WebSockets Active Scanning

About SQL Injections

Kirtas Emmanouil (a.k.a Manos)
manolis.kirt@gmail.com
irc: manos

March 27, 2018

1 A few words about myself and maybe even more

First of all, for the past 6 years I have been studying and practicing the Computer Science. My interests include, development and operation of software, as well as research. More specifically, I am very interested in security of systems and privacy, particularly after having participated in relevant conferences these past 2 years. Furthermore, I am very fond of artificial intelligence techniques and machine learning. On account of this, for the past year and a half I have been implementing on java an academic paper, titled "A framework for Access Control with Inference Constraints". [1] Subsequently, my personal goal for the future, is to combine software security with machine learning and fuzz logic techniques.

In addition to the above, I have worked as a member of a photo-shooting cooperative. This experience has given me the opportunity to work on data base management (MySQL) and web application development. Also, I had the opportunity to learn how to work as an equal part of a group in a work environment.

Additionally, all these years I have contributed to various communities, such as ACM in Greek student chapter in order to develop a game using game engine unity. Furthermore, I have participated in ELLAK's summer school of code for two years in a row. [2], [3] [4] Lately, I have been part of a student group, which deals with Security issues. Currently, we have been working on basic Pen Test and Reverse Engineering techniques through OverTheWire and HatchTheBox platforms. Hence, we aspire to participate, in the future, in some CTFs.

Last but not least, I have been a Linux user for as long as I can remember, however I do not consider myself as a SuperUser, as I have further elaborated on Linux systems only these past 3 years. I am also very fond of configuring the open source programs I use, such as the i3 windows manager, Emacs Editor, tmux, zsh, as well as Arch, which I still run on Virtual Machine (I hope that it will soon become my basic OS). Finally, I use InteliJ IDE to develop and

[1] http://ieeexplore.ieee.org/abstract/document/6032355/?anchor=authors

[2] https://ma.ellak.gr/unit/%CE%B1%CF%81%CE%B9%CF%83%CF%84%CE%BF%CF%84%CE%AD%CE%BB%CE%B5%CE%B9%CE%BF-%CF%80%CE%B1%CE%BD%CE%B5%CF%80%CE%B9%CF%83%CF%84%CE%AE%CE%BC%CE%B9%CE%BF-%CE%B8%CE%B5%CF%83%CF%83%CE%B1%CE%BB%CE%BF%CE%BD%CE%AF/

[3] https://ma.ellak.gr/software/%CF%83%CF%8D%CF%83%CF%84%CE%B7%CE%BC%CE%B1-%CE%B4%CE%B9%CE%B1%CF%87%CE%B5%CE%AF%CF%81%CE%B9%CF%83%CE%B7%CF%82-%CE%BF%CF%85%CF%81%CF%8E%CE%BD/

[4] https://github.com/ManosMagnus/Easy-Ticket-App

build Java application.

My skills:

- I am well experienced in Java and I am familiar with Java 8 and Lamba expressions.
- I have good knowledge of C++, Python, HTML/CSS, JSON, MYSQL, MongoDB and Git
- I am familiar with javascript and prolog
- Also, I have good knowledge working on a remote server via ssh
- I am in love with LaTeX

1.1 In reference to my contact with OWASP organization

My first contact with OWASP organization was when I was looking for information about my thesis. Since then, I frequently read articles and I even own some books from the organization. Hence, the moment that this organization was announced to be part of GSoC, it was the first (of the participating organizations) that I wanted to look into. In addition, when I read OWASP's "philosophy" I realized that I wanted to get involved solely with the organization's projects. In my opinion, OWASP's trend promotes the development of secure software, thus becoming integral for developers who respect users and their personal data.

1.2 In reference to my contact with ZAP

As a junior developer who cares about the security of applications I find ZAP an essential tool in developing and testing lifecycle. I like the philosophy of an easy to use tool, which could be used by non security experts. Of course, besides that, ZAP is a powerful tool for resolving many advanced issues. That is the reason I got immediately involved with ZAP by learning how to use it from their many beginners' tutorial videos, wikis and also by asking for help from the community. I also started to use ZAP in real case scenarios with the help of DVWS[5]. At the same time, I have started to familiarize with code base by implementing a FirstIdealIssue #1382[6] (I made pull request #1506 [7]) as well as noting a bug in community about AjaxSpider (issue #5521[8]).

[5] https://www.owasp.org/index.php/OWASP_Damn_Vulnerable_Web_Sockets_(DVWS)
[6] https://github.com/zaproxy/zaproxy/issues/1382
[7] https://github.com/zaproxy/zap-extensions/pull/1506
[8] https://github.com/zaproxy/zaproxy/issues/4521

2 In reference to the Idea I am going to Implement

2.1 The Basic Idea

For my contribution to the OWASP organization and more specifically to the ZAP project I am going to implement an active web socket scan, which was proposed on the Ideas' page. Specifically, I will develop an extension for WebSocket add-on by adding some Active Scans. The Active scan will be include tests to web application in Error[9], Blind[10] and Login by pass[11] SQL Injections vulnerabilities. To accomplish my idea I will test web application using Time Based techniques [12]. The scans will be able to test either specific or already established connections. At this moment, we are not able to automatically establish connection with AjaxSpider (unless the issue is resolved until May). So, my long term goal about that is to establish and scan all websocket connections under a specific endpoint.

