Düzce University Journal of Science & Technology

Research Article

VinJect: Toolkit for Penetration Testing and Vulnerability Scanning

Akhan AKBULUT [a,b*]

[a] *Department of Computer Engineering, Istanbul Kültür University, Istanbul, TURKEY*
[b] *Department of Computer Science, North Carolina State University, Raleigh, USA*
** Corresponding author's e-mail address: a.akbulut@iku.edu.tr, aakbulu@ncsu.edu*

ABSTRACT

Penetration testing plays an important role in the development of secure software products and electronic systems. Sustainability of commercial systems is ensured through the regular scans of vulnerability. In this era where quality assurance and testing organizations become increasingly widespread, the effectiveness of the used tools and methods are critical. This article describes the architecture of the software named VinJect, which is developed for efficient penetration testing and vulnerability scanning. The primary goal of this application is to detect vulnerable locations in a shorter time with running in a multi-threaded structure. Our proposed application uses Wapiti and SQLmap applications' services in the background. With user-friendly interfaces, it is also aimed to remove the bad user experience (UX) that these applications running on the command line have. In the tests we performed, WinJect was found to be more efficient in completing the vulnerability scans in a much shorter time.

Keywords: *Penetration testing tool, Vulnerability detection, Security*

VinJect: Sızma Testi ve Güvenlik Açığı Taraması Aracı

ÖZET

Güvenilir yazılım ürünleri ve elektronik sistemlerin geliştirilmesinde sızma testi önemli rol oynamaktadır. Zaafiyet taramalarının düzenli olarak yapılması sayesinde, ticari sistemlerin sürdürülebilirliği sağlanmaktadır. Kalite güvence ve test firmalarının günümüzde yaygınlıklarını arttırdıkları bu dönemde, kullanılan araç ve yöntemlerin etkinlikleri çok kritiktir. Bu makalede etkin bir sızma testi ve güvenlik açığı taraması için geliştirilmiş VinJect ismindeki yazılımın mimarisi anlatılmaktadır. Amaç, çok iş parçacıklı yapıda çalışan bu uygulama ile zaafiyet barındıran yerlerin tespitinin daha kısa sürede yapılmasıdır. Önerdiğimiz uygulama, arka planda Wapiti ve SQLmap uygulamalarına ait servisleri kullanmaktadır. Kullanıcı dostu arayüzler ile çoğunlukla komut satırında çalışan uygulamaların verdiği olumsuz kullanıcı tecrübesinin (UX) ortadan kaldırılması hedeflenmiştir. Yaptığımız testlerde, WinJect'in daha etkin bir kullanım sunduğu ve zaafiyet taramalarını çok daha kısa sürede tamamladığı görülmüştür.

Anahtar Kelimeler: *Sızma testi aracı, Güvenlik açığı taraması, Güvenlik*

Received: 21/05/2018, Revised: 26/05/2018, Accepted: 28/05/2018

I. INTRODUCTION

As technology has progressed, security problems have arisen in both hardware and systems used to it. That's way, security systems have to switch from a defensive to offensive approach, so penetration testing became on the agenda. A penetration test is a simulation of attacks and interventions using methods to attack a malicious person (hacker) against the network infrastructures, hardware, software and applications that make up the information systems of companies and to try to detect and disclose these vulnerabilities to the system [1]. It is important to keep in mind that it is not possible to capture and steal systems, to blackmail money, to infringe on competitor information, to change internet pages, to seize financial information, to retrieve e-mail accounts, to make servers unusable, to disclose personal or corporate information, only a few of the security risks.

In order to minimize or eliminate these risks, it is necessary to make sure that the systems are durable and healthy in every direction by performing periodic penetration tests (pen test). The methods of application of infiltration tests may change as needed. For some institutions over the internet; servers, applications, and devices, as well as for specific applications that are used directly in the IT environments of the internal network and on the internal network in some organizations. If the need arises, scenarios covering both internal and external threats can be created. Penetration test has some technical processes. The processes of most pen test are testing, reporting, verification [2]. All those steps are very important to reach the result.

