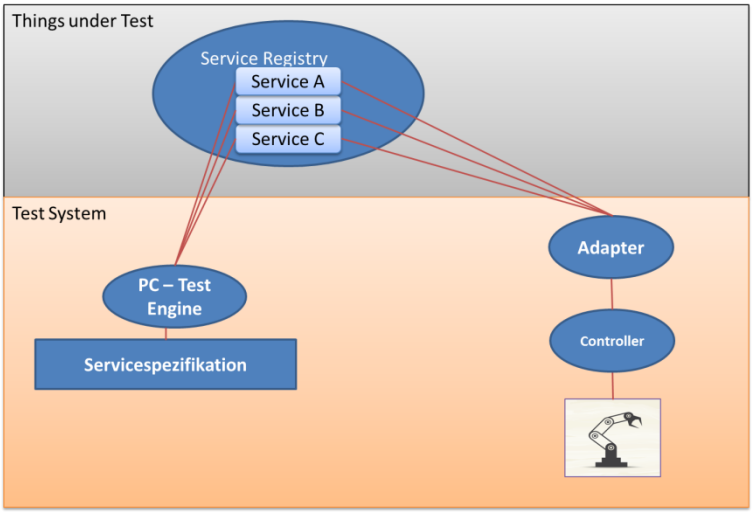

Fig. 6: Test auf Vorhandensein von Services in einer Service Registry

2. Stand der Prüfmethoden

Die für IoT benötigten Testmethoden und -werkzeuge sind grundsätzlich nicht unbedingt neu, sondern ergeben sich vielmehr aus bekannten Ansätzen und Werkzeugen [Bucsics15]. Bei der Auswahl sind die für IoT typischen Eigenschaften als auch die im Projekt über die Anforderungsanalyse ermittelten Besonderheiten zu berücksichtigen.

2.1. Funktionalität / Konformität

2.1.1. Monitoring

Zu den einfachen Prüfmethoden zählt das passive Beobachten der Prüfobjekte während des Betriebs in einer Laborumgebung oder im realen Einsatz beim Anwender, d.h. die Prüfgegenstände werden in Ihrem Zustand nicht beeinflusst, weder durch die Entgegennahme von Testnachrichten noch durch die Anforderung von Statusinformationen.

Das Monitoring kann rein manuell durch einen Operator erfolgen, ggf. unter Einbeziehung von passiven Beobachtungswerkzeugen als auch vollautomatisch durch den Einsatz eins Frameworks, der an mehreren Beobachtungspunkten gleichzeitig und auf einander abgestimmt Ablaufinformationen bezieht, in Bezug setzt und auswertet.

2.1.2. Konnektivitätstests

Im Rahmen von aktiven Testszenarien stehen die grundlegenden Tests zur Ermittlung der Fähigkeit der Kommunikation der Testobjekte mit ihrer Umgebung am Anfang der Testkataloge. Hierbei handelt es sich oft um reine Verbindungstests über die beteiligten Kommunikationsprotokolle.

2.1.3. Protokolltests

Neben den elementaren Konnektivitätstest spielen die speziellen Funktionen, Daten und Features aller beteiligten Protokolle von der Netzwerkschicht bis hin zur Anwendungsschicht eine wichtige Rolle in IoT

Infrastrukturen und Szenarien. Während bewährte und lang erprobte Netzwerkprotokolle wie TCP/IP etc. keine neuen Tests bestehen müssen liegt der Fokus derzeit bei den für IoT typischen Protokollen CoAP, MQTT, 6LowPAN oder LPWAN in Verbindung mit IoT Testobjekten in spezifischen Umgebungen und unter realen Betriebsbedingungen.

Für zahlreiche Protokolle aus der Standardisierung (z.B. in der Telekommunikation bei ETSI) werden neben der Definition der Protokolle zugleich auch Testzielkataloge (test purposes) bzw. (semi)-formale Testbeschreibungen (z.B. mit TTCN-3) veröffentlicht. Die Testkataloge beinhalten in der Regel die Einhaltung der Nachrichten- bzw. Datenformate (bis zur einzelnen Bitfeldern) unter Verwendung regulärer als auch fehlerhafter Dateninhalte. Darüber hinaus gilt es die protokollspezifischen Abläufe durch Tests sicherzustellen, z.B. Auf-/Abbau bzw. Modifikation von Anwendungssessions oder die Prozeduren zu Subscription / Notification von Informationsereignissen. Ggf. sind hier die spezifischen Rollen der Kommunikationspartner zu beachten und zu trennen (z.B. Server/Client).

Im Gegensatz zu Interoperabilitätstests nehmen die Test bei Funktions-/Konformitätstests eine gleichwertige Position in der Testarchitektur ein und beschränken sich nicht auf die Steuerung von Referenzimplementierungen.

2.1.4. Anwendungsszenarien

IoT Anwendungen umfassen häufig eine hohe Anzahl von Geräten, heterogene Infrastrukturen sowie verschiedenartige Endgeräte der Benutzer ein. In der Regel reicht es daher nicht aus einzelne Schnittstellen gezielt zu untersuchen, sondern es wird erforderlich das korrekte Zusammenspiel an den beteiligten verteilten Beobachtungspunkten zu passiv oder aktiv zu kontrollieren und reale Anwendungsfälle nachzuvollziehen (vgl. Kapitel zu Prüfarchitekturen).