Another significant clue about websocket vulnerabilities is that the WebSocket protocol doesn't handle authentication. Practically this means that, a WebSocket opened from a page behind authentication doesn't "automatically" receive any sort of authentication; you need to take steps to also secure the WebSocket connection. That makes possible to establish connection without authentication making use of http and origin headers field of the client[13]. "WebSocket is a nightmare because it does not follow Same-origin Policy browser restrictions"[14]. That vulnerability was known as WebSocket Hijacking[15].

Some others posible attacks and scans at WebSockets are the classic bruteforce attack[16], Command execution[17], Local[18] and Remote[19] file inclusion. In addition, some more challenging attacks at WebSockets are the Stored and Reflected Cross-Site Scripting[20] .

I am going to implement some of the above scans, (as more as possible) starting with SQls Injections. I strongly believe that we could easily add the scans if the basic infrastructure was build.

The general aim of this implementation will be to respect the basic principals of usability and consistency making ZAP easy to use. I, also, intent to reuse and extend classes and methods following the Development Rules and Guidelines[21]. Finally, my implementation will aim to facilitate the internationalization of ZAP.

To begin with, the extensions will come with appropriate user interface in the standards of the existing Active Scan. Also, the appropriate dialogs, popups, new components in attack category and new tab panel in tool menu will make the extension usable. In addition, I intent to add triggers for alert panel with existing descriptions, information and solution guides. Last

[9] https://www.owasp.org/index.php/SQL_Injection
[10] https://www.owasp.org/index.php/Blind_SQL_Injection
[11] https://www.owasp.org/index.php/SQL_Injection_Bypassing_WAF
[12] https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OTG-INPVAL-005)Time_delay_Exploitation_technique
[13] https://devcenter.heroku.com/articles/websocket-security
[14] https://gist.github.com/subudeepak/9897212
[15] http://www.christian-schneider.net/CrossSiteWebSocketHijacking.html
[16] https://www.owasp.org/index.php/Brute_force_attack
[17] https://www.owasp.org/index.php/Command_Injection
[18] https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion
[19] https://www.owasp.org/index.php/Testing_for_Remote_File_Inclusion
[20] https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
[21] https://github.com/zaproxy/zaproxy/wiki/DevGuidelines

but not least, the extension will come with API.

2.2 Reasons for choosing this particular idea

WebSockets is a very recent protocol, which was standardized in 2011[22]. One of the basic features of WebSocket is establishing a connection over TCP and remaining open. In that way it allows the server and the client to create a full-duplex communication channel, thus allowing both of them to truly communicate asynchronously. This it the biggest advantage of WebSocket contrary to traditional Ajax Request/Response. That is the reason the WS is suitable for chats, messages notifications, monitoring etc. Furthermore, recent years WS potocol was used widely over the web applications. With the current version of ZAP we are able to intercept and show WS payloads, set breakpoint on specific types of WS's payloads and fuzz payloads. Also ZAP WS add-on is considered as a reliable tool for WS communication analysis and debuging. I strongly believe that active scan extension of ZAP will be make it an even more powerful tool.

On the other hand, I choose specific Vulnerabilities because in most cases, the ones I will be working on are the most critical issues that arise on a web application. In addition, we find those vulnerabilities always on the top of 10 most critical security risks. [23] [24] [25] In conclusion, I believe SQL injections is a good start for me to get involved with ZAP.

2.3 How am I going to implement the Idea?

For active scans implementation, I am going to use the existing fuzzing rules of ZAP. These rules use common techniques to find and exploit sql injection vulnerabilities. More specifically, I will extend the classes which relate to active scans like AbastractHostPlugin[26] and SQLInjectionsTest[27]. Also, I am going to use the existing class of messages representation[28], WebSocket proxy[29], HSQLDB storing[30] and API generator[31].

However, one of the problems I am going to face is the difference between WebSocket messages' format and traditional protocols. As it is known, WebSocket protocol uses JSON strings format. Therefore, I will either use Java's [32] or quick-json[33] API.

[22] https://en.wikipedia.org/wiki/WebSocket
[23] https://www.owasp.org/index.php/Top_10-2017_Top_10
[24] https://www.owasp.org/index.php/Top_10_2013-Top_10
[25] https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#OWASP_Top_10_for_2010
[26] https://github.com/zaproxy/zaproxy/blob/develop/src/org/parosproxy/paros/core/scanner/AbstractHostPlugin.java
[27] https://github.com/zaproxy/zap-extensions/blob/master/src/org/zaproxy/zap/extension/ascanrules/TestSQLInjection.java
[28] https://github.com/zaproxy/zap-extensions/blob/master/src/org/zaproxy/zap/extension/websocket/WebSocketFuzzMessageDTO.java
[29] https://github.com/zaproxy/zap-extensions/blob/master/src/org/zaproxy/zap/extension/websocket/WebSocketProxy.java
[30] https://github.com/zaproxy/zap-extensions/blob/master/src/org/zaproxy/zap/extension/websocket/db/TableWebSocket.java
[31] https://github.com/zaproxy/zap-extensions/blob/master/src/org/zaproxy/zap/extension/ApiGenerator.java
[32] https://www.owasp.org/index.php/OWASP_Damn_Vulnerable_Web_Sockets_(DVWS)
[33] https://code.google.com/archive/p/quick-json/

As for graphical user interface, I am going to use javax.swing library. More specifically, I will extend classes like ExtenionPopupMenu, WebSocketPopupMenuItem etc. Of course, always with respect to encapsulation and extensibility principals.

