Dipartimento di Ingegneria e Scienza dell'Informazione
Departement of Information Engineering ans Computer Science

Bachelor's Degree in
Computer Science

Final dissertation

HTTP Security testing language and implementation

Sottotitolo (alcune volte lungo - opzionale)

Supervisor
Silvio Ranise

Student
Matteo Bitussi

Co-Supervisors
Andrea Bisegna
Roberto Carbone

Academic year 2021/2022

Ringraziamenti

...thanks to... TODO (in italiano)

Contents

Abstract **2**

1 Background **3**
1.1 Burp Suite . . . 3

2 Design **3**
2.1 The Language . . . 3
2.1.1 Language structure . . . 4
2.2 The oracle . . . 5

3 Implementation **5**
3.1 SAML use case . . . 5

4 Related work **6**
4.1 Wendy's language . . . 6
4.2 The old plugin . . . 6
4.3 Last plugin version . . . 6

5 Uses cases **7**
5.1 SAML Use-Case . . . 7
5.2 OAuth & OIDC Use-Case . . . 7

6 Conclusions **7**

Bibliografia **7**

A Titolo primo allegato **9**
A.1 Titolo . . . 9
A.1.1 Sottotitolo . . . 9

B Titolo secondo allegato **10**
B.1 Titolo . . . 10
B.1.1 Sottotitolo . . . 10

Abstract

This thesis covers the work started in my internship at FBK in the context of Single-Sign-On (SSO) protocols testing. SSO protocols are becoming more and more poular these days, and are being used in very sensitive applications such as SPID or CIE. Seeing the vast number of different implementations that are being used from whatever type of service, there is the need to test them in order to ensure that (at least) the known most common vulnerabilities are avoided.
To avoid having to manually test each implementation, an automaic tool is necessary. Moreover, a standard language to define these test suites has to be defined. This is what will be shown in this paper.

1 Background

The idea of a pentesting tool used to test SSO implementations such as OAuth, OIDC and SAML was previously discussed by my colleagues Stefano Facchini [2], Claudio Grisenti [3] and Wendy Barreto [1], which developed a plugin in Burp with the intent of automating the the testing of OIDC and OAuth protocols.

1.1 Burp Suite

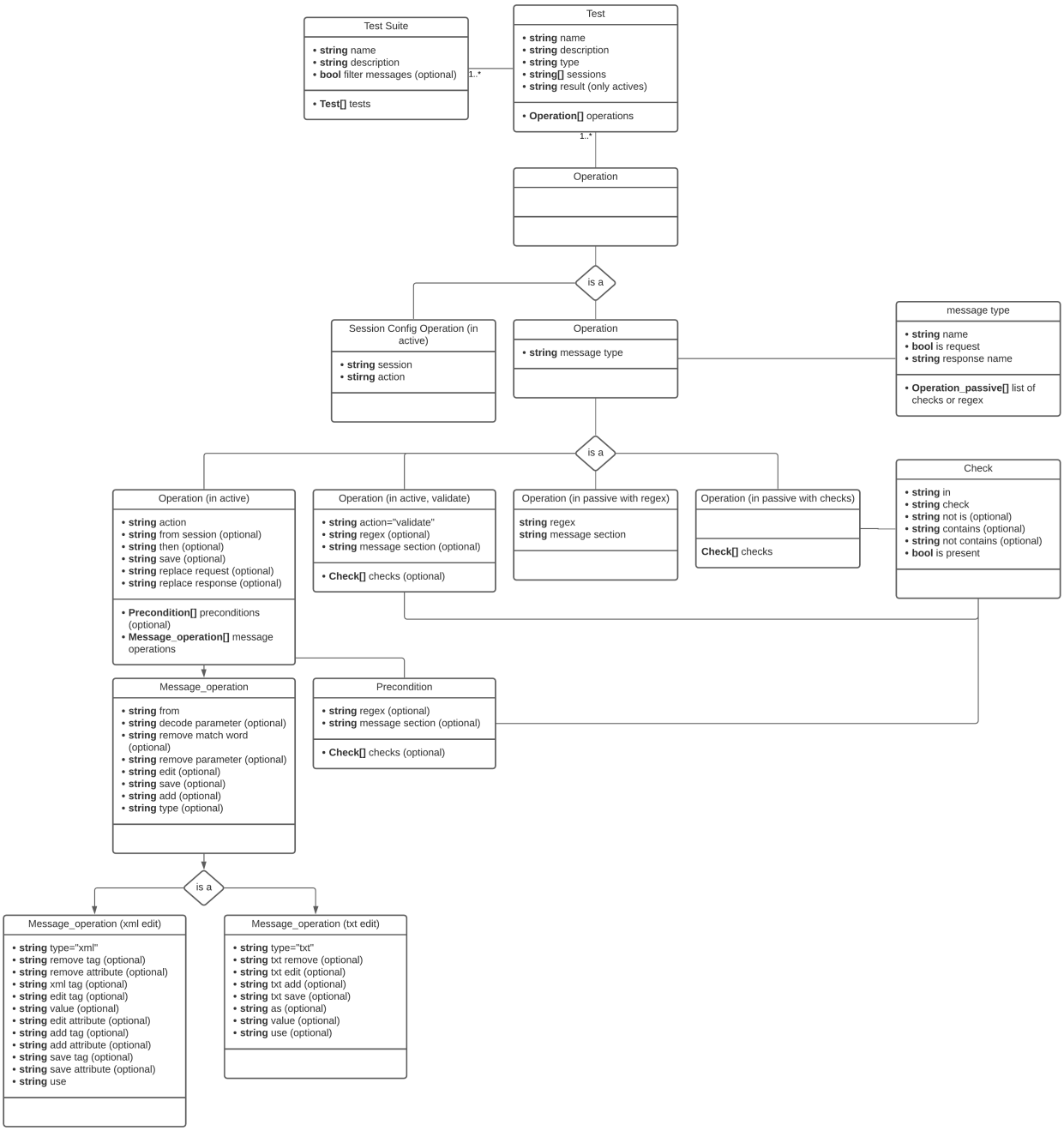
Burp is one of the most used application security testing software for web security testing. It works by the use of a proxy server over which a browser redirect the traffic to. Burp has access to the proxy, it can sniff HTTP packets and can edit them. Burp also gives the possibility of creating custom plugins giving to the developers access to the java API.