2.1.5. Weitere (z.B. crowd-testing)

Aufgrund der zeitlichen Zwänge bei der Entwicklung von IoT-Lösungen ist eine systematische und hinreichend vollständige Testentwicklung häufig schwer möglich. Aus diesem Grund sind neue pragmatische Ansätze neben den klassischen Testmethoden hinzugekommen. Zu den bekannten Vertretern gehört das sog. Crowd-testing bei dem eine virtuelle Gruppe von Testern in die Rolle der Anwender eintritt und mit ihren unterschiedlichsten Endgeräten und Konfigurationen, u.a. auch über mehrere Länder verteilt Anwendungsszenarien ausführt. Auf diese Weise sollen in kürzester Zeit Fehlerquellen (bugs) erkannt und zusammengetragen werden.

2.2. Interoperabilität

2.2.1. Plugfest

In der Industrie und Standardisierung werden Plugfest Events häufig dann organisiert, wenn die Kompatibilität zwischen Produkten einer größeren Gruppe verschiedener Hersteller zur Ermittlung ihrer Interoperabilität innerhalb kürzester Zeit an einem Ort getestet werden soll. Nach einem vorab festgelegten Zeitplan führen jeweils zwei der teilnehmenden Hersteller gemeinsame Test durch die im Vorfeld von einem unabhängigen Gremium festgelegt und beschrieben wurden. Derartige Veranstaltungen gab z.B. zur Einführung von IPv6 oder zu den IoT Protokollen CoAP oder dem oneM2M Service layer. Die Ergebnisse der Plugfests sind vertraulich und dienen den Herstellern in erster Linie

dazu direkt und schnell mögliche Implementierungs- oder Interpretationsfehler bei der Realisierung ihrer Produkte zu finden.

2.2.2. Test Suites

Interoperabilitätstests während als auch unabhängig von organisierten Plugfests folgen in der Regel klar festgelegten Testszenarien. Diese Vorgaben enthalten ähnlich den funktionalen Konformitätstests Angaben zum Testaufbau und den Testszenarien unter Einbeziehung relevanter Testdaten. Die Möglichkeiten der Interoperabilitätstests sind allerdings dadurch eingeschränkt, dass die beteiligen Produktimplementierungen nur die regulären Kommunikationsabläufe realisieren und nicht die Bandbreite der Fehlerquellen abdeckt wie es z.B. standardisierte Protokolltests erlauben.

2.2.3. Referenz Implementierungen

Bei den Herstellern werden Interoperabilitätstests zur Qualitätskontrolle auch außerhalb von Plugfests eingesetzt. Im technische Ansatz erfolgt hier die Prüfung einer kompatiblen Zusammenarbeit des zu testenden Systems des Herstellers mit einer als korrekt angenommenen Referenzimplementierung. Die Testabläufe können entsprechend oder unabhängig ggf. vorhandener Interoperabilitäts-Testbeschreibungen folgen. Wie bei den Plugfests bleiben auch hier die Einschränkungen gegenüber den Konformitätstests bestehen, hinzu kommt das Risiko, dass die genutzte Referenz Implementierung nicht hinreichend auf ihre eigene Konformität geprüft wurde.

2.3. Sicherheit

2.3.1. Schwachstellen Datenbanken und Scannen

Für die Öffentlichkeit relevante erkannte Schwachstellen von Betriebssystemen und verbreiteten Anwenderprogrammsystemen werden systematisch erfasst und in frei zugänglichen Datenbanken bereitgestellt. Anhand detaillierter Informationen zu den betroffenen Versionen und verfügbarer Patches können interessierte Personen und Organisationen ihre Systeme aktualisieren und Schwachstellen beheben. Das Durchsuchen („Scannen“) bekannter Sicherheitslücken gehört daher zu den weit verbreiteten Methoden bei Sicherheitsprüfungen (u.a. bei der Schwachstellenanalyse einer Common Criteria Prüfung).

2.3.2. Zero-Day/Fuzzing

Unbekannte oder unveröffentlichte Schwachstellen stellen eine unmittelbare Gefahr für die betroffenen Systeme und ihre Benutzer dar. Mögliche Schwachstellen müssen im Rahmen einer Prüfung aktiv aufgespürt und analysiert werden. Zu den anerkannten und weit verbreiteten Prüfmethoden zählen solcher Zero-Day-Schwachstellen gehören die sog. Fuzzing-Tests. Hierbei werden an offenen Schnittstellen ungültige bzw. insbesondere von gültigen Daten ableitete Eingaben vorgenommen. Das Spektrum der Eingaben kann in einfachen Fällen durch einen Zufallsgenerator erzeugt werden oder bei gezielten Fuzz-Tests unter Einbeziehung der jeweiligen Anwendungsszenarien und bekannten ggf. Zustandsübergänge der Zielsysteme punktuell verändert werden.

2.3.3. Analyse des Verkehrs

An standardisierte Netzwerkschnittstellen können in verteilten Systemen umfangreiche Datenströme aufgezeichnet und in bestimmten Fällen analysiert und mit einander in Bezug gesetzt werden. Die

Auswertung von erwarteten (bzw. statistisch ermittelten) Daten gegenüber den aktuellen Beobachtungen können der Erkenntnisse über Gefahrensituationen und zu deren Aufdeckung führen. Beispielsweise können die standardisieren Sicherheitsindikatoren der ETSI ISG ISI verwendet werden [http://www.etsi.org/technologies-clusters/technologies/information-security-indicators].