For testing purposes, the penetration testers use some tools and technical ways. The tools facilitate the work of tester. It may be open source software and proprietary software. VinJect is a useful and automated open source tool that achieves detection of web application vulnerabilities and injection. With using our proposed tool, the tester determines and edits the target to our program. Then, Vinject start to find out vulnerabilities in the target system, and list to a user the vulnerable URLs. The tester passes to the injection step on the program, select and continues with the URLs. On the injection step, the tester uses some injection parameters and try to dump system's stored information.

Many penetration testers use lots of tools in their vulnerability resource. Before Vinject application, they first try to find out vulnerabilities on the target system with some tools, then they use another tool for injection. In addition, the tester has to know the details and properties of the tools which are used in pen testing.

The Vinject application involves two powerful features and put them together. One of them is a detection of vulnerabilities on the target system and another one is the injection to the system with detected vulnerability. In this way, the penetration tester is going to save self-time and the web application security testing is going to be easier. So, web application testing is becoming protective, easy and faster status.

The contribution of this paper is two-fold shown as follows:

- We have developed a toolkit for penetration testing and vulnerability scanning with improved user experience (UX) for pen testers.
- We have demonstrated the practical use of the multi-threaded application with user-friendly interfaces in decreasing the time of vulnerability scans.

The rest of this paper is organized as follows. Section II shows the related work. Section III explains the architecture of WinJect. Then, experimental results are given. Section V describes the threats to validity, and finally, Section VI presents our conclusion.

II. Literature Review

In recent decades, a large number of software systems and platforms are moving to the Web to offer popular new services over the Internet such as e-commerce, e-health, e-government etc. However, at the same time, receiving such forms of web applications is under attack by hackers with the aim of causing unauthorized access to the system, accessing private information, or simply denying service. Securing databases as a priority target against frequent attacks is a major concern while intruders usually intend to steal personal information and make it inoperable by damaging databases.

The need for web applications has grown rapidly over the years with compulsory loyalty, which is technological. This turns the security into a very critical issue and inevitable trend. The number of security vulnerabilities detected in such applications is constantly increasing, especially with SQL injection attacks. For this reason, it is necessary to check Web applications regularly to verify the existence of exploitable vulnerabilities. Looking at application security vulnerabilities, SQL Injection is classified as the most common and dangerous web application failure.

Patil et al. [3] present a clustering method to detect the Cross Site Scripting attacks, SQL and XPath Injection. Their objective is to increase the detection capability of the vulnerability scanner while preserving low false positive and false negative rates. Aliero and Ghani [4] propose a reusable component-based SQL injection attack detection instrument to allow several processes quickly and with low cost. According to authors, their tool was tested with three different vulnerable web applications and was compared its effectivity with seven different SQL injection detection tool. They claim that their tool detected all potential injections on various scenarios.

Parvez et al. [5] examined for the detection ability of some black box web application security scanners against stored SQL injection and stored Cross Site Scripting. For this purpose, they construct test-bed and the results show that black box scanners still required enhancement in identifying stored SQL injection and stored Cross Site Scripting vulnerabilities. Another similar study [6] evaluated the state of the three black box scanners which have support for identifying SQL injection vulnerabilities with custom test bed. They explored scanning, parameters and penetration style, user input, analysis of server reactions, incorrect classification of findings and automatic transaction functionality. In consequence of poor detection rate, they discuss the different stages of black box scanners browsing cycle and suggest a set of offers that can improve the detection proportion of stored SQL injection vulnerabilities.

