Another Effective Approach to Detect SQL Injection and Business Intelligence Using Splunk

Hung-Yu Chien, Meng-Yuan Chiang, Yi-Ting Hsin, Chun-Pei Hung
Department of information management, National ChiNan University

Abstract

Accessing various web applications has become one of the must activities of the general public nowadays, and these accessing activities not only generate lots of valuable data but also open possible security weaknesses to adversaries. SQL injection is one of the most popular web security issues. Conventionally, web programmers used to detect possible SQL injection attacks by embedding detecting codes in their programs. The effectiveness of this approach depends on the secure detection and protection of all web pages in a system, and any single neglect in any pages would open serous security holes to the whole system. In addition to possible security holes, lots of web access activities also contain lots of valuable information. In addition to possible security holes, web activities also provide potential valuable data. Conventional approach to mining these web activities is through analyze the database records that these activities generate; even though this approach works, it has several limitations in some cases; some valuable data might not be recorded in database and it is sometimes not convenient or not authorized to mine data cross several cross-authority database. In this paper, we challenge these issues by using Splunk. Splunk is a big data indexing and searching tool. We apply Splunk on indexing and searching the web logs and network packets. This approach has several potential advantages: (1) we can detect possible SQL injection attacks in an effective and whole system-wise way, and (2) it is easier to explore business intelligence cross-authority data sources.

Keywords: SQL injection, Splunk, Web, security, business intelligence.

1 Introduction

1.1 Background

Computers and Internet have made a great impact on our daily lives; People use them for entertainment and work. Accessing various web applications has become one of the must activities of many people. In order to record users' activities and personal data, the web developers usually record these data in their databases in their backend servers in some concise formats. Programmers might use these databases to explore or

mine valuable information for business intelligence including better services or more competitive strategies. However, these application recorded data in the databases usually record only seem necessary data (like user name, action, products, and so so) for the application layers but neglect or unable to record some implicit data (like machine IP address, software used, lower-layer packet data). Unfortunately, these implicit data play some critical factors to both the web security and the business intelligence. For example, the software (it is usually called software agent) a client uses provide the information whether the client access the Internet through computer or mobile phone, or which operation system he use. And, the content of the lower-layer packets from users transmission traffics provide more subtle information for mining any security threats.

In addition to the above concerns, each web developer is confined to those databases they created or are authorized, but are not allowed (or very in-convenient) to access those databases belonging to different developing groups, even though they all belong to the same institution or company.

SQL injection is one kind of attacks by inputting un-expected commands or control characters in the input forms [7]. If the system fails in detecting and excluding these malicious codes or commands, then the attacker might get un-authorized rights or get private data of the system. Even though several web programming languages like PHP [9] have provide several useful application program interfaces (API) for developers to detect these malicious inputs. Its effectiveness still depends on each programmer's awareness and profession. In a company-wise system, there are many web pages developed by different groups and sometimes even the tasks of developing web pages are out-sourced. Furthermore, embedding detecting code in individual web page cannot allow the system to explore cross-system-wise tracts of attacking.

These limitations or barriers build big obstacles to get better security detection and better business intelligence exploration for company-wise purposes.

1.2 Purpose

Logs and packet records are the historical records of what systems and devices have done or communicated. Correct logging not only facilitates the detection of suspected SQL injection attacks but also provide abundant data for exploring business intelligence. Therefore, correct logging and keeping logs secure is very important.

Accessing various web applications is a must for many people these days, and these accessing activities not only generate tons of logs but also tons of transmission packets. Applications used to neglect these logs and packets because it was very difficult or in-efficient to explore information in these logs and packets without proper big data tool.

As there are more and more big-data tools available like [1, 2], it is an interesting and challenging to apply suck kinds of tool to solve the challenges. Splunk [2, 3] is one of such tools that are available and is free for its trial version. In this study, we choose Splunk to solve this challenge and evaluate its performance. Splunk is a universal log analyzer. The merits of this tool include the following:

(1) It can index data from different platforms and in different data formats no matter whether these formats are well studied (there are built-in parsers) or new formats;
(2) It provides a flexible and scalable architecture which could adapt to various organization size and structure;
(3) It provides powerful search commands that can help IT technicians to design efficient and effective search scenarios and scripts;
(4)) The search scenarios and reports could be automatically executed and properly presented to users of different professional backgrounds.