2.3.4. Analyse des Source Codes

Bei Vorliegen des Quellcodes von Softwaresystemen oder Anwendungen besteht die Möglichkeit einer automatischen Programmanalyse (white-box testing). Diese Prüfmethode ermittelt Schwachstellen z.B. infolge von typischen Programmierfehlern als auch die Aufdeckung von bekannten Sicherheitsmängeln aus zugänglichen Datenbanken.

2.3.5. Social Engineering

Bekannte Angriffsszenarien schließen die Ausnutzung von menschlichen Schwächen mit ein (z.B. Phishing) und ihr Auftreten bzw. die Versuche zur Erlangung unerlaubter Zugangsinformationen etc. sind daher mit in den Katalog von Prüfmethoden einzubeziehen.

2.4. Performanz / Robustheit

Die Untersuchung der Leistungsfähigkeit eines Systems unter steigender Last (Skalierung) bzw. die Robustheit eines Systems unter Höchstbelastung zählt neben Funktionalität und Sicherheit zu den wichtigen Prüfaspekten. Die Prüfmethoden benötigen hierbei neben den Kommunikationsabläufen und Daten weitere Parameterwerte zum Umfang der Belastung (z.B. Anzahl von Benutzeranfragen pro Zeiteinheit). Funktionale Prüfmethoden und -werkzeuge stoßen dabei häufig an ihre Grenzen (limitierte Speicherkapazität etc.) und müssen optimiert werden.

3. Prüflösungen und Werkzeuge (Open Source)

3.1. Funktionalität / Konformität

Name	Wireshark	Hersteller	Wireshark
Beschreibung	Wireshark ist ein Open-Source-Netzwerkanalyse- und Sniffer-Tool (d.h. Paketanalysator) für Open-Source-Netzwerkanalyse. Es hilft Analysten zu Überwachen und Leistung zu verbessern. Computer-Netzwerk-Traffic-Analyse sollte nicht nur Leistung verbessern, sondern überwachen, analysieren und potenzielle Sicherheitsprobleme reparieren.		
Link	https://www.wireshark.org/		
Lizenz	Open source (GNU General Public Lizenz, version 2)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Desktopplattformen (Windows, Linux, Mac)		

Besondere Merkmale und Funktionalität	- Überwacht nahezu 1,000 verschiedene Protokolltypen - Liest und schreibt Netzwerkdaten von verschiedenen Anwendungen - Live Aufzeichnung und offline Analyse		
Aktuelle Version	2.2.6	Seit	April 2017
Verbreitung und Haupteinsatzgebiet	Network Protocol Analyzer		
Probleme / Nachteile	Unbekannt		

Name	SikuliX	Hersteller	RaiMan
Beschreibung	SikuliX automatisiert alles, was Sie auf dem Bildschirm Ihres Desktop-Computers mit Windows, Mac oder einem Linux / Unix sehen. Es verwendet Bilderkennung von OpenCV, um GUI-Komponenten zu identifizieren und zu steuern. Dies ist praktisch in Fällen, in denen es keinen einfachen Zugriff auf GUI-Elemente, den Quellcode der Anwendung oder Web-Seite gibt.		
Link	http://sikulix.com/		
Lizenz	Open Source (MIT Lizenz)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Desktopplattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. You want to test applications or web pages that are under development. 2. You want to create usage documentation or training material that run live on the addressed application or web page.		
Aktuelle Version	1.1.1	Seit	5.4.2017
Verbreitung und Haupteinsatzgebiet	Functional Testing		
Probleme / Nachteile	- Änderungen im Layout oder Style der Anwendung haben negative Auswirkungen auf die Testfälle - ClearType sollte abgeschaltet sein (Pixelgenauigkeit wichtig)		

Name	CasperJS	Hersteller	
Beschreibung	CasperJS ist ein in Javascript geschriebenes Navigations-Scripting & Test-Dienstprogramm für Headless-Browser wie PhantomJS (WebKit) und SlimerJS (Gecko).		
Link	http://casperjs.org/		
Lizenz	Open Source (MIT Lizenz)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Desktopplattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	Konstrukte zur Vereinfachung der Erstellung von Browsertests		
Aktuelle Version	1.1.0	Seit	2012
Verbreitung und Haupteinsatzgebiet	Functional Testing		
Probleme / Nachteile	PhantomJS als Basis wird nicht weiterentwickelt		

Name	Selenium 3.0	Hersteller	Testing Freak
Beschreibung	Selenium ist ein Open-Source-Tool, mit dem Sie funktionale Tests für Webanwendungen und Desktop-Anwendungen durchführen können.		
Link	http://testingfreak.com/selenium/		
Lizenz	Open Source (Apache 2.0)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Browser, Desktop und Mobileplattformen (Windows, Linux, Mac, Android, iOS, Chrome, Firefox, Safari, IE, Edge)		
Besondere Merkmale und Funktionalität	1. Akzeptiert mehrere Sprachen für Test-Scripts wie z.B.: C #, Java, Perl, PHP, Python, Ruby und Groovy		

Aktuelle Version	3.0	Seit	13.10.2016
Verbreitung und Haupteinsatzgebiet	Functional Testing		
Probleme / Nachteile	Es werden WebDriver für die verschiedenen Browser benötigt		

