SIP Proxy
User Documentation

Philipp Haupt
Matthias Hürlimann

December 14, 2006

Contents

List of Figures

Chapter 1

User Documentation

1.1 Document Information

1.1.1 History

Date	Version	Author	Description
07.12.2006	1.00	PH	Created document
08.12.2006	1.01	PH	Review and correction
12.12.2006	1.02	MH	Proofreading

1.2 Introduction

1.2.1 Purpose

This document will describe how to use the SIP Proxy application.

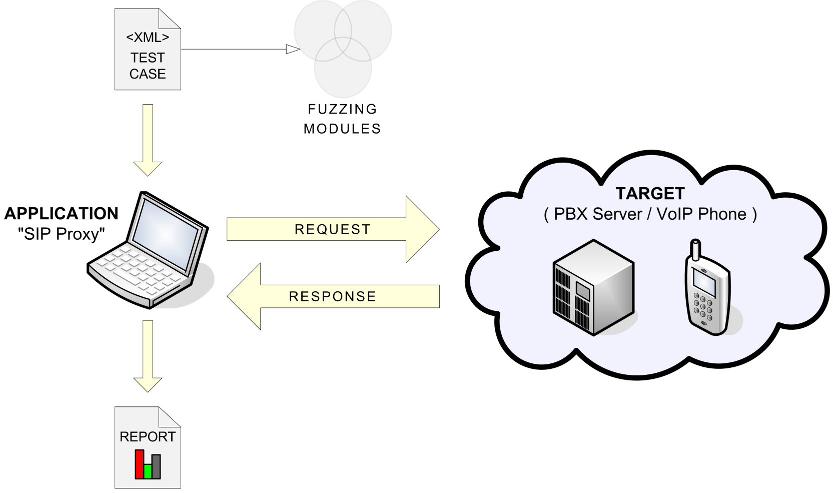
1.2.2 Scope

This document is valid over the whole project life time cycle.

1.3 User Documentation

1.3.1 What is SIP Proxy ?

SIP Proxy is an open source testing tool which can sniff SIP traffic or perform SIP related security tests. This tool should assist security analysts in finding security flaws within a VoIP environment. Security engineers have the opportunity to add custom test cases. SIP Proxy includes fuzzing technology which is a kind of black-box testing. A fuzzed attack may include random generated data to discover security flaws which are hard to find with conventional testing techniques. Hence it can help to improve the security of VoIP infrastructures. Since SIP Proxy is published under the “GNU Public License”, its source and software releases are freely available at SourceForge.net.



1.3.2 Program features

- **Proxy Mode**
  This mode can be used to sniff the SIP traffic between to participants.
  - SIP traffic sniffer
  - Send customized SIP messages
  - Dynamic message mutations with customized regular expression rules
- **Test Case Mode**
  This mode can be used to execute customized test cases against a specific VoIP target.
  - Execution of customized test cases
  - Different kinds of fuzzing modules
  - Graphical test report
  - PDF export of executed test cases
  - Sample test case files (XML-format)

1.3.3 Program Limitations

- SIP Proxy usage is limited to UDP traffic only

1.3.4 Minimum Requirements

- PC Pentium III 1GHz; >256 MB Memory
- OS: Windows, Solaris, Linux, OS X
- Java Runtime Engine Version 5.0 or higher

1.4 User Documentation

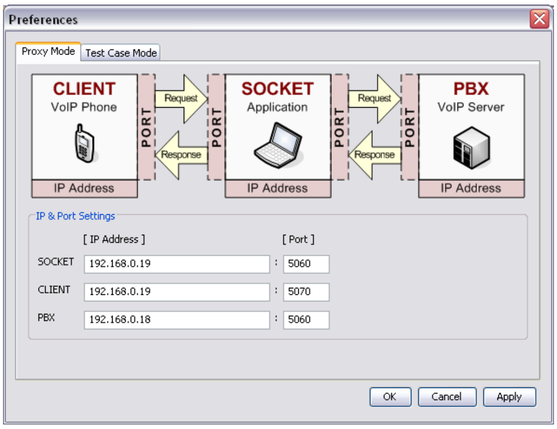
1.4.1 Installation

Extract the ZIP archive and start the JAR file (SIPProxy.jar) with the following command (shell or cmd):

```
java -jar SIPProxy.jar
```