Escrow [7] is a large-scale SQL Injection detection tool with a platform-independent exploitation module developed by Delamore and Ko. It uses a custom search application along with a static code analysis module to discover possible target web applications supported with the graphical user interface. According to their study, they can scan 100 databases per 100 min and detect lots of vulnerable web applications. Liban and Hilles [8] proposed the SQ L injection vulnerability scanning tool by using a time-based attack with the Inference Binary Search Algorithm to automatically generate SQL injection attacks. That tool contains an order by attacks, true error, time-based, and true/false SQL injection attack types. But this tool only works with PHP-based websites that use MySQL databases. Their announced

results show 93% accuracy for detecting the vulnerability and 84% accuracy for detecting the MySQL injections.

Singh and Roy [9] introduced a network-based security vulnerability scanner approach that provides better coverage and with no false positive. They have used Java and MySQL in their tests which contains Scanner Implementation, Attack Implementation, and Network Implementation. Their claim for the other tools, they detect less than 85% of the vulnerabilities but their tool finds all of it. Lounis et al. [10] propose a method that was built in a perception to enhance the logic reflected in modeling WASAPY [11] which web vulnerability scanner tool. They claim that their proposed method can precisely identify successful injection requests that the WASAPY approach cannot detect.

Another vulnerability scanner study was presented by Salas and Martins [12], they used SOAP UI vulnerability scanner to simulate the attacks and add fraudulent intention for testing the web service request. SOAP messages contain a security problem such as; Injection Attacks, password phishing attacks, social media threats, Denial of Service attacks, and others. To overcome these security issues, a set of rules was developed to analyze responses to reduce false positives and negatives. The results show that 97.1% of requests have at least one security vulnerability.

In addition to academic research projects, there are also several vulnerability scanners used to overcome the security issues such as; Webinspect [13], Gamja [14], N-Stalker Security Scanner [15], IBM Security AppScan [16], BrupSuite [17], Acunetix [18], and ImmuniWeb [19].

III. METHODOLOGY

This section explains the software architecture and the implementation details of the proposed application VinJect.

A. OVERVIEW

VinJect is a tool that is developed with Java programming language and it aims to provide easiness to a tester for SQL injection assessments with the help of two open source software: Wapiti [20] and SQLmap [21]. Basically, the process is performed by trying the all possible combinations available in the target website according to the specified parameters. It is often probable to detect errors resulting from errors that do not understand the SQL injection logic or are caused by the old code libraries. Our proposed tool provides the ability to scan and test the SQL security vulnerability of the destination URL in a practical way. WinJect benefits from Wapiti and SQLmap services.

Wapiti provides security controls over web applications and performs black-box scans. Basically, it is not used to examine the source code of the application, but it can search the web pages of the distributed web application for forms that can inject script and data. After SQLmap receives this list, it loads to see if any script is vulnerable. Wapiti provides infrastructure support for our application.

We intended to bring out the power by bringing these two practical tools together and making improvements in our project. Pen tester defines the target URL that it has specified on Wapiti and starts the pen test process. Wapiti displays URLs that can contain weaknesses as a result of this attack being launched towards the destination URL. Users select the displayed links to inquire and examine. After

the selected links are copied, the query is started in VinJect. It checks these links and displays the results of the injection or non-injection on the screen. If there is no weakness in the links, the program terminates itself. However, if any injection is detected in the link, it is transmitted to the pen tester and it requires parameters about how to proceed from the user. Pen tester will first see Database Platforms that VinJect detects. It will then be able to access the data (tables, columns, etc.) on the target system with the parameters to add. After this process, the data can be dumped and all information can be handed down. Use cases of our proposed software are shown in Figure 1.

