Du kommst hier net rein! (Oder doch?)

Messung der Code-Abdeckung im Rahmen von Penetrationstests

Dr. Sabine Poehler, Verifysoft Technology GmbH

Im Bereich der sicherheitskritischen Softwareentwicklung (Safety) ist die Messung der Codeabdeckung im Rahmen des Testens schon lange ein Standardinstrument. Sie wird in den gängigen Sicherheitsnormen gefordert. Ein neueres Einsatzgebiet für die Abdeckungsanalyse ist der Einsatz während der Durchführung von Penetrationstests.

Im Rahmen einer Bachelorarbeit untersuchen wir, wie sich die Auswertung von Penetrationstests durch die Messung der Code Coverage vereinfachen lässt. Gleichzeitig wird mit dieser parallelen Analyse die Qualität des Penetrationstests selbst kontrolliert.

Code Coverage

Um nachzuweisen, dass Software ausreichend getestet wurde, werden verschiedene Code Coverage Maße verwendet, z.B.:

- Function Coverage: jede Funktion wurde aufgerufen,
- Statement Coverage: jede Anweisung wurde ausgeführt,
- Decision Coverage: Jede Entscheidung, z.B. in einem if-Statement, wurde als wahr und als falsch ausgewertet.

Im Bereich der sicherheitskritischen Softwareentwicklung gibt es weitere relevante Coverage Maße, als höchste Stufe die Modified Condition/ Decision Coverage (MC/DC), die allerdings nicht im Fokus unserer Untersuchung liegen.

TER % - decision	TER % - statement	File
Directory: C:\Projects\hcontrol		
79 % - (22/28)	83 % - (20/24)	regulators.c
100 % (8/8)	100 % (4/4)	sensors.c
71 % - (5/7)	93 % - (14/15)	service_functions.c
75 % - (12/16)	75 % - (15/20)	zhome.c
80 % - (47/59)	**84 %** - (53/63)	**DIRECTORY OVERALL**
80 % - (47/59)	**84 %** - (53/63)	**OVERALL**

Abbildung 1: Testwell CTC++ Coverage Report - Decision und Statement Coverage pro Datei

Penetrationstests

Software, die „von außen“ auf irgendeinem Weg erreichbar ist, ist der Gefahr durch kriminelle Angreifer ausgesetzt.

Softwarefirmen, die geschäftskritische Webanwendungen entwickeln, führen daher klassischerweise im Rahmen ihrer Qualitätssicherung Penetrationstest durch – oder

lassen sie von Dritten durchführen. Gegenüber den eigenen Kunden oder Auditoren kann dadurch der Nachweis erbracht werden, dass die Software definierte Angriffe abblockt.

Werden Sicherheitslücken entdeckt, ist die nachfolgende Analyse für die Entwicklungsabteilung nicht in allen Fällen einfach. Bei einem selbst durchgeführten Penetrationstest, z.B. mit geeigneten Tools, besteht noch eine gewisse Kontrolle über seinen Ablauf – wobei auch hier ein hohes Detailverständnis für die Funktionsweise von Penetrationstests allgemein und vom konkret eingesetzten Tool erforderlich ist.

Ein anderer Aspekt kommt dazu, wenn Dritte den Penetrationstest durchführen: Hier besteht zunächst wenig Kontrolle darüber, was genau wann getestet wurde. Wird das testende Unternehmen darüber hinaus nicht selbst beauftragt, sondern von einem Kunden, dann ist die direkte Zusammenarbeit nicht zwangsweise konstruktiv und vertrauensvoll. In solchen Fällen entsteht durch die Analyse des übermittelten Berichts ein hoher Arbeitsaufwand in der Entwicklungsabteilung. Wurden tatsächlich oder vermeintlich kritische Schwachstellen gefunden wurden, dann ist der zeitliche und inhaltliche Druck groß.

Für dieses Security-Thema sind also technische, organisatorische und menschliche Herausforderungen eng verwoben. Da heute auch eingebettete Software zahlreichen Angriffsmöglichkeiten von außen ausgesetzt ist, wird das Instrument Penetrationstest absehbar eine ähnliche Bedeutung in der Entwicklung und Qualitätskontrolle solcher Software spielen, wie es z.B. für Webanwendungen schon sehr lange der Fall ist.

Messung der Codeabdeckung während eines Penetrationstest

Wie kann man dem Entwickler oder der Entwicklungsabteilung, die mit der Analyse eines Penetrationstests betraut wird, helfen?

Unsere Idee ist, von vorneherein eine Variante des Software zu testen (oder testen zu lassen) die für die Messung der Codeabdeckung instrumentiert wurde.

Neben des Berichts zum Penetrationstest liegt dann automatisch auch ein Bericht über die aufgerufenen Teile der getesteten Software vor. Das kann auf verschiedene Art hilfreich sein:

Im Voraus können Bereiche der Software definiert werden, die während des Penetrationstests auf keinen Fall aufgerufen werden sollten. Wird z.B. für eine Anwendung mit notwendiger Benutzeranmeldung ein Penetrationstest mit nichtangemeldetem Benutzer durchgeführt, dann sind das im Wesentlichen alle Komponenten der Software, die nichts mit der Prüfung von Benutzername und Passwort zu tun haben. Für solche Bereiche des Software dreht sich das klassische Coverage-Ziel um: Statt 100% Coverage ist hier 0% Coverage ideal.

Wurde andererseits eine Schwachstelle entdeckt, dann hilft der Coverage-Bericht, den Weg des Angreifers durch die Software nachzuvollziehen.

Und nicht zuletzt wird auch die Qualität des Penetrationstests selbst analysiert: Durch die Abdeckungsmessung wird transparent, für welche Teile der Software überhaupt Angriffe versucht wurden. Das hilft in jedem der beschriebenen

Durchführungsszenarien: Beim eigenen Einsatz eines Penetrationstools kann dessen Einsatzbereich erweitert werden – ein beauftragter Dienstleister wird auf die vollständige Durchführung seines Auftrags kontrolliert.

Exemplarisches Projekt

Dieser Ansatz wird aktuell (Stand: Oktober 2018) im Rahmen einer Bachelorarbeit praktisch in folgendem Setup untersucht:

1. Das Heimautomatisierungssystem Domoticz
   Zahlreiche Geräte und Sensoren können mit diesem Open Source System überwacht und gesteuert werden.
   Technisch ist Domoticz im Kern in C++ implementiert und besitzt als Benutzerschnittstelle ein Web-Frontend.
   Domoticz dient als Testobjekt für die Penetrationstests und läuft für diesen Zweck auf einem Raspberry Pi.

2. Arachni
   „Web Application Security Scanner Framework“, wird zur Durchführung der Penetrationstests über das Web-Frontend von Domoticz verwendet.

3. Weitere Penetrationstests
   Geplant ist sowohl der Einsatz weiterer Tools als auch selbstentwickelter Penetrationstests.

4. Testwell CTC++
   Die Ermittlung der Code Coverage erfolgt mit dem Code Coverage Analyzer Testwell CTC++.

In diesem Setup werden die Grundideen der kombinierten Durchführung von Penetrationstests und Coverage-Messungen nachvollzogen und auf ihre Anwendbarkeit untersucht.

Autorin

Sabine Poehler arbeitet bei der Verifysoft Technology GmbH als Produktmanagerin für die Produktlinie Testwell, darunter insbesondere für den Code Coverage Analyzer Testwell CTC++. Sie ist für die strategische Weiterentwicklung der Testwell-Tools verantwortlich und leitet die Support- und Entwicklungsabteilung.

Kontakt: www.verifysoft.com/de_contact.html