Name	Firebase Test Lab for Android	Hersteller	Firebase
Beschreibung	Mit einer Operation können Sie das Testen Ihrer App über eine Vielzahl von Geräten und Gerätekonfigurationen initiieren. Testergebnisse - einschließlich Protokolle, Videos und Screenshots - werden in Ihrem Projekt in der Firebase-Konsole zur Verfügung gestellt. Auch wenn Sie keinen Testcode für Ihre App geschrieben haben, kann Test Lab Ihre App automatisch ausüben und sucht nach Abstürzen.		
Link	https://firebase.google.com/docs/test-lab/		
Lizenz	Open Source (Apache 2.0)	Preis ~	Frei
Besondere Merkmale und Funktionalität	1. Test auf realen Geräten, Workflow-Integration (Test Lab ist mit Android Studio, der Firebase-Konsole und der gcloud-Befehlszeile integriert. Sie können auch Test Lab mit Continuous Integration (CI) -Systemen verwenden.) 2. Sie können Robo-Test verwenden, um Probleme mit Ihrer App zu finden, damit Sie Ihre App testen können, auch wenn Sie keine App-Tests geschrieben haben. Robo-Test analysiert die Struktur der App-Benutzeroberfläche und erforscht sie dann automatisch, um Benutzeraktivitäten zu simulieren.		
Aktuelle Version	Software as a Service	Seit	Juni 2016
Verbreitung und Haupteinsatzgebiet	Android App		
Probleme / Nachteile	Bindung an Google als einzigen Anbieter		

Name	SmartBear TestLeft	Hersteller	SmartBear
Beschreibung	TestLeft ist ein leistungsfähiges, aber schlankes Funktionstest-Tool für Entwickler-Tester, die in Agile Teams arbeiten. Es bettet sich in Standard-Entwicklungs-IDEs ein. Ein eingebauter Zugriff auf Objekt- und Methodenbibliothek ist auch bei TestLeft verfügbar.		
Link	https://smartbear.com/product/testleft/overview/		
Lizenz	Kommerziell (*Node-Locked* und *Floating User*.)	Preis ~	430-1000 Euro
Unterstützte Plattformen	Jegliche 64bit Betriebssysteme		
Besondere Merkmale und Funktionalität	1. Unterstützung von Snapshots 2. Integration mit TestComplete 3. Vorinstallierte Entwicklerwerkzeuge 4. Wiederverwendbare Testmodule		
Aktuelle Version	TestLeft 2.3	Seit	2009
Verbreitung und Haupteinsatzgebiet	Functional Testing		
Probleme / Nachteile	1. Benötigt (Microsoft Visual Studio 2015 (jede Edition) und Microsoft .NET Framework 4.5 oder später) 2. Funktioniert nur mit 64Bit Betriebssystemen.		

Name	Eclipse Titan	Hersteller	Eclipse
Beschreibung	Titan ist eine TTCN-3 Kompilierungs- und Ausführungsumgebung mit einer Eclipse-basierten IDE. TTCN-3 ist eine modulare Sprache, die speziell für die Prüfung konzipiert wurde (das Akronym steht für Test und Test Conformance Notation), standardisiert durch ETSI (siehe www.ttcn-3.org) und von ITU unterstützt.		
Link	https://projects.eclipse.org/projects/tools.titan		

Lizenz	Eclipse Public Lizenz 1.0	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Desktopplattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. Gute Dokumentation 2. Viele Beispiele 3. Der Benutzer des Tools kann Testfälle entwickeln, Testausführungslogik ausführen und die ausführbare Test-Suite für mehrere Plattformen erstellen.		
Aktuelle Version	6.1.0	since	27.03.2015
Verbreitung und Haupteinsatzgebiet	Protocol Testing		
Probleme / Nachteile	Nicht immer einfach aufzusetzen (benötig Cygwin unter Windows)		

3.2. Interoperabilität

Name	Cooper (Add-on)	Hersteller	Matthias Kovatsch
Beschreibung	Der Copper (Cu) CoAP User-Agent für Firefox installiert einen Handler für das 'coap'-URI-Schema und ermöglicht es Benutzern, mit dem Internet der Dinge zu interagieren.		
Link	https://addons.mozilla.org/de/firefox/addon/copper-270430/		
Lizenz	Open Source (BSD 3-Clause)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Desktopplattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. Implementiert RFC 7252 und URI Handhabung für 'CoAP' scheme 2. Umsetzung von GET, POST, PUT, and DELETE 3. Unterstützung für Resource Discovery 4. Unterstützung für Blockwise Transfers		

	5. Überwachung von Resources		
Aktuelle Version	1.0.1	since	18.12.2016
Verbreitung und Haupteinsatzgebiet	Debugging, Experimentieren und Entwickeln		
Probleme / Nachteile	Benötigt Firefox		

Name	Coap Sandboxes	Hersteller	Eclipse
Beschreibung	Es sollte für jeden interessant sein, der eine CoAP-Client-Implementierung gegen einen anderen Endpunkt zu testen will und daran interessiert ist das Schlüsselkonzepte des CoAP-Protokolls zu verstehen. Auf dem Coap Sandboxes Server läuft Eclipse Californium.		
Link	https://projects.eclipse.org/projects/technology.californium		
Lizenz	Eclipse Public Lizenz 1.0	Preis ~	Frei
Unterstützte Plattformen	Bereitgestellt als Software as a Service (läuft auf jedem Java unterstützendem System)		
Besondere Merkmale und Funktionalität	1. Implementation von CoAP (RFC 7252) 2. Implementation von CoAP (RFC 7252) 3. Implementation von CoAP (RFC 7252) 4. Implementation von CoAP-HTTP cross-proxy Unterstützung mittels HttpCore-NIO und Guava		
Aktuelle Version	1.0.0	Seit	11.04.2015
Verbreitung und Haupteinsatzgebiet	Conformance Testing		
Probleme / Nachteile			