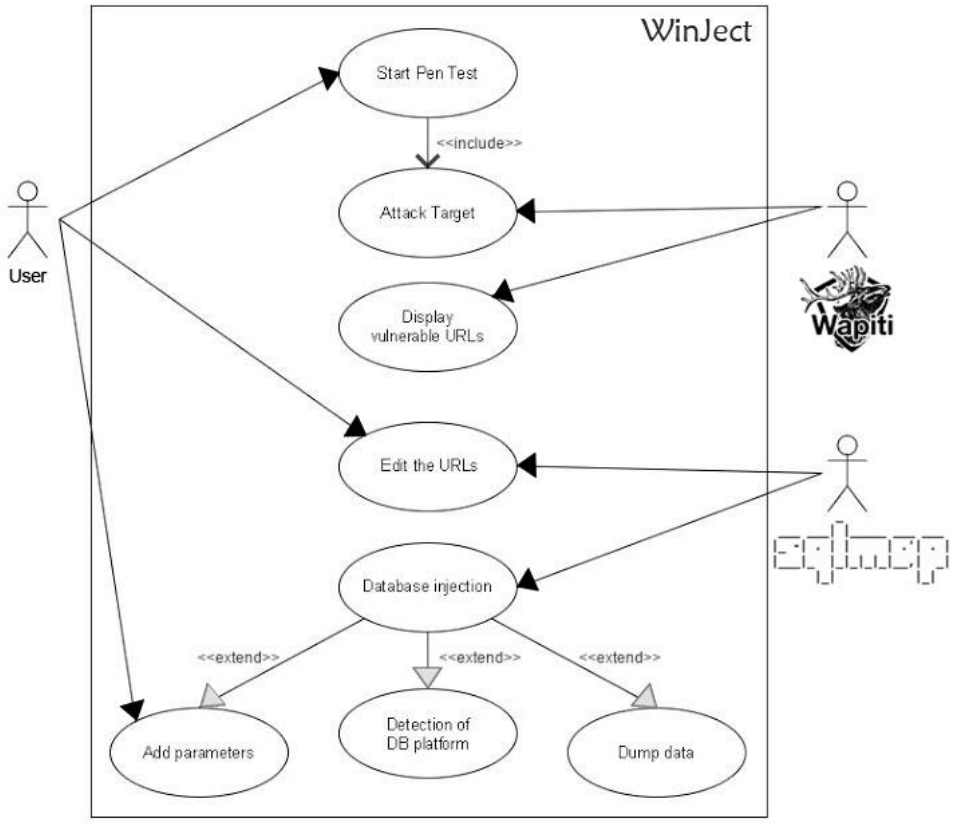

Figure 1. *Pen testing scenarios' use case diagram*

As shown in Figure 1, the three main actors of the WinJect application can be listed as a pen tester, Wapiti, and SQLmap. Pen tester triggers the entire process by giving the target URL to the system. Then WinJect uses the Wapiti's and SQLmap's necessary functions to complete the assessment. The details about use cases are given in Table 1 below.

Table 1. *Use Case Scenarios of WinJect*

Use Case Element	Description
Attack Target	Wapiti starts to reach all URL and identify which are vulnerable
Display vulnerable URLs	Display the URLs that are detected to the pen tester
Send the URLs	The tester can send URLs to SQLmap for injection
Database Injection	SQLmap start to injection on the vulnerable URLs
Add Parameter	The tester includes to SQLmap parameter that detected commands

B. APPLICATION LOGIC

In order to explain how our proposed application work, first we should explain the main components that WinJect uses. Wapiti and SQLmap play an important role in providing the necessary functions and services in the background.

Wapiti accesses pages of web applications; checks whether there are script directories and forms in which the data can be injected. Thus, the application will not run or affect the source code. Wapiti can detect vulnerabilities which are file disclosure, XSS, CRLF Injection, Database Injection, wrong htaccess configurations etc.

SQLmap is the other tool that we use it is also the open-source application which is automated the process of exploiting SQL injection flaw on database servers. This tool has a powerful structure of detection engine which many features as database fingerprinting, over data fetching, accessing to file system, executing commands by out-of-band connections. Sqlmap supports a lot of database management system as SQLite, Oracle, Mysql, Microsoft SQL, SAP MaxDB etc. In addition, it has powerful SQL injection on error-based, out-of-band, UNION query-based, stacked queries, and time-based blind.

