Information and Software Technology 140 (2021) 106703

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

HYDRA: Feedback-driven black-box exploitation of injection vulnerabilities ☆

Manuel Leithner, Bernhard Garn, Dimitris E. Simos *

SBA Research, Floragasse 7, 1040 Wien, Austria

ARTICLE INFO

Keywords:
Security testing
Combinatorial testing
Black-box testing
Injection vulnerabilities
Web application security
Cross-site scripting
SQL injection

ABSTRACT

Context: Injection vulnerabilities remain an omnipresent threat to web application security. These issues arise when user-supplied input is included in commands constructed by the application without applying adequate validation and filtering, permitting attackers to modify the resulting instructions.

Objective: Tools used in real-world security assessments commonly employ a static list of malicious input strings to be submitted to the system under test (SUT) to gauge the presence of vulnerabilities. However, sanitizing filters may cause these simulated attacks to fail, even if they only mitigate a subset of potentially harmful values. This may result in a false sense of security. This work introduces HYDRA, a feedback-driven black-box security testing approach for the exploitation of injection vulnerabilities. It is capable of constructing inputs designed to evade such imperfect filters while allowing users to define and rank *output contexts*, abstract locations in the output of the SUT that are associated with desirable semantics (for instance, allowing the execution of JavaScript code).

Method: Starting with an innocuous initial input string that is submitted to the SUT and appears anywhere in the output, HYDRA identifies the initial output context. It extends the input string with the goal of reaching contexts that are deemed "better" according to domain knowledge. This process continues until an "ideal" output context is reached, usually corresponding to an exploit that impacts the security of the SUT. In addition to this dynamic approach, we present a static variant based on combinatorial security testing.

We instantiate our approach by targeting cross-site scripting (XSS) vulnerabilities, detailing the unique challenges posed by HTML parsing, and implement this application of HYDRA in a prototype tool.

Results: The evaluation shows that our implementation is able to evade faulty filters and is effective at identifying injection vulnerabilities while remaining more flexible than existing approaches by allowing users to define desirable output contexts.

Conclusion: Based on the results of our evaluation, we are confident that including the HYDRA approach in security assessments will increase the number of identified XSS vulnerabilities, particularly those that are difficult to exploit. We anticipate that an application to other classes of vulnerabilities such as SQL injections will significantly advance the state of the art.

1. Introduction

Few classes of vulnerabilities have had such a widespread, long-lasting and deep impact on web application security as injection flaws [1–3].

An injection flaw may occur whenever a piece of software accepts user-supplied input and uses it to construct a command in the input language of another system, which acts as an interpreter. A wide variety of such input languages exist; perhaps most relevant in the context of web application security are SQL (often used to provide persistent structured storage for data underlying an application), OS commands, and the combination of HTML, JavaScript and CSS that is commonly used to construct web sites. If appropriate measures are not taken to secure the application against injection flaws, an attacker may be able to craft an input string that modifies the structure of the command sent to the underlying interpreter and extend or replace the behavior of the intended command. For instance, an adversary may be able to retrieve data they are not authorized to access, alter sensitive information or deceive a user into accessing a resource controlled by the malicious entity, thus paving the way for further exploitation.

☆ The research presented in this paper has been funded in part by the Austrian Research Promotion Agency (FFG) under grant 865248 (SECuring Web technologies with combinatorial Interaction Testing - SecWIT). The competence center SBA Research (SBA-K1) is funded within the framework of COMET – Competence Centers for Excellent Technologies by BMVIT, Austria, BMDW, Austria, and the federal state of Vienna, managed by the FFG, Austria.

* Corresponding author.
E-mail addresses: mleithner@sba-research.org (M. Leithner), bgarn@sba-research.org (B. Garn), dsimos@sba-research.org (D.E. Simos).

https://doi.org/10.1016/j.infsof.2021.106703
Received 7 January 2021; Received in revised form 29 July 2021; Accepted 31 July 2021
Available online 8 August 2021