I will test the usability of extension in DVWS[34] and if is necessary I will develop some small javascript web applications.

2.4 Why is this goal feasible?

- It is not based in theoretical guesses or scenarios but in real situations (Websocket could be established from different kind of network clients, sql injections could be transmitted via WebSocket connections etc)
- There are also other implementations in Javascript[35] and Python (sqlmap)[36]
- codebase support additions of active scans extensions easily
- ZAP is well documended (wikis, videos, blogs)
- Developers of community are active and very helpful

2.5 What am I going to commit on the 9th of August

- Basic Infrastructure which supports adding Active Scans
- Actives Scan about
  - Error SQL Injection
  - Blind SQL Injection
  - Sql injection login by pass
  - Classic Brute Force
  - Local and Remote file inclusion
  - (Optional) WebSocket Hijacking
  - (Optional) Stored and Reflected Cross-Site Scripting
- Users Graphical Interface
  - New tad in tools tab
  - New entry in Attack's category
  - Suitable dialogs
  - Triggers for alert tab
- Active Scanner API
- JUnits Tests

[34] https://www.owasp.org/index.php/OWASP_Damn_Vulnerable_Web_Sockets_(DVWS)
[35] https://github.com/mmmds/walkthroughs/blob/master/dvws/error_sql.html
[36] https://github.com/RicterZ/websocket-injection

3 Timeline Table

- 14 May - 19 May
  - Design the strategy of the implementation I am going to use
  - Create a very basic class diagram
  - Use case scenarios abouts scans
  - If it is necessary, Develop small javascript applications for test
  - Feedback about my Ideas from Developers Community
  - Collect all the necessary libraries
  - Create necessary abstract classes
  - Create necessary data structures
- 21 May - 26 May
  - (Continue to) create necessary data structures and abstract classes
  - Develop necessary components and modules
  - Editing the existing classes (if necessary)
- 28 May - 2 June
  - Start extending AbstractHostPluging and TestSQLInjection for SQL Blind Injection
- 4 June - 9 June
  - Add methods for Erros SQL Injections
- 11 June - 16 June
  - Add method for SQL Injection login bypass
- 18 June - 23 June
  - Add Methods for BruteForce Scans
- 25 June - 29 June
  - This week Is for unexpected bug resolving
  - Test in web applications
  - Feedback from community
  - (Alternatively/Optional) Stored and Reflected Cross-Site Scripting
- 1 July - 7 July
  - WebSocket Hijacking
- 9 July - 13 July

- Methods for Remote and Local and Remote file inclusion scans

- 16 July - 21 July
  - Development for basic ui like tabs, dialogs and popups and Alert Feature

- 23 July - 28 July
  - API Development

- 30 July - 9 Aug
  - Feedback from community
  - This week Is for unexpected bug resolving

4 In reference with my participation in GSoC program

In the past, I have cooperated with many people. In my opinion, one of the most remarkable cooperations I ever had is the professor who is currently supervising my thesis. In fact, he and I hope to publish a paper together. However, to be honest, I have never cooperated with people from abroad. That fact makes me uncomfortable sometimes when writing or speaking English, but that is something I'm working on constantly. After all, I aspire to cooperate with people from abroad in the future and more specifically I really hope to be given this chance through GSoC, if I were to be elected.

My home country is Greece where I was born and have been living ever since. I am always open to meeting others from OWASP community.

In reference to the hours I am able to work on this project, I am going to work 8 hours per day, 5-6 days a week. The only commitments I have this period are this semester's exams and the presentation of my thesis. However, my exams won't take a great deal off of my time, as I only need to take exams in two subjects, for which I am already studying, so last minute work won't become an obstacle in my work on the project. Overall, I am very enthusiastic about this project, so working on it will be my main occupations

I want to remain ZAP's contributor even after GSoC. One of the issues I want to involved, is the Ajax Spider bug, if the issue isn't resolved until then. Also I want to develop another add-on which will give me the ablity to export ZAP's reports in Tex file. In addition, I want to involved with othe OWASP's Projects like owasp-mstg[37] and OWASP Security Knowledge Framework[38] as well.

5 Contact me!

Irc at mozilla server especially in channel #websectools with nickname "manos"
Email: manolis.kirt@gmail.com
Github: `https://github.com/ManosMagnus`

[37] `https://github.com/OWASP/owasp-mstg`
[38] `https://www.owasp.org/index.php/OWASP_Security_Knowledge_Framework`