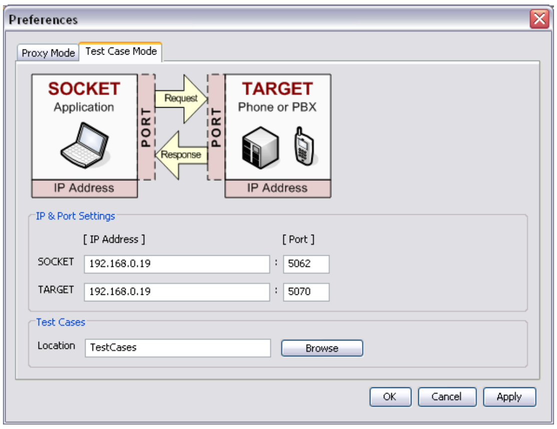
1.4.2 Preferences

As soon as the application is running, please check the IP and Port settings for the Proxy- and Test Case Mode (Figures: 1.1, 1.2).
Also check the path for the test case directory where your test case files (XML-format) are located (Figure: 1.2).



	[IP Address]		[Port]
SOCKET	192.168.0.19	:	5060
CLIENT	192.168.0.19	:	5070
PBX	192.168.0.18	:	5060

Figure 1.1: Proxy Mode settings



	[IP Address]		[Port]
SOCKET	192.168.0.19	:	5062
TARGET	192.168.0.19	:	5070

Location: TestCases

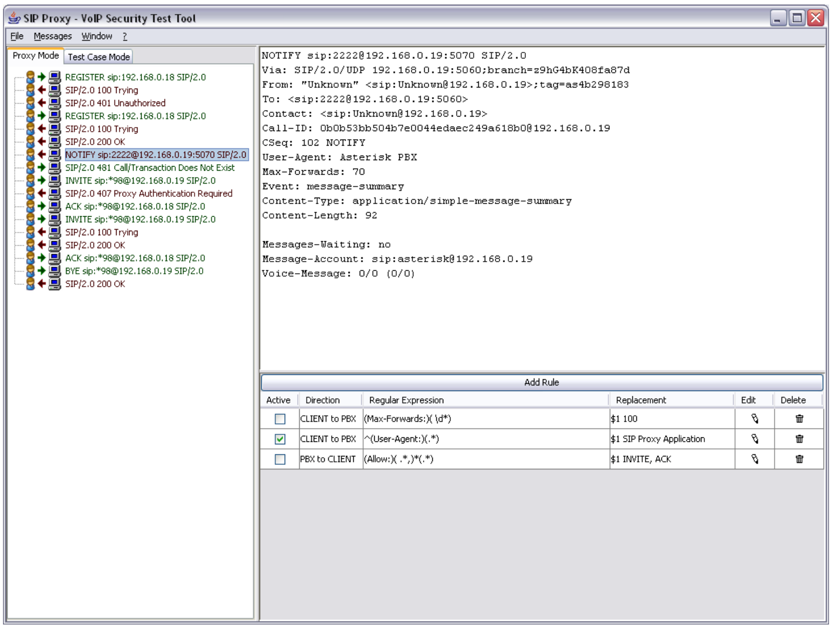
Figure 1.2: Test Case Mode settings

1.4.3 Proxy Mode

Message History

The message history tree will display captured incoming SIP messages from both directions. A left-click on a specific message in the history tree will show the content of the message in the upper right part of the application (Figure: 1.3). A right-click on a specific message in the history tree will open up a pop up menu which allows specific operations.

Note: The view of the message history can be sorted by the **message timestamp** or by the **CSeq number** ("MENU:Proxy Mode:Sort History").



```
REGISTER sip:192.168.0.18 SIP/2.0
SIP/2.0 100 Trying
SIP/2.0 401 Unauthorized
REGISTER sip:192.168.0.18 SIP/2.0
SIP/2.0 100 Trying
SIP/2.0 200 OK
NOTIFY sip:2222@192.168.0.19:5070 SIP/2.0
SIP/2.0 481 Call/Transaction Does Not Exist
INVITE sip:*98@192.168.0.19 SIP/2.0
SIP/2.0 407 Proxy Authentication Required
ACK sip:*98@192.168.0.18 SIP/2.0
INVITE sip:*98@192.168.0.19 SIP/2.0
SIP/2.0 100 Trying
SIP/2.0 200 OK
ACK sip:*98@192.168.0.18 SIP/2.0
BYE sip:*98@192.168.0.19 SIP/2.0
SIP/2.0 200 OK
```