2 Design

2.1 The Language

The idea was to think of a language that could implement all the possible actions which a security tester would be wanting to do on a multipart webapp test.
I had to decide how to write and define the actual tests, i thought i could define a proper language with a dedicated parser, but it was not worth the effort, as there are already some well-tested alternatives available. I found a great alternative: i used JSON as a base over which write the tests. It is a convinient way of defining gerarchical sturctures like tests could be. The gerarchical structure and the details of the language will be discussed in the next charapter.

2.1.1 Language structure



Test suite

The test suite is the main component which contains all the other one, it is composed by:

- Test suite name, the name of the test suite
- Test suite description, the description of the test suite
- Tests, which is a list containing the tests to be executed

Test

The Test object is the one that actually defines a test. As said earlier, a test is contained in a Test Suite, and has various items:

- name

- description
- type, it can be "active" or "passive"
- sessions, which is a list of the sessions which are needed in this test
- result, (only for actives) it defines the conditions over which the test is considered passed or not.
- operations, a list of operation objects which will be executed in the Test object

it can be defined either as an active or a passive test, depending on the type of actions it has to do on the intercepted messages. If a doesn't need to manipulate the flow or the content of the messages, then it is considered passive, otherwise it is considered active. The list of operations is executed iteratively one after the other.

Operation

The operation object is the thing that define what a test actually does. As shown in the image above, an operation could be either a standard operation or a session config operation, the latter is used to manage the sessions for the active tests (i.e. start, stop, pause). Depending on the type of test which an Operation is defined into, the standard Operation can be active or passive.
A **passive** operation should contain one of the following options:

- A list of Check objects
- A regex inspection

A regex operaton executes an inspection in

2.2 The oracle

3 Implementation

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

3.1 SAML use case

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis.

Phasellus gravida tellus velit, non eleifend justo lobortis eget.

4 Related work

4.1 Wendy's language

Action	Old language	New language
Custom message filtering	Only on active tests	supported
Edit string	only by regex	supported with regex and check construct
Remove string	only by regex	supported with regex and check construct
Add string	not supported	supported
Check parameter	only with regex	with regex and check construct
Multiple operations in single message	not supported	supported
Saving and reusing of values and messages	not supported	supported
Multiple sessions in single test	not supported	supported
Custom oracle definition	not supported	by using regex and checks

4.2 The old plugin

The old plugin of Stefano and Claudio was based on a track, which defined some actions to be done by the browser, which was a selenium istance. The plugin checks the messages and based on the test defined in the plugin tells if there is a vulnerability or not. Starting from the first version of Stefano, to the last of Wendy, the plugin was improved. In the first two versions of Stefano and Claudio the plugin had its tests hard-coded, in a way that ony the supported tests could be executed, with little settings to change. If a new test had to be implemented, the plugin had to be recompiled. This version of the plugin worked well, but as said, the tests could not be customized or adapted by the user.

4.3 Last plugin version

Wendy improved the plugin by removing the staticity of the test, adding the possibility to customly define all of the tests with the use of a JSON language.
This is the last version of the plugin which i started working to. The plugin supported the definition of passive and active tests: passive tests are tests where the messages are not edited, active tests are tests where there could be an edit of one or more messages. The availabe test actions worked well, but there were some limitations on the possible actions, especially in the active tests. For example:

- Limited oracle for the verification of active tests, having just the verification of the correct execution of the operation and a check for the string "error" on the last page of the browser
- The filtering of which message to check or edit for static tests was limited: (only "Authorization grant message", "Response messages", "Request messages" and "All messages")
- Only regex were supported to search something in a message
- Unable to work over encoded parameters
- Impossibility of doing multiple operations on a single message
- Impossibility of saving a parameter and using it somewhere else

Some of which stated as future works in Wendy's thesis. Moreover, the language was thought to be used with tests for OIDC and OAuth, other SSO protocols such as SAML could not be tested, because

of the fact that SAML parameters are encoded, so editing them or verifying them is not possible. This is the biggest limitation that made me decide to redesign the language.

5 Uses cases

5.1 SAML Use-Case

5.2 OAuth & OIDC Use-Case

6 Conclusions

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Bibliography

[1] Wendy Barreto. Design and implementation of an attack pattern language for the automated pentesting of oauth/oidc deployments. Master's thesis, Università degli Studi di Trento, 2021.

[2] Stefano Faccini. Design and implementation of an automated tool for checking saml sso vulnerabilities and spid compliance. Master's thesis, Università degli Studi di Trento, 2019/2020.

[3] Claudio Grisenti. A pentesting tool for oauth and oidc deployments. Master's thesis, Università degli Studi di Trento, 2019/2020.

Allegato A Titolo primo allegato

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

A.1 Titolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

A.1.1 Sottotitolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Allegato B Titolo secondo allegato

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

B.1 Titolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

B.1.1 Sottotitolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.