The application logic is started with the users setting the target to WinJect. Then, Wapiti starts to detection of vulnerable URLs on the target system. WinJect display results that are found by Wapiti. After those, there should be a decision node because the user would want to exit from WinJect without the next step. Then, the user needs to select an URL which is found and displayed and injection starts on SQLmap with selected URL. In the end, the user can directly exit or after dump operation, can exit as we show Figure 2 from WinJect.

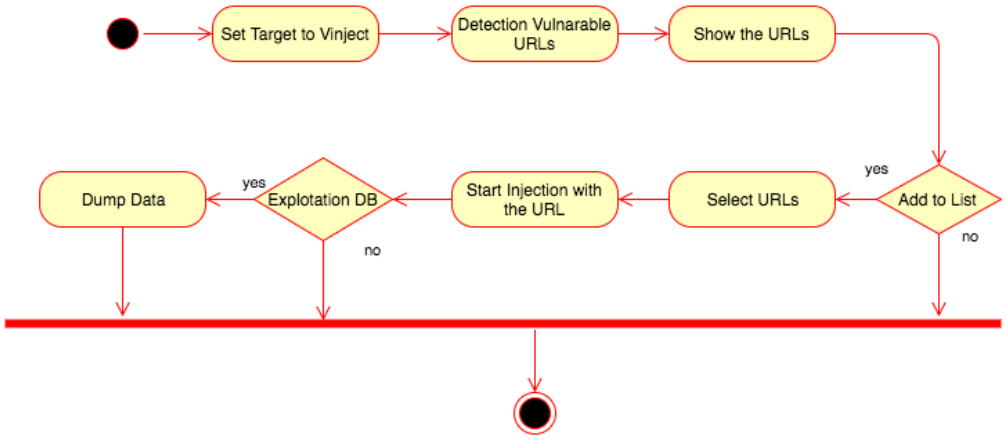

Figure 2. *Activity Diagram of Assessment Process*

Perhaps the most important architectural design decision of the Vinject is its multi-threaded structure [22]. Since the proposed system runs lots of processes (different points to asses) at the same time, we prefer to implement a multi-threaded approach as an efficient way to reduce long execution cycles. We implemented a *JFrame* approach with a *SwingWorker* function to trigger multiple working threads in

the background. In this way we provide parallelism and assessment periods are reduced dramatically. The interactions between system elements are depicted in Figure 3.