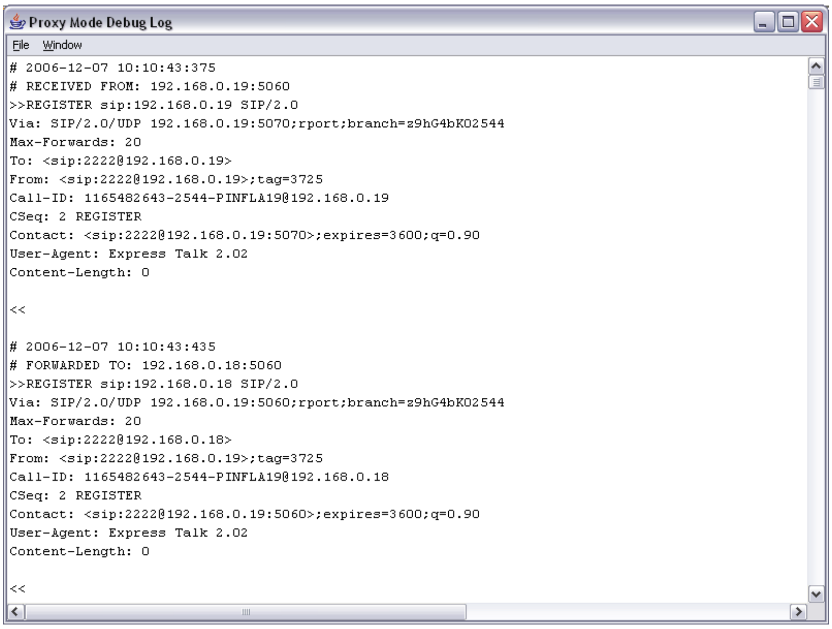
```
NOTIFY sip:2222@192.168.0.19:5070 SIP/2.0
Via: SIP/2.0/UDP 192.168.0.19:5060;branch=z9hG4bK408fa87d
From: "Unknown" <sip:Unknown@192.168.0.19>;tag=as4b298183
To: <sip:2222@192.168.0.19:5060>
Contact: <sip:Unknown@192.168.0.19>
Call-ID: 0b0b53bb504b7e0044edaec249a618b0@192.168.0.19
CSeq: 102 NOTIFY
User-Agent: Asterisk PBX
Max-Forwards: 70
Event: message-summary
Content-Type: application/simple-message-summary
Content-Length: 92

Messages-Waiting: no
Message-Account: sip:asterisk@192.168.0.19
Voice-Message: 0/0 (0/0)
```

Active	Direction	Regular Expression	Replacement	Edit	Delete
☐	CLIENT to PBX	(Max-Forwards:)(\d*)	$1 100	✎	🗑
☑	CLIENT to PBX	^(User-Agent:)(.*)	$1 SIP Proxy Application	✎	🗑
☐	PBX to CLIENT	(Allow:)(.*,)*(.*)	$1 INVITE, ACK	✎	🗑

Figure 1.3: Proxy Mode: SIP traffic sniffing

Proxy Log

The Proxy Log File (“MENU:Proxy Mode:View Log...”) shows the incoming SIP messages before and after the applied transformation (Figure: 1.4). The current content of the Proxy Log Window can be updated (“MENU:Window:Update”), cleared (“MENU:Window:Clear”) or saved as a ASCII file (“MENU:File:Save as”).