Name	Passive Validation Tool For Coap	Hersteller	IRISA
Beschreibung	Es führt eine passive Analyse in einer pcap-Datei durch und ordnet automatisch jede CoAP-Konversation den relevanten Testzielen zu, die in der Testspezifikation beschrieben sind. Mit anderen Worten, alles, was Sie tun müssen, ist, eine Capture-Datei im pcap-Format zur Verfügung zu stellen, und das Tool wird eine detaillierte Analyse seines Inhalts zur Verfügung stellen.		
Link	http://senslab2.irisa.fr/coap/ git clone git://scm.gforge.inria.fr/t3devkit/ttproto.git		
Lizenz	Open Source (CeCILL FREE SOFTWARE LICENSE AGREEMENT)	Preis ~	Frei
Unterstützte Plattformen	Alle als Software as a Service		
Besondere Merkmale und Funktionalität	Implementierung von ETSI COAP#4 Plugtest scenarios		
Aktuelle Version	CoAP.12.08.2015-8-g0d1bad2	Seit	März 2012
Verbreitung und Haupteinsatzgebiet	Interoperability Testing		
Probleme / Nachteile	Es ermöglicht die Durchführung von CoAP-Interoperabilitätstests (mit aufgelisteten verfügbaren Testszenarien) auf den bereitgestellten Traces von CoAP Client-Server-Interaktionen. Dokumentation		

Name	Graphical MQTT Client Tools (Paho)	Hersteller	Eclipse
Beschreibung	Das Eclipse-Paho-Projekt bietet eine Open-Source-Client-Implementierung vom MQTT- und MQTT-SN-Messaging-Protokoll, als Grundlage für neue, bestehende und aufkommende Anwendungen für das Internet der Dinge (IoT).		
Link	https://eclipse.org/paho/		

Lizenz	Open Source (EPL 1.9)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Desktopplattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. Für beschränkte Netzwerke 2. Verlässlich		
Aktuelle Version	1.2.0	Seit	März 2013
Verbreitung und Haupteinsatzgebiet	Conformance Testing und Entwicklung		
Probleme / Nachteile			

Name	Leshan	Hersteller	Eclipse
Beschreibung	Leshan bietet Bibliotheken, die Menschen helfen, ihren eigenen Lightweight M2M Server und Client zu entwickeln.		
Link	https://eclipse.org/leshan/		
Lizenz	Open Souce (EPL 1.0)	Preis ~	Frei
Unterstützte Plattformen	Alle (Java basiert)		
Besondere Merkmale und Funktionalität	1. Modulare Bibliotheken 2. Basiert auf der Californium CoAP Implementierung 3. Basiert auf der Scandium DTLS Implementierung 4. IPSO Objekte werden unterstützt		
Aktuelle Version	1.0.0-M1	Seit	2014
Verbreitung und Haupteinsatzgebiet	OMA Lightweight M2M Server und Client		

Probleme / Nachteile	

3.3. Sicherheit

Name	Vega	Hersteller	Subgraph
Beschreibung	Vega ist ein kostenloser und Open-Source-Web-Security-Scanner und Web Security Test-Plattform, um die Sicherheit von Web-Anwendungen zu testen. Es ist in Java geschrieben, GUI basiert und läuft auf Linux, OS X und Windows.		
Link	https://subgraph.com/vega/		
Lizenz	Open Source (MIT)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Desktopplattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. Vega kann Ihnen helfen, Schwachstellen wie: reflektierte Cross-Site-Scripting, gespeicherte Cross-Site-Scripting, blind SQL-Injektion, enthaltene Remote-Dateien und Shell-Injektion zu enddecken. 2. Vega prüft auch die TLS / SSL-Sicherheitseinstellungen und identifiziert Möglichkeiten zur Verbesserung der Sicherheit Ihrer TLS-Server.		
Aktuelle Version	1.0	Seit	-
Verbreitung und Haupteinsatzgebiet	Security Testing		
Probleme / Nachteile			

Name	Zed Attack Proxy	Hersteller	OWASP
Beschreibung	ZAP bietet automatisierte Scanner sowie eine Reihe von Tools, mit denen Sie Sicherheitslücken manuell finden können.		

Link	https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project		
Lizenz	Open Source (Apache 2.0)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Plattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. Intercepting Proxy 2. Automated Scanner 3. Passive Scanner 4. Forced Browsing 5. Fuzzer		
Aktuelle Version	2.6.0	Seit	2013
Verbreitung und Haupteinsatzgebiet	Security Testing		
Probleme / Nachteile			

Name	Wapiti	Hersteller	
Beschreibung	Wapiti erlaubt Ihnen, die Sicherheit Ihrer Web-Anwendungen zu auditieren. Es führt "Black-Box" -Scans durch, d. H. Es lernt nicht den Quellcode der Anwendung, sondern scannt die Webseiten des eingesetzten Webapps und sucht nach Scripts und Formularen, wo es Daten injizieren kann.		
Link	http://wapiti.sourceforge.net/		
Lizenz	Open Source (GPL 2.0)	Preis ~	Frei
Unterstützte Plattformen	Linux und Windows		
Besondere Merkmale und Funktionalität	1. Generiert Vulnerability Reports in verschiedenen Formaten (HTML, XML, JSON, TXT...) 2. Pausieren ist unterstützt		