In this study, we focus on the effectiveness of applying Splunk, wireshark and tshark on detecting SQL injection attacks and mining business intelligence from tons of logs and packet traffics. Since packet logs belong to the whole institution (or the whole company) and not confine themselves to any single developing group. The effectiveness of this approach has significant impacts: (1) it allows IT technicians much easier way to tackle their challenges in a cross-group and company-wise way [8], (2) it opens a new and way for IT technicians to mining the information in a deeper and cross-application manner, because logs and packet records contain many activities from many applications from the whole organization. The rest of this paper is organized as follows. In Section 2, we introduce Splunk main modules with key functions, wireshark [4] and tshark [5]. We describe how we apply these tools in solving the above challenges- the methodology, the procedure, and the search scenario demonstration in Section 3, and discuss its effectiveness in Section 4. Finally, we give our conclusions in Section 5.

2 The Tools and SQL Injection

2.1 Splunk

Splunk is an integrated package of modules that facilitate analysts the jobs of collecting various logs, analyzing the logs and reporting key information across various platforms and various log-formats. Real-time monitoring and friendly reporting utilities are powerful features of this tool.

Splunk’s main key features are discussed as follows.

- Splunk can parse, tag and index any data from any devices and applications, no matter what formats they owns. For example, as shown in Fig. 1, it can parse, tag and index Windows and Linus system logs, various network device logs, and any applications' log and data.
- Powerful, flexible and scalable structure: Splunk can real-time monitor remote data, and use an encrypted connection to forward data so that logs can be saved and indexed in secure indexers. It provides web interface and commands for users to inspect and audit their log data. It provides flexible and scalable architecture to adapt to different organization structures and requirements .

Figure 1. Splunk function overview

- Powerful search engine

  In addition to collecting data from cross-platforms, powerful searching is an another great feature to derive valuable information from big data. If we use general search engines on the market, the loading on computers will deteriorate its performance. On the contrary, Splunk search engine solves this problem, and it owns some merits. (1) fast indexing: data will be instantly tagged and indexed in an efficient way when the data are fed into Splunk; (2) it is easy to derive valuable information cross various data source using interrelated data fields; (3) Math calculation and analysis: when Splunk indexes and searches its data, it can instantly analyze the fields, generate new fields using interrelated data, calculate statistics and report immediately.

Slpunk’s flexible and scalable architecture includes several key modules that are introduced as follows and depicted in Fig. 2.

- Forwarder: A forwarder is a Splunk Enterprise instance that forwards data to another Splunk Enterprise instance (an indexer or another forwarder) or to a third-party system. It facilitates indexers real-time monitor remote data and securely archive the data.
- Indexer：An indexer is the Splunk Enterprise instance that indexes data. The indexer tags the raw data into events and stores the events into an index. The indexer also searches the indexed data in response to search requests.
- Searcher: In a distributed search environment, the search head is the Splunk Enterprise instance that handles search management functions, directs search requests to a set of search peers and then merges the results back to the user. And a search peer is a Splunk Enterprise instance that performs indexing and fulfills search requests originating from the search head.

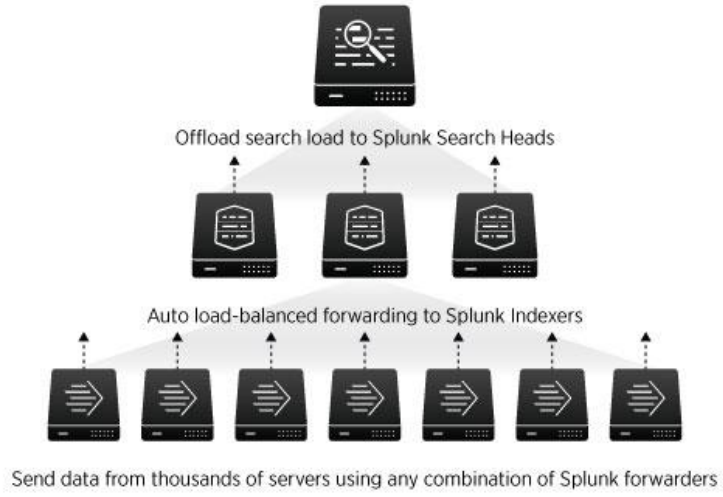

Figure 2. Splunk architectures and deployment

2.2 Wireshark and tshark

Wireshark [4] is an open-source program that can record the contents of packets, and examine these contents according to respective network protocol layers. For example, one can see a packet header and its contents, according to medium access layer, IP packet layer, TCP layer, HTTP layer [6], and so on. In this study, we apply wireshark to record all transmission traffics of our experiments. Wireshark provide friendly graphic user interfaces and is easy to use.