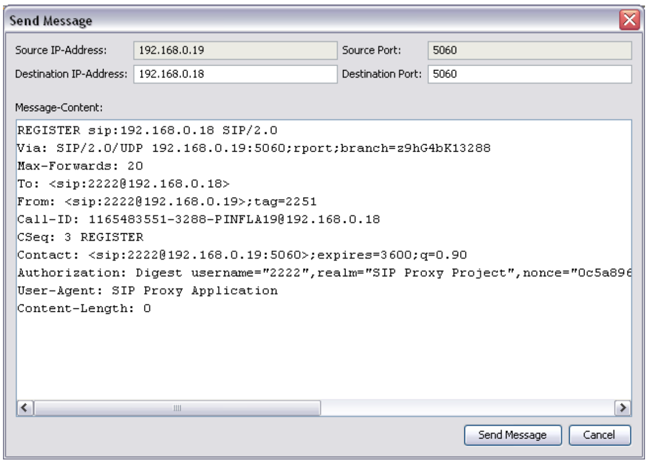


```
# 2006-12-07 10:10:43:375
# RECEIVED FROM: 192.168.0.19:5060
>>REGISTER sip:192.168.0.19 SIP/2.0
Via: SIP/2.0/UDP 192.168.0.19:5070;rport;branch=z9hG4bK02544
Max-Forwards: 20
To: <sip:2222@192.168.0.19>
From: <sip:2222@192.168.0.19>;tag=3725
Call-ID: 1165482643-2544-PINFLA19@192.168.0.19
CSeq: 2 REGISTER
Contact: <sip:2222@192.168.0.19:5070>;expires=3600;q=0.90
User-Agent: Express Talk 2.02
Content-Length: 0

<<

# 2006-12-07 10:10:43:435
# FORWARDED TO: 192.168.0.18:5060
>>REGISTER sip:192.168.0.18 SIP/2.0
Via: SIP/2.0/UDP 192.168.0.19:5060;rport;branch=z9hG4bK02544
Max-Forwards: 20
To: <sip:2222@192.168.0.18>
From: <sip:2222@192.168.0.19>;tag=3725
Call-ID: 1165482643-2544-PINFLA19@192.168.0.18
CSeq: 2 REGISTER
Contact: <sip:2222@192.168.0.19:5060>;expires=3600;q=0.90
User-Agent: Express Talk 2.02
Content-Length: 0

<<
```

Figure 1.4: Proxy Log

Sending Customized SIP Messages

Customized SIP message can be sent (“MENU:Proxy Mode:Send Message...”) (Figure: 1.5). Moreover it is possible to resend a captured SIP message by right-clicking the specific message in the message history tree and choosing “Resend...”.



Source IP-Address: 192.168.0.19
Source Port: 5060
Destination IP-Address: 192.168.0.18
Destination Port: 5060

Message-Content:

```
REGISTER sip:192.168.0.18 SIP/2.0
Via: SIP/2.0/UDP 192.168.0.19:5060;rport;branch=z9hG4bK13288
Max-Forwards: 20
To: <sip:2222@192.168.0.18>
From: <sip:2222@192.168.0.19>;tag=2251
Call-ID: 1165483551-3288-PINFLA19@192.168.0.18
CSeq: 3 REGISTER
Contact: <sip:2222@192.168.0.19:5060>;expires=3600;q=0.90
Authorization: Digest username="2222",realm="SIP Proxy Project",nonce="0c5a896
User-Agent: SIP Proxy Application
Content-Length: 0
```

Figure 1.5: Sending customized SIP messages

Dynamic Message Mutation Rules

A custom rule can be defined to dynamically mutate incoming messages (Figure: 1.6). Regular expressions must be used to define a valid rule. Since a regular expression can consist of different groups, a replacement string can be applied. Refer to the following sample rule to understand the general usage:

```
Regular Expression = "^(User-Agent:)(.*)"
Replacement        = "$1 SIP Proxy"
```

The regular expression in the sample above is divided into two groups. The replacement string will then refer the first group with $1 and leave the second group out. This will result in the following mutation:

```
Input string = "User-Agent: Asterisk PBX"
```

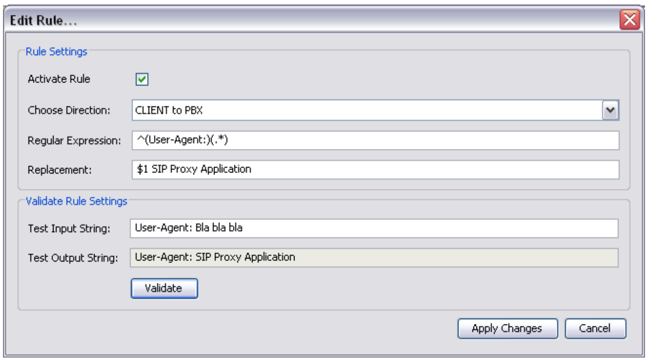
Will be marked as:

```
$1 = "User-Agent:"
$2 = " Asterisk PBX"
```

And then replaced as:

```
"User-Agent: SIP Proxy"
```

Use the “validate rule” to verify the expected result.