Aktuelle Version	2.3.0	Seit	20.10.2013
Verbreitung und Haupteinsatzgebiet	Security Testing		
Probleme / Nachteile	Weitere Installationen benötigt		

Name	IronWASP	Hersteller	IronWasp
Beschreibung	IronWASP ist anpassbar konzipiert, so dass Benutzer ihre eigenen benutzerdefinierten Sicherheitsscanner verwenden können. Obwohl ein fortgeschrittener Benutzer mit Python / Ruby Scripting-Know-how in der Lage ist, die Plattform voll auszunutzen, sind viele Funktionen des Tools einfach genug, um von absoluten Anfängern benutzt zu werden		
Link	https://ironwasp.org/about.html		
Lizenz	Open Source (GPL 3)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Plattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. GUI basiert und einfach zu benutzen 2. Keine Vorkenntnisse benötigt 3. Unterstütz das Aufnehmen der Loginsequenz		
Aktuelle Version	0.9.8.6	Seit	9/02/2015
Verbreitung und Haupteinsatzgebiet	Security Testing		
Probleme / Nachteile	Benötigt .NET 2.0 SP2 Für den vollen Funktionsumfang werden Python/Ruby scripting Erfahrung benötigt		

Name	Nikto	Hersteller	Chris Sullo, David Lodge

Beschreibung	Web-Server-Assessment-Tool. Es wurde entwickelt, um verschiedene unsichere und Standarddateien, Konfigurationen und Programme auf jeder Art von Webserver zu finden.		
Link	https://cirt.net/nikto2/		
Lizenz	Open Source (GPL)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Plattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. Volle HTTP Proxy Unterstützung		
Aktuelle Version	2.1.5	Seit	2012
Verbreitung und Haupteinsatzgebiet	Security Testing, Vulnerability Detection		
Probleme / Nachteile	Einige POSIX Features stehen eventuell unter Windows nicht zur Verfügung.		

Name	OpenVas	Hersteller	Greenbone Networks
Beschreibung	Der OpenVAS Scanner ist ein umfassendes Vulnerability Assessment System, das Sicherheitsprobleme in allen Arten von Servern und Netzwerkgeräten erkennen kann. Eine gehostete Version der OpenVAS Software kann genutzt werden um eigene Internet-Infrastruktur zu testen.		
Link	http://openvas.org/		
Lizenz	Open source under GNU GPL	Preis ~	Frei
Unterstützte Plattformen	Linux oder als Image für VirtualBox, ESXi, Hyper-V		
Besondere Merkmale und Funktionalität	1. Benachrichtigung per Email 2. Erweiterbar		

Aktuelle Version	4.0.5	Seit	2008
Verbreitung und Haupteinsatzgebiet	Vulnerability Assessment		
Probleme / Nachteile	Erfahrung für Installation und Konfiguration nötig		

Name	BeEF	Hersteller	IT Security Solutions.org
Beschreibung	BeEF ist kurz für das Browser Exploitation Framework. Es ist ein Penetrationstest-Tool, das sich auf den Webbrowser konzentriert.		
Link	http://beefproject.com/		
Lizenz	Open source (GNU 2.0)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Desktopplattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. BeEF ermöglicht es dem professionellen Penetrationstester, die tatsächliche Sicherheitsposition einer Zielumgebung zu beurteilen, indem sie clientseitige Angriffsvektoren verwendet. 2. Im Gegensatz zu anderen Sicherheits-Frameworks sieht BeEF an dem gehärteten Netzwerk-Perimeter- und Client-System vorbei und untersucht die Verwertbarkeit im Rahmen der offenen Tür: den Webbrowser		
Aktuelle Version	4.6.1	Seit	2011
Verbreitung und Haupteinsatzgebiet	Penetration Testing (Client Side / Browser)		
Probleme / Nachteile			

Name	Paros	Hersteller	Paros Proxy
Beschreibung	Ein Java-basierter Web-Proxy zur Beurteilung der Anfälligkeit der Webanwendung. Es unterstützt das Bearbeiten / Betrachten von HTTP / HTTPS-Nachrichten on-the-fly, um Elemente wie Cookies und Formularfelder zu ändern		
Link	https://sourceforge.net/projects/paros/		
Lizenz	Artistic Lizenz	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Desktopplattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. Client-Zertifikat, Proxy-Verkettung, intelligentes Scannen für XSS- und SQL-Injektionen		
Aktuelle Version	3.2.13	Seit	14.8.2013
Verbreitung und Haupteinsatzgebiet	Vulnerability Assessment		
Probleme / Nachteile	Keine Updates Fork: OWASP Zed Attack ProxyProject		

Name	Skipfish	Hersteller	-
Beschreibung	Actives Web Application Sicherheitsaufklärungswerkzeug.		
Link	https://code.google.com/archive/p/skipfish/		
Lizenz	Open Source (Apache 2.0)	Preis ~	Frei
Unterstützte Plattformen	All		
Besondere Merkmale und Funktionalität	1. Skipfish ist schnell, einfach zu bedienen und basiert auf modernster Sicherheitslogik. 2. hohe Qualität, niedrige falsch positive, differentielle Sicherheitskontrollen, die in der Lage sind, eine Reihe von subtilen Mängeln, einschließlich blinden Injektionsvektoren aufzuspüren		

