Evaluation of Automated Testing Coverage: a Case Study
Testing of Wireless Secure Connection Software

S. Vilkomir\textsuperscript{1}, P. Tips\textsuperscript{2}, D. Parnas\textsuperscript{1},
J. Monahan\textsuperscript{2}, T. O'Connor\textsuperscript{2}

\textsuperscript{1}Software Quality Research Laboratory (SQRL),
University of Limerick, Ireland
\textsuperscript{2}Dell Products, Limerick, Ireland

sergiy.vilkomir at ul.ie
http://www.csis.ul.ie/staff/SergiyVilkomir/

ISSRE 2005, November 8–11, 2005, Chicago, USA
Case Study – Secure Wireless Connection

Wireless Security Test Setup

Wireless Access Point

Wired Connection (for testing purposes only)

Wired Link

Computer with WLAN Card

& Wireless LAN Software

Wireless LAN Card Interface
Project features and aims

- With every new wireless product extensive testing is required & performed
- Wireless security testing forms an especially important part
- New wireless securities become available
  - Initially WEP & WPA
  - Now WPA2
- Result: more to test & testing will take more effort
- Security testing is very repetitive

- Support Dell’s constant efforts to assure quality
- Increase and expand test coverage
- Reduce labour intensive effort
- Improve test time
New approach to Wireless Connection Testing

BEFORE

• Regression testing:
  • Manual
  • Fixed set of predefined test cases
    Example:
    • 0102030405060708
    • 11121314151617181920
    • 212223242526272829303
    • asdhjkl;'ASDFG:"zxcv
    • /ZXCVBNM<>?

AFTER

• Regression testing:
  • Automated
  • Modified fixed set of predefined test cases
  +
  • Statistical testing
  • Based on satisfaction of coverage requirements
Tool for Automated Testing

• A 3rd party commercial scripting tool was used.

• The criteria for selection was as follows:
  ➢ Scalable and Extensible solution – allowed the use of external routines.
  ➢ Ease of Use – leading to reduced development cycle.
  ➢ Non proprietary programming language - VBScript.
Approach to Automated Testing

• Test is divided into number of separate hierarchal scripts

• Requires only one script change when using a different WLAN Client Manager or Wireless Access Point
  – One script per WLAN Client Manager
  – One script per Access Point

• Devices may have different defaults and behaviours
  – These device specifics are declared in the Declarations script
  – Allows simple selection of the device by editing the declarations script, without other script changes

• Generated function library
Approach to Automated Testing

High Level Script File Hierarchy

- Main Script
- Top Menu
- Declarations Script
- Function Library Script
- Security Script
- Data Table
- Access Point Script
- WLAN Client Manager Script
- Connectivity Check Script
- Error Recovery Script
Notes: Code example that removes any configured Wireless Access Points in the Wireless Client Manager

CMName is a variable

SystemUtil.Run CMAPP,"","",""
Dialog(CMName).Activate
NoAPs = Dialog(CMName).WinListView("Automatically connect").GetItemsCount
Do While NoAPs > 0
    Dialog(CMName).WinListView("Automatically connect").Select 0
    Dialog(CMName).WinButton("Remove").Click
    NoAPs = NoAPs - 1
Loop
Dialog(CMName).WinButton("OK").Click
Testing coverage - general approach

1. Predefined test cases
2. Attributes of test cases
3. Tabular requirements
4. Informal requirements
5. Regression testing

Statistical testing
Informal testing requirements (fragment)

1 WEP key should be used
   1.1 Valid WEP keys should include
       1.1.1 40 bit ASCII keys with the length of 5 characters
       1.1.2 40 bit HEX keys with the length of 10 digits
       1.1.3 104 bit ASCII keys with the length of 13 characters
       1.1.4 104 bit HEX keys with the length of 26 digits
   1.2 Invalid WEP keys should include
       1.2.1 40 bit ASCII keys with a key length of 5 < and > 5 characters
       1.2.2 40 bit HEX keys with a key length of 10 < and > 10 digits
       1.2.3 40 bit HEX keys that contain non hex characters
       1.2.4 104 bit ASCII keys with a key length of 13 < and > 13 char.
       1.2.5 104 bit Hex keys with a key length of 26 < and > 26 digits

2 WPA key should be used
   ...
Attributes of Test Cases

- Type of security algorithm (WEP - WPA)
- Symbols of security keys (ASCII - HEX)
- Length of security keys (40 bit - 104 bit)
- Validity (valid - invalid)
- Sequence of symbols (sequential - random)
- Case of symbols (lower case - upper case - mixed)
- Type of symbols (numeric - characters - mixed)
- Invalidity (too short - too long - invalid symbols)
# Tabular requirements

<table>
<thead>
<tr>
<th></th>
<th>Valid keys</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>numbers</td>
<td>characters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>seq</td>
<td>rand</td>
<td>seq</td>
<td>rand</td>
<td>seq</td>
<td>rand</td>
<td>seq</td>
</tr>
<tr>
<td>WEP</td>
<td>ASCII</td>
<td>40</td>
<td>seq</td>
<td>rand</td>
<td>seq</td>
<td>rand</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HEX</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WPA</td>
<td>ASCII</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HEX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- The attributes are divided into two orthogonal classes (row headers and column headers)
- Cells represent the different groups of tests
- To require a number of test cases – put this number into the corresponding cell
# Tabular requirements