Edit Rule...

Rule Settings

Activate Rule ☑

Choose Direction: CLIENT to PBX

Regular Expression: ^(User-Agent:)(.*)

Replacement: $1 SIP Proxy Application

Validate Rule Settings

Test Input String: User-Agent: Bla bla bla

Test Output String: User-Agent: SIP Proxy Application

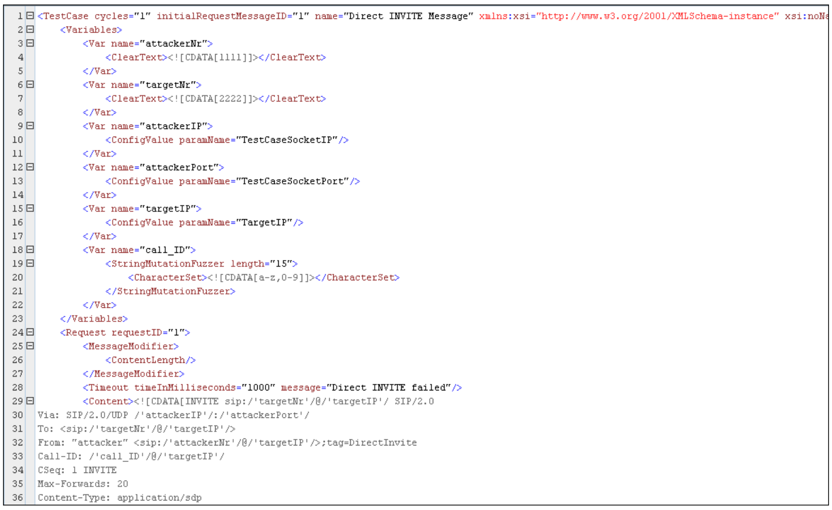
Figure 1.6: Dynamic Message Mutation Rules

1.4.4 Test Case Mode

Writing a Customized Test Case

First of all, a custom test case scenario can be defined by the user. Create an XML file according to the "**Test Specification Reference**" **??**document to define your own test case (Figure: 1.7). Place all your test case XML files in the directory which you have defined in the preferences settings 1.4.2 of the application. The application will then try to load all valid test cases within this directory. The test cases will then appear in the drop down menu. If you added new test case files after loading the application, please press the "Reload"-Button to reload the directory.

Note: Please include the XML Schema "**TestCaseSchema.xsd**" in all your test case XML files to validate it. Any invalid test case files will not be accepted by the SIP Proxy application. The schema is located in the /TestCase directory.