Aktuelle Version	2.10b	Seit	2012
Verbreitung und Haupteinsatzgebiet	Vulnerability Scanner / Security Assesment		
Probleme / Nachteile	Nicht mehr gepflegt		

Name	BFB Tester	Hersteller	Mike Heffner
Beschreibung	BFBTester ist gut für schnelle, proaktive Sicherheitskontrollen von Binärprogrammen.		
Link	http://bfbtester.sourceforge.net/		
Lizenz	Open Source (GNU 2.0)	Preis ~	Frei
Unterstützte Plattformen	Linux		
Besondere Merkmale und Funktionalität	BFBTester führt Prüfungen von Einzel- und Mehrfachbefehlszeilenargumentüberläufen und Umgebungsvariablenüberläufen durch.		
Aktuelle Version	2.0.1	Seit	15.04.2001
Verbreitung und Haupteinsatzgebiet	Non networked Vulnerability Scanner		
Probleme / Nachteile	Nicht mehr gepflegt		

Name	Brakeman	Hersteller	Brakeman
Beschreibung	Brakeman is a security scanner for Ruby on Rails applications.		
Link	http://brakemanscanner.org/docs/introduction/		
Lizenz	Open Source (MIT)	Preis ~	Frei

Unterstützte Plattformen	Linux		
Besondere Merkmale und Funktionalität	1. Keine Konfiguration notwendig 2. Flexible Prüfung 3. Im Gegensatz zu vielen Websicherheitsscannern betrachtet Brakeman den Quellcode Ihrer Applikation. Die macht ein Aufsetzen der Anwendung überflüssig.		
Aktuelle Version	3.6.1	Seit	11.02.2014
Verbreitung und Haupteinsatzgebiet			
Probleme / Nachteile	1. Ist nicht allwissend und nur die Entwickler einer Anwendung können verstehen, ob bestimmte Werte gefährlich sind oder nicht. Standardmäßig ist Brakeman äußerst argwöhnisch. Das kann zu vielen "falschen Positiven" führen. 2. Nicht alle Funktionen im Werkzeug. Brauchen Sie die Pro-Version zu kaufen 3. Nur für Ruby on Rails		

Name	Gendarme	Hersteller	Mono-Project
Beschreibung	Gendarme ist ein erweiterbares, regelbasiertes Tool, um Probleme in .NET-Anwendungen und Bibliotheken zu finden		
Link	http://www.mono-project.com/docs/tools+libraries/tools/gendarme/		
Lizenz	Open Source (MIT/X11)	Preis ~	Frei
Unterstützte Plattformen	Linux und Windows		
Besondere Merkmale und Funktionalität	1. Gendarme prüft Programme und Bibliotheken, die Code im ECMA-CIL-Format enthalten (Mono und .NET) und sucht nach häufigen Problemen		

	mit dem Code, Probleme, die der Compiler normalerweise nicht überprüft oder historisch nicht überprüft hat.		
Aktuelle Version	4.8.0	Seit	2007
Verbreitung und Haupteinsatzgebiet	Statical Code Analysis (Security)		
Probleme / Nachteile	Nur für .Net		

Name	FlawFinder	Hersteller	David A. Wheeler
Beschreibung	Ein einfaches Programm, das den C / C ++ - Quellcode untersucht und mögliche Sicherheitsschwächen ("Fehler") nach Risikostufe sortiert.		
Link	https://www.dwheeler.com/flawfinder/		
Lizenz	Open Source (GPL 2.0)	Preis ~	Frei
Unterstützte Plattformen	Linux und Windows		
Besondere Merkmale und Funktionalität	1. It's very useful for quickly finding and removing at least some potential security problems *before* a program is widely released to the public		
Aktuelle Version	1.31	Seit	21.05.2001
Verbreitung und Haupteinsatzgebiet	Statical Code Analysis (Security)		
Probleme / Nachteile	Nur für C / C++		

Name	WebScarab-NG	Hersteller	Owasp
Beschreibung	Ein Framework für die Analyse von Anwendungen, die mit den HTTP- und HTTPS-Protokollen kommunizieren.		

Link	https://www.owasp.org/index.php/Category:OWASP_WebScarab_Project		
Lizenz	Open Source (GNU 2.0)	Preis ~	Frei
Unterstützte Plattformen	Linux und Windows		
Besondere Merkmale und Funktionalität	1. WebScarab hat mehrere Betriebsarten, die durch eine Reihe von Plugins implementiert werden. WebScarab ist in der Lage, sowohl HTTP- als auch HTTPS-Kommunikation abzufangen. 2. Kann als WebScarab Abhörproxy betrieben werden.		
Aktuelle Version	5.27	Seit	30.10.2002
Verbreitung und Haupteinsatzgebiet	Monitoring, Fuzzing, Development		
Probleme / Nachteile	Nicht mehr in Entwicklung		

Name	W3af	Hersteller	Andres Riancho
Beschreibung	W3af ist ein Web Application Attack und Audit Framework.		
Link	http://w3af.org/download		
Lizenz	Open Source (GPL 2.0)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Desktopplattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. schnelle HTTP-Anfragen 2. Integration von Web- und Proxy-Servern in den Code 3. Injektion von Nutzlasten in verschiedene Arten von HTTP-Anfragen etc.		
Aktuelle Version	1.0	since	1.03.2015
Verbreitung und Haupteinsatzgebiet	Penetration Testing		