Tshark [5] is also a command-line packet recording tool like wire shark and it further provides more subtle command options to filter, transform and extract

specified data fields within packets, and these specified data fields can be stored in several popular file formats. We apply this tool to extract valuable fields in specified formats, and then load these files into Splunk.

2.3 SQL Injection

In web application, developers usually design some kind of forms to allow users to input data to the system so that the system can record personal profiles or provide specified services. SQL injection is one kind of attacks by inputting un-expected commands or control characters in the input forms. If the system fails in detecting and excluding these malicious codes or commands, then the attacker might get un-authorized rights or get private data of the system. Even though several web programming languages like PHP [9] have provide several useful application program interfaces (API) for developers to detect these malicious inputs. Its effectiveness still depends on each programmer's awareness and profession. In a company-wise system, there are many web pages developed by different groups and sometimes even the tasks of developing web pages are out-sourced. Furthermore, embedding detecting code in individual web page cannot allow the system to explore cross-system-wise tracts of attacking.

3 Detecting SQL injection attacks and exploring business intelligence from packet traffics

In order to verify the effectiveness of detecting SQL injection and exploring business intelligence from a company-wise view using transmission traffics, we design an experiment environment, develop a new approach, implement the experiment and evaluate its effectiveness in this section.

3.1 The experiment environment

The experiment environment consists of a personal computer on which a shopping web application is provided, several clients from several computers launch their request to the web application, and the tools-wireshark and tshark- are installed on the computer to record transmission traffics. The tool Splunk can be installed on the same computer or another computer. Fig. 3 shows the experiment environment.