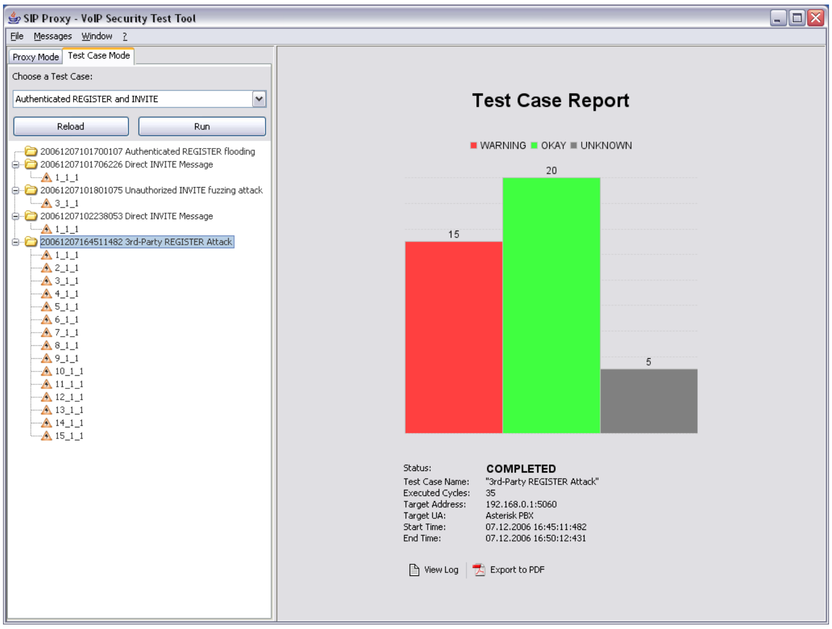


```
<TestCase cycles="1" initialRequestMessageID="1" name="Direct INVITE Message" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noN
    <Variables>
        <Var name="attackerNr">
            <ClearText><![CDATA[1111]]></ClearText>
        </Var>
        <Var name="targetNr">
            <ClearText><![CDATA[2222]]></ClearText>
        </Var>
        <Var name="attackerIP">
            <ConfigValue paramName="TestCaseSocketIP"/>
        </Var>
        <Var name="attackerPort">
            <ConfigValue paramName="TestCaseSocketPort"/>
        </Var>
        <Var name="targetIP">
            <ConfigValue paramName="TargetIP"/>
        </Var>
        <Var name="call_ID">
            <StringMutationFuzzer length="15">
                <CharacterSet><![CDATA[a-z,0-9]]></CharacterSet>
            </StringMutationFuzzer>
        </Var>
    </Variables>
    <Request requestID="1">
        <MessageModifier>
            <ContentLength/>
        </MessageModifier>
        <Timeout timeInMilliseconds="1000" message="Direct INVITE failed"/>
        <Content><![CDATA[INVITE sip:/'targetNr'/@/'targetIP'/ SIP/2.0
Via: SIP/2.0/UDP /'attackerIP'/:/'attackerPort'/
To: <sip:/'targetNr'/@/'targetIP'/>
From: "attacker" <sip:/'attackerNr'/@/'targetIP'/>;tag=DirectInvite
Call-ID: /'call_ID'/@/'targetIP'/
CSeq: 1 INVITE
Max-Forwards: 20
Content-Type: application/sdp
```

Figure 1.7: Part of a Test Case sample file (XML)

Execution of Test Cases

Choose a specific test case in the drop down menu and click on the "Run"-button. The test case will be started afterwards. The test case history tree will list all executed and running test cases. If you just started a test case run or if you clicked on a specific test case item in the history tree, a test case report will be shown in the right part of the application (Figure: 1.8). The history tree will also list all incoming warning messages with its ID number [1.4.4]. A right-click on a test case entry will open up a pop up menu which allows more specific operations.



Choose a Test Case: Authenticated REGISTER and INVITE

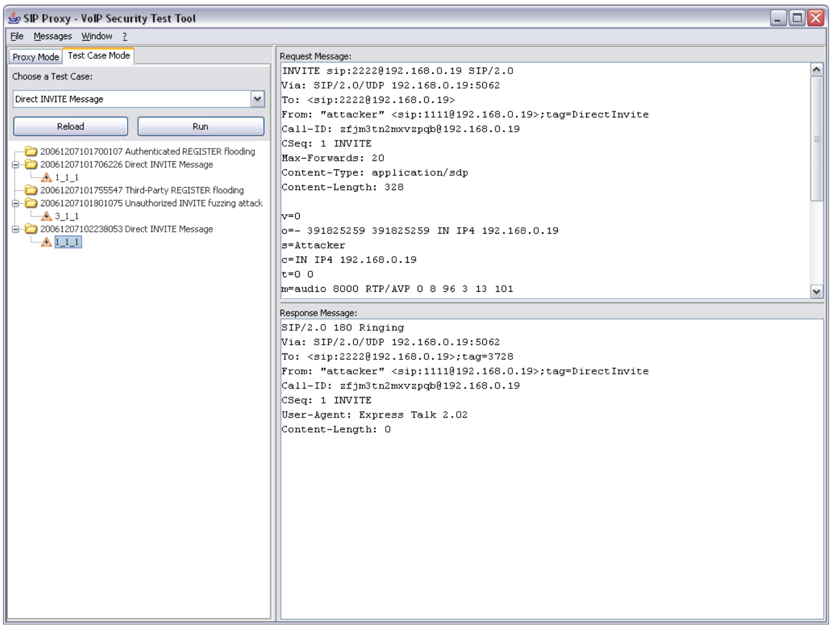
- 20061207101700107 Authenticated REGISTER flooding
- 20061207101706226 Direct INVITE Message
  - 1_1_1
- 20061207101801075 Unauthorized INVITE fuzzing attack
  - 3_1_1
- 20061207102238053 Direct INVITE Message
  - 1_1_1
- 20061207164511482 3rd-Party REGISTER Attack
  - 1_1_1
  - 2_1_1
  - 3_1_1
  - 4_1_1
  - 5_1_1
  - 6_1_1
  - 7_1_1
  - 8_1_1
  - 9_1_1
  - 10_1_1
  - 11_1_1
  - 12_1_1
  - 13_1_1
  - 14_1_1
  - 15_1_1