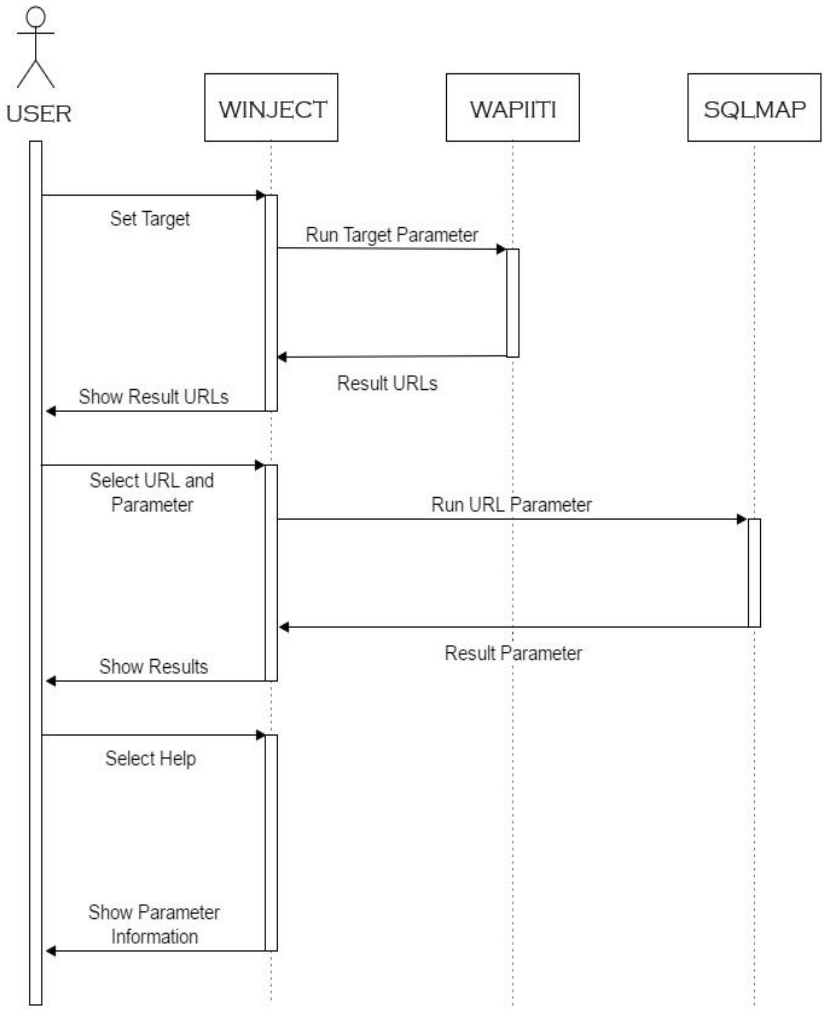

Figure 3. *Sequence diagram of Components' interaction*

IV. EXPERIMENTS

For testing the effectiveness of VinJect application, we used virtual Kali Linux distributions to host the testing environments. Two different test were completed; a black-box testing and an acceptance test.

A. FUNCTIONAL TESTING

4 Undergraduate students were invited to perform a black-box testing to verify the functionality of the proposed application. Test cases of this testing activity are given below.

Table 2. *Detection in Vinject*

TEST CASE 01: DETECTION IN VINJECT	
Preconditions	Set Target
Scenario	Pen Testers set target on Vinject
Steps	- Pen Testers specifies the target URLs. - Add the URL to the relevant place and click "Start" button. - Vinject detects the vulnerability URLs.
Expected Outputs	The user can see and detect which vulnerability URLs on the target

Table 3. *Injection with Vinject*

TEST CASE 02: INJECTION WITH VINJECT	
Preconditions	Select URLs and add parameters
Scenario	Pen Testers select vulnerability URLs and add parameters on Vinject
Steps	- Pen Testers select vulnerability URL and click "Add List" button. - Pen Testers add parameters to vulnerability URL and click the “Run” button. - Vinject will inject databases and dump data. - The dumped data will save in the .txt file.
Expected Outputs	The user can dump any data from the vulnerability URL with Vinject

After applying the given two test cases, all testers were able to get the expected results and WinJect achieved a 100% success in functional testing.

B. USABILITY & PERFORMANCE TESTING

Functional testing showed the operational capabilities of WinJect, however, its usefulness remained out of focus. So, in order to demonstrate the benefits that can be gained from our proposed application, we have organized a usability & performance test as a secondary validation job [23]. The aim of this test is to find the vulnerabilities of given two web sites using WinJect. One team used our proposed application whereas the other use Wapiti and SQLmap. Tests are conducted with one entry-level and two moderate-level experienced, in total 3 testers. One of the moderate levels experienced undergraduate student asked to use WinJect, while others use Wapiti and SQLmap standalone.

Table 4. *Test Results*

		Web Site 1	Web Site 2
Pen Tester 1♊	Coverage	38 / 40	148 / 163
	Duration	2 min 17 sec	11 min 09 sec
Pen Tester 2♌	Coverage	9 / 40	25 / 163
	Duration	45 min	3 hour 48 min
Pen Tester 3♊	Coverage	37 / 40	151 / 163
	Duration	1 hour 07 min	5 hour 29 min

♊ : Moderate Level Experienced User, ♌ : Entry Level Experienced User

WinJect Usage, Wapiti and SQLmap Usage

Usability and performance tests showed that WinJect increases the efficiency of using Wapiti and SQLmap. Tester 1 accomplished to finish its tasks 25 times faster than Tester 2 and 3. The fact that the application was developed in a multithreaded structure and has user-friendly interfaces has made this possible. The faster testing of WinJect users to complete tests helps optimize resource utilization.

V. THREATS TO VALIDITY

The validity of VinJect approach can be effected from several matters. This section explains potential threats against proper usage [24].

- Experience of the tester: The experience of the pen tester who uses the application is directly affecting the assessment results. If the user does not have sufficient basic information in the domain of cybersecurity, the usage capacity will be limited. Especially testers who do not know the parameters which can be used in tuning SQLmap can be less effective and they will find a lesser amount of vulnerabilities.

- Version control: As we explained in the Methodology section, our proposed solution uses the Wapiti and SQLmap applications in the background. In order to benefit from services of these applications, they should be installed on the system that will run WinJect. We have developed our application by using the Wapiti v3.0.0 and SQLmap v1.1.4. If new versions of these applications are published with modified/changed services, WinJect should be modified as well. Otherwise, the application may not work due to invalid service calls.

- Platform: The recommended platform for WinJect is Linux distributions. On our development tests, we observed that Mac OS platforms can be used as an alternative. But using the Microsoft Windows operating system causes some crashes that disable effective usage.

VI. CONCLUSION

In this paper, we proposed a penetration tool which is developed for a better user experience. The proposed application WinJect uses the Wapiti and SQLmap services inside and aims to make more efficient for pen testers. Our motivation is to combine the necessary applications of pen testers' tools under a single roof and reduce the workload and save time. Penetration testing can be done in various ways and many different tools can be used. But the efficiency of the assessment is directly related to the selected tool. We implemented a multi-threaded approach to maximize the resource allocation to the assessment. This allowed tests resulting in faster than other tools.

While we created this application, we aimed to reduce the workload on the pen tester and complete the long penetration testing process in less time. WinJect will make things easier when doing penetration testing to pen testers. Vinject is powered by Wapiti and SQLmap infrastructure, allowing faster testing. Detection can be performed on the target with Wapiti. Wapiti services' output can be directed to SQLmap for further inspections without any difficulties. Database injection can be performed on the current target detected by SQLmap. It is hardly possible to avoid long periods of time when the individual tools are being processed separately. WinJect detects the target firstly, then inject on the detected targets and reports it by dumping the databases through the parameters. In today's world, where even 1 second is so precious, it is of utmost importance that things are completed in a serial manner. In WinJect, operations can be performed up to 5 injection tests at the same time. WinJect became a functional vulnerability detection injection tool using Wapiti and SQLmap's infrastructure. Vinject is not an application that the pen tester can develop or learn from. It is a platform developed for more practical work. This means big gains for pen tester in terms of time and workload. It has been prioritized to use the interface effectively to keep the use of the keyboard at a minimum level and to use the mouse as much as possible. WinJect is designed with a clean and understandable interface for easy use of pen tester. Our experimental tests show that with using WinJect, pen testers are able to accomplish the assessments 25 times faster than traditional methods.

ACKNOWLEDGEMENTS: Author would like to thank Ali Rıza Selçuk for his assistance in testing the WinJect application.

VII. REFERENCES

[1] L. Allen, T. Heriyanto, and S. Ali, *Kali Linux–Assuring security by penetration testing*. Packt Publishing Ltd, 2014.

[2] W. Stallings, L. Brown, M. D. Bauer, and A. K. Bhattacharjee, *Computer security: principles and practice*. Pearson Education, 2012.

[3] S. Patil, N. Marathe, and P. Padiya, "Design of efficient web vulnerability scanner.", *Inventive Computation Technologies (ICICT), International Conference on*. vol. 2. IEEE, 2016, pp. 1–6.

[4] M. S. Aliero and I. Ghani, "A component based SQL injection vulnerability detection tool.", *Software Engineering Conference (MySEC), 2015 9th Malaysian*. IEEE, 2015, pp. 224-229.

[5] M. Parvez, P. Zavarsky, and N. Khoury, “Analysis of effectiveness of black-box web application scanners in detection of stored SQL injection and stored XSS vulnerabilities.”, *Internet Technology and Secured Transactions (ICITST), 2015 10th International Conference for*. IEEE, 2015, pp. 186-191.

[6] N. Khoury, P. Zavarsky, D. Lindskog, and R. Ruhl, “An analysis of black-box web application security scanners against stored SQL injection.”, Privacy, *Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 1095-1101.

[7] B. Delamore and R. K. Ko, “Escrow: A large-scale web vulnerability assessment tool.”, *Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on*. IEEE, 2014, pp. 983-988.

[8] A. Liban and S. M. Hilles, “Enhancing Mysql Injector vulnerability checker tool (Mysql Injector) using inference binary search algorithm for blind timing-based attack.”, *Control and System Graduate Research Colloquium (ICSGRC), 2014 IEEE 5th*. IEEE, 2014, pp. 47-52.

[9] A. K. Singh and S. Roy, “A network based vulnerability scanner for detecting sqli attacks in web applications.”, *Recent Advances in Information Technology (RAIT), 2012 1st International Conference on*. IEEE, 2012, pp. 585-590.

[10] O. Lounis, S. E. B. Guermeche, L. Saoudi, and S. E. Benaicha, “A new algorithm for detecting SQL injection attack in Web application.”, *Science and Information Conference (SAI), 2014*. IEEE, 2014, pp. 589-594.

[11] A. Dessiatnikoff, R. Akrout, E. Alata, M. Kaâniche, and V. Nicomette, “A clustering approach for web vulnerabilities detection.”, *Dependable Computing (PRDC), 2011 IEEE 17th Pacific Rim International Symposium on*. IEEE, 2011, pp. 194-203.

[12] M. I. P. Salas and E. Martins, “A black-box approach to detect vulnerabilities in web services using penetration testing.”, *IEEE Latin America Transactions* 13.3, pp. 707-712, 2015.

[13] Fortify WebInspect, (2018, 20 Mayıs). [Online]. Available: http://www8.hp.com/us/en/software-solutions/webinspect-dynamic-analysis-dast/

[14] Gamja: Web vulnerability scanner, (2018, 20 Mayıs). [Online]. Available: https://sourceforge.net/projects/gamja/

[15] N-Stalker The Web Security Specialists, (2018, 20 Mayıs). [Online]. Available: http://www.nstalker.com/

[16] IBM Security AppScan, (2018, 20 Mayıs). [Online]. Available: https://www.ibm.com/developerworks/downloads/r/appscan/index.html/

[17] Burp Suite Scanner | PortSwigger, (2018, 20 Mayıs). [Online]. Available: http://portswigger.net/suite/

[18] Acunetix, (2018, 20 Mayıs). [Online]. Available: https://www.acunetix.com/web-vulnerability-scanner/

[19] ImmuniWeb Application Security Testing Platform, (2018, 20 Mayıs). [Online]. Available: https://www.htbridge.com/immuniweb/

[20] Wapiti: a Free and Open-Source web-application vulnerability scanner in Python for Windows, Linux, BSD, OSX, (2018, May 20). [Online]. Available: http://wapiti.sourceforge.net/

[21] sqlmap: automatic SQL injection and database takeover tool, (2018, 20 Mayıs). [Online]. Available: http://sqlmap.org/

[22] V. Pankratius, A. R. Adl-Tabatabai, and W. Tichy, eds. *Fundamentals of multicore software development*. CRC Press, 2011.

[23] P. Ammann and J. Offutt, *Introduction to software testing*. Cambridge University Press, 2016.

[24] H. K. Wright, M. Kim, and D. E. Perry, "Validity concerns in software engineering research.", *Proceedings of the FSE/SDP workshop on Future of software engineering research*. ACM, 2010, pp. 411-414.