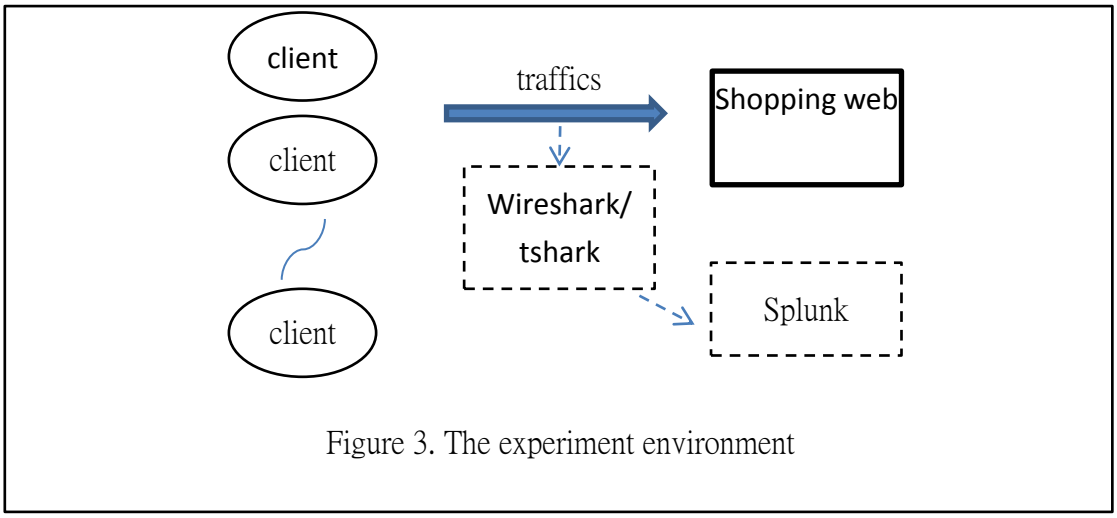

Figure 3. The experiment environment

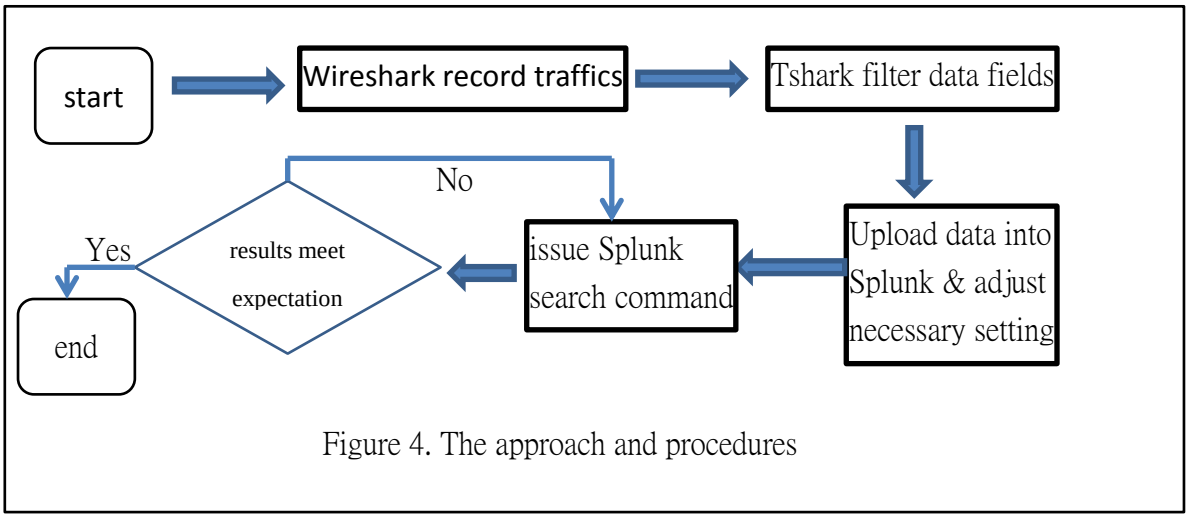

Figure 4. The approach and procedures

3.2 The methodology and procedures

The general approach and procedure is depicted in Fig. 4, and the details are described as follows.

1. Use Wireshark to record the transmission traffics towards the shopping web server.
   We use Wireshark to record the transmission traffics, filter the HTTP packets with the specified IP address, and store the recording in files.
2. Use Tshark to filter specified data fields.
   There are many data fields that are un-relevant to our experiment. For each application, it is better to extract those interested data fields only to save space and time.
3. Transform the data format to Splunk-friendly format and load it into Splunk.

Splunk can read any data format. But, to facilitate the later searching, we delete redundant blanks among keywords. Here keywords are those interested terms in logs that are to be searched later.

4. Issue Splunk search command to detect SQL injection or business intelligence.
   Depending on each individual goal and domain, we issue the corresponding commands to accomplish the tasks.

```
C:\Program Files\Wireshark>Tshark  -r c:\tmp\sqlinjection.pcapng  -T fields  -E
header=y  -E separator=, -E occurrence=a  -E quote=d -e frame.time -e ip.src -e
ip.dst  -e data > c:\tmp\sqlinjection
```

Figure 5. tshark command for the SQL injection detection experiment

3.3 Case study

3.3.1 The SQL inject detection

In this case, we applied Tshark to we to filter interested data fields. These fields include “frame.time”, “ip.src”, “ip.dst”, and “data” in the SQL injection detection experiment, and Fig. 5 shows the command, where “data” corresponds to the input form of the web page.

After uploading the files into Splunk, we need to detect those suspected commands in the packet contents. Fig. 6 showed one example of which the search filters the special character “ ’ ” and “ ” ” in the input form. Fig. 7 shows the suspected SQL injection commands (in ASCII format) and their corresponding IP sources.

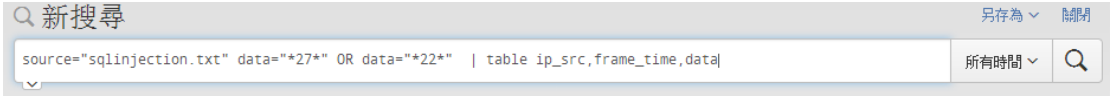

Figure 6. Filtering special character “’” and “”” in the input form.

