Report of Findings

This report examines the Automated Website Vulnerability Detection toolkit provided by group six. Our team performed manual testing of the toolkit against a minimally-configured DVWA instance running on an Ubuntu VM, as well as source code review on specific areas of the toolkit. This analysis revealed an appreciable breadth of the tool's own testing, but a handful of small implementation issues as well: the results of processing a URL varied based on the protocol (HTTP/S) and the capitalization of the string, and some tests had unexpected failures which impacted functionality.

1. General Findings

Overall, Group Five's findings on Group Six's project show that the group was successful in creating a useful vulnerability detector for web applications. When tested against DVWA, the toolkit had no problems brute-forcing its way through the site's insecure login process and was able to continue testing for the application's numerous vulnerabilities. The object-oriented design of the toolkit lends itself to further maintenance and development of vulnerability tests, and already identifies a number of significant attack vectors in its current state.

2. URL Evaluation

Our testing has identified two issues regarding the toolkit's URL evaluations. The toolkit functions as expected when tested against DVWA over HTTP, but is unable to connect to the same application over HTTPS. This may be a simple issue resulting from the absence of a valid server certificate, but without support or diagnostics for this issue the toolkit may struggle to meaningfully evaluate legitimate web applications.

The toolkit's evaluation of the DVWA application also differs on whether the application is hosted under the directory .../dvwa or .../DVWA, with no other changes to the application itself. This difference suggests some degree of inflexibility in evaluating the application's structure, which is similarly problematic.

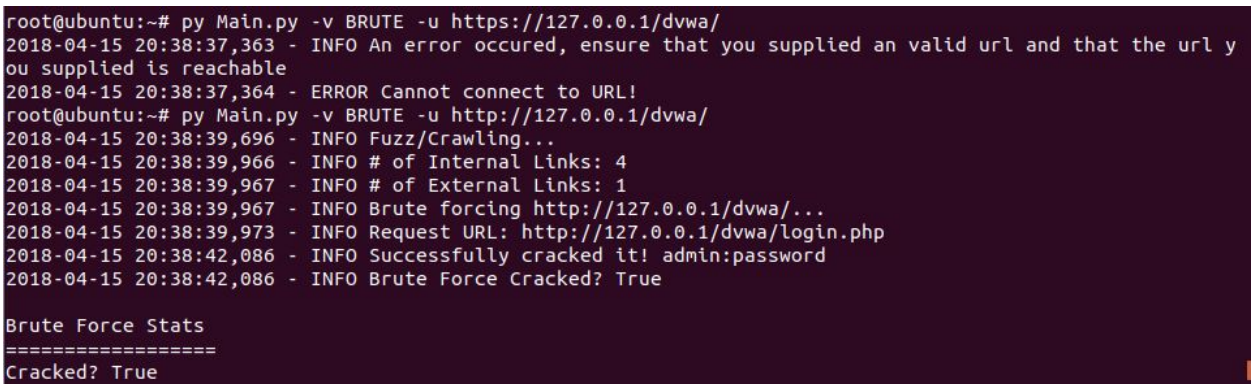
3. Attack Completion

As Group Six had previously identified, one issue that our team came across was with the Cross Site Request Forgery (CSRF) command. Part way through running this attack, the toolkit asks the user to input a CSRF token, but provides the option to continue without entering one. If no CSRF token is provided, the toolkit crashes and the process is not completed.

A second issue in this vein relates to the Active SQL Injection (A-SQL) command. While successful attacks against a vulnerable web application yields useful feedback, if the attack is unsuccessful it has been observed crashing with a stack trace or abruptly end without returning feedback to the user.

Appendix

- (good) Focuses on common vulnerabilities identified by OWASP. Results include documentation on how to resolve any issues found.
- (good) Modular OOP design allows for further development and extensions.
- (good) Tests cover a variety of common patterns and designs, not hardcoded to particular test cases.
- (bad?) Difficulties connecting to HTTPS URLs.



- (bad) Crawler findings change based on whether the URL is upper or lower case, even if the server being tested hosts the site at the URL provided. (testing w/ DVWA on Ubuntu)

```
root@ubuntu:~# py Main.py -v XSS -u http://127.0.0.1/DVWA
2018-04-15 20:56:37,999 - INFO Fuzz/Crawling...
2018-04-15 20:56:38,018 - INFO # of Internal Links: 0
2018-04-15 20:56:38,019 - INFO # of External Links: 1
2018-04-15 20:56:38,028 - INFO Request URL: http://127.0.0.1/DVWA/login.php
2018-04-15 20:56:40,595 - INFO Successfully cracked it! admin:password
2018-04-15 20:56:40,598 - INFO Fuzz/Crawling...
2018-04-15 20:56:40,603 - INFO Performing reflect XSS attack
2018-04-15 20:56:40,603 - INFO No vulnerability in XSS reflected!
2018-04-15 20:56:40,603 - INFO Performing stored XSS attack
2018-04-15 20:56:40,603 - INFO No vulnerability in XSS stored!
root@ubuntu:~# mv /opt/lampp/htdocs/DVWA /opt/lampp/htdocs/dvwa
root@ubuntu:~# py Main.py -v XSS -u http://127.0.0.1/dvwa
2018-04-15 20:57:09,302 - INFO Fuzz/Crawling...
2018-04-15 20:57:09,589 - INFO # of Internal Links: 4
2018-04-15 20:57:09,589 - INFO # of External Links: 1
2018-04-15 20:57:09,597 - INFO Request URL: http://127.0.0.1/dvwa/login.php
2018-04-15 20:57:11,794 - INFO Successfully cracked it! admin:password
2018-04-15 20:57:11,796 - INFO Fuzz/Crawling...
2018-04-15 20:57:18,001 - INFO Performing reflect XSS attack
2018-04-15 20:57:18,829 - INFO No vulnerability in XSS reflected!
2018-04-15 20:57:18,829 - INFO Performing stored XSS attack
2018-04-15 20:57:19,550 - INFO XSS Stored Vulnerability found!

Cross-Site Scripting (XSS) Stats
================
Vulnerability Found True
--- Completed in 9.961 ms
```

- (bad) A-SQL crashes in some cases, and abruptly exits in others:

```
root@ubuntu:~# py Main.py -v A-SQL -u http://127.0.0.1/dvwa/vulnerabilities/
2018-04-15 20:10:05,408 - INFO Fuzz/Crawling...
2018-04-15 20:10:05,878 - INFO # of Internal Links: 27
2018-04-15 20:10:05,878 - INFO # of External Links: 2
2018-04-15 20:10:05,881 - INFO Request URL: http://127.0.0.1/dvwa/vulnerabilities/
Traceback (most recent call last):
  File "Main.py", line 298, in <module>
    main()
  File "Main.py", line 138, in main
    url.args.url
AttributeError: 'str' object has no attribute 'args'
```

- (bad, known) CSRF exits prematurely with no token provided.