Status:	**COMPLETED**
Test Case Name:	"3rd-Party REGISTER Attack"
Executed Cycles:	35
Target Address:	192.168.0.1:5060
Target UA:	Asterisk PBX
Start Time:	07.12.2006 16:45:11:482
End Time:	07.12.2006 16:50:12:431

Figure 1.8: Test Case Mode: Test Case Report

Analyze Warnings

A left-click on a warning entry in the history tree will show the appropriate request- and response message which have been categorized as a warning (Figure: 1.9). Right-click on a warning entry will open up a PopUp-Menu which allows more specific operations.


SIP Proxy - VoIP Security Test Tool
File Messages Window ?
Proxy Mode
Test Case Mode
Choose a Test Case:
Direct INVITE Message
Reload
Run
20061207101700107 Authenticated REGISTER flooding
20061207101706226 Direct INVITE Message
1_1_1
20061207101755547 Third-Party REGISTER flooding
20061207101801075 Unauthorized INVITE fuzzing attack
3_1_1
20061207102238053 Direct INVITE Message
1_1_1
Request Message:
Response Message:

```
INVITE sip:2222@192.168.0.19 SIP/2.0
Via: SIP/2.0/UDP 192.168.0.19:5062
To: <sip:2222@192.168.0.19>
From: "attacker" <sip:1111@192.168.0.19>;tag=DirectInvite
Call-ID: zfjm3tn2mxvzpqb@192.168.0.19
CSeq: 1 INVITE
Max-Forwards: 20
Content-Type: application/sdp
Content-Length: 328

v=0
o=- 391825259 391825259 IN IP4 192.168.0.19
s=Attacker
c=IN IP4 192.168.0.19
t=0 0
m=audio 8000 RTP/AVP 0 8 96 3 13 101
```

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.0.19:5062
To: <sip:2222@192.168.0.19>;tag=3728
From: "attacker" <sip:1111@192.168.0.19>;tag=DirectInvite
Call-ID: zfjm3tn2mxvzpqb@192.168.0.19
CSeq: 1 INVITE
User-Agent: Express Talk 2.02
Content-Length: 0
```

Figure 1.9: Test Case Mode: Test Case Warning

View Log File

Just click on the "View Log"-Button in the test case report to open the test case log file. The log file will show all outgoing and incoming messages of the selected test case run.

Export and Print Test Case Reports

Just click on the "Export to PDF"-Button in the test case report to export it into a printable PDF document.

Message ID format

Every request and response message of a test case run will be logged with a specific ID number. The ID number has the following format:

```
<current cycle number>_<request message ID>_<unique ID per cycle>
```

- **current cycle number**
  The current cycle number of the test case run
- **request message ID**
  The request message ID number which have been defined in the test case specification
- **unique ID per cycle**
  A incrementing ID number which is unique within the current cycle. A new cycle will reset this number to zero.

1.5 WARNING

Please be aware that the we take no responsibility for any damage caused by the usage of SIP Proxy. Do not use this tool unless you know what you are doing. SIP Proxy was designed for testing purpose only.

1.6 Problem Section

1.6.1 The application will not start

Please check the minimum requirements at the beginning and make sure the Java runtime engine version 5.0 or higher has been installed properly.

1.6.2 Known bugs

There are no known bugs for the current release version 2.0 stable.
Please report any bugs via SourceForge.