<table>
<thead>
<tr>
<th></th>
<th>numbers</th>
<th>characters</th>
<th>Valid keys</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Low case</td>
<td>Up case</td>
</tr>
<tr>
<td></td>
<td>seq</td>
<td>rand</td>
<td>seq</td>
</tr>
</tbody>
</table>

| WEP | ASCII | 40         | 5          |
|     | HEX   | 104        |            |

| WPA | ASCII | 104        |            |
|     | HEX   |            |            |

Five test cases:  
WEP, ASCII, 104 bit,  
only characters, only UP case,  
random sequence
Test Generation according to tabular requirements

- Test Generation according to formal requirements
- All Characters, symbols & numbers used in test case are contained in an array
- Formal requirements according to tabular representation are contained in array
- Tests will sequentially work through array
- Tests stats & individual character usage in tests are saved for statistical purposes
- Individual tabular test options are contained in a function
- The characters in each test case are randomly selected according to the specific criteria

Example:
  ASCII, 64 Bit, Validkey, Characters, Mix, Random
  AjKxQ, pMrCz
Tabular and informal requirements

<table>
<thead>
<tr>
<th>WEP 1.</th>
<th>Valid keys</th>
<th>numbers</th>
<th>characters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>seq</td>
<td>rand</td>
</tr>
<tr>
<td>WEP 1.</td>
<td>ASCII</td>
<td>1.8.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>seq</td>
<td>rand</td>
</tr>
<tr>
<td></td>
<td></td>
<td>seq</td>
<td>rand</td>
</tr>
<tr>
<td>1.1.1</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.3</td>
<td>104</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.2</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.4</td>
<td>104</td>
<td>1.9.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>seq</td>
<td>rand</td>
</tr>
<tr>
<td></td>
<td></td>
<td>seq</td>
<td>rand</td>
</tr>
<tr>
<td></td>
<td></td>
<td>seq</td>
<td>rand</td>
</tr>
<tr>
<td>HEX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>seq</td>
<td>rand</td>
</tr>
<tr>
<td></td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCII</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WPA</td>
<td>ASCII</td>
<td></td>
<td>1.8.1</td>
</tr>
<tr>
<td>HEX</td>
<td>1.9.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1.8.1 – number of the informal requirements for corresponding group of tests

? – informal requirement is missed
Final stages of the project

• Correction of the informal requirements
• Modification of the set of predefined test cases for regression testing
• Establishment of the tabular coverage requirements to reflect a user’s profile
• Using the tabular coverage requirements for statistical testing
• Reliability evaluation based on results of statistical testing
Reliability evaluation
(traditional well-known approach)

• $p$ - probability of a failure in a test
  $(1 - p)$ - probability of a test without failure

• $n$ - number of test cases
  $(1 - p)^n$ - probability of $n$ test without failure

\[
(1 - p)^n = c
\]

To be sure that probability of a failure is less than $p$, $c$ should be small

$C = 1 - c$ - confidence level
Reliability evaluation

Let C be fixed (for example, C = 0.99)

\[(1 - p)^n = 1 - C\]

Question 1: We would like guarantee that the probability of a failure in a test less then \(p\). How many test cases without failures would we need?

\[n = \frac{\log(1 - C)}{\log(1 - p)}\]

Question 2: we run \(n\) test cases without failures. What is a probability of a failure in a test ?

\[p = 1 - (1 - C)^{1/n}\]
## Required number of test cases

<table>
<thead>
<tr>
<th>Reliability, $1-p$</th>
<th>Confidence level $C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.95</td>
<td>0.99</td>
</tr>
<tr>
<td>0.99</td>
<td>299</td>
</tr>
<tr>
<td>0.999</td>
<td>2 995</td>
</tr>
<tr>
<td>0.9999</td>
<td>29 956</td>
</tr>
<tr>
<td>0.99999</td>
<td>299 572</td>
</tr>
<tr>
<td>0.999999</td>
<td>2 995 734</td>
</tr>
</tbody>
</table>
Testing results

- Work in progress. Total 13,168 test cases (No failures)

<table>
<thead>
<tr>
<th></th>
<th>Valid Key Test</th>
<th>Invalid Key Test</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WEP</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCII</td>
<td>40 742</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>104 742</td>
<td>53</td>
</tr>
<tr>
<td>HEX</td>
<td>40 3374</td>
<td>240</td>
</tr>
<tr>
<td></td>
<td>104 3374</td>
<td>240</td>
</tr>
<tr>
<td><strong>WPA</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCII</td>
<td>2030</td>
<td>145</td>
</tr>
<tr>
<td>HEX</td>
<td>2030</td>
<td>145</td>
</tr>
</tbody>
</table>

- Note: Valid Key => ASCII/Hex key expected to pass
  Invalid Key => ASCII/Hex key expected to fail