Probleme / Nachteile	Schwache Windows Unterstützung

Name	Sqlmap	Hersteller	Bernardo Damele A. G., Miroslav Stampar
Beschreibung	Dieses Tool wird hauptsächlich für die Erkennung und Nutzung von SQL-Injection-Problemen in einer Anwendung als auch Hacking über Datenbank-Servern verwendet. Es kommt mit einer Kommandozeilenschnittstelle.		
Link	http://sqlmap.org		
Lizenz	Open Source (GPL 2.0)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Desktopplattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. Vollständige Unterstützung für sechs SQL-Injektionstechniken: 1. Full support for six SQL injection techniques: Boolean-based blind, time-based blind, error-based, UNION query-based, stacked queries und out-of-band.		
Aktuelle Version	1.1.4-28	Seit	2009
Verbreitung und Haupteinsatzgebiet	Penetration Testing		
Probleme / Nachteile			

Name	Kali Linux	Hersteller	Offensive Security
Beschreibung	Kali Linux is a Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing. Kali contains several hundred tools which are geared towards various information security tasks, such as Penetration Testing, Security research, Computer Forensics and Reverse Engineering		
Link	http://docs.kali.org/category/introduction		

Lizenz	Open Source (GPL)	Preis ~	Frei
Unterstützte Plattformen	Als Image (basierend auf Linux)		
Besondere Merkmale und Funktionalität	1. Mehr als 600 Penetrationstest-Tools enthalten, Weitreichende Wireless-Geräte-Unterstützung		
Aktuelle Version	2017.1	Since	13.03.2013
Verbreitung und Haupteinsatzgebiet	Penetration Testing, Security research, Computer Forensics und Reverse Engineering		
Probleme / Nachteile			

Name	OSSEC	Hersteller	OSSEC
Beschreibung	OSSEC ist eine Plattform zur Überwachung und Steuerung Ihrer Systemaktivität mit Dateiintegritätsüberwachung, Protokollüberwachung, Rootcheck und Prozessüberwachung		
Link	http://ossec.github.io/		
Lizenz	Open Source (GPL 2.0)	Preis ~	Frei
Unterstützte Plattformen	Für alle großen Desktopplattformen (Windows, Linux, Mac)		
Besondere Merkmale und Funktionalität	1. Wenn Angriffe passieren, können Sie mit OSSEC über Alert-Logs und E-Mail-Benachrichtigungen informiert werden. 2. OSSEC exportiert auch Alerts zu jedem SIEM-System über syslog, so dass Sie Echtzeit-Analytik und Einblicke in Ihre System-Security-Events erhalten können.		
Aktuelle Version	2.8.3	since	5.11.2015
Verbreitung und Haupteinsatzgebiet	Intrusion Detection Testing		

Probleme / Nachteile	

3.4. Performanz

Name	Apache JMeter	Hersteller	Apache
Beschreibung	Apache JMeter eine reine Java-Anwendung um Test-Funktionsverhalten zu laden und Leistung zu messen. Sie wurde ursprünglich für das Testen von Web-Applikationen entworfen, ist aber seither auf andere Testfunktionen erweitert		
Link	http://jmeter.apache.org/		
Lizenz	Open Source (Apache 2.0)	Preis ~	Frei
Unterstützte Plattformen	Alle Plattformen die Java unterstützen		
Besondere Merkmale und Funktionalität	1. Fähigkeit des Ladens und Ausführens von Test auf verschiedenen Ebenen Tests Applikation/Server/Protokoll 2. Komplette Test IDE welche das Planen, Ausführen und Aufnehmen von Test erlaubt 3. Einfache Korrelation durch die Fähigkeit, Daten aus den meisten populären Antwortformaten, HTML, JSON, XML oder irgendeinem Textformat zu extrahieren		
Aktuelle Version	3.2	Seit	15.12.1998
Verbreitung und Haupteinsatzgebiet	Performance Measurment and Loadtesting		
Probleme / Nachteile	benötigt Java		

4. Zusammenfassung und Ausblick

Die Sicherung des IoT benötigt nahezu das volle Spektrum der bekannten Tests für netzwerkbasierte Systeme mit besonderen Randbedingungen. Diese Tests und Prüfungen fallen in die Bereiche Performanz, Sicherheit, Funktionalität, Skalierbarkeit, Robustheit, Konformität und Interoperabilität.

Hierfür haben sich verschiedene Werkzeuge und Methoden etabliert. Um diese mit den neuen Protokollen und Anforderungen des IoT verwenden zu können, werden Erweiterungen oder Anpassungen nötig sein. Der Anfang wird in diesem Projekt bei neuen, umfänglichen Testsuiten für CoAP und MQTT gemacht. Anpassungen für Sicherheitsprüfwerkzeuge werden folgen, um die Zertifizierte Prüfung von IoT Produkten zu ermöglichen. Zukünftige Versionen dieses Dokumentes werden die Insbesondere auf prüfrelevante Werkzeuge im Detail eingehen.

5. Referenzen

[Rennoch16] Rennoch, A., Wagner, M.: Challenges and ideas for IoT testing. In: GL & IS (ISSN 1265-1397), No. 119, Paris (F), December 2016.

[ETSI12] ETSI CTI Plugtests Guide First Draft V0.0.16 (2012-03), IoT CoAP Plugtests; Paris, France; 24 - 25 March 2012

[Bucsics15] Bucsics, T.; Baumgartner, M.; Seidl, R.; Gwihs, S.: Basiswissen Testautomatisierung. Dpunkt Verlag, 2. Aktualisierte und überarbeitete Auflage, Heidelberg, 2015.