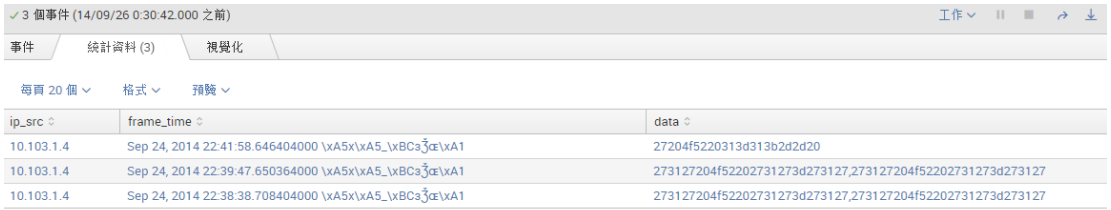

Figure 7. The filtering of suspected packets and ip source

3.3.2 The business intelligence exploration

In our experiment, we develop a shopping web application on which customers can purchase various products. Customer can browse the product descriptions, put their selected ones in a shopping cart, and place the order. In this experiment, we would to derive various business intelligences, using the transaction traffics. Because each transaction in a web application might contain several sub-transactions with the same field name (it is the product name in our examples). Without proper pre-processing the raw data, it would generate several records of the same field name, and would generate incorrect statistics. Splunk provides several configuration files to tailor the pre-processing process for each individual application. In this case, we design the transforms.conf file with a setting of "REGEX = \s(.+?)="(.+?)"\s", which is a regular expression indicating the format for matching and extracting sub-field. Fig. 8 shows the listing of correct transactions. Fig. 9 shows one example of transaction statistics by product name, and the corresponding search command is {source="BI_test1.txt" checkout-product | rex field=_raw "product_name=\"(?<a>[\w|\d]+)"\" max_match=5 | rex field=_raw "quantity=\"(?<b>[\w|\d]+)"\" max_match=5 | eval fields = mvzip(a,b) | mvexpand fields | rex field=fields "(?<name>[\w|\d]+),(?<number>[\w|\d]+)" | stats sum(number) by name}, where the main meaning is to break the incoming raw data into individual fields of "product_name", "quantity", "number" and "allmoney", and finally sum up the values of product. Now we can easily depict the statistics as various visual chart to show the statistics and transaction trends.

事件 統計資料 (9) 視覺化

每頁 20 個 格式 預覽

_time	name	number	money	allmoney
2014/10/11 03:53:00	iMac	1	100	100
2014/10/11 03:53:00	iPhone	1	101	101
2014/10/11 00:32:00	iPhone	2	101	202
2014/10/11 20:13:55	iPhone	1	101	101
2014/10/11 20:13:55	iMac	1	100	100
2014/10/11 20:13:55	SamsungSyncMaster941BW	1	200	200
2014/10/11 20:13:55	CanonEOS5D	2	80	160
2014/10/11 20:13:55	iMac	1	100	100
2014/10/11 20:13:55	iPhone	1	101	101

Figure 8. Splunk correctly extracts related fields from packets

name	sum(number)
CanonEOS5D	2
SamsungSyncMaster941BW	1
iMac	3
iPhone	5

Figure 9. Transaction statistics by product name

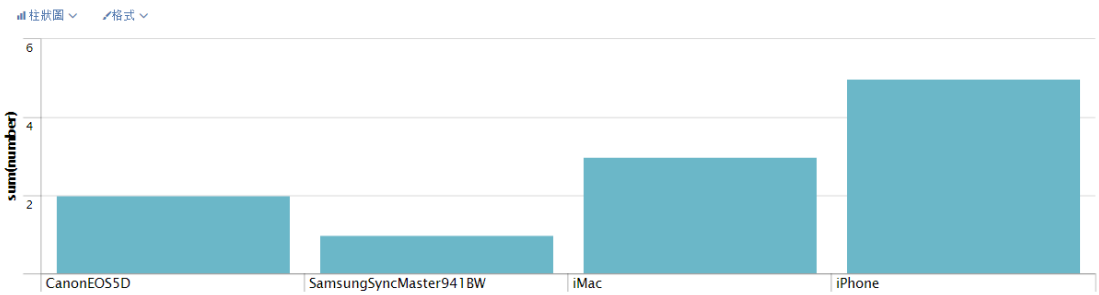

Figure 10. Splunk shows transaction statistics by product name.

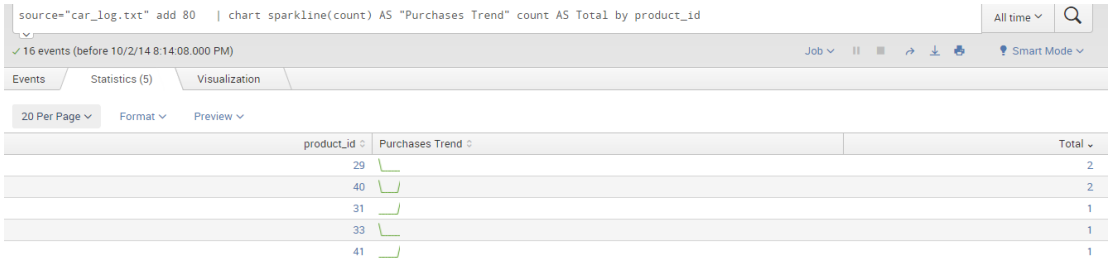

Figure 11. Splunk shows each product transaction trend.

The result of this experiment verifies that our new approach can easily extract business intelligence from network traffics.

4 Comparison between the conventional approach and the new approach

The conventional approach of detecting SQL injection is to embed detection codes in all possible web pages. The effectiveness of this approach depends on embedding proper codes in all web pages that accept users' input. As various web projects are usually maintained by different groups, and some web projects might be out-sourced. The conventional approach incurs a big burden and challenge to secure all the web applications. On the contrary, the new approach detects possible SQL injection directly from the network traffics. This approach easily covers all web accesses, when the network traffics are recorded in some critical points like routers or gateways. It also points out one new potential: detecting more subtle attacks that crosses several applications and protocols.

The conventional approach of mining business intelligence is analyzing the transaction databases. This approach can only extract the information that is confined to those authorized database accesses, and is difficult to explore the information that need to inter-correlate many databases cross several applications or several authority domains. On the contrary, mining business intelligence directly from network traffics can easily solve these limitations, because all traffics no matter from which applications all being recorded in traffic logs.

5 Conclusions and discussions

In this paper, we have discussed the limitations of the conventional approach of detecting SQL injection and exploring business intelligence. We have design a new approach of tracking the above challenges. In the experiments, we have verified the effectiveness of the new approaches, and have pointed out its further potentials. One of our future works is to extending the current experiments to cover those traffics coming from many applications sources, many authorities and many protocols. One limitation of the new approach is that wireshark and tshark should be able to read the traffic contents. So one another challenge is to applying this approach on those encrypted channels. .

Acknowledgements: This project is partially supported by the National Science Council, Taiwan, R.O.C., under grant no. MOST 103-2221-E-260 -022.

6 References

[1] Gabriel, R., Hoppe, T., Pastwa, A., Sowa, S.: Analyzing Malware Log Data to Support Security Information and Event Management: Some Research Results, Advances in Databases, Knowledge, and Data Applications, DBKDA '09. First International Conference, 108 – 113(2009).

[2] Wikipedia, Splunk, http://en.wikipedia.org/wiki/Splunk.

[3] Splunk official website document -- About Splunk Enterprise deployments, http://docs.splunk.com/Documentation/Splunk/latest/Overview/AboutSplunkEnterprisedeployments.

[4] Wikipedia, Wireshark, http://zh.wikipedia.org/wiki/Wireshark.

[5]Tshark- The Wireshark Network Analyzer 1.12.2, https://www.wireshark.org/docs/man-pages/tshark.html.

[6] Wikipedia, The Hypertext Transfer Protocol, http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol.

[7] Wikipedia, SQL injection, http://en.wikipedia.org/wiki/SQL_injection.

[8] Hung-Yu Chien, Meng-Yuan Chiang, Yi-Ting Hsin, Chun-Pei Hung, “Efficiently tracking TANET suspicious packets and identifying the devices using Splunk”, International Computer Symposium (ICS2014), Dec 14-17, Taichung, Taiwan.

[9] Wikipedia, PHP, http://en.wikipedia.org